

Multiobjective Optimal Control Methods for Fluid Flow Using Reduced Order Modeling

Sebastian Peitz · Sina Ober-Blöbaum ·
Michael Dellnitz

Received: date / Accepted: date

Abstract In a wide range of applications it is desirable to optimally control a dynamical system with respect to concurrent, potentially competing goals. This gives rise to a multiobjective optimal control problem where, instead of computing a single optimal solution, the set of optimal compromises, the so-called Pareto set, has to be approximated. When the problem under consideration is described by a partial differential equation (PDE), as is the case for fluid flow, the computational cost rapidly increases and makes its direct treatment infeasible. Reduced order modeling is a very popular method to reduce the computational cost, in particular in a multi query context such as uncertainty quantification, parameter estimation or optimization. In this article, we show how to combine reduced order modeling and multiobjective optimal control techniques in order to efficiently solve multiobjective optimal control problems constrained by PDEs. We consider a global, derivative free optimization method as well as a local, gradient based approach for which the optimality system is derived in two different ways. The methods are compared with regard to the solution quality as well as the computational effort and

This research was funded by the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (its OWL) and the DFG Priority Programme 1962 "Non-smooth and Complementarity-based Distributed Parameter Systems". Calculations leading to the results presented here were performed on resources provided by the Paderborn Center for Parallel Computing (PC^2)

Sebastian Peitz

Department for Mathematics, University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany

E-mail: speitz@math.upb.de

Sina Ober-Blöbaum

Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

Michael Dellnitz

Department for Mathematics, University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany

they are illustrated using the example of the two-dimensional incompressible flow around a cylinder.

1 Introduction

In many applications from industry and economy, one is interested in simultaneously optimizing several criteria. For example, in transportation one wants to reach a destination as fast as possible while minimizing the energy consumption. This example illustrates that in general, the different objectives contradict each other. Therefore, the task of computing the set of optimal compromises between the conflicting objectives, the so-called *Pareto set*, arises. This leads to a multiobjective optimization problem (MOP) or, if the control variable is a function, a multiobjective optimal control problem (MOCP). Based on the knowledge of the Pareto set, a *decision maker* can use this information either for improved system design or for changing control parameters during operation as a reaction on external influences or changes in the system state itself.

Multiobjective optimization is an active area of research. Different approaches exist to address MOPs, e.g. deterministic approaches [44, 19], where ideas from scalar optimization theory are extended to the multiobjective situation. In many cases, the resulting solution method involves solving multiple scalar optimization problems consecutively. Continuation methods make use of the fact that under certain smoothness assumptions the Pareto set is a manifold that can be approximated by continuation methods known from dynamical systems theory [30]. Another prominent approach is based on evolutionary algorithms [1, 12], where the underlying idea is to evolve an entire set of solutions (population) during the optimization process. Set oriented methods provide an alternative deterministic approach to the solution of MOPs. Utilizing subdivision techniques, the desired Pareto set is approximated by a nested sequence of increasingly refined box coverings [17, 56].

When dealing with control functions, a multiobjective optimal control problem (MOCP) needs to be solved. Similar to MOPs, ideas from scalar optimal control theory (see e.g. [31] for an overview of optimal control methods for PDE-constrained problems) can be extended to take into account multiple objectives. By applying a direct method, the MOCP is transformed into a high-dimensional, nonlinear MOP such that the methods mentioned before can be applied. Another approach is based on the transformation of the MOCP into a sequence of scalar optimal control problems and the use of well established optimal control techniques for their solution. Examples for MOCPs can be found e.g. in [15, 41, 48, 56] with ODE constraints. In [42] as well as [2], nonlinear PDE constraints are taken into account but the model is treated as a black box, i.e. no special treatment of the constraints is required. The first articles explicitly taking into account PDE constraints are [34, 33], where multiobjective optimal control problems are solved with a weighted sum approach and model order reduction techniques subject to linear and semilinear PDE

constraints, respectively. Fluid flow applications have been considered in [47, 49].

All approaches to MOP / MOCP have in common that a large number of function evaluations is typically required. Thus, the direct computation of the Pareto set can quickly become numerically infeasible. This is frequently the case for problems described by (nonlinear) partial differential equations such as the Navier-Stokes equations. Standard optimization methods for PDEs [31] often make use of a discretization by finite elements, finite volumes or finite differences which results in a high-dimensional system of ODEs. In a multi query context (such as optimization, parameter identification, etc.), this approach often exceeds the limits of today's computing power. Hence, one aims for methods which reduce the computational cost significantly. This can be achieved by approximating the PDE by a reduced order model of low dimension.

In recent years, major achievements were made in the field of reduced order modeling [55, 6]. In fact, different methods for creating low dimensional models exist for linear systems (e.g. [60]) as well as for nonlinear systems (e.g. [5, 26]), see [4] for a survey. Many researchers focus their attention to the Navier-Stokes equations (e.g. [14, 54, 61]), where *Proper Orthogonal Decomposition* (POD) [32] has proved to be a powerful tool. In the context of fluids, this technique has first been introduced by Lumley [43] to identify coherent structures. Reduced order models using POD modes and the *method of snapshots* go back to Sirovich [58].

Due to the wide spectrum of potential applications (optimal mixing, drag reduction, HVAC (*Heating, Ventilation and Air Conditioning*), etc.) and the progress in computational capabilities, a lot of research is devoted to the control of the Navier-Stokes equations, either directly [28, 38] or via reduced order modeling [25, 20, 35, 9, 8, 49]. In many cases, the energy consumption plays an important role. Thus, ideally, one wants to consider the control cost in addition to the main objective in order to choose a good trade-off between the achievement of the desired objective and the respective control cost. Since this causes a considerable computational effort in case of systems described by the Navier-Stokes equations, we show in this article how reduced order modeling and multiobjective optimal control methods can be combined to solve MOCPs with nonlinear PDE constraints. Using model order reduction via POD and Galerkin projection, we present several methods to compute the Pareto set for the conflicting objectives *minimization of flow field fluctuations* and *control cost* for the two-dimensional flow around a cylinder at $Re = 200$. We discuss the advantages and disadvantages of the different approaches and comment on the respective computational cost. In fact, it is shown that the different methods strongly differ in their respective efficiency and that gradient based approaches have a better performance, provided that the gradient is sufficiently accurate. We also show that the choice of the algorithm and the corresponding particular numerical realization of the control function lead to different optimal control behavior.

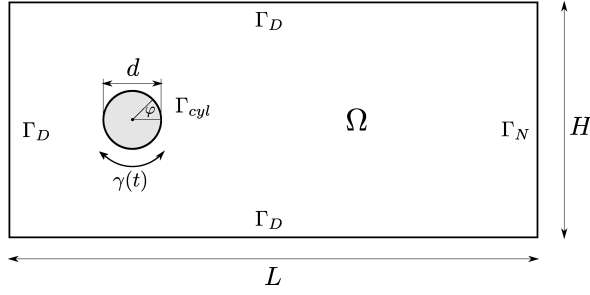


Fig. 1 Sketch of the domain $\Omega \subset \mathbb{R}^2$. The length is $L = 25d$, the height $H = 15d$, and the cylinder center is placed at $(5d, 7.5d)$.

The article is organized as follows. In Section 2 we present the problem setting and formulate the MOCP. After an introduction to multiobjective optimal control in Section 3, the reduced order model and the resulting reduced MOCP are derived in Section 4. We then present our results in Section 5 and draw a conclusion in Section 6.

2 Problem formulation

The two-dimensional, viscous flow around a cylinder is one of the most extensively studied cases for separated flows in general as well as for flow control problems [25, 23, 9]. In this paper, we consider the laminar case described by the incompressible Navier-Stokes equations at a Reynolds Number $Re = \frac{U_\infty d}{\nu} = 200$ computed with respect to the far field velocity U_∞ , the kinematic viscosity ν and the cylinder diameter d :

$$\frac{\partial U(x, t)}{\partial t} + (U(x, t) \cdot \nabla) U(x, t) = -\frac{\nabla p(x, t)}{\rho} + \frac{1}{Re} \nabla^2 U(x, t), \quad (2.1a)$$

$$\nabla \cdot U(x, t) = 0, \quad (2.1b)$$

$$(U(x, 0), p(x, 0)) = (U_0(x), p_0(x)), \quad (2.1c)$$

$$\text{for } x \in \Omega, t \in [t_0, t_e],$$

where $U \in H^2(\Omega \times [t_0, t_e], \mathbb{R}^2)$ is the two-dimensional fluid velocity and $p \in H^1(\Omega \times [t_0, t_e], \mathbb{R})$ the pressure. H^k is the standard Sobolev space $W^{k,2}$ (cf. e.g. [31]). The domain (cf. Figure 1) is denoted by Ω . We impose Dirichlet boundary conditions at the inflow as well as the upper and lower walls Γ_D . At the outflow Γ_N , we impose a standard *no shear stress* condition [29]:

$$U(x, t) = (U_\infty, 0) \quad \text{for } x \in \Gamma_D, \quad (2.1d)$$

$$p(x, t) n = \frac{1}{Re} \frac{\partial U(x, t)}{\partial n} \quad \text{for } x \in \Gamma_N, \quad (2.1e)$$

where $n \in \mathbb{R}^2$ is the outward normal vector of the boundary. On the cylinder Γ_{cyl} , we prescribe a time-dependent Dirichlet BC such that it performs a

rotation around its center with the angular velocity $\gamma(t)$:

$$U(x, t) = \frac{d}{2}\gamma(t) \begin{pmatrix} -\sin(\varphi) \\ \cos(\varphi) \end{pmatrix} \quad \text{for } x \in \Gamma_{cyl}, \quad (2.1f)$$

with φ according to Figure 1. The cylinder rotation $\gamma(t) \in L^2([t_0, t_e], \mathbb{R})$ serves as the control mechanism for the flow. The Hilbert space L^2 is equipped with the inner product $(u, v)_{L^2} = \int_{t_0}^{t_e} u(t) \cdot v(t) dt$ and the norm $\|u\|_{L^2} = \left(\int_{t_0}^{t_e} u(t) \cdot u(t) dt \right)^{1/2}$.

Following [20, 9], we introduce the weak formulation of (2.1a). Consider the Hilbert space $V = \{\psi \in H^1(\Omega \times [t_0, t_e], \mathbb{R}^2) \mid \nabla \cdot \psi = 0\}$ of divergence free test functions. Then, a function $U \in H^1(\Omega \times [t_0, t_e], \mathbb{R}^2)$ which satisfies

$$\begin{aligned} \left(\frac{\partial U}{\partial t} + (U \cdot \nabla) U, \psi \right) &= \left(\frac{p}{\rho}, \nabla \cdot \psi \right) - \left[\frac{p}{\rho} \psi \right] - \frac{1}{Re} (\nabla \psi, (\nabla U)^\top) \\ &\quad + \frac{1}{Re} [(\nabla U)^\top \psi] \end{aligned} \quad (2.2)$$

for all $\psi \in V$ is called a *weak solution* of (2.1a). Here, $[\cdot]$ is the boundary integral (e.g. $[U] = \int_\Gamma U(x) \cdot n dx$) and (\cdot, \cdot) is the inner product for vector-valued quantities (e.g. $(\nabla \psi, (\nabla U)^\top) = \int_\Omega \sum_{i,j} \partial \psi_i / \partial x_j \cdot \partial U_i / \partial x_j dx$). Note that (2.1b) is automatically satisfied by design of the test function space V .

2.1 Finite volume discretization

The system (2.1a – 2.1f) is solved by the software package *OpenFOAM* [36] using a finite volume discretization and the *PISO* scheme [21]. We have chosen *OpenFOAM* because it contains a variety of efficient solvers for various fluid flow applications. Since we will utilize finite elements for the computation of the reduced order model (evaluation of inner products etc.), we then map our solution to a finite element mesh with $N = 17.048$ degrees of freedom (Figure 2(a)). This is done in the spirit of data driven modeling, where we collect data which does not necessarily have to come from a numerical method. Finally, the velocity field can be written in terms of the FEM basis:

$$U(x, t) = \begin{pmatrix} \sum_{j=1}^N U_j^d(t) \phi_j(x) \\ \sum_{j=1}^N U_{j+N}^d(t) \phi_j(x) \end{pmatrix}, \quad (2.3)$$

where $\{\phi_j(x)\}_{j=1}^N$ are the FEM basis functions and $U^d(t) \in \mathbb{R}^{2N}$ are the nodal values of the two velocity components, the superscript d denoting that this is a quantity defined on the grid nodes. In the following, the nodal values of all quantities will be denoted by a superscript d . All finite element related computations are performed with the *FEniCS* toolbox [40] using linear basis functions.

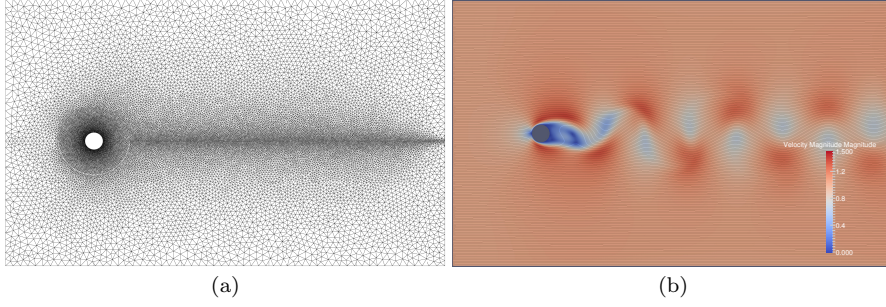


Fig. 2 (a) FEM discretization of the domain Ω by a triangular mesh ($N = 17,048$). (b) A snapshot of the solution to (2.1a) – (2.1f) for a non-rotating cylinder ($\gamma(t) = 0$), the coloring is according to the velocity magnitude. The pattern is the well-known *von Kármán vortex street*.

For a non-rotating cylinder, i.e. $\gamma(t) = 0$, the system possesses a periodic solution, the well-known *von Kármán vortex street* (Figure 2(b)), where vortices separate alternately from the upper and lower surface of the cylinder, respectively. The effect is observed frequently in nature and is one of the most studied phenomena in fluid mechanics, also in the context of flow control, where the objective is to stabilize the flow and to reduce the drag.

2.2 Multiobjective optimal control problem

In many applications, the control cost is of great interest. This is immediately clear when the goal of the optimization is to save energy such that in this case, the control effort needs to be taken into account. In scalar optimization problems, this is often done by adding an additional term of the form $\beta \int_{t_0}^{t_e} \gamma^2(t) dt$ to the cost functional where $\beta \in \mathbb{R}_{\geq 0}$ is a weighting parameter. Here, we want to consider the two objectives flow stabilization, i.e. the minimization of the fluctuations $u(x, t) = U(x, t) - \langle U(x) \rangle$ around the mean flow field $\langle U(x) \rangle = \frac{1}{T} \int_0^T U(x, t) dt$, and the minimization of the control cost separately which leads to the following multiobjective optimal control problem:

$$\min_{U, \gamma} \hat{J}(U, \gamma) = \min_{U, \gamma} \left(\frac{\int_{t_0}^{t_e} \|u(\cdot, t)\|_{L^2}^2 dt}{\|\gamma\|_{L^2}^2} \right),$$

where $\hat{J} : H^2(\Omega \times [t_0, t_e], \mathbb{R}^2) \times L^2([t_0, t_e], \mathbb{R}) \rightarrow \mathbb{R}^2$ and $U(x, t)$ satisfies (2.1a – 2.1f). In contrast to bounded domains (cf. [22]), the proof of existence of a solution is an open problem for cases with *no-shear* or *do nothing* boundary conditions [50]. Nevertheless, based on numerical experiences [50], we will from now on assume that there exists a unique solution $U(x, t)$ for each $\gamma(t)$ and hence, we denote by $U(\gamma)$ the solution $U(x, t)$ for a fixed $\gamma \in L^2([t_0, t_e], \mathbb{R})$ and consider the reduced cost functional $J : L^2([t_0, t_e], \mathbb{R}) \rightarrow \mathbb{R}^2$ which leads

to the following multiobjective optimal control problem:

$$\min_{\gamma} J(\gamma) = \min_{\gamma} \left(\int_{t_0}^{t_e} \frac{\|u\|_{L^2}^2}{\|\gamma\|_{L^2}^2} dt \right). \quad (\text{MOCP})$$

In general, the solution to this problem does not consist of isolated points but a set of *optimal compromises* between the two objectives. In the following section, we give a short introduction to multiobjective optimal control theory and solution methods.

3 Multiobjective Optimal Control

This section is concerned with the treatment of general multiobjective optimal control problems. We will give a short introduction to the general theory before addressing the two algorithms used later on in combination with model order reduction techniques.

3.1 Theory of multiobjective optimal control

Consider the general multiobjective optimal control problem:

$$\min_{\gamma} J(\gamma) = \min_{\gamma} \begin{pmatrix} J_1(\gamma) \\ \vdots \\ J_k(\gamma) \end{pmatrix}, \quad (1)$$

with $J : L^2([t_0, t_e], \mathbb{R}) \rightarrow \mathbb{R}^k$ and $J_i : L^2([t_0, t_e], \mathbb{R}) \rightarrow \mathbb{R}$, $i = 1, \dots, k$. The space in which the control functions live is denoted as the *decision space* and the function J is a mapping to the k -dimensional *objective space*. The set of feasible functions γ is the *feasible set in decision space*. We denote its image as the *feasible set in objective space* which consists of the *feasible points* $J(\gamma)$. In contrast to single objective optimization problems, there exists no total order of the objective function values in \mathbb{R}^k , $k \geq 2$. Therefore, the comparison of values is defined in the following way [44]:

Definition 1 Let $v, w \in \mathbb{R}^k$. The vector v is *less than* w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \dots, k\}$. The relation \leq_p is defined in an analogous way.

A consequence of the lack of a total order is that we cannot expect to find isolated optimal points. Instead, the solution to (1) is the set of optimal compromises, the so-called *Pareto set* named after Vilfredo Pareto:

Definition 2

- (a) A function γ^* *dominates* a function γ , if $J(\gamma^*) \leq_p J(\gamma)$ and $J(\gamma^*) \neq J(\gamma)$.
- (b) A feasible function γ^* is called *(globally) Pareto optimal* if there exists no feasible function γ dominating γ^* . The image $J(\gamma^*)$ of a (globally) Pareto optimal function γ^* is a *(globally) Pareto optimal point*.

- (c) The set of nondominated feasible functions is called the *Pareto set*, its image the *Pareto front*.

Consequently, for each function that is contained in the Pareto set (cf. the red line in Figure 3(a)), one can only improve one objective by accepting a trade-off in at least one other objective. Figuratively speaking, in a two-dimensional problem, we are interested in finding the "lower left" boundary of the feasible set in objective space (cf. Figure 3(b)). More detailed introductions to multiobjective optimization can be found in [44, 19].

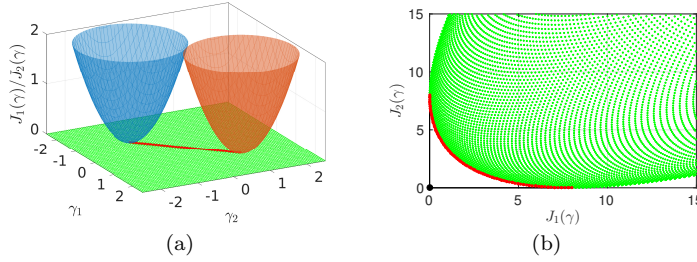


Fig. 3 Red lines: Sketch of a Pareto set (a) and Pareto front (b) of the multiobjective optimization problem $\min_{\gamma \in \mathbb{R}^2} J(\gamma)$, $J : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. The objectives are to minimize the two functions represented by the blue and the orange paraboloid, respectively, where the third axis represents the function values.

3.2 Solution methods

Various methods exist to solve problem (1). In this work, we present two approaches that are fundamentally different. The first one is a reference point method [53] for which the distance between a feasible point (i.e. an objective value that lies in the feasible set in objective space) and an infeasible target in objective space is minimized. The method yields a moderate number of single objective optimization problems that are solved consecutively. The second approach is a global, derivative free subdivision algorithm [17] for which the Pareto set is approximated by a nested sequence of increasingly refined box coverings. However, its applicability depends critically on both the decision space dimension and the numerical effort of function evaluations. The main properties are summarized in Table 1.

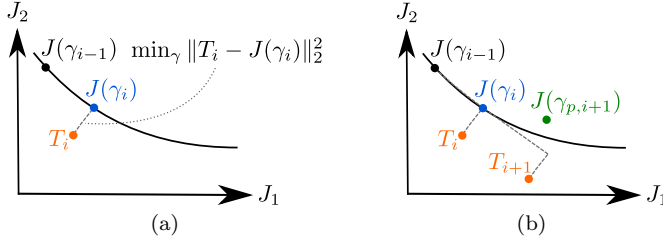
3.2.1 Reference point method

The reference point method presented here belongs to the category of scalarization techniques for which the solution set of (1) is approximated by a finite set of points, each computed by solving a scalar optimization problem. In the beginning, one Pareto optimal point γ_0 has to be known. This can be

Table 1 Most important properties of the two algorithms presented in this section.

	Reference point method	Subdivision algorithm
Optimality	local	global
Gradients	yes	no
Solution concept	Boundary of feasible set of $J(\gamma)$	Nondominated subsets
Parameter Dim.	high	moderate

achieved by solving a scalar optimization problem for some weighted sum of all objectives (i.e. $\min_{\gamma_0} sJ_1(\gamma_0) + (1-s)J_2(\gamma_0)$, $s \in [0, 1]$ for the case of two objectives), including the scalar optimization with respect to any of the objectives of (MOCP). Then, a so-called *target* $T_1 \in \mathbb{R}^k$ is chosen such that it lies outside the feasible set in objective space, e.g. by shifting the solution of the first Pareto point ($T_1 = J(\gamma_0) - (h_{\parallel}, 0, \dots, 0)^{\top}$, $h_{\parallel} > 0$). We then solve the scalar optimization problem $\min_{\gamma_1} \|T_1 - J(\gamma_1)\|_2^2$. As a result, the corresponding optimal point $J(\gamma_1)$ lies on the boundary of the feasible set and it is not possible to further improve all objectives at the same time. Thus, the point is (locally) Pareto optimal. By adjusting the target position based on targets and Pareto points already known, multiple points on the Pareto front (i.e. $J(\gamma_2), J(\gamma_3), \dots$) are computed recursively (cf. Figure 4 for an illustration). For J being continuous, the change in decision space is small when the target position changes only slightly [30] and hence, the current solution is a good initial guess for the next scalar problem which accelerates the convergence considerably.

**Fig. 4** Reference point method in image space. (a) Determination of the i -th point on the Pareto front by solving a scalar optimization problem. (b) Computation of new target point T_{i+1} and predictor step in decision space ($\gamma_{p,i+1}$, cf. line 16 in Algorithm 1).

Remark 1 The distance between two consecutive targets can be influenced by the choice of the parameter h_{\parallel} . A larger value of h_{\parallel} increases the distance between two points on the Pareto front and hence decreases the numerical effort. On the other hand, a smaller value of h_{\parallel} is advisable in regions where the Pareto front possesses a strong curvature. In practice, h_{\parallel} can be chosen by computing the outermost points (i.e. the scalar minima) on the Pareto front. The desired number of points on the Pareto front then allows us to

Algorithm 1 (Reference point method for $J \in \mathbb{R}^2$)

Require: Initial solution γ_0 , step length parameters $h_{\parallel}, h_{\perp}, h_p > 0$, index $i = 0$

```

1: Compute the first target point  $T_1 = J(\gamma_0) - (h_{\parallel}, 0)^{\top}$ .
2: for  $run = 1, 2$  do
3:   loop
4:      $i = i + 1$ 
5:     Solve scalar optimization problem  $\min_{\gamma_i} \|T_i - J(\gamma_i)\|_2^2$ 
6:     if extremal point of Pareto front is passed ( $J_{run}(\gamma_i) > J_{run}(\gamma_{i-1})$ ) then
7:       if  $run = 1$  (first direction is completed) then
8:          $\gamma_{p,i+1} = \gamma_0$  (Go back to the initial solution)
9:          $T_{i+1} = J(\gamma_0) - h_{\parallel} \frac{J(\gamma_1) - J(\gamma_0)}{\|J(\gamma_1) - J(\gamma_0)\|_2} + h_{\perp} \frac{T_1 - J(\gamma_1)}{\|T_1 - J(\gamma_1)\|_2}$  (Opposite direction)
10:        break
11:      else
12:        STOP
13:      end if
14:    else
15:       $T_{i+1} = J(\gamma_i) + h_{\parallel} \frac{J(\gamma_i) - J(\gamma_{i-1})}{\|J(\gamma_i) - J(\gamma_{i-1})\|_2} + h_{\perp} \frac{T_i - J(\gamma_i)}{\|T_i - J(\gamma_i)\|_2}$ 
16:       $\gamma_{p,i+1} = \gamma_i + h_p (\gamma_i - \gamma_{i-1})$  (Predictor step)
17:    end if
18:  end loop
19: end for

```

determine an appropriate value for h_{\parallel} . Alternatively, h_{\parallel} can be adjusted during the computation in order to take into account the local curvature of the Pareto front (cf. [3] for a detailed introduction of this concept).

Theoretically, the method is not restricted to low objective space dimensions. However, setting the targets properly to get a proper approximation of the Pareto front (i.e. an approximation of the entire front by evenly distributed points) becomes complicated in higher dimensions. In our case, we are dealing with two objectives and the targets can be determined easily using linear extrapolation (cf. Figure 4 and Algorithm 1) as proposed in [53]. This also allows us to compute the whole front in at most two search directions ($run = 1, 2$, line 2 in Algorithm 1). From the initial point, we first proceed in one direction, e.g. decreasing J_1 . When, at some point, J_1 is increasing again (line 6 in Algorithm 1), we have reached the *extremal point* of the feasible set (cf. Figure 3(b)) and the scalar optimum of J_1 . We then return to the initial point and proceed in the opposite direction (lines 8, 9 in Algorithm 1) until the other extremal point is reached.

Remark 2 Using system knowledge, the algorithm can be further simplified. In the case of (MOCP), for example, we know that $\gamma(t) = 0$ is Pareto optimal since it is the scalar minimum of $J_2(\gamma)$ and therefore has the lowest possible value of J_2 . Knowing this, we do not need to solve the initial optimization problem and moreover, we only need to move along the front in one direction until the other extremal point is reached.

The scalar optimization problems can be solved using any suitable method. Here, we use a line search approach [46] and compute the derivatives using an

adjoint approach. Note that since the scalar optimization routine often is of local nature, the method overall is also local and depends on the initial guesses as well as the choice of the target points. However, under certain smoothness assumptions, the Pareto set is a $(k - 1)$ -dimensional manifold [30] such that once the first point is computed correctly, the method is promising to find the globally optimal Pareto set for many problems.

Remark 3 For the reference point method, it turned out to be numerically beneficial that $(T_i - J_i)/(T_j - J_j) = \mathcal{O}(1) \forall i, j \in \{1, \dots, k\}$. Otherwise, the computation of new target points may become sensitive to the step length parameters.

3.2.2 Global subdivision algorithm

When the image space dimension increases or, alternatively, the Pareto front is disconnected (see e.g. [15]), continuation methods like the reference point method may fail. Moreover, the resulting scalar optimization problems are often solved by algorithms of local nature which may also not be sufficient if global optima are desired. In addition to that, derivatives are hard to compute or not available at all in many applications. The subdivision algorithm presented here overcomes these problems. It is described in more detail in [17] including a proof of convergence. The version using gradient information is based on concepts for the computation of attractors of dynamical systems [16], it is however not considered here. Instead, we focus on the derivative free approach. It is very robust and since we directly utilize the concept of dominance when comparing solutions, we theoretically do not need to make any assumptions about the problem except that the control is real-valued, i.e. $\gamma \in \mathbb{R}^m$. In the numerical realization however, we approximate sets by sample points and hence, we further require that the objective J is a continuous function of γ .

The subdivision algorithm (cf. Algorithm 2 and Figure 5) is a set oriented method where the Pareto set is approximated by a nested sequence of increasingly refined box coverings (cf. Figure 5). During the algorithm, a sequence of box collections $\{\mathcal{B}_i\}_{i=0, \dots}$ covering the Pareto set is constructed, starting with a sufficiently large initial set $\mathcal{B}_0 \subset \mathbb{R}^m$. (This results in box constraints for the control γ .) In each iteration, the algorithm performs the steps *subdivision* and *selection* until a given precision criterion is satisfied. This way, a subset of the previous box collection remains that is a closer covering of the desired set. In the selection step, the boxes are compared pairwise and all boxes that are dominated by another box are eliminated from the collection. Numerically, this is realized by a representative set of sampling points in each box (cf. Figure 5(a)), at which the objectives J_i are evaluated. In this case, we say that a box is dominated if all sampling points are dominated by at least one point from another box. The remaining boxes are then subdivided into two boxes of half the size and we proceed with the next selection step. The subdivision step is performed cyclically in the decision space dimensions 1 to m . Hence,

in order to divide each dimension q times, we require $N_{sub} = mq$ subdivision steps in total.

Algorithm 2 (Global, derivative free subdivision algorithm)

Require: Box constraints $\gamma^{min}, \gamma^{max} \in \mathbb{R}^m$, number of subdivision steps N_{sub}

- 1: Create initial box collection \mathcal{B}_0 defined by the constraints $\gamma^{min}, \gamma^{max}$, i.e. $\mathcal{B}_0 = B = [\gamma_1^{min}, \gamma_1^{max}] \times \dots \times [\gamma_m^{min}, \gamma_m^{max}]$
- 2: **for** $i = 1, \dots, N_{sub}$ **do**
- 3: Subdivision step: $\bigcup_{B \in \hat{\mathcal{B}}_i} B = \bigcup_{B \in \mathcal{B}_{i-1}} B$ such that $\max_{B \in \hat{\mathcal{B}}_i} (\text{diam}(B)) < \Theta \max_{B \in \mathcal{B}_{i-1}} (\text{diam}(B))$, $\Theta \in (0, 1)$
- 4: Insert S sampling points $\gamma_1, \dots, \gamma_S$ each box and evaluate the cost functional $J(\gamma_s), s = 1, \dots, S$
- 5: Selection step: Eliminate all boxes that contain only dominated points: $\mathcal{B}_i = \left\{ B \in \hat{\mathcal{B}}_i \mid \text{There exists no } \gamma^* \in \bigcup_{\hat{B} \in \hat{\mathcal{B}}_i \setminus B} \hat{B} \text{ such that } J(\gamma^*) \leq_p J(\gamma) \forall \gamma \in B \right\}$
- 6: **end for**

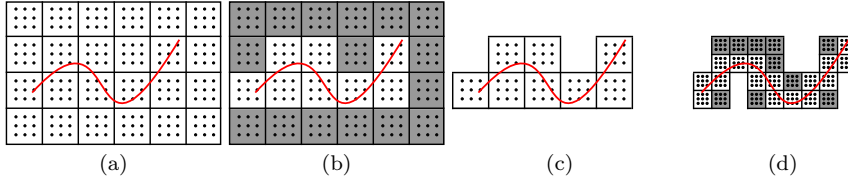


Fig. 5 Global subdivision algorithm, example with $\gamma \in \mathbb{R}^2$. (a) Sampling. (b) Nondominance test. (c) Elimination of dominated boxes. (d) Subdivision, sampling and nondominance test.

Since the applicability of the subdivision algorithm depends critically on both the decision space dimension m and the numerical effort of function evaluations, it is in practice limited to low decision space dimensions. Therefore, we introduce a sinusoidal control, i.e. $\gamma(t) = A \sin(2\pi\omega t + \tau)$. The choice is motivated by the fact that the uncontrolled dynamics of the problem at hand are also periodic and we include a phase shift to allow for controls with non-zero initial conditions. This way, we transform (MOCP) into an MOP with $\tilde{J}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$:

$$\min_{A, \omega, \tau \in \mathbb{R}} \tilde{J}(A, \omega, \tau) = \min_{A, \omega, \tau \in \mathbb{R}} \left(\int_{t_0}^{t_e} \|u\|_{L^2}^2 dt, \int_{t_0}^{t_e} (A \sin(2\pi\omega t + \tau))^2 dt \right). \quad (\text{MOP-3D})$$

4 Reduced Order Modeling

The problem (MOCP) could now be solved using either of the methods presented in Section 3.2. However, both require a large number of evaluations of the cost functional and consequently, evaluations of the system (2.1a – 2.1f).

A finite element or finite volume discretization yields a large number of degrees of freedom such that solving (MOCP) quickly becomes computationally infeasible. Hence, we need to reduce the cost for solving the dynamical system significantly. This is achieved by means of reduced order modeling where we replace the state equation by a reduced order model of coupled, nonlinear ordinary differential equations that can be solved much faster.

4.1 Reduced Order Model

The standard procedure for deriving a reduced order model is by introducing a Galerkin ansatz [32]:

$$U(x, t) = \sum_{j=1}^l \alpha_j(t) \psi_j(x), \quad (x, t) \in \Omega \times [t_0, t_e], \quad (2)$$

where $\alpha(t)$ are time-dependent coefficients and $\psi_j(x)$ are basis functions. In contrast to FEM, these basis functions may in general have full support, i.e. they are global. The objective is now to find a basis as small as possible ($l \ll N$) while allowing for a good approximation of the flow field.

Using (2), we can derive a reduced model of dimension l . By selecting a basis that is divergence-free, the mass conservation equation (2.1b) is automatically satisfied. (When computing this basis using data from a divergence-free flow field, the resulting basis elements are also divergence-free [10].) Inserting (2) into the weak formulation of the Navier-Stokes equations (2.2) yields a system of l ordinary differential equations of the form

$$\dot{\alpha}(t) = \tilde{\mathcal{B}}\alpha(t) + \tilde{\mathcal{C}}(\alpha(t)),$$

where the coefficient matrices are computed by evaluating the L^2 inner products:

$$\begin{aligned} \left(\tilde{\mathcal{B}}\right)_{ij} &= \frac{1}{Re} (\nabla \psi_i, \nabla \psi_j)_{L^2}, \\ \left(\tilde{\mathcal{C}}(\alpha(t))\right)_j &= \alpha(t)^\top \mathcal{Q}_j \alpha(t), \\ \text{with } (\mathcal{Q}_j)_{ik} &= ((\psi_i \cdot \nabla) \psi_k, \psi_j)_{L^2}. \end{aligned}$$

Remark 4 In the above as well as all following formulations, we have neglected the influence of the pressure term. In [45], it was shown that including the pressure term is favorable for the accuracy of the reduced model for open systems. Since we consequently perform a model stabilization in order to increase the accordance with the flow field data, this results in a further modification of the model coefficients. Consequently, we have (following [13]) decided to neglect the influence of the pressure term in this work.

This model, however, is not controllable in the sense that we can no longer influence the solution by choosing $\gamma(t)$. Moreover, the basis functions need to be computed from data with homogeneous boundary conditions ($U(x, t) = 0$ for $x \in \Gamma$) such that they also possess homogeneous boundary conditions [58]. Otherwise, it cannot be guaranteed that the Galerkin ansatz (2) satisfies the PDE boundary conditions. The idea is therefore to introduce the Galerkin expansion for a modified flow field $u(x, t)$ which satisfies homogeneous boundary conditions for all $t \in [t_0, t_e]$. In order to realize this, we introduce the following flow field decomposition [51, 9]:

$$U(x, t) = \langle U(x, t) \rangle + \gamma(t)U_c(x) + u(x, t), \quad (3)$$

where $U_c(x)$ is a so-called *control function* which is a solution to (2.1a), (2.1b), (2.1c) with a constant cylinder rotation γ_c and zero boundary conditions elsewhere. The choice of the rotating velocity for the computation of $U_c(x)$ is arbitrary but has an influence on the resulting reduced model such that the model performance might be influenced by this choice. Since the main intention of this article is to develop multiobjective optimal control algorithms for PDE-constrained problems, we do not address this problem further and choose $\gamma(t) = \gamma_c = 2$ such that the cylinder surface rotates with a velocity of 1. The decomposition is now computed in two steps:

$$\begin{aligned} \tilde{U}(x, t) &= U(x, t)|_{\gamma(t)=\gamma_{ref}(t)} - \gamma_{ref}(t)U_c(x), \\ u(x, t) &= \tilde{U}(x, t) - \langle \tilde{U}(x, t) \rangle, \end{aligned}$$

where γ_{ref} is the *reference control* for which the data was collected.

In this way, $u(x, t) = \sum_{j=1}^l \alpha_j(t)\psi_j(x)$ satisfies homogenous boundary conditions for all $t \in [t_0, t_e]$. (Note that this does not hold exactly on Γ_N , however, the resulting error is small and hence neglected [9].) Inserting (3) into (2.2) yields an extended reduced order model:

$$\dot{\alpha}(t) = \mathcal{A} + \mathcal{B}\alpha(t) + \mathcal{C}(\alpha(t)) + \mathcal{D}\dot{\gamma}(t) + (\mathcal{E} + \mathcal{F}\alpha(t))\gamma(t) + \mathcal{G}\gamma^2(t), \quad (4)$$

$$\alpha_j(t_0) = (u(\cdot, t_0), \psi_j)_{L^2} =: \alpha_{0,j}, \quad (5)$$

where $\alpha \in H^1([t_0, t_e], \mathbb{R}^l)$. The expressions for the coefficient matrices \mathcal{A} to \mathcal{G} are given in Appendix A, their size depends on the dimension l of the ROM: $\mathcal{A}, \mathcal{D}, \mathcal{E}, \mathcal{G} \in \mathbb{R}^l$; $\mathcal{B}, \mathcal{F} \in \mathbb{R}^{l,l}$; $(\mathcal{C}(\alpha(t)))_j = \alpha(t)^\top \mathcal{Q}_j \alpha(t)$ with $\mathcal{Q} \in \mathbb{R}^{l,l,l}$.

Remark 5 Since the time derivative of the control occurs in (4), we require higher regularity, i.e. $\gamma \in H^1([t_0, t_e], \mathbb{R})$.

Due to the truncation of the POD basis, the higher modes covering the small scale dynamics, i.e. the dynamics responsible for dissipation, are neglected. This can lead to incorrect long term behavior and hence, the model needs to be modified to account for this. Several researchers have addressed this problem and proposed different solutions, see e.g. [52, 57, 45, 13, 7]. Here, we achieve stabilization of the model (4) by solving an augmented optimization

problem [23]. In this approach the entries of the matrix \mathcal{B} are manipulated in order to minimize the difference between the ROM solution and the projection of the data onto the reduced basis ($\alpha_j^p(t) = (U(\cdot, t), \psi_j)_{L^2}$). The resulting initial value problem is solved using a Runge-Kutta method of fourth order.

Finally, we replace the objectives in (MOCP) by their equivalents in the reduced order model. Since the reference point method is more stable for objectives of the same order of magnitude (cf. Remark 3), we additionally multiply the second component by l (i.e. the number of basis functions)

$$\min_{\gamma} \left(\int_{t_0}^{t_e} \int_{\Omega} \left\| \sum_{i=1}^l \alpha_i(t) \psi_i(x) \right\|_2^2 dx dt, \quad l \int_{t_0}^{t_e} \gamma^2(t) dt \right). \quad (6)$$

4.2 Proper orthogonal decomposition (POD)

In order to achieve high numerical efficiency, we want to compute a basis as small as possible. To this end, we utilize the well-known Proper Orthogonal Decomposition (POD). Using POD, we can compute the orthonormal basis $\{\psi_i(x)\}_{i=1}^l$ which, for every l , optimally represents a set of flow field realizations $\{u(x, t_1), \dots, u(x, t_m)\}$ [58]. Basically, the computation is realized via a singular value decomposition of the so-called *snapshot matrix* $S \in \mathbb{R}^{2N, m}$, $S = [u^d(t_1), \dots, u^d(t_m)]$, where we collect $2N$ measurements (i.e. the velocity components at N nodes) at m different time instants. If $m \ll N$, as it is often the case for FEM discretizations, it is more efficient to solve the m -dimensional eigenvalue problem [37, 20]:

$$S^T M S v_i = \sigma_i v_i, \quad i = 1, \dots, m, \quad (7)$$

where $M \in \mathbb{R}^{2N, 2N}$ is the finite element mass matrix. Using the eigenvalues and eigenvectors from (7), the POD modes can be computed by making use of the FEM basis functions $\{\phi_j(x)\}_{j=1}^N$:

$$\psi_i^d = \frac{1}{\sqrt{\sigma_i}} S v_i, \\ \psi_i(x) = \left(\frac{\sum_{j=1}^N \psi_{i,j}^d \phi_j(x)}{\sum_{j=1}^N \psi_{i,j+N}^d \phi_j(x)} \right).$$

The eigenvalues σ are a measure for the information contained in the respective modes and hence, the error between the Galerkin ansatz and the snapshot data is determined by the ratio of the truncated eigenvalues [58], i.e. $\epsilon(l) = \sum_{j=1}^l \sigma_j / \sum_{j=1}^m \sigma_j$. By choosing a value for ϵ , typically 0.99 or 0.999, the basis size can be determined. For many applications, the eigenvalues decay fast such that a truncation to a small basis is possible.

The error ϵ is only known for the reference control at which the data was collected. If one allows for multiple evaluations of the finite element solution, the reduced model can be updated when necessary. This is realized using,

e.g., trust-region methods [20, 8]. An alternative approach is to use a reference control that yields sufficiently rich dynamics such that the model can be expected to be valid for a larger variety of controls [9]. Since it is computationally cheaper, we follow the second approach and take 1201 snapshots in the interval $[0, 60]$ with $\Delta t = 0.05$ for a *chirping* reference control (cf. Figure 6). This leads to a basis of size $l = 38$ for $\epsilon(l) \geq 99\%$. Figure 7 depicts the comparison between the uncontrolled solution (i.e. $\gamma(t) = 0$) of the reduced model and the projection of the finite volume solution for reduced models computed with a reference control $\gamma_{ref} = 0$ and $\gamma_{ref} = \gamma_{chirp}$, respectively.

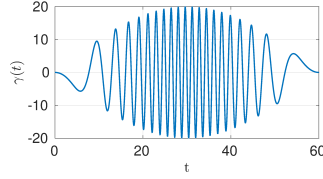


Fig. 6 Chirping function ($\gamma_{chirp} = -4 \sin(2\pi t/120) \cos(2\pi t/3 - 18 \sin(2\pi t/60))$) used for the computation of the snapshot matrix [9].

Figure 8 shows the first four POD modes for the uncontrolled solution. The modes occur in pairs, the second one slightly shifted downstream. This is due to symmetries (see the double eigenvalues in Figure 9(a)) in the problem, namely in the horizontal axis through the cylinder as well as on the upper and lower boundary, respectively. The first four modes already account for $\approx 98\%$ of the information, cf. Figure 9, where in (a) the eigenvalues are shown for both the uncontrolled flow and the flow controlled by the chirping function. The error, i.e. the ratio of the truncated eigenvalues, is visualized in (b).

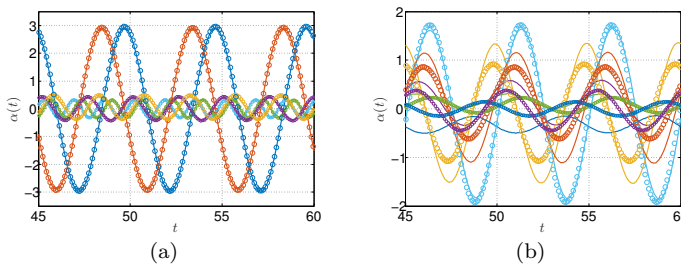


Fig. 7 First six modes of (4), (5) (—) and projection of FEM solution (o) for two reduced order models with different reference controls. (a) $\gamma_{ref} = 0$. (b) $\gamma_{ref} = \gamma_{chirp}$.

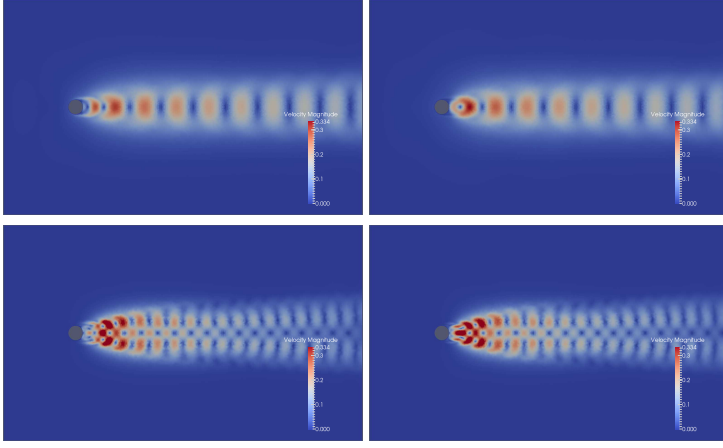


Fig. 8 The first four POD modes for an uncontrolled solution.

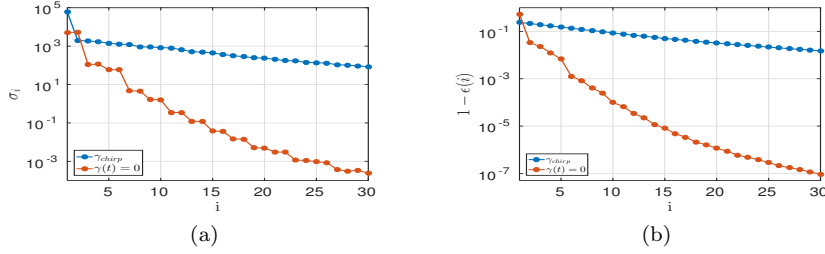


Fig. 9 Eigenvalues (a) and error at current basis size (b) for an uncontrolled solution and a solution controlled by the chirping function.

Making use of the orthonormality of the POD basis, the first objective in (6) can be simplified:

$$\begin{aligned}
 \int_{\Omega} \left\| \sum_{i=1}^l \alpha_i(t) \psi_i(x) \right\|_2^2 dx &= \int_{\Omega} \left(\sum_{i=1}^l \alpha_i(t) \psi_i(x) \right) \cdot \left(\sum_{i=1}^l \alpha_i(t) \psi_i(x) \right) dx \\
 &= \sum_{i=1}^l \alpha_i^2(t) \\
 &\text{with } (\psi_i, \psi_j)_{L^2} = \delta_{i,j},
 \end{aligned}$$

which allows us to replace (6) by a simpler formulation:

$$\min_{\gamma} J(\alpha, \gamma) = \min_{\gamma} \left(\begin{array}{c} \int_{t_0}^{t_e} \sum_{i=1}^l \alpha_i^2(t) dt \\ l \int_{t_0}^{t_e} \gamma^2(t) dt \end{array} \right). \quad (\text{R-MOCP})$$

4.3 Adjoint systems

We would like to use a gradient based algorithm for the scalar optimization problems occurring in the reference point method (Section 3.2.1). We realize this with an adjoint approach. The first step is to transform (R-MOCP) into a single objective optimization problem where the euclidean distance to a target T is to be minimized:

$$\begin{aligned} \min_{\gamma} \bar{J}(\alpha, \gamma) &= \min_{\gamma} \|T - J(\alpha, \gamma)\|_2^2, & (\text{SOP}) \\ \text{with} \quad \bar{J}(\alpha, \gamma) &= (T_1 - J_1(\alpha, \gamma))^2 + (T_2 - J_2(\alpha, \gamma))^2 \\ &= \left(T_1 - \int_{t_0}^{t_e} \sum_{i=1}^l \alpha_i^2(t) dt \right)^2 + \left(T_2 - l \int_{t_0}^{t_e} \gamma^2(t) dt \right)^2, & (8) \end{aligned}$$

where $J(\alpha, \gamma)$ according to (R-MOCP) and $\alpha(t)$ satisfies the reduced order state equation (4), (5).

The task is now to solve a sequence of scalar optimization problems with varying targets T . We choose a line search strategy [46] to address (SOP), where the direction is computed with a conjugate gradient method and the step length via backtracking and a valid length has to satisfy the Armijo rule. The necessary gradient information is computed with an adjoint approach. To this end, we introduce the adjoint state $\lambda \in H^1([t_0, t_e], \mathbb{R}^l)$ and the Lagrange functional:

$$\begin{aligned} L(\alpha, \lambda, \gamma) &= \int_{t_0}^{t_e} \bar{J}(\alpha, \gamma) - \lambda^\top (\dot{\alpha} - \mathcal{A} - \mathcal{B}\alpha - \mathcal{C}(\alpha) - \mathcal{D}\dot{\gamma} \\ &\quad - (\mathcal{E} + \mathcal{F}\alpha)\gamma - \mathcal{G}\gamma^2) dt, & (9) \end{aligned}$$

which is stationary for optimal values of $\gamma(t)$ and the corresponding state $\alpha(t)$ and adjoint state $\lambda(t)$. Using integration by parts, this leads to the following system of equations¹:

$$\dot{\alpha} = \mathcal{A} + \mathcal{B}\alpha + \mathcal{C}(\alpha) + \mathcal{D}\dot{\gamma} + (\mathcal{E} + \mathcal{F}\alpha)\gamma + \mathcal{G}\gamma^2, \quad (10)$$

$$\dot{\lambda} = -\frac{\partial \bar{J}}{\partial \alpha} - \left(\mathcal{B} + \frac{\partial \mathcal{C}(\alpha)}{\partial \alpha} + \mathcal{F}\gamma \right)^\top \lambda, \quad (11)$$

$$0 = \frac{\partial \bar{J}}{\partial \gamma} + (\mathcal{E} + \mathcal{F}\alpha + 2\mathcal{G}\gamma)^\top \lambda - \mathcal{D}^\top \dot{\lambda} =: D_\gamma \bar{J}, \quad (12)$$

with the Fréchet derivatives $\partial \bar{J} / \partial \alpha_j = 4 \left(\int_{t_0}^{t_e} \sum_{i=1}^l \alpha_i^2(t) dt - T_1 \right) \alpha_j(t)$ and $\partial \bar{J} / \partial \gamma = 4l \left(l \int_{t_0}^{t_e} \gamma^2(t) dt - T_2 \right) \gamma(t)$ and the respective boundary conditions

¹ Note, that since this approach makes use of variational calculus, it is formally only applicable in the case of stronger assumptions, namely for $\alpha, \lambda, \gamma \in C^1$ (since we apply integration by parts).

$\alpha(0) = \alpha_0$ and $\lambda(t_e) = 0$. A detailed derivation of the optimality system is given in Appendix B.

Equations (10) – (12) form the so-called optimality system which is seldom solved explicitly. Instead, the state equation (10) and the adjoint equation (11) are solved by forward and backward integration, respectively. The optimality condition (12) provides the derivative information $D_\gamma \bar{J}$ of the cost functional with respect to the control γ . This information can be used to improve an initial guess for the optimal control within an iterative optimization scheme as described in Algorithm 3.

Algorithm 3 (Adjoint based scalar optimization)

Require: Target point T , initial guess $\gamma^{(0)}$

- 1: **for** $i = 0, \dots$ **do**
 - 2: Compute $\alpha^{(i)}$ by solving the reduced order model (10) with the control $\gamma^{(i)}$
 - 3: Solve the adjoint equation (11) in backwards time with $\gamma^{(i)}$, $\alpha^{(i)}$, $\partial \bar{J} / \partial \alpha^{(i)}$
 - 4: Compute the gradient $D_\gamma \bar{J}^{(i)}$ by evaluating the optimality condition (12) with $\gamma^{(i)}$, $\alpha^{(i)}$ and $\lambda^{(i)}$
 - 5: Update the control: $\gamma^{(i+1)} = \gamma^{(i)} + a^{(i)} d^{(i)}$, where $a^{(i)}$ is the step length computed by the backtracking Armijo method [46] and $d^{(i)}$ is the conjugate gradient direction $d^{(i)} = -D_\gamma \bar{J}^{(i)} + \beta^i d^{(i-1)}$, $d^{(0)} = D_\gamma \bar{J}^{(0)}$ with $\beta^{(i)} = \frac{D_\gamma \bar{J}^{(i)} \cdot D_\gamma \bar{J}^{(i)}}{D_\gamma \bar{J}^{(i-1)} \cdot D_\gamma \bar{J}^{(i-1)}}$
 - 6: **end for**
-

In [9], the condition $\lambda(t_e) = 0$ is the only boundary condition for the adjoint equation and the condition $\lambda^\top(t_0) \mathcal{D} \delta \gamma(t_0) = 0$ (cf. Appendix B) is neglected. In our computations, using this optimality system caused convergence issues. The reason for this is that the gradient obtained by the adjoint approach appears to be wrong, cf. Figure 10(a) for a comparison to a finite difference approximation. (Since the state equation is very accurate, the gradient can be approximated well this way.) The reason for this might be that we neglect the additional condition for the adjoint equation. However, it has been reported before [20, 18] that it is favorable to include information of the adjoint state of the PDE in the reduced order model. In fact, the adjoint based gradient can easily become inaccurate or even incorrect if based on information about the state only. Consequently, it is difficult to say whether the convergence problems stem from the missing boundary condition or are a flaw of the model reduction approach itself. To further investigate this, we derive a second optimality system based on an alternative formulation of the reduced model. Following [51], we define an augmented state $(\alpha, \gamma)^\top$ and introduce a new control $v \in L^2([t_0, t_e], \mathbb{R})$, $v(t) = \dot{\gamma}(t)$. It is known from optimal control theory that if a dynamical system depends linearly on the control, the control is bounded by box constraints and does not appear in the cost functional, the optimal control is of *bang bang* type and might include singular arcs [24]. This is exactly the situation for the second formulation and since the computation of such solutions is numerically challenging, we avoid this behavior by adding a regularization term to the second objective in (R-MOCP),

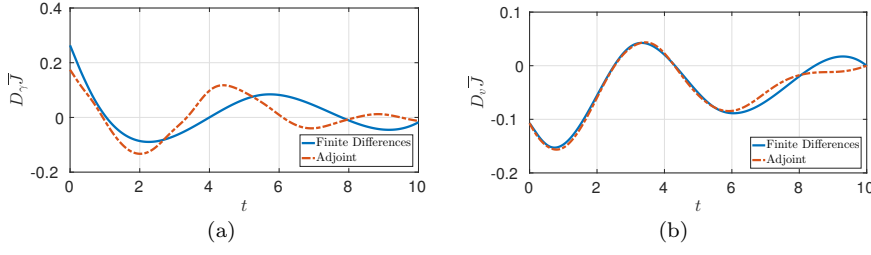


Fig. 10 Comparison of the gradient computed using the adjoint approaches with an approximation by finite differences (FD). (a) Optimality system (10) – (12) (FD: $D_{\gamma,i}\bar{J} = (\bar{J}(\gamma + he_i) - \bar{J}(\gamma))/h$, where he_i is the variation by h of the i -th entry of the discretization of γ). (b) Optimality system (13) – (17) (FD: $D_{v,i}\bar{J} = (\bar{J}(v + he_i) - \bar{J}(v))/h$).

i.e. $J_2 = l\|\gamma\|_{L^2}^2 + \beta\|v\|_{L^2}^2$, where β is a small number (here: $\beta = 1e^{-5}$). The optimality system can be derived in an analogous way to the one described above, where the adjoint states are now denoted by $\lambda \in H^1([t_0, t_e], \mathbb{R}^l)$ and $\mu \in H^1([t_0, t_e], \mathbb{R})$:

$$\dot{\alpha} = \mathcal{A} + \mathcal{B}\alpha + \mathcal{C}(\alpha) + \mathcal{D}v + (\mathcal{E} + \mathcal{F}\alpha)\gamma + \mathcal{G}\gamma^2, \quad (13)$$

$$\dot{\gamma} = v, \quad (14)$$

$$\dot{\lambda} = -\frac{\partial \bar{J}}{\partial \alpha} - \left(\mathcal{B} + \frac{\partial \mathcal{C}(\alpha)}{\partial \alpha} + \mathcal{F}\gamma \right)^\top \lambda, \quad (15)$$

$$\dot{\mu} = -\frac{\partial \bar{J}}{\partial \gamma} - (\mathcal{E} + \mathcal{F}\alpha + 2\mathcal{G}\gamma)^\top \lambda, \quad (16)$$

$$0 = \frac{\partial \bar{J}}{\partial v} + \mathcal{D}^\top \lambda + \mu =: D_v \bar{J}, \quad (17)$$

with the Fréchet derivatives:

$$\begin{aligned} \partial \bar{J} / \partial \gamma &= 4l \left(l \int_{t_0}^{t_e} \gamma^2(t) dt + \beta \int_{t_0}^{t_e} v^2(t) dt - T_2 \right) \gamma(t), \\ \partial \bar{J} / \partial v &= 4\beta \left(l \int_{t_0}^{t_e} \gamma^2(t) dt + \beta \int_{t_0}^{t_e} v^2(t) dt - T_2 \right) v(t), \end{aligned}$$

$\partial \bar{J} / \partial \alpha_j$ as before and the respective boundary conditions

$$\alpha(0) = \alpha_0, \quad \lambda(T) = 0, \quad \mu(0) = 0, \quad \mu(T) = 0. \quad (18)$$

This yields a boundary value problem for the state equations (13), (14) and the adjoint equations (15), (16) which can, for a given control v , be solved by an iterative scheme. Applying a shooting method, the initial value $\gamma(0)$ is computed such that $\mu(0) = 0$ is satisfied using Newton's method (cf. Algorithm 4). This is computationally more expensive than the simple forward-backward integration in Algorithm 3. However, the system (13) – (17) yields strongly improved convergence and decreased sensitivity to the optimization parameters in comparison to the system (10) – (12), which stems from the improved

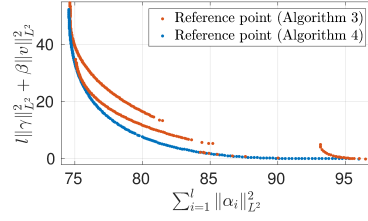


Fig. 11 Comparison of the Pareto fronts computed with the two optimality systems (10) – (12) and (13) – (17).

gradient accuracy (cf. Figure 10(b)). This is also evident from Figure 11, where the Pareto front based on (13) – (17) was computed executing Algorithm 4 once, starting in the already known point of zero control cost. In contrast to that, multiple attempts were made with Algorithm 3, starting from different points on the Pareto front that were computed with Algorithm 4. All of them either divert quickly from the front or stop prematurely. The results shown in Section 5 are therefore based on (13) – (17). From our experience, the approach based on the adjoint equation can be very sensitive, also with respect to stabilization methods where the coefficients are manipulated to better match the state equation. Hence, it would be favorable for future work to utilize direct approaches where only accuracy of the state equation is of importance.

Algorithm 4 (Adjoint based scalar optimization with shooting)

Require: Target point T , initial guess v^0

- 1: **for** $i = 0, \dots$ **do**
 - 2: Shooting step: Determine $\gamma^{(i)}(0)$ by solving an internal root finding problem in order to enforce the condition $\mu^{(i)}(0) = 0$. This step requires forward solves of (13), (14) and backward solves of (15), (16).
 - 3: Compute $\alpha^{(i)}, \gamma^{(i)}$ by solving the reduced order model (13), (14) with $v^{(i)}$ as the control and $\gamma^{(i)}(0)$ as computed in step 2.
 - 4: Solve the adjoint equations (15), (16) in backwards time with $v^{(i)}, \alpha^{(i)}, \gamma^{(i)}, (\partial \bar{J} / \partial \alpha)^{(i)}$ and $(\partial \bar{J} / \partial \gamma)^{(i)}$.
 - 5: Compute the gradient $(D_v \bar{J})^{(i)}$ by evaluating the optimality condition (17) with $\lambda^{(i)}$ and $\mu^{(i)}$.
 - 6: Update the control: $v^{(i+1)} = v^{(i)} + a^{(i)} d^{(i)}$, with $a^{(i)}$ and $d^{(i)}$ as in Algorithm 3.
 - 7: **end for**
-

5 Results

In this section we present solutions to (R-MOCP) obtained with the algorithms presented in Section 3. For all cases, we fix the time interval $[t_0, t_e]$ to $[0, 10]$ which corresponds to roughly two vortex shedding cycles. In order to use the subdivision method, we transform (MOCP) into a multiobjective optimization problem with moderate decision space dimension ($m \leq 10$). One way to do

this is by introducing a sinusoidal control (cf. Section 3.2.2) which then yields (MOP-3D). Another way is to approximate γ by a cubic spline with m break points. In this case, the parameters to be optimized are the spline values at these break points. For the reference point method, we discretize γ with a constant step length of $\Delta t = 0.05$ and solve Equations (13) – (16) with a fourth order Runge-Kutta scheme.

Figure 12 shows the box covering of the Pareto set for (MOP-3D), the respective Pareto front is shown in Figure 13. It is noteworthy that, apart from a few spurious points at $A \approx 0$ caused by numerical errors (Figure 12(a)), the largest part of the set is restricted to a small part of the frequency domain. This is comprehensible since the rotation counteracts the natural dynamics of the von Kármán vortex street. The trade-off between control cost and stabilization is almost exclusively realized by changing the amplitude of the rotation. When moving towards the two scalar optimal solutions, i.e. minimal fluctu-

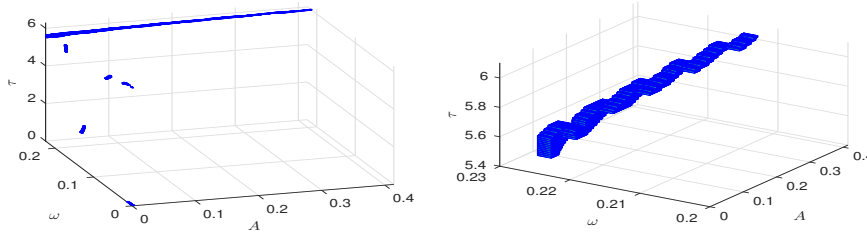


Fig. 12 Box covering of the solution of (MOP-3D), obtained with global subdivision algorithm after 27 subdivision steps. There are spurious solutions with almost zero amplitude, the right plot shows only the physically relevant part which is constrained to a small part of the frequency domain. The trade-off between the two objectives is mainly realized by changing the amplitude.

ations and minimal control cost, respectively, further improvements are only possible by accepting large trade-offs in the other objective which is a common phenomenon in multiobjective optimization. When designing a system, one could now accept a small increase in the main objective, i.e. flow stabilization, in order to achieve a large decrease in the control cost. This becomes more clear when looking at Figure 14(b), where different optimal compromises for the control are shown. For a relatively small improvement of J_1 from 85 to 81.05 ($\approx -5\%$), the control cost increases by 322%.

Figures 13(a) and 14 also show the Pareto fronts and Pareto optimal controls, respectively, computed with the other methods, i.e. two different spline approximations ($m = 5$ and $m = 10$) computed with the subdivision algorithm on the one hand and arbitrary control functions determined by the reference point method on the other hand. We observe that a spline with 5 breakpoints is too restrictive to find control functions of acceptable quality. When using 10 breakpoints on the other hand, the Pareto front clearly surpasses the solution of (MOP-3D). It is, however, numerically expensive to compute. The best so-

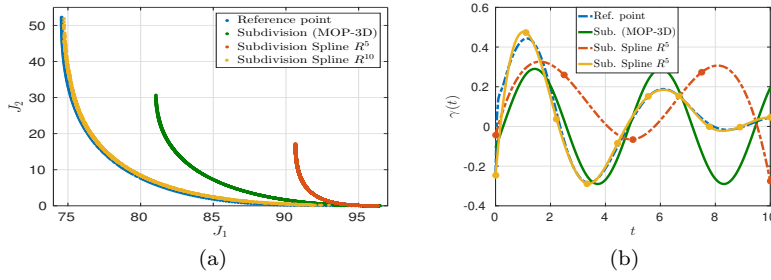


Fig. 13 (a) Comparison of Pareto fronts for different MOCP solution methods. (b) Comparison of the Pareto optimal controls with equal cost ($J_2 = 15$) for different MOCP solution methods.

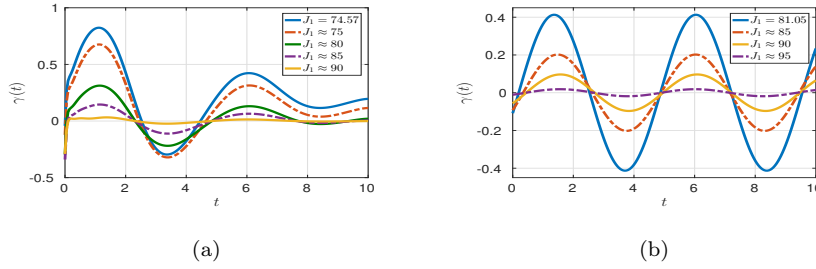


Fig. 14 Different Pareto points for the reference point method (a) and (MOP-3D) (b).

lution is computed with the reference point method. The improvement over the spline based solution is relatively low, the reason being that the controls are almost similar (see Figure 13(b), where solutions of the different methods with the same control cost $J_2 = 15$ are compared). When considering scenarios with larger time intervals than 10 seconds, the spline dimension would have to be increased further, leading to again much higher computational cost whereas the cost for the reference point method increases only linearly with time due to the fact that only the number of time steps in the forward and backward integration (13) – (16) is affected. Finally, Figure 15 shows two solutions of (2.1a) – (2.1f) with controls computed by the reference point method, corresponding to different values of the control cost J_2 and hence different degrees of stabilization. Due to the relatively short integration time of $t_e = 10$ seconds, changes are only apparent closely past the cylinder. Still, we observe increased stabilization when allowing higher control cost.

In order to evaluate the quality of the solution obtained with the reduced order modeling approach, we have evaluated the cost function using the system state obtained from a PDE evaluation. This is depicted in Figure 16. Since we only introduce an error in the first objective by the reduced order model, the error results in a horizontal shift of the Pareto points. In general, we observe a good agreement between the solutions, the error being less than 4% for all points that were tested. However, especially in regions with a steep gradient $\partial J_2 / \partial J_1$ in the Pareto front, we see that this small error can result in a different

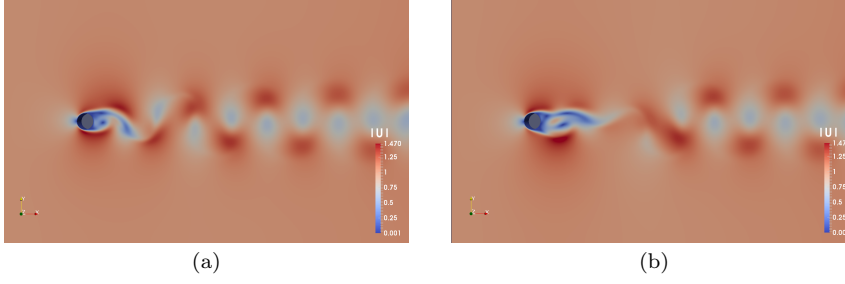


Fig. 15 FV solution at $t = 10$ with two different Pareto optimal solutions computed with the reference point method. (a) Relatively low control cost ($J_2 = 1.84$), almost no reduction of fluctuations ($J_1 = 85$). (b) Larger control cost ($J_2 = 30.12$), stronger reduction of fluctuations ($J_1 = 75$). Due to the short integration time, the effect is only visible in the vicinity of the cylinder.

shape of the front. Note that the two points with the highest values for J_2 are now dominated by the next lower point. This gives a strong motivation for further efforts to decrease the error of the reduced order model (e.g. by Trust-Region methods [20]) and investigate the influence of the inaccuracies in the gradient obtained by the adjoint approach. In the latter case, the subdivision algorithm has the clear advantage over the reference point method since it depends on the accuracy of the state equation only.

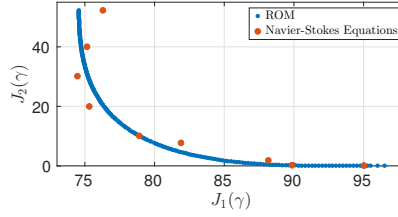


Fig. 16 Comparison of the objective values based on the state of the reduced model and of the PDE solution. Since there is no error in J_2 , the points are only shifted horizontally. Although the overall agreement is acceptable, the PDE based Pareto front appears to have a different shape, especially for solutions with larger control cost.

From a computational point of view, the reference point method is significantly faster, cf. Table 2. This is not surprising since, in order to have a good representation of a box, a large number of sample points needs to be evaluated. This is already numerically expensive in 3D and more so for a spline based approximation, i.e. we need to pay a price for global optimality and the derivative free approach. For future work, it is therefore advisable to utilize the gradient based version of the subdivision algorithm in order to decrease the computational cost, provided that high accuracy can be achieved for the gradient.

Table 2 Comparison of CPU time of different methods (Subdivision algorithm in parallel: CPU time = Number of CPUs times wall clock time).

	# boxes / points	# Function evaluations	# Adjoint evaluations	CPU time [h]
Subdivision (MOP-3D)	1383	$\approx 8.3 \cdot 10^6$	0	≈ 1132
Subdivision Spline \mathbb{R}^5	6057	$\approx 11.1 \cdot 10^6$	0	≈ 1512
Subdivision Spline \mathbb{R}^{10}	1880	$\approx 42 \cdot 10^6$	0	≈ 5875
Reference point (13) – (17)	256	26781	1002	≈ 12.2

6 Conclusion

In this article, we present different approaches for numerically solving multi-objective optimal control problems involving boundary control of the Navier-Stokes equations. In order to reduce the computational effort, model order reduction via proper orthogonal decomposition and Galerkin projection is used to compute a surrogate reduced model of ODEs. Different multiobjective optimal control algorithms are introduced and their respective advantages and disadvantages are discussed. The subdivision algorithm yields a box covering of globally optimal Pareto sets and can also deal with problems where the set is disconnected. Since its applicability depends critically on both the decision space dimension and the numerical effort of function evaluations, it is restricted to moderate decision space dimensions. To solve optimal control problems, the control function therefore has to be represented by a low number of parameters, e.g. by a harmonic function or spline coefficients. The reference point method converts the multiobjective optimal control problem into a sequence of scalar optimal control problems that can be solved using well known procedures from single objective optimal control theory, also for high decision space dimensions. To incorporate gradient information, the optimality system is derived for the scalar problem. In this case, the accuracy of the approximated gradient is of great importance, which becomes evident when comparing the two optimality systems that were derived. Provided the gradient accuracy is sufficient, the Pareto set can be approximated very efficiently, using the solution from previous scalar problems as initial guesses. The method is of local nature, however, such that these initial guesses as well as the computation of the first Pareto point are important. Additionally, the selection of target points becomes tedious for higher objective space dimensions.

The computed Pareto set yields considerably more information about the system than the solution of a single objective optimization. A decision maker can use this information to make well-founded decisions on how to control the system or even to devise adaptive control strategies reacting to changing priorities such as the need to save energy. There are numerous applications where multiobjective optimal control for systems described by PDEs is of great interest, e.g. minimizing the drag and maximizing the lift of airplanes, minimizing the weight and maximizing the stability of structures, optimal mixing with

minimal cost, control of flow patterns and vortices in HVAC applications, to name a few. Moreover, applications with higher Reynolds numbers are certainly of great interest [11], which adds difficulties especially concerning the model order reduction. A question that needs to be addressed in the future is the error control for the reduced model and the respective gradient in order to guarantee optimality for the PDE based problem. Several authors have addressed this for scalar optimization with reduced order models [20, 37, 59, 8, 27, 39] and additional questions arise in the context of multiobjective optimal control.

References

1. Abraham, A., Lakshmi, J., Goldberg, R. (eds.): *Evolutionary Multiobjective Optimization*. Springer (2005)
2. Albus, M.N., Rischmuller, V., Fritzsche, T., Lohmann, B.: Multiobjective Optimization of the Design of Nonlinear Electromagnetic Systems Using Parametric Reduced Order Models. *IEEE Transactions on Magnetics* **45**(3), 1474–1477 (2009)
3. Allgower, E.L., Georg, K.: *Introduction to Numerical Continuation Methods*. Society for Industrial and Applied Mathematics (2003)
4. Antoulas, A.C., Sorensen, D.C., Gugercin, S.: A survey of model reduction methods for large-scale systems. *Contemporary mathematics* **280**, 193–220 (2001)
5. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: Numerical Analysis An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique* **339**(9), 667–672 (2004)
6. Benner, P., Mehrmann, V., Sorensen, D.C. (eds.): *Dimension Reduction of Large-Scale Systems*. Springer (2005)
7. Bergmann, M., Bruneau, C.H., Iollo, A.: Enablers for robust POD models. *Journal of Computational Physics* **228**(2), 516–538 (2009)
8. Bergmann, M., Cordier, L.: Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *Journal of Computational Physics* **227**, 7813–7840 (2008)
9. Bergmann, M., Cordier, L., Brancher, J.P.: Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids* **17**, 1–21 (2005)
10. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics* **25**, 539–575 (1993)
11. Brunton, S.L., Noack, B.R.: Closed-Loop Turbulence Control: Progress and Challenges. *Applied Mechanics Reviews* **67**(050801), 1–48 (2015)
12. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer (2007)
13. Cordier, L., Abou El Majd, B., Favier, J.: Calibration of POD reduced-order models using Tikhonov regularization. *International Journal for Numerical Methods in Fluids* **63**(2), 269–296 (2009)
14. Couplet, M., Basdevant, C., Sagaut, P.: Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics* **207**(1), 192–220 (2005)
15. Dellnitz, M., Eckstein, J., Flaßkamp, K., Friedel, P., Horenkamp, C., Köhler, U., Ober-Blöbaum, S., Peitz, S., Tiemeyer, S.: Multiobjective Optimal Control Methods for the Development of an Intelligent Cruise Control. In: G. Russo, V. Capasso, G. Nicosia, V. Romano (eds.) *Progress in Industrial Mathematics at ECMI 2014*. Taormina, Italy (2016)
16. Dellnitz, M., Hohmann, A.: A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik* **75**(3), 293–317 (1997)
17. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications* **124**(1), 113–136 (2005)

18. Diwoky, F., Volkwein, S.: Nonlinear Boundary Control for the Heat Equation Utilizing Proper Orthogonal Decomposition. In: K.H. Hoffmann, R.H.W. Hoppe, V. Schulz (eds.) *Fast Solution of Discretized Optimization Problems*, pp. 73–87. Springer Basel (2001)
19. Ehrgott, M.: *Multicriteria optimization*, 2nd edn. Springer (2005)
20. Fahl, M.: *Trust-region Methods for Flow Control based on Reduced Order Modelling*. Phd thesis, University of Trier (2000)
21. Ferziger, J.H., Peric, M.: *Computational Methods for Fluid Dynamics*, 3rd edn. Springer (2002)
22. Fursikov, A.V., Gunzburger, M.D., Hou, L.S.: Boundary value problems and optimal boundary control for the Navier-Stokes system: the two-dimensional case. *SIAM Journal on Control and Optimization* **36**(3), 852–894 (1998)
23. Galletti, B., Bruneau, C.H., Zannetti, L., Iollo, A.: Low-order modelling of laminar flow regimes past a confined square cylinder. *Journal of Fluid Mechanics* **503**, 161–170 (2004)
24. Gerdts, M.: *Optimal control of ODEs and DAEs*. Walter de Gruyter (2012)
25. Graham, W.R., Peraire, J., Tang, K.Y.: Optimal Control of Vortex Shedding Using Low Order Models. Part I: Open-Loop Model Development. *International Journal for Numerical Methods in Engineering* **44**(7), 945–972 (1999)
26. Grepl, M.A., Maday, Y., Nguyen, N.C., Patera, A.T.: Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis* **41**(3), 575–605 (2007)
27. Gubisch, M., Volkwein, S.: Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control. In: P. Benner, A. Cohen, M. Ohlberger, K. Willcox (eds.) *Model Reduction and Approximation: Theory and Algorithms*. SIAM (2016)
28. Gad-el Hak, M.: Flow Control. *Applied Mechanics Reviews* **42**(10), 261–292 (1989)
29. Heywood, J.G., Rannacher, R., Turek, S.: Artificial Boundaries and Flux and Pressure Conditions for the Incompressible Navier-Stokes Equations. *International Journal for Numerical Methods in Fluids* **22**(5), 325–352 (1996)
30. Hillermeier, C.: *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Birkhäuser (2001)
31. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: *Optimization with PDE Constraints*. Springer Science+Business Media (2009)
32. Holmes, P., Lumley, J.L., Berkooz, G.: *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press (1998)
33. Iapichino, L., Trenz, S., Volkwein, S.: Multiobjective optimal control of semilinear parabolic problems using POD. In: B. Karasözen, M. Manguoglu, M. Tezer-Sezgin, S. Goktepe, Ö. Ugur (eds.) *Numerical Mathematics and Advanced Applications (ENUMATH 2015)*, pp. 389–397. Springer (2016)
34. Iapichino, L., Ulbrich, S., Volkwein, S.: *Multiobjective PDE-Constrained Optimization Using the Reduced-Basis Method*. *Advances in Computational Mathematics* (2017)
35. Ito, K., Ravindran, S.S.: Reduced Basis Method for Optimal Control of Unsteady Viscous Flows. *International Journal of Computational Fluid Dynamics* **15**(2), 97–113 (2001)
36. Jasak, H., Jemcov, A., Tukovic, Z.: OpenFOAM : A C ++ Library for Complex Physics Simulations. *International Workshop on Coupled Methods in Numerical Dynamics* pp. 1–20 (2007)
37. Kunisch, K., Volkwein, S.: Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis* **40**(2), 492–515 (2002)
38. Lachmann, G.V.: *Boundary Layer and Flow Control. Its Principles and Application*. Elsevier (2014)
39. Lass, O.: *Reduced order modeling and parameter identification for coupled nonlinear PDE systems*. Ph.D. thesis, University of Konstanz (2014)
40. Logg, A., Mardal, K.A., Wells, G.N.: *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS book*, vol. 84. Springer Science & Business Media (2012)
41. Logist, F., Houska, B., Diehl, M., van Impe, J.: Fast Pareto set generation for nonlinear optimal control problems with multiple objectives. *Structural and Multidisciplinary Optimization* **42**(4), 591–603 (2010)

42. Lotov, A.V., Kamenev, G.K., Berezkin, V.E., Miettinen, K.: Optimal control of cooling process in continuous casting of steel using a visualization-based multi-criteria approach. *Applied Mathematical Modelling* **29**, 653–672 (2005)
43. Lumley, J.L.: The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation* pp. 166–178 (1967)
44. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Springer Science+Business Media (2012)
45. Noack, B.R., Papas, P., Monkewitz, P.A.: The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics* **523**, 339–365 (2005)
46. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Science & Business Media (2006)
47. Ober-Blöbaum, S., Padberg-Gehle, K.: Multiobjective optimal control of fluid mixing. In: *Proceedings in Applied Mathematics and Mechanics (PAMM)* (2015)
48. Ober-Blöbaum, S., Ringkamp, M., zum Felde, G.: Solving Multiobjective Optimal Control Problems in Space Mission Design using Discrete Mechanics and Reference Point Techniques. In: *51st IEEE International Conference on Decision and Control*, pp. 5711–5716. Maui, HI, USA (2012)
49. Peitz, S., Dellnitz, M.: Multiobjective Optimization of the Flow Around a Cylinder Using Model Order Reduction. In: *Proceedings in Applied Mathematics and Mechanics (PAMM)*, pp. 613–614 (2015)
50. Rannacher, R.: Finite Element Methods for the Incompressible Navier-Stokes Equations. In: *Fundamental directions in mathematical fluid mechanics*, pp. 191–293. Birkhäuser (2000)
51. Ravindran, S.S.: A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids* **34**, 425–448 (2000)
52. Rempfer, D.: On Low-Dimensional Galerkin Models for Fluid Flow. *Theoretical and Computational Fluid Dynamics* **14**, 75–88 (2000)
53. Romaus, C., Böcker, J., Witting, K., Seifried, A., Znamenshchykov, O.: Optimal energy management for a hybrid energy storage system combining batteries and double layer capacitors. In: *2009 IEEE Energy Conversion Congress and Exposition*, pp. 1640–1647 (2009)
54. Rowley, C.W.: Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition. *International Journal of Bifurcation and Chaos* **15**(3), 997–1013 (2005)
55. Schilders, W., van der Vorst, H.A., Rommes, J. (eds.): *Model Order Reduction*. Springer Berlin Heidelberg (2008)
56. Schütze, O., Witting, K., Ober-Blöbaum, S., Dellnitz, M.: Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems. In: E. Tantar, A.A. Tantar, P. Bouvry, P. Del Moral, P. Legrand, C.A. Coello Coello, O. Schütze (eds.) *EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation, Studies in Computational Intelligence*, vol. 447, pp. 187–219. Springer Berlin Heidelberg (2013)
57. Sirisup, S., Karniadakis, G.E.: A spectral viscosity method for correcting the long-term behavior of POD models. *Journal of Computational Physics* **194**, 92–116 (2004)
58. Sirovich, L.: Turbulence and the dynamics of coherent structures Part I: coherent structures. *Quarterly of Applied Mathematics* **XLV**(3), 561–571 (1987)
59. Veroy, K., Patera, A.T.: Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids* **47**, 773–788 (2005)
60. Willcox, K., Peraire, J.: Balanced Model Reduction via the Proper Orthogonal Decomposition. *AIAA Journal* **40**(11), 2323–2330 (2002)
61. Xiao, D., Fang, F., Buchan, A.G., Pain, C.C., Navon, I.M., Du, J., Hu, G.: Non-linear model reduction for the Navier-Stokes equations using residual DEIM method. *Journal of Computational Physics* **263**, 1–18 (2014)

A Reduced Order Model

Here we state the model coefficients for the reduced order model (4), see e.g. [20, 9] for details. We refer to the time average of the modified flow field by $\mathbf{U}_m = \langle \tilde{\mathbf{U}}(x, t) \rangle$.

$$\begin{aligned}\mathcal{A}_i &= -((\mathbf{U}_m \cdot \nabla) \mathbf{U}_m, \boldsymbol{\psi}_i)_{L^2} - \frac{1}{Re} (\nabla \mathbf{U}_m, \nabla \boldsymbol{\psi}_i)_{L^2}, \\ \mathcal{B}_{ij} &= -((\mathbf{U}_m \cdot \nabla) \boldsymbol{\psi}_j, \boldsymbol{\psi}_i)_{L^2} - ((\boldsymbol{\psi}_j \cdot \nabla) \mathbf{U}_m, \boldsymbol{\psi}_i)_{L^2} - \frac{1}{Re} (\nabla \boldsymbol{\psi}_i, \nabla \boldsymbol{\psi}_j)_{L^2}, \\ \mathcal{Q}_{jik} &= -((\boldsymbol{\psi}_i \cdot \nabla) \boldsymbol{\psi}_k, \boldsymbol{\psi}_j)_{L^2}, \\ \mathcal{D}_i &= -(\mathbf{U}_c, \boldsymbol{\psi}_i)_{L^2}, \\ \mathcal{E}_i &= -((\mathbf{U}_m \cdot \nabla) \mathbf{U}_c, \boldsymbol{\psi}_i)_{L^2} - ((\mathbf{U}_c \cdot \nabla) \mathbf{U}_m, \boldsymbol{\psi}_i)_{L^2} - \frac{1}{Re} (\nabla \mathbf{U}_c, \nabla \boldsymbol{\psi}_i)_{L^2}, \\ \mathcal{F}_{ij} &= -((\mathbf{U}_c \cdot \nabla) \boldsymbol{\psi}_j, \boldsymbol{\psi}_i)_{L^2} - ((\boldsymbol{\psi}_j \cdot \nabla) \mathbf{U}_c, \boldsymbol{\psi}_i)_{L^2}, \\ \mathcal{G}_i &= -((\mathbf{U}_c \cdot \nabla) \mathbf{U}_c, \boldsymbol{\psi}_i)_{L^2}.\end{aligned}$$

B Derivation of the optimality system

Here we derive the optimality system (10) – (12) for the reference point method with the scalarized cost functional (8). In order to satisfy the necessary condition for optimality, we require that all variations of the Lagrange functional (9) are zero:

$$\begin{aligned}\delta L &= \frac{\partial L}{\partial \alpha} \delta \alpha + \frac{\partial L}{\partial \dot{\alpha}} \delta \dot{\alpha} + \frac{\partial L}{\partial \lambda} \delta \lambda + \frac{\partial L}{\partial \gamma} \delta \gamma + \frac{\partial L}{\partial \dot{\gamma}} \delta \dot{\gamma} = 0 \\ \Leftrightarrow \int_{t_0}^{t_e} \left(\frac{\partial \bar{J}}{\partial \alpha} + \lambda^\top \left(\mathcal{B} + \frac{\partial \mathcal{C}(\alpha)}{\partial \alpha} + \mathcal{F} \gamma \right) \right) \delta \alpha + \left(\frac{\partial \bar{J}}{\partial \gamma} + \lambda^\top (\mathcal{E} + \mathcal{F} \alpha + 2\mathcal{G} \gamma) \right) \delta \gamma + \\ &+ (\dot{\alpha} - \mathcal{A} - \mathcal{B} \alpha - \mathcal{C}(j \alpha) - \mathcal{D} \dot{\gamma} - (\mathcal{E} - \mathcal{F} \alpha) \gamma - \mathcal{G} \gamma^2) \delta \lambda + \lambda^\top \mathcal{D} \delta \dot{\gamma} - \lambda^\top \delta \dot{\alpha} dt = 0.\end{aligned}$$

Using partial integration, this leads to

$$\begin{aligned}& \int_{t_0}^{t_e} \left(\dot{\lambda}^\top + \frac{\partial \bar{J}}{\partial \alpha} + \lambda^\top \left(\mathcal{B} + \frac{\partial \mathcal{C}(\alpha)}{\partial \alpha} + \mathcal{F} \gamma \right) \right) \delta \alpha + \\ &+ (\dot{\alpha} - \mathcal{A} - \mathcal{B} \alpha - \mathcal{C}(j \alpha) - \mathcal{D} \dot{\gamma} - (\mathcal{E} - \mathcal{F} \alpha) \gamma - \mathcal{G} \gamma^2) \delta \lambda + \\ &+ \left(-\dot{\lambda}^\top \mathcal{D} + \frac{\partial \bar{J}}{\partial \gamma} + \lambda^\top (\mathcal{E} + \mathcal{F} \alpha + 2\mathcal{G} \gamma) \right) \delta \gamma - \lambda^\top \delta \alpha \Big|_{t_0}^{t_e} + \lambda^\top \mathcal{D} \delta \gamma \Big|_{t_0}^{t_e} = 0.\end{aligned}$$

Since we require each variation to be zero individually, we obtain the equations (10) – (12) and two boundary conditions at t_0 and t_e , respectively. We have a fixed value $\alpha(t_0) = \alpha_0$ such that $\delta \alpha(t_0) = 0$ and hence, we have no initial condition $\lambda(t_0)$. In contrast, $\alpha(t_e)$ is arbitrary such that $\delta \alpha(t_e) \neq 0$ which results in the terminal condition $\lambda(t_e) = 0$. Since we want to accept arbitrary initial values $\gamma(t_0)$ and $\lambda(t_0)$ is also free, we obtain an additional condition $\lambda^\top(t_0) \mathcal{D} = 0$. The second optimality system (13) – (17) is derived analogously.