

MOX–Report No. 22/2013

**Recent Results in Shape Optimization and Optimal
Control for PDEs**

FALCONE, M.; VERANI, M.

MOX, Dipartimento di Matematica “F. Brioschi”
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it

<http://mox.polimi.it>

Recent Results in Shape Optimization and Optimal Control for PDEs

Maurizio Falcone^a and Marco Verani^b

April 15, 2013

^a Dipartimento di Matematica, SAPIENZA - Università di Roma
P.za Aldo Moro, 2, 00185 Roma, Italy
E-mail: falcone@mat.uniroma1.it

^b MOX, Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
E-mail: marco.verani@polimi.it

Abstract

In this paper we will present some recent advances in the numerical approximation of two classical problems: shape optimization and optimal control for evolutive partial differential equations. For shape optimization we present two novel techniques which have shown to be rather efficient on some applications. The first technique is based on multigrid methods whereas the second relies on an adaptive sequential quadratic programming. With respect to the optimal control of evolutive problems, the approximation is based on the coupling between a POD representation of the dynamical system and the classical Dynamic Programming approach. We look for an approximation of the value function characterized as the weak solution (in the viscosity sense) of the corresponding Hamilton-Jacobi equation. Several tests illustrate the main features of the above methods.

1 Introduction

In this survey we will present some recent advances in the numerical approximation of two classical problems: shape optimization and optimal control for evolutive partial differential equations. These results have been achieved with the contributions of the researcher working in the teams at Milano Politecnico and Roma "La Sapienza" within the ESF OPTPDE project.

Shape optimization problems are ubiquitous in science, engineering and industrial applications. Indeed, starting with the foundation of PDE-based optimization [30], shape design has become one of the most frequent application in technologies and it is nowadays one main focus of aerodynamics simulation (see, e.g., [31, 42]).

A central role in the formulation and development of computational frameworks for shape optimization has been played by elliptic shape optimization problems [36] that correspond to cases of potential flow allowing simpler investigation. Nevertheless, these problems arise in many important applications as nozzle and airfoil design, and in the design of beams and plates. Along this development, one of the most remarkable advances in shape design has been to replace the approach of parametric optimization with the concept of continuous

shape design (see, e.g., the books [17, 22, 31, 36, 38, 28]). In fact, in the former approach the control variable (i.e., the shape) is restricted to belong to a finite dimensional space spanned by suitable basis functions, while in the latter case it is an element of an infinite-dimensional space. This second approach opens enormous perspective in the formulation of more accurate and sophisticated shape optimization problems.

The possibility of formulating the shape optimization problems at the infinite-dimensional level poses new challenges to the design and implementation of numerical optimization schemes that properly accommodate the infinite dimensionality of the control function. In particular, a successful and effective algorithm must allow the control function to be *adaptively* approximated and optimized to any desired degree of accuracy.

With respect to shape optimization, the purpose of this paper is twofold. We first formulate and analyze a multigrid shape optimization framework that extends principles and techniques of the multigrid strategy for PDE solvers and accommodates the infinite-dimensionality of the control variables; then we introduce an adaptive strategy able to automatically deal with the approximation of the optimal geometry combined with the approximation of the underlying PDE. As we said, our second problem will be the approximation of a finite horizon optimal control problem for an evolutive partial differential equation, e.g. the advection–diffusion equation. The basic ingredient of the method is the coupling between an adaptive reduced basis representation of the solution and a Dynamic Programming scheme for the evolutive Hamilton-Jacobi equation characterizing the value function. Since the theory of weak solutions for Hamilton-Jacobi equation is rather complete in any dimension, the method can in principle solve a rather general class of optimal control problems. The approach described here is clearly different from the more classical approach based on the solution of the system of necessary conditions obtained via the Pontryagin maximum principle. The main advantage is that we naturally obtain optimal control in feedback form but the price we pay is related to the well know curse of dimensionality of Dynamic Programming. We try to circumvent this problem using new tools which have emerged in recent years to deal with optimal control problems in infinite dimension. In particular, we will use new techniques to reduce the number of dimensions in the description of the dynamical system or, more in general, of the solution of the problem that one is trying to optimize. These methods are generally called *reduced-order methods* and include for example the POD (Proper Orthogonal Decomposition) method and reduced basis approximation (see [35]). In some particular case, as for the heat equation, even 5 basis functions will suffice to have a rather accurate POD representation of the solution. Having this in mind, it is reasonable to start thinking to a different approach based on Dynamic Programming (DP) and Hamilton-Jacobi-Bellman equations (HJB). In this new approach we will first develop a reduced basis representation of the solution along a reference trajectory and then use this basis to set-up a control problem in the new space of coordinates. Then, the corresponding Hamilton-Jacobi equation will just need 3-5 variables to represent the state of the system. It is well known that the solution of HJB equation is not an easy task from the numerical point of view since viscosity solutions of the HJB equation are typically non regular (typically, just Lipschitz continuous). Optimal control problems for ODEs were solved by Dynamic Programming, both analytically and numerically (see [4] for a general presentation of this theory). From the numerical point of view, this approach has been developed for many classical control problems obtaining convergence results and a-priori error estimates ([19], [21] and the book [20]). We should mention that a first tentative in this direction has been made by Kunisch and co-authors in a series of papers [23, 24] for diffusion dominated equations.

In particular, in the paper by Kunisch, Volkwein and Xie [26] one can see a feedback control approach based on coupling between POD basis approximation and HJB equations for the viscous Burgers equation.

Note that restricting the dimension to a rather low number of basis functions (typically 4) naturally affects the accuracy of the POD approximation. In fact, under this restriction, the POD method does not always have enough informations to follow correctly the solution of the evolutive problem. We circumvent this problem updating our POD basis during the evolution and splitting the problem into subproblems. Every sub-problem is set in an interval $I_j = [t_j, t_{j+1}]$ where we recompute the POD basis. Behind the adaptive method and the choice of the t_j there are two important *a-posteriori* estimators: the first is related to the computation of the POD basis function while the second takes into account the residual of the dynamic.

2 Two approaches for Shape Optimization: Multigrid and Adaptivity

Shape optimization problems governed by partial differential equations (PDE) can be formulated as constrained minimization problems with respect to the shape of a domain Ω in \mathbb{R}^d . If $u = u(\Omega)$ is the solution of a PDE in Ω , the state equation,

$$\mathcal{A}u(\Omega) = f, \quad (1)$$

and $J(\Omega, u(\Omega))$ is a cost functional, then we consider the minimization problem

$$\Omega^* \in \mathcal{U}_{ad} : \quad J(\Omega^*, u(\Omega^*)) = \inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega, u(\Omega)), \quad (2)$$

where \mathcal{U}_{ad} is a set of admissible domains in \mathbb{R}^d . This is a constrained minimization problem for J .

In this Section we review two different shape optimization algorithms, namely the Multigrid Sequential Quadratic Programming (MSQP) presented in [3] and the Adaptive Sequential Quadratic Programming algorithm (ASQP) introduced in [32]. Such algorithms build a sequence of domains $\{\Omega^{(\ell)}\}_{\ell \geq 0}$ converging to a local minimizer of the shape optimization problem (1)-(2). To motivate and briefly describe the ideas underlying MSQP and ASQP, we need the concept of shape derivative $\nabla J(\Omega; \mathbf{w})$ of $J(\Omega)$ in the direction of a normal velocity \mathbf{w} . By resorting to the celebrated Hadamard-Zolésio structure theorem (see, e.g., [38, 17]), it is well known that the shape derivative $\nabla J(\Omega; \mathbf{w})$ can be always written as

$$\nabla J(\Omega; \mathbf{w}) = \int_{\Gamma} G(\Omega) \mathbf{w}, \quad (3)$$

for a proper choice of the function $G(\Omega)$, named the *Riesz representation* of the shape derivative, that in general depends on the solution $u(\Omega)$ of the state equation (1). To review MSQP and ASQP, we preliminary introduce an infinite dimensional Sequential Quadratic Programming (∞ -ASQP) algorithm. Let $\Omega^{(\ell)}$ be the current iterate and $\Omega^{(\ell+1)}$ be the new one. We let $\Gamma^{(\ell)} := \partial\Omega^{(\ell)}$ and $\mathbb{V}(\Gamma^{(\ell)})$ be a Hilbert space defined on $\Gamma^{(\ell)}$, with norm $\|\cdot\|_{\mathbb{V}(\Gamma^{(\ell)})}$. We further let $b_{\Gamma^{(\ell)}}(\cdot, \cdot) : \mathbb{V}(\Gamma^{(\ell)}) \times \mathbb{V}(\Gamma^{(\ell)}) \rightarrow \mathbb{R}$ be a continuous and coercive bilinear form

with respect to the norm $\|\cdot\|_{\mathbb{V}(\Gamma^{(\ell)})}$, which gives rise to the elliptic selfadjoint operator $\mathcal{B}^{(\ell)}$ on $\Gamma^{(\ell)}$ defined by $\langle \mathcal{B}^{(\ell)}\mathbf{v}, \mathbf{w} \rangle_{\Gamma^{(\ell)}} = b_{\Gamma^{(\ell)}}(\mathbf{v}, \mathbf{w})$. We then consider the following *quadratic model* $Q^{(\ell)} : \mathbb{V}(\Gamma^{(\ell)}) \rightarrow \mathbb{R}$ of J at $\Omega^{(\ell)}$

$$Q^{(\ell)}(\mathbf{w}) := J(\Omega^{(\ell)}) + \nabla J(\Omega^{(\ell)}; \mathbf{w}) + \frac{1}{2} \langle \mathcal{B}^{(\ell)}\mathbf{w}, \mathbf{w} \rangle. \quad (4)$$

We denote by $\mathbf{v}^{(\ell)}$ the minimizer of $Q^{(\ell)}(\mathbf{w})$, namely $\mathbf{v}^{(\ell)}$ satisfies

$$\mathbf{v}^{(\ell)} \in \mathbb{V}(\Gamma^{(\ell)}) : \quad b_{\Gamma^{(\ell)}}(\mathbf{v}^{(\ell)}, \mathbf{w}) = -\langle G^{(\ell)}, \mathbf{w} \rangle_{\Gamma^{(\ell)}} \quad \forall \mathbf{w} \in \mathbb{V}(\Gamma^{(\ell)}), \quad (5)$$

with $g^{(\ell)} := g(\Omega^{(\ell)})$. It is easy to check that $\mathbf{v}^{(\ell)}$ is the unique minimizer of $Q^{(\ell)}(\mathbf{w})$ and that the coercivity of the form $b_{\Gamma^{(\ell)}}(\cdot, \cdot)$ implies that $\mathbf{v}^{(\ell)}$ is an admissible descent direction; i.e. $\nabla J(\Omega^{(\ell)}; \mathbf{v}^{(\ell)}) < 0$.

Once $\mathbf{v}^{(\ell)}$ has been found, we need to determine a stepsize that is not too small and guarantees sufficient decrease of the functional J . To accomplish this goal we identify a range of admissible stepsizes by adapting the classical Armijo-Wolfe conditions in \mathbb{R}^n : given constants $0 < \alpha < \beta < 1$, we seek a stepsize $\mu \in \mathbb{R}^+$ satisfying

$$J(\Omega^{(\ell)} + \mu\mathbf{v}^{(\ell)}) \leq J(\Omega^{(\ell)}) + \alpha\mu \nabla J(\Omega^{(\ell)}; \mathbf{v}^{(\ell)}), \quad (6)$$

$$\nabla J(\Omega^{(\ell)} + \mu\mathbf{v}^{(\ell)}; \mathbf{v}^{(\ell)}) \geq \beta \nabla J(\Omega^{(\ell)}; \mathbf{v}^{(\ell)}), \quad (7)$$

where $\Omega^{(\ell)} + \mu\mathbf{v}^{(\ell)} := \{\mathbf{y} \in \mathbb{R}^d : \mathbf{y} = \mathbf{x} + \mu\mathbf{v}^{(\ell)}(\mathbf{x}), \mathbf{x} \in \Omega^{(\ell)}\}$ is the updated domain and $\mathbf{v}^{(\ell)} = v^{(\ell)}\boldsymbol{\nu}^{(\ell)}$ is a normal vector field.

We are now ready to introduce the infinite dimensional Sequential Quadratic Programming algorithm (∞ -SQP) for solving the constrained optimization problem (1)-(2):

∞ -SQP Algorithm

Given the initial domain $\Omega^{(0)}$, set $\ell = 0$ and iterate:

- (a) Compute $u^{(\ell)} = u(\Omega^{(\ell)})$ by solving (1)
- (b) Compute the Riesz representation $G^{(\ell)} = G(\Omega^{(\ell)})$ of (3)
- (c) Compute the search direction $\mathbf{v}^{(\ell)}$ by solving (5)
- (d) Determine an admissible stepsize $\mu^{(\ell)}$ satisfying (6)-(7)
- (e) Update: $\Omega^{(\ell+1)} = \Omega^{(\ell)} + \mu^{(\ell)}\mathbf{v}^{(\ell)}$; $\ell := \ell + 1$

It is important to note that, the ∞ -SQP algorithm is not feasible as it stands, because it requires the exact computation of the following quantities at each iteration: the solution $u^{(\ell)}$ to the state equation (1); the solution $\mathbf{v}^{(\ell)}$ to the linear subproblem (5); the values of the functional J and of its derivative dJ in the line search routine. In the following, we review the Adaptive Sequential Quadratic Programming algorithm (ASQP) (see Section 2.1) and the Multigrid Sequential Quadratic Programming (MSQP) (see Section 2.2) as possible feasible variants of the ∞ -SQP algorithm.

2.1 MSQP: a multigrid shape optimization algorithm

In this Section we sketch the ideas underlying the construction of MSQP, we present the algorithm and we report some enlightening numerical results (see [3] for more details). Generally speaking, in MSQP the boundary of the domain, i.e., the control variable, is represented at various levels k of discretization and the resulting multigrid shape optimization scheme acts directly on the geometry of the domain combining a single-grid shape gradient optimizer with a coarse-grid correction (minimization) step, recursively within a hierarchy of levels.

As we focus on multigrid concepts, we need to define an iterative optimization process that can be applied at every level of discretization with the aim of improving the shape towards the optimum. In our case, this is a shape-gradient optimizer, denoted by SQP_k , that acts similarly to a Jacobi smoother in a classical multigrid scheme. In practice, SQP_k is a feasible variant of the ∞ -SQP stated at the level k of discretization (see e.g. [16, 18]).

In addition to the iterative scheme mentioned above, the formulation of a multigrid scheme requires to define a coarse-grid correction step that complements the action of the single-grid optimization procedure. To construct this step, suitable intergrid transfer operators are required together with the formulation of a coarse optimization problem that correctly approximates the fine-level shape optimization problem. On the other hand, to define the coarse shape optimization problem, the multigrid optimization framework introduced in [29, 33] is extended to the present case where the optimization variable is a geometrical object.

The approach presented in [3] is in contrast to previous attempts [7, 12, 13, 14, 15, 11] to define a consistent multigrid framework for shape optimization where the computational domain is discretized by finite elements and the control boundary is represented through parameterized shape functions. Therefore, within the hierarchy of levels defined by the multigrid strategy, the approach of MSQP allows to construct a coarse-grid correction step that can be understood from the geometrical [40] and optimization [5, 6, 29, 33, 41] point of views, whereas the idea in [7, 12] of coarsening by taking a subset of shape parameters appears based on heuristic consideration.

To prepare the description of the MSQP algorithm, we first introduce the hierarchy of spaces \mathcal{U}_{ad}^k of *discrete* admissible configurations. According to this, we denote by Ω_k an element of \mathcal{U}_{ad}^k and by Γ_k the corresponding boundary (see Figure 1 for an example where the deformable part of the domain is the graph of a function). Then we introduce the finite element space to approximate the solution of the PDE on Ω_k : let $\mathcal{T}_k(\Omega_k)$ be a conforming and shape-regular triangulation of Ω_k and $\mathbb{V}(\Omega_k)$ denote the associated space of finite elements. If we define the discrete reduced functional $\hat{J}_k(\Gamma_k)$ at k -level as

$$\hat{J}_k(\Gamma_k) := J(\Omega_k, y_k(\Omega_k)) \quad (8)$$

then the reduced discrete shape optimization problem at level k reads as follows:

$$\Gamma_k^* = \operatorname{argmin}_{\Gamma_k \in \mathcal{U}_{ad}^k} \hat{J}_k(\Gamma_k) . \quad (9)$$

Finally, for multigrid purpose, we need to define intergrid transfer operators acting on: functions in \mathcal{U}_{ad}^k ; geometric boundaries Γ_k ; and functions defined on geometric boundaries (i.e., shape gradients). To simplify the exposition (see [3] for more rigorous definitions) we will not use different symbols to distinguish among the above operators: we will always denote by I_k^{k-1} the restriction operators and by I_{k-1}^k the corresponding prolongation operators, the difference being clear from the context (see Figure 1).

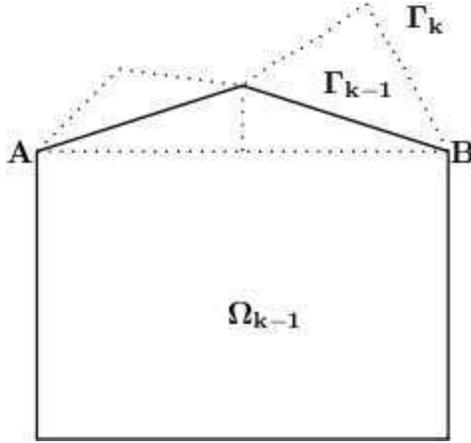


Figure 1: An example of discrete control boundary represented at different levels of discretization: Γ_k (dotted) and $\Gamma_{k-1} = I_k^{k-1}\Gamma_k$ (solid).

Finally, we introduce a hierarchy of nested shape optimization problems that will be solved at different levels of discretization. At k -level of discretization, we consider a function g_k to be defined iteratively in terms of g_{k+1} , where we set $g_K = 0$, being K the finest level of discretization (see below, Step 4 of the MSQP algorithm, for the precise recursive definition of g_k). The corresponding shape optimization problem at k -level reads as follows:

$$\min_{\Gamma_k \in \mathcal{U}_{ad}^k} F_k(\Gamma_k) := \widehat{J}_k(\Gamma_k) - \int_{\Omega_k} g_k d\Omega . \quad (10)$$

It is clear that at the finest level K , the problem (10) corresponds to the original discrete shape optimization problem. Our aim is to formulate a multigrid shape optimization scheme for solving the minimization problem (10) for all levels k .

Let $\Gamma_k^{(0)}$ be the initial optimization boundary at level k and g_k be given. The following steps define one multigrid V -cycle that will be denoted by $\Gamma_k^{(new)} = \text{MSQP}(\Gamma_k^{(old)}, k, g_k)$.

MSQP Algorithm

If $k = 1$ (coarsest resolution) then the minimization problem (10) is solved exactly. Else if $k > 1$:

(1) Apply one-grid optimization

$$\Gamma_k^{(\ell+1)} = \text{SQP}_k(\Gamma_k^{(\ell)}), \quad \ell = 0, 1, \dots, m_1 - 1.$$

(2) Compute the gradient residual

$$r_k = g_k - \nabla \widehat{J}_k(\Gamma_k^{(m_1)}).$$

(3) Restrict the residual and the approximate solution to coarse levels

$$r_{k-1} = I_k^{k-1} r_k, \quad \widehat{\Gamma}_{k-1} = I_k^{k-1} \Gamma_k^{(m_1)}.$$

(4) Setup the coarse-grid problem

$$g_{k-1} = \nabla \widehat{J}_{k-1}(\widehat{\Gamma}_{k-1}) + r_{k-1}.$$

(5) Call the MSQP scheme to compute the coarse-grid minimizer for $\min F_{k-1}(\Gamma_{k-1})$: $\tilde{\Gamma}_{k-1} = \text{MSQP}(\widehat{\Gamma}_{k-1}, k-1, g_{k-1})$ such that

$$\tilde{\Gamma}_{k-1} \approx \text{argmin} F_{k-1}(\Gamma_{k-1}).$$

(6) Construct the multigrid coarse-to-fine descent direction

$$\gamma_k = I_{k-1}^k (\tilde{\Gamma}_{k-1} - \widehat{\Gamma}_{k-1}).$$

(7) Optimize along γ_k with α -linesearch

$$\Gamma_k^{m_1+1} = \Gamma_k^{(m_1)} + \alpha \gamma_k$$

(8) Apply one-grid optimization

$$\Gamma_k^{(\ell+1)} = \text{SQP}_k(\Gamma_k^{(\ell)}), \quad \ell = m_1 + 1, \dots, m_1 + m_2.$$

(9) End.

In [3] it is proved that the multigrid coarse-to-fine direction γ_k built in Step 6 is indeed a descent direction. Moreover, it should be clear that the MSQP scheme given above will be applied iteratively, thus resulting in a sequence of V-cycles with finest level K and $g_K = 0$. Therefore, we also refer to the following algorithm as the MSQP scheme.

MSQP Algorithm

Input finest level K , initial Γ_K^0 , $g_K = 0$, Tolerance ϵ , iteration counter $\ell = 0$, max number iterations ℓ_{max} and iterate:

- (1) Compute $\Gamma_K^{\ell+1} = \text{MSQP}(\Gamma_K^\ell, K, g_K)$
- (2) Check convergence: if $\|\nabla \widehat{J}(\Gamma_K^\ell)\| > \epsilon$ and $\ell < \ell_{max}$ then $\ell := \ell+1$ and go to Step 1.
- (3) End

In the following, we report some numerical results, originally presented in [3], where a shape optimization problem governed by an elliptic PDE has to be solved. In particular, let $y = y(\Omega)$ be the unique solution to the following elliptic partial differential equation

$$-\Delta y = f \quad \text{in } \Omega \tag{11}$$

$$y = y_b \quad \text{on } \partial\Omega, \tag{12}$$

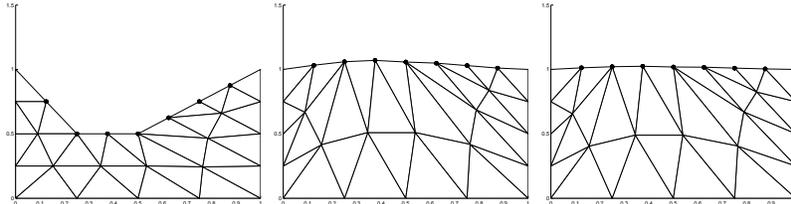
where y_b is a given function defined in \mathbb{R}^2 . Let r be a given function and $\lambda_1, \lambda_2, A, P > 0$ be given positive parameters. We consider the following cost functional

$$J(y, \Omega) := \int_{\Omega} r(y) \, d\Omega + \frac{\lambda_1}{2} \left(\int_{\partial\Omega} d\Gamma - P \right)^2 + \frac{\lambda_2}{2} \left(\int_{\Omega} d\Omega - A \right)^2, \tag{13}$$

which depends on the solution y of the problem (11)-(12), on the difference between the perimeter of $\partial\Omega$ and a given target value P and on the difference between the area of Ω and a given target value A .

The set \mathcal{U}_{ad}^k of the admissible configurations is obtained by deforming the upper part of the domain, which is described by the graph of a piecewise linear function defined on a one dimensional grid. Increasing k amounts to decrease the mesh-size of the grid. As shown in Figure 2, the MSQP algorithm, for different values of the finest level K of discretization, is able to efficiently approximate the optimal domain (in this case represented by the unitary square).

(a) Finest level of discretization $K = 1$. Initial configuration (left), after 1 iteration (middle) and 12 iterations (right).



(b) Finest level of discretization $K = 3$. Initial configuration (left), after 1 iteration (middle) and 9 iterations (right).

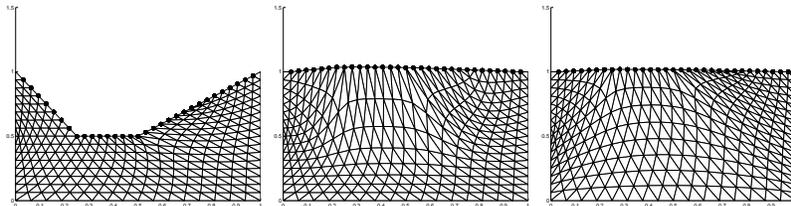


Figure 2: Performance of the MSQP scheme for different values of the finest level K of discretization. The algorithm converges towards the optimal shape represented by the unitary square.

2.2 ASQP: an adaptive shape optimization algorithm

An alternative feasible variant of the ∞ -SQP is represented by the Adaptive Sequential Quadratic Programming (ASQP) algorithm originally introduced in [32]. The ASQP scheme replaces all the non-computable operations of ∞ -SQP (the solution to the state equation (1), the solution to the linear subproblem (5), the values of the functional J and of its derivative dJ in the line search routine) by adaptive finite dimensional approximations, whose accuracies are adjusted relative to the energy decrease for each iteration. It is worth noticing that the adaptive procedure driving ASQP has to deal with two distinct sources of error:

- *PDE Error*: this hinges on the approximation of (1), the values of the functional J and its derivative (3);
- *Geometric Error*: this relates to the approximation of (5) which yields the new domain.

Since it is wasteful to impose a PDE error finer than the expected geometric error, we have a natural mechanism to balance the computational effort.

In the following, we briefly describe the ASQP algorithm (see [32] for more details). Recall that $\ell \geq 1$ stands for the adaptive counter and $\Omega^{(\ell)}$ is the current domain produced by ASQP with deformable boundary $\Gamma^{(\ell)}$. Let $\mathbb{S}^{(\ell)} = \mathbb{S}_{\mathcal{T}^{(\ell)}}(\Omega^{(\ell)})$ and $\mathbb{V}^{(\ell)} = \mathbb{V}_{\mathcal{T}^{(\ell)}}(\Gamma^{(\ell)})$ be the finite element spaces on the bulk and boundary, which are compatible and fully determined by one underlying mesh $\mathcal{T}^{(\ell)}$ of $\Omega^{(\ell)}$. We define ASQP as follows:

ADAPTIVE SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM (ASQP)

Given the initial domain $\Omega^{(0)}$, a triangulation $\mathcal{T}^{(0)}$ of $\Omega^{(0)}$, and the parameters $0 < \theta \leq \frac{1}{5}$, set $\gamma = \frac{1}{2} - \theta(1 + \theta)$, $k = 0$, $\varepsilon^{(0)} = +\infty$, $\mu^{(0)} = 1$, repeat the following steps:

- (1) $[\mathcal{T}^{(\ell)}, U^{(\ell)}, Z^{(\ell)}, J^{(\ell)}, G^{(\ell)}] = \text{APPROXJ}(\Omega^{(\ell)}, \mathcal{T}^{(\ell)}, \varepsilon^{(\ell)})$
- (2) $[\mathbf{V}^{(\ell)}, \mathcal{T}^{(\ell)}] = \text{DIRECTION}(\Omega^{(\ell)}, \mathcal{T}^{(\ell)}, G^{(\ell)}, \theta)$
- (3) $[\Omega^{(\ell+1)}, \mathcal{T}^{(\ell+1)}, \mu^{(\ell+1)}] = \text{LINESEARCH}(\Omega^{(\ell)}, \mathcal{T}^{(\ell)}, \mathbf{V}^{(\ell)}, J^{(\ell)}, \mu^{(\ell)})$
- (4) $\varepsilon^{(\ell+1)} := \gamma \mu^{(\ell+1)} \|\mathbf{V}^{(\ell+1)}\|_{\Gamma^{(\ell)}}^2$; $\ell \leftarrow \ell + 1$.

In theory this algorithm is an infinite loop giving a more accurate approximation as the iterations progress, but in practice we implement a stopping criteria in **LINESEARCH**.

The modules **APPROXJ** and **DIRECTION** are driven by different adaptive strategies and corresponding different tolerances, say a PDE tolerance γ and a geometric tolerance θ . Their relative values allow for different distributions of the computational effort in dealing with the PDE and the geometry.

The routine **DIRECTION** enriches/coarsens the space $\mathbb{V}^{(\ell)}$ to control the quality of the descent direction, guaranteeing a geometric error proportional to $\mu^{(\ell)} \|V^{(\ell)}\|_{\Gamma^{(\ell)}}^2$, namely

$$|J(\Omega^{(\ell)} + \mu^{(\ell)} \mathbf{V}^{(\ell)}) - J(\Omega^{(\ell)} + \mu^{(\ell)} \mathbf{v}^{(\ell)})| \leq \delta \mu^{(\ell)} \|V^{(\ell)}\|_{\Gamma^{(\ell)}}^2, \quad (14)$$

with $\delta := \theta(1 + \theta) \leq \frac{3}{2}\theta$. On the other hand, the module **APPROXJ** enriches/coarsens the space $\mathbb{S}^{(\ell)}$ to control the error in the approximate functional value $J^{(\ell)}(\Omega^{(\ell)} + \mu^{(\ell)} \mathbf{V}^{(\ell)})$ to the prescribed tolerance $\gamma \mu^{(\ell)} \|V^{(\ell)}\|_{\Gamma^{(\ell)}}^2$,

$$|J(\Omega^{(\ell)} + \mu^{(\ell)} \mathbf{V}^{(\ell)}) - J^{(\ell)}(\Omega^{(\ell)} + \mu^{(\ell)} \mathbf{V}^{(\ell)})| \leq \gamma \mu^{(\ell)} \|V^{(\ell)}\|_{\Gamma^{(\ell)}}^2, \quad (15)$$

where $\gamma = \frac{1}{2} - \delta \geq \delta$ prevents excessive numerical resolution relative to the geometric one. This is achieved within the module **APPROXJ** via the *Dual Weighted Residual* method (DWR) [8], tailored to the approximation of the functional value J . The remaining modules perform the following tasks. The module **SOLVE** finds approximate solutions $U^{(\ell)} \in \mathbb{S}^{(\ell)}$ of (1) and $Z^{(\ell)} \in \mathbb{S}^{(\ell)}$ of an adjoint equation (necessary for the computation of the shape derivative) while **RIESZ** builds on $\mathbb{S}^{(\ell)}$ an approximation $G^{(\ell)}$ to the shape derivative. Finally, the module **LINESEARCH** enforces an inexact version of Wolfe's conditions.

We observe that the test (15) is not very demanding for DWR. So we expect coarse meshes at the beginning, and a combination of refinement and coarsening later as DWR detects geometric singularities, such as corners, and sorts out whether they are genuine to the problem or just due to lack of numerical resolution. This aspect of ASQP is a novel paradigm in adaptivity and is detailed in [32].

In the following, we report some numerical examples originally presented in [32] to highlight the main features of the adaptive algorithm. In particular, we consider the drag reduction shape optimization problem governed by Stokes equation (see e.g. [37]). Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with its boundary subdivided into an *inflow* part Γ_{in} , an *outflow* part Γ_{out} , a

part considered as *walls* Γ_w , and an obstacle Γ_s which is the deformable part to be optimized. The velocity $\mathbf{u} := \mathbf{u}(\Omega)$ and the pressure $p := p(\Omega)$ solve the following problem:

$$\begin{aligned} -\operatorname{div}(\mathbf{T}(\mathbf{u}, p)) &= 0 && \text{in } \Omega \\ \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega \\ \mathbf{u} &= \mathbf{u}_d && \text{on } \Gamma_{in} \cup \Gamma_s \cup \Gamma_w \\ \mathbf{T}(\mathbf{u}, p) \boldsymbol{\nu} &= 0 && \text{on } \Gamma_{out} \end{aligned} \tag{16}$$

where $\mathbf{T}(\mathbf{u}, p) := 2\mu\epsilon(\mathbf{u}) - p\mathbf{I}$ is the Cauchy tensor with $\epsilon(\mathbf{u}) = \frac{\nabla\mathbf{u} + \nabla\mathbf{u}^T}{2}$, $\mu > 0$ is the viscosity, and

$$\mathbf{u}_d = \begin{cases} \mathbf{v}_\infty & \text{on } \Gamma_{in} \\ \mathbf{0} & \text{on } \Gamma_w \cup \Gamma_s, \end{cases}$$

with $\mathbf{v}_\infty = V_\infty \hat{\mathbf{v}}_\infty$, $\hat{\mathbf{v}}_\infty$ being the unit vector pointing in the direction of the incoming flow and V_∞ a scalar function.

We let the cost functional measuring the obstacle *drag* be

$$J[\Omega, (\mathbf{u}, p)] := - \int_{\Gamma_s} (\mathbf{T}(\mathbf{u}, p) \boldsymbol{\nu}) \cdot \hat{\mathbf{v}}_\infty \, dS, \tag{17}$$

where (\mathbf{u}, p) solves (16). We would like to minimize the linear boundary functional J subject to the state constraint (16) among all admissible configurations with *fixed volume* that can be obtained by piecewise smooth perturbations of the obstacle boundary Γ_s .

In Figure 3 we report the initial and final optimal configuration. As an effect of the DWR error indicator, the mesh refinement takes place mostly around the deformable shape, whereas in the rest of the domain Ω the mesh is rather coarse.

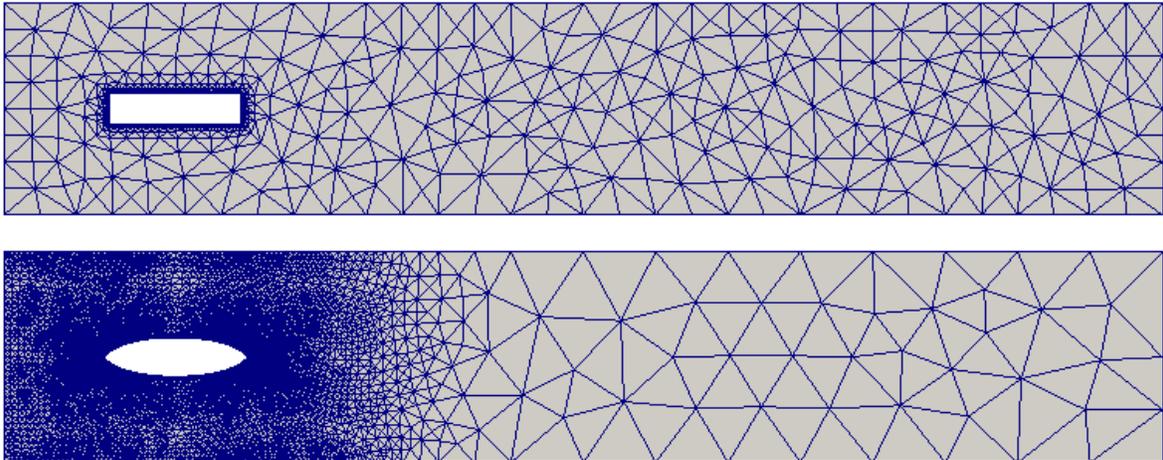


Figure 3: Initial (top) and final (bottom) configuration. The ASQP algorithm obtains the optimal "rugby ball" shape [37]. The mesh refinement takes place mostly around the deformable shape, whereas in the rest of Ω the mesh is rather coarse: this is related to DWR mesh refinement (and coarsening).

In Figures 4-6, we show the efficacy of the adaptive ASQP method to sort out whether a geometric singularity is genuine to the problem or just due to the lack of numerical resolution. In the first case (genuine singularities) the method preserves the singularities and further refine them, whereas in the latter case (non-genuine singularities) the algorithm coarsens the (unnecessarily) over refined regions.

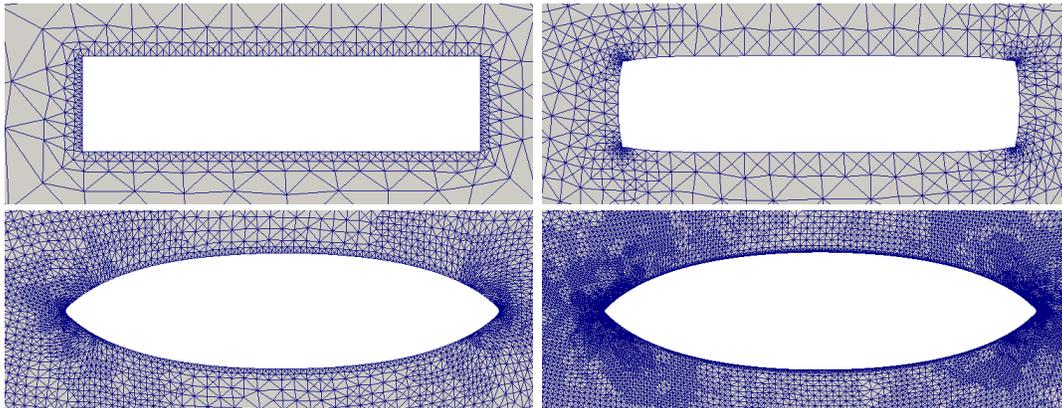


Figure 4: Zoom of the evolution of the deformable shape. The initially refined corners (top) are subsequently smoothed out and coarsened (see Figure 5). The new corners of the rugby ball, instead, are genuine singularities and are preserved and further refined by ASQP (bottom).

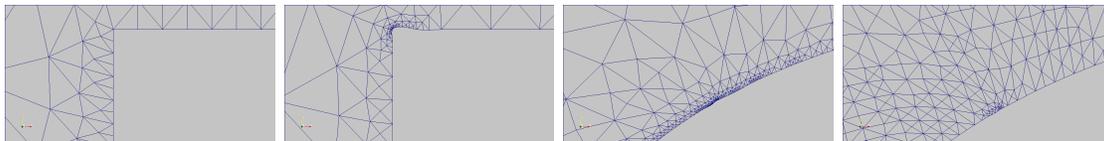


Figure 5: Detection of genuine geometric singularities. Evolution of the initial upper-left corner of the deformable shape (see top of Figures 3 and 4). The adaptive ASQP method is able to sort out whether geometric singularities are genuine to the problem or just due to lack of numerical resolution and to coarsen overrefined regions of the computational grid.

3 Optimal control for evolutive pdes

Let us now turn our attention to optimal control problems for evolutive partial differential equations. The classical approach is based on open-loop controls and on the Pontryagin maximum principle which leads to a backward-forward system characterizing the optimal couple state-control. We have followed a different idea, trying to apply the Dynamic Programming approach. The results presented here are illustrated in [1, 2].

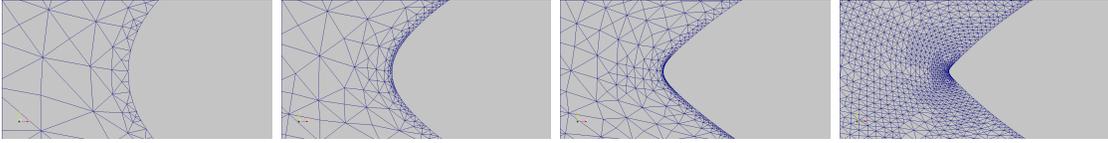


Figure 6: Detection of genuine geometric singularities. Zoom on the evolution of the left-hand part of the deformable shape (see top of Figure 3 and bottom of Figure 4). The adaptive ASQP method is able to recognize the corner of the rugby ball as genuine singularity of the problem and to refine the mesh to improve both the PDE and the Geometric approximation.

3.1 The POD approximation method for evolutive PDEs

We briefly describe some important features of the POD approximation, more details as well as precise results can be found in the notes by Volkwein [43]. Let us consider a matrix $Y \in \mathbb{R}^{m \times n}$, with rank $d \leq \min\{m, n\}$. We will call y_j the j -th column of the matrix Y . We are looking for an orthonormal basis $\{\psi_i\}_{i=1}^\ell \in \mathbb{R}^m$ with $\ell \leq n$ such that the minimum of the following functional is reached:

$$J(\psi_1, \dots, \psi_\ell) = \sum_{j=1}^n \left\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle \psi_i \right\|^2. \quad (18)$$

The solution of this minimization problem is given in the following theorem

Theorem 3.1 *Let $Y = [y_1, \dots, y_n] \in \mathbb{R}^{m \times n}$ be a given matrix with rank $d \leq \min\{m, n\}$. Further, let $Y = \Psi \Sigma V^T$ be the Singular Value Decomposition (SVD) of Y , where $\Psi = [\psi_1, \dots, \psi_m] \in \mathbb{R}^{m \times m}$, $V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$ are orthogonal matrices and the matrix $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal, $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_m\}$. Then, for any $\ell \in \{1, \dots, d\}$ the solution to (18) is given by the left singular vectors $\{\psi_i\}_{i=1}^\ell$, i.e., by the first ℓ columns of Ψ .*

The vectors $\{\psi_i\}_{i=1}^\ell$ will be indicated as the *POD basis of rank ℓ* . This idea is really useful, in fact we get a representation of a solution for the original dynamics solving an equation of lower dimension. Whenever it is possible to compute a POD basis of rank ℓ , we get a problem of lower dimension ℓ which will be of manageable size provided ℓ is very small.

Let us consider the following ODEs system

$$\begin{cases} \dot{y}(s) = Ay(s) + f(s, y(s)), & s \in (0, T] \\ y(0) = y_0 \end{cases} \quad (19)$$

where $y_0 \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times m}$ and $f : [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ is continuous and locally Lipschitz to ensure uniqueness.

The system (19) can be also interpreted as a semidiscrete problem, where the matrix A represents the discretization in space of an elliptic operator, e.g. the Laplace operator. To compute the POD basis functions, first of all we have to construct a time grid $0 \leq t_1 \leq \dots \leq t_n = T$ and we assume to know the solution of (19) at given time t_j , $j = 1, \dots, N$. We call *snapshots* the solution at those fixed times, they will be used to find a proper POD basis. For the moment, let us skip the problem of selecting the snapshots sequence to obtain an

efficient POD basis since this is a rather difficult problem, we refer the interested reader to [25]). Given a snapshots sequence, Theorem 3.1 allows to compute our POD basis, namely, $\{\psi_j\}_{j=1}^\ell$.

Assume we can write the solution in reduced form as

$$y^\ell(s) = \sum_{j=1}^{\ell} y_j^\ell(s) \psi_j = \sum_{j=1}^{\ell} \langle y^\ell(s), \psi_j \rangle \psi_j, \quad \forall s \in [0, T]$$

substituting this formula into (19) we obtain the equivalent dynamics in the reduced coordinate space

$$\begin{cases} \sum_{j=1}^{\ell} \dot{y}_j^\ell(s) \psi_j = \sum_{j=1}^{\ell} y_j^\ell(s) A \psi_j + f(s, y^\ell(s)), & s \in (0, T] \\ \sum_{j=1}^{\ell} y_j^\ell(0) \psi_j = y_0. \end{cases} \quad (20)$$

Our new problem (20) has $\ell \leq m$ unknown coefficient functions which are indicated by $y_j^\ell(s)$, $j = 1, \dots, \ell$. The problem is now in low dimension, using a compact notation we get:

$$\begin{cases} \dot{y}^\ell(s) = A^\ell y^\ell(s) + F(s, y^\ell(s)) \\ y^\ell(0) = y_0^\ell \end{cases}$$

where

$$A^\ell \in \mathbb{R}^{\ell \times \ell} \quad \text{with } (A^\ell)_{ij} = \langle A \psi_i, \psi_j \rangle,$$

$$y^\ell = \begin{pmatrix} y_1^\ell \\ \vdots \\ y_\ell^\ell \end{pmatrix} : [0, T] \rightarrow \mathbb{R}^\ell$$

$$F = (F_1, \dots, F_\ell)^T : [0, T] \times \mathbb{R}^\ell \rightarrow \mathbb{R}^\ell,$$

$$F_i(s, y) = \left\langle f \left(s, \sum_{j=1}^{\ell} y_j \psi_j \right), \psi_i \right\rangle \quad \text{for } s \in [0, T] \quad y = (y_1, \dots, y_\ell) \in \mathbb{R}^\ell,$$

finally obtaining the representation of y_0 in \mathbb{R}^ℓ

$$y_0^\ell = \begin{pmatrix} \langle y_0, \psi_1 \rangle \\ \vdots \\ \langle y_0, \psi_\ell \rangle \end{pmatrix} \in \mathbb{R}^\ell.$$

In order to apply the POD method to our optimal control problem, the number ℓ of POD basis functions plays a crucial role. In fact, we would like to keep ℓ as low as possible still capturing the behavior of the original dynamics. Then, the main question is: how can we

measure the accuracy of our POD approximation? We need to define an accuracy parameter and a good choice is given by the following ratio

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \sigma_i}{\sum_{i=1}^d \sigma_i}. \quad (21)$$

where the σ_i are the the singular value obtained by the SVD. Clearly, when $\mathcal{E}(\ell)$ is close to one this means that the approximation is rather accurate because it keeps the main features of the original dynamics. This is also strictly related to the truncation error due to the projection of y_j onto the space generated by the orthonormal basis $\{\psi_i\}_{i=1}^{\ell}$, in fact:

$$J(\psi_1, \dots, \psi_{\ell}) = \sum_{j=1}^n \left\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle \psi_i \right\|^2 = \sum_{i=\ell+1}^d \sigma_i^2$$

3.2 An optimal control problem via POD approximation

Following [1] we present this approach for the finite horizon control problem. Consider the controlled system

$$\begin{cases} \dot{y}(s) = f(y(s), u(s), s), & s \in (t, T] \\ y(t) = x \in \mathbb{R}^n, \end{cases} \quad (22)$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, we will denote by $y : [t, T] \rightarrow \mathbb{R}^n$ its solution, by u the control $u : [t, T] \rightarrow \mathbb{R}^m$, and by

$$\mathcal{U} = \{u : [0, T] \rightarrow U\}$$

the set of admissible controls where $U \subset \mathbb{R}^m$ is a compact set. Whenever we want to emphasize the dependency of the solution on the control u we will write $y(t; u)$. Assume that there exists a unique solution trajectory for (22) provided the controls are measurable (a precise statement can be found in [4]). For the finite horizon optimal control problem the cost functional will be given by

$$\min_{u \in \mathcal{U}} J_{x,t}(u) := \int_t^T L(y(s, u), u(s), s) e^{-\lambda s} ds + g(y(T)) \quad (23)$$

where $L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathcal{R}$ is the running cost, (x, t) is the initial condition and $\lambda \geq 0$ is the discount factor.

The goal is to find a state-feedback control law $u(t) = \Phi(y(t), t)$, in terms of the state equation $y(t)$, where Φ is the feedback map. To derive optimality conditions we use the well-known *dynamic programming principle* due to Bellman (see [4]). We first define the value function:

$$v(x, t) := \inf_{u \in \mathcal{U}} J_{x,t}(u) \quad (24)$$

Proposition 3.1 (DPP) *For all $x \in \mathbb{R}^n$ and $t \leq \tau \leq T$ then:*

$$v(x, t) = \min_{u \in \mathcal{U}} \left\{ \int_t^{\tau} L(y(s), u(s), s) e^{-\lambda s} ds + v(y(\tau), T - \tau) \right\}. \quad (25)$$

Due to (25) we can derive the *Hamilton-Jacobi-Bellman* equations (HJB):

$$-\frac{\partial v}{\partial t}(y, t) = \min_{u \in U} \{L(y, u, t) + \nabla v(y, t) \cdot f(y, u, t)\}. \quad (26)$$

complemented by the terminal condition $v(x, T) = g(x)$. This is a nonlinear partial differential equation of the first order which is hard to solve analitically although a general theory of weak solutions is available [4]. Rather we can solve it numerically by means of a finite differences or semi-Lagrangian schemes (see the book [20] for a comprehensive analysis of approximation schemes for Hamilton-Jacobi equations). For a semi-Lagrangian discretization one starts by a discrete version of (HJB) by discretizing the underlined control problem and then project the semi-discrete scheme on a grid obtaining the fully discrete scheme

$$\begin{cases} v_i^{n+1} = \min_{u \in U} [\Delta t L(x_i, n\Delta t, u) + I[v^n](x_i + \Delta t F(x_i, t_n, u))] \\ v_i^0 = g(x_i). \end{cases}$$

with $x_i = i\Delta x$, $t_n = n\Delta t$, $v_i^n := v(x_i, t_n)$ and $I[\cdot]$ is an interpolation operator which is necessary to compute the value of v^n at the point $x_i + \Delta t F(x_i, t_n, u)$ (in general, this point will not be a node of the grid). The interested reader will find in [21] a detailed presentation of the scheme and a priori error estimates for its numerical approximation.

It is also important to note that we need to compute the minimum in order to get the value v_i^{n+1} . Since, in general, v^n is not a smooth function, we compute the minimum by means of a minimization method which does not use derivatives (this can be done by the Brent algorithm as in [10]).

The main advantage of this approach is that it allows to compute the optimal feedback via the value function. However, there are two major difficulties: our weak solutions (in the viscosity sense) are in general non-smooth and the approximation in high dimension is not feasible due to the huge amount of data required. The request to solve an HJB in high dimension comes up naturally whenever we want to control evolutive PDEs. Just to give an idea, if we build a grid in $[0, 1] \times [0, 1]$ with a discrete step $\Delta x = 0.01$ we have 10^4 nodes: to solve an HJB in that dimension is simply impossible. The POD method allows us to obtain reduced models even for rather complicated dynamics opening the way to a feasible solution of the HJB equation.

Consider the following abstract problem:

$$\begin{cases} \frac{d}{ds} \langle y(s), \varphi \rangle_H + a(y(s), \varphi) = \langle B(u(s), \varphi) \rangle_{V', V} \quad \forall \varphi \in V \\ y(t) = y_0 \in H, \end{cases} \quad (27)$$

where $B : U \rightarrow V'$ is a linear and continuous operator. We assume that a space of admissible controls \mathcal{U}_{ad} is given in such a way that for each $u \in \mathcal{U}_{ad}$ and $y_0 \in H$ there exists a unique solution y of (27). V and H are two Hilbert spaces, with $\langle \cdot, \cdot \rangle_H$ we denote the scalar product in H ; $a : V \times V \rightarrow \mathcal{R} : is symmetric coercive and bilinear. Then, we introduce the cost functional of the finite horizon problem$

$$\mathcal{J}_{y_0, t}(u) := \int_t^T L(y(s), u(s), s) e^{-\lambda s} ds + g(y(T)),$$

where $L : V \times U \times [0, T] \rightarrow \mathcal{R}$. The optimal control problem is

$$\begin{aligned} & \min_{u \in \mathcal{U}} \mathcal{J}_{y_0, t}(u) & (28) \\ \text{subject to the constraint: } & y \in W_{loc}(0, T; V) \times \mathcal{U} \text{ solves (27)} \end{aligned}$$

with $W_{loc}(0, T) = \bigcap_{T>0} W(0, T)$, where $W(0, T)$ is the standard Sobolev space:

$$W(0, T) = \{\varphi \in L^2(0, T; V), \varphi_t \in L^2(0, T; V')\}.$$

The model reduction approach for an optimal control problem (28) is based on the Galerkin approximation of dynamic with some informations on the controlled dynamic (snapshots). To compute a POD solution for (28) we make the following ansatz

$$y^\ell(x, s) = \sum_{i=1}^{\ell} w_i(s) \psi_i(x). \quad (29)$$

where $\{\psi\}_{i=1}^{\ell}$ is the POD basis. The computation of the POD basis functions follows three easy steps:

1. Computation of the snapshots for the solution at given times, $y(s_j)$.
2. Collect the snapshots into a matrix Y and compute the singular value decomposition of $Y = U\Sigma V^T$
3. Take the first ℓ columns of U , they will be the POD basis of rank ℓ .

Now let us introduce mass and stiffness matrix:

$$\begin{aligned} M &= ((m_{ij})) \in \mathbb{R}^{\ell \times \ell} \text{ with } m_{ij} = \langle \psi_j, \psi_i \rangle_H, \\ S &= ((s_{ij})) \in \mathbb{R}^{\ell \times \ell} \text{ with } s_{ij} = a(\psi_j, \psi_i), \end{aligned}$$

and the control map $b : U \rightarrow \mathbb{R}^\ell$ is defined by:

$$u \rightarrow b(u) = (b(u)_i) \in \mathbb{R}^\ell \text{ with } b(u)_i = \langle Bu, \psi_i \rangle_H.$$

The coefficients of the initial condition $y^\ell(0) \in \mathbb{R}^\ell$ are determined by $w_i(0) = (w_0)_i = \langle y_0, \psi \rangle_X$, $1 \leq i \leq \ell$, and the solution of the reduced dynamic problem is denoted by $w^\ell(s) \in \mathbb{R}^\ell$. Then, the Galerkin approximation is given by

$$\min J_{w_0^\ell, t}^\ell(u) \quad (30)$$

with $u \in \mathcal{U}$ and w solves the following equation:

$$\begin{cases} \dot{w}^\ell(s) = F(w^\ell(s), u(s), s) & s > 0, \\ w^\ell(0) = w_0^\ell. \end{cases} \quad (31)$$

The cost functional is defined as:

$$J_{w_0, t}^\ell(u) = \int_0^T L(w^\ell(s), u(s), s)e^{-\lambda s} dt + g(w^\ell(T)),$$

with w^ℓ and y^ℓ linked to (29) and the nonlinear map $F : \mathbb{R}^\ell \times U \rightarrow \mathbb{R}^\ell$ is given by

$$F(w^\ell, u, s) = M^{-1}(-Sw^\ell(s) + b(u(s))).$$

The value function v^ℓ , defined for the initial state $w_0 \in \mathbb{R}^\ell$, reads as

$$v^\ell(w_0, t) = \inf_{u \in \mathcal{U}} J_{w_0, t}^\ell(u)$$

and w^ℓ solves (30) with the control u and initial condition w_0 .

To complete the secenario, let us explain how we have computed the intervals defining the domain where we are going to solve the HJB equation in reduced coordinate. Clearly we need to restrict the computation to a bounded domain Υ_h in \mathbb{R}^ℓ . We would like to find an invariant domain for the discrete dynamics, i.e. a domain Υ_h such that $y + \Delta t F(y, u) \in \Upsilon_h$ for each $y \in \Upsilon_h$ and $u \in \mathcal{U}$. We can choose $\Upsilon_h = [a_1, b_1] \times [a_2, b_2] \times \dots [a_\ell, b_\ell]$ with $a_1 \geq a_2 \geq \dots \geq a_\ell$. How should we compute the intervals $[a_i, b_i]$?

Ideally the intervals should be chosen so that the dynamics contains all the components of the controlled trajectory. Moreover, they should be encapsulated because we expect that their importance should decrease monotonically with their index and that our interval lengths decrease quickly.

Let us suppose to discretize the space control $U = \{u_1, \dots, u_M\}$ where U is symmetric with respect to the origin, i.e. $\bar{u} \in U$ implies $-\bar{u} \in U$.

Hence, if $y^\ell(s) = \sum_{i=1}^{\ell} \langle y(s), \psi_i \rangle \psi_i = \sum_{i=1}^{\ell} w_i(s) \psi_i$, as a consequence, the coefficients $w_i(s) \in [a_i, b_i]$. We consider the trajectories solution $y(s, u_j)$ such that the control is constant $u(s) \equiv u_j$ for each $t_j, j = 1, \dots, M$. Then, we have

$$y^\ell(s, u_j) = \sum_{i=1}^{\ell} \langle y(s, u_j), \psi_i \rangle \psi_i.$$

We write $y^\ell(s, u_j)$ to stress the dependence on the constant control u_j . Each trajectory $y^\ell(s, u_j)$ corresponds to a set of coefficients $w_i^{(j)}(t)$ for $i = 1, \dots, \ell, j = 1, \dots, M$. Every coefficient $w_i^{(j)}(s)$ belongs to an interval $[\underline{w}_i^{(j)}, \bar{w}_i^{(j)}]$ so, for $i = 1, \dots, \ell,$, we define:

$$a_i \equiv \min\{\underline{w}_i^{(1)}, \dots, \underline{w}_i^{(M)}\}$$

$$b_i \equiv \max\{\bar{w}_i^{(1)}, \dots, \bar{w}_i^{(M)}\}.$$

Note that when we do not find an invariant domain to set up our computation we must introduce appropriate boundary conditions to manage the trajectories leaving the domain (see [19, 20] for more details on this technical problem).

3.3 Numerical experiments

In this section we present some numerical tests for the controlled heat equation and for the advection-diffusion equation with a quadratic cost functional. Consider the following advection-diffusion equation:

$$\begin{cases} y_s(x, s) - \varepsilon y_{xx}(x, s) + cy_x(x, s) = u(s) \\ y(x, 0) = y_0(x), \end{cases} \quad (32)$$

with $x \in [a, b]$, $s \in [0, T]$, $\varepsilon \in \mathbb{R}_+$ and $c \in \mathbb{R}$.

Note that changing the parameters c and ε we can obtain the heat equation ($c = 0$) and the advection equation ($\varepsilon = 0$). The functional to be minimized is

$$J_{y_0, t}(u(\cdot)) = \int_0^T \|y(x, s) - \hat{y}(x, s)\|^2 + R\|u(s)\|^2 ds, \quad (33)$$

i.e. we want to stay close to a reference trajectory \hat{y} while minimizing the norm of u . Note that we dropped the discount factor setting $\lambda = 0$. Typically in our test problems \hat{y} is obtained by applying a particular control \hat{u} to the dynamics. The numerical simulations reported here have been made on a server SUPERMICRO 8045C-3RB with 2 cpu INTEL Xeon Quad-Core 2.4 Ghz and 32 GB RAM under SLURM (<https://computing.llnl.gov/linux/slurm/>).

Test 1: Heat equation with smooth initial data We compute the snapshots with a centered/forward Euler scheme with space step $\Delta x = 0.02$, and time step $\Delta t = 0.012$, $\varepsilon = 1/60$, $c = 0$, $R = 0.01$ and $T = 5$. The initial condition is $y_0(x) = 5x - 5x^2$, and $\hat{y}(x, s) = 0$. In Figure 7 we compare four different approximations concerning the heat equation: (a) is the solution for $\hat{u}(t) = 0$, (b) is its approximation via POD (non-adaptive), (c) is the direct LQR solution computed by MATLAB without POD and, finally, the approximate optimal solution obtained coupling POD and HJB. The approximate value function is computed for $\Delta t = 0.1$ $\Delta x = 0.1$ whereas the optimal trajectory as been obtained with $\Delta t = 0.01$. Test 1, and even Test 2, have been solved in about half an hour of CPU time.

Note that in this example the approximate solution is rather accurate because the regularity of the solution is high due to the diffusion term. Since in the limit the solution tends to the average value, the choice of the snapshots will not affect too much the solution, i.e. even a rough choice of the snapshots will give us a good approximation. The difference between Figure 2c and Figure 2d is due to the fact that the control space is continuous for 2c and discrete for 2d.

Test 2: Heat equation with no-smooth initial data In this section we change the initial condition with a function which is only Lipschitz continuous: $y_0(x) = 1 - |x|$. According to Test 1, we consider the same parameters. (see Figure 8).

Riccati's equation has been solved by a MATLAB LQR routine. Thus, we have used the solution given by this routine as the correct solution in order to compare the errors in L^1 and L^2 norm between the reduced Riccati's equation and our approach based on the reduced HJB equation. Since we do not have any information, the snapshots are computed for $\hat{u} = 0$. This is only a guess, but in the parabolic case it fits well due to the diffusion term.

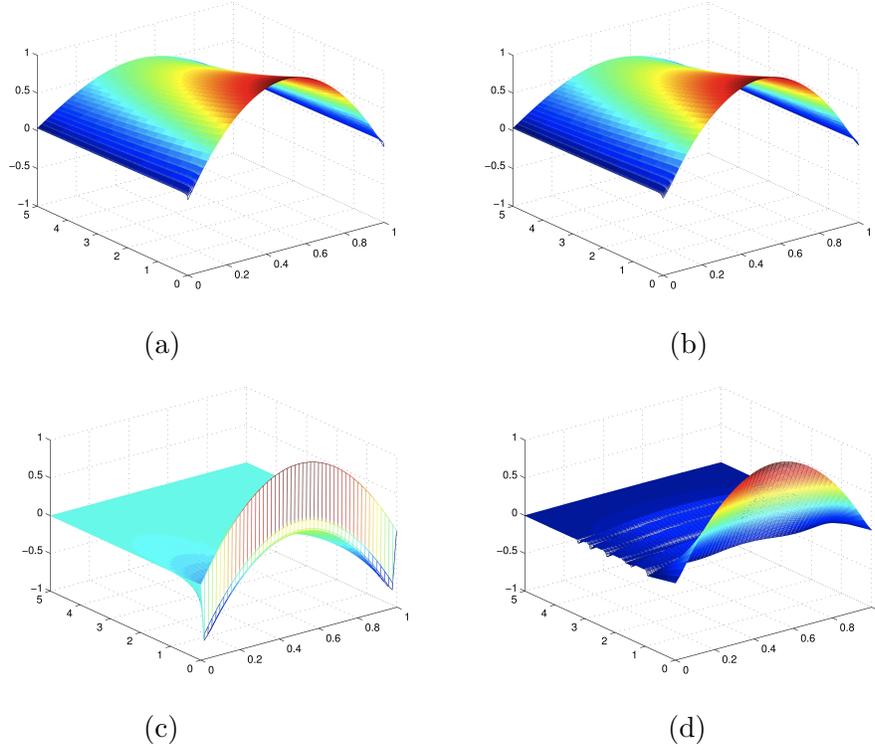
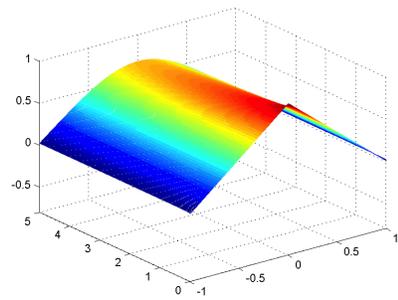


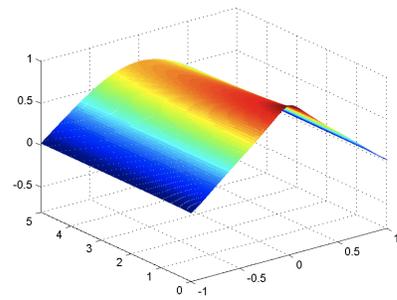
Figure 7: Test 1:(a) Heat Equation without control; (b) Heat Equation without control, 3 POD basis; (c) Controlled solution with LQR-MATLAB; (d) Approximate solution POD (3 basis functions) + HJB.

	L^1	L^2
$y^{LQR} - y^{POD+LQR}$	0.0221	0.0172
$y^{LQR} - y^{POD+HJB}$	0.0204	0.0171

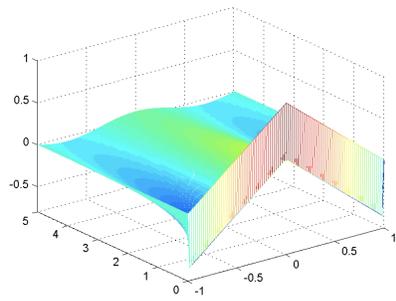
Table 1: Test 2: L^1 and L^2 errors at time T for the optimal approximate solution.



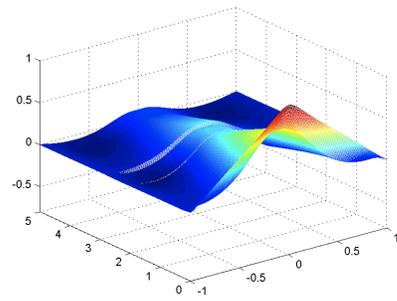
(a)



(b)



(c)



(d)

Figure 8: Test 2: (a) exact solution for $\hat{u} = 0$; (b) Exact solution for $\hat{u} = 0$ POD (3 basis functions); (c) Approximate optimal solution for LQR-MATLAB; (d) Approximate solution POD (3 basis functions)+ HJB.

As in Test 1, the choice of the snapshots does not affect strongly the approximation due to the asymptotic behavior of the solution. The presence of a Lipschitz continuous initial condition has almost no influence on the global error (see Table 1).

4 The adaptive POD approximation method

We now present an adaptive method to compute POD basis. As we have seen in Section 3 we have a big constraint on the number of variables in the state space for numerical solution of an HJB.

For a parabolic equation, one can try to solve the problem with only three/four POD basis functions; they are enough to describe the solution in a rather accurate way. In fact the singular values decay pretty soon and it is rather easy to work with a really low-rank dimensional problem.

On the contrary, hyperbolic equations do not have this nice property and they will need more POD basis functions to get accurate results. Then, it is quite natural to split the problem into subproblems having different POD basis functions. The crucial point is to decide the splitting in order to have the same number of basis functions in each subdomain with a guaranteed accuracy in the approximation.

4.1 Numerical experiments for the adaptive POD approximation method

Let us first give an illustrative example for the parabolic case, considering a 1D advection-diffusion equation:

$$\begin{cases} y_s(x, s) - \varepsilon y_{xx}(x, s) + cy_x(x, s) = 0 \\ y(x, 0) = y_0(x), \end{cases} \quad (34)$$

with $x \in [a, b]$, $s \in [0, T]$, $\varepsilon, c \in \mathbb{R}$.

We use a finite difference approximation for this equation based on an explicit Euler method in time combined with the standard centered approximation of the second order term and with an up-wind correction for the advection term. The snapshots will be taken from the sequence generated by the finite difference method. The final time is $T = 5$, moreover $a = -1$, $b = 4$. The initial condition is $y_0(x) = 5x - 5x^2$, when $0 \leq x \leq 1$, 0 otherwise.

For $\varepsilon = 0.05$ and $c = 1$ with only 3 POD basis functions, the approximation fails (see Figure 9). Note that in this case the advection is dominating the diffusion, a low number of POD basis functions will not suffice to get an accurate approximation (Figure 9.b). However, the adaptive method which only uses 3 POD basis functions will give accurate results (Figure 9.d).

The idea which is behind the adaptive method is rather simple and easy to implement. Instead of taking into account the whole interval $[0, T]$, we prefer to split it in sub-intervals

$$[0, T] = \cup_{k=0}^K [T_k, T_{k+1}]$$

where K is a-priori unknown, $T_0 = 0, T_K = T$ and $T_k = t_i$ for some i . In this way, choosing properly the length of the k -th interval $[T_k, T_{k+1}]$, we consider only the snapshots falling in

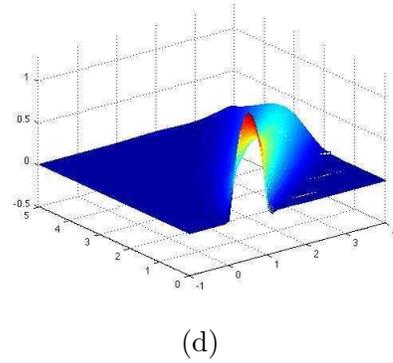
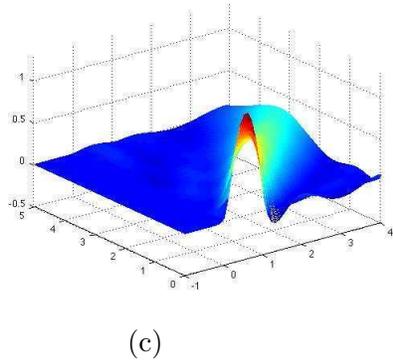
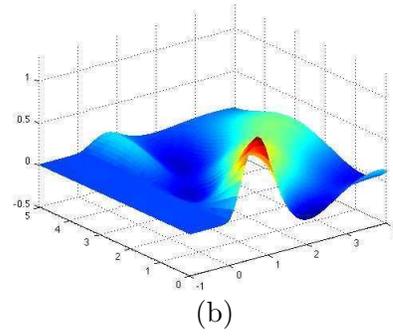
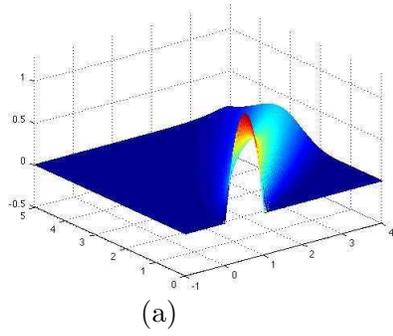


Figure 9: Equation (34):(a) solved with finite difference; (b) POD-Galerkin approximation with 3 basis functions; (c) solved via POD-Galerkin approximation with 5 basis functions; (d) Adaptive POD 3 basis functions.

that sub-interval, typically there will be at least three snapshots in every sub-interval. In this way we will have enough informations in every sub-interval and we can apply the standard routines (explained in Section 3) to get a "local" POD basis.

Now let us explain how to divide our time interval $[0, T]$. We will choose a parameter to check the accuracy of the POD approximation and define a threshold. Above that threshold we loose in accuracy and we need to compute a new POD basis. A good parameter to check the accuracy is $\mathcal{E}(\ell)$ (see (21)), as it was suggested by several authors. The method to define the splitting of $[0, T]$ and the size of every sub-interval works as follows. We start computing the SVD of the matrix Y that gives us informations about our dynamics in the whole time interval. We check the accuracy at every t_i , $i = 1, \dots, N$, and if at t_k the indicator is above the tolerance we set $T_1 = t_k$ and we divide the interval in two parts, $[0, T_1]$ and $(T_1, T]$. Now we just consider the snapshots related the solution up to the time T_1 . We iterate this idea until the indicator is below the threshold. When the first interval is found, we restart the procedure in the interval $[T_1, T]$ and we stop when we reach the final time T . Note that the extrema of every interval coincide by construction with one of our discrete times $t_i = i\Delta t$ so that the global solution is easily obtained linking all the sub-problems which always have a snapshot as initial condition. A low value for the threshold will also guarantee that we will not have big jumps passing from one sub-interval to the next. Once we know we got nice POD basis functions we compute the solution of the problem in each sub-intervals. Moreover, in each intervals $[T_k, T_{k+1}]$ we check the residual of the solution previously computed. If the residual is not below a given threshold, we split again the problem into two subproblems. This two subproblems need to update their own basis functions that will satisfy, of course, the error estimator applied to the POD method, since we are considering only a subset of the snapshots.

This idea can be applied also when we have a controlled dynamic (see [2]). First of all we have to decide how to collect the snapshots, since the control $u(t)$ is completely unknown. One can make a guess and use the dynamics and the functional corresponding to that guess, by these informations we can compute the POD basis. Once the POD basis is obtained we will get the optimal feedback law after having solved a reduced HJB equation as we already explained. Let us summarize the POD adaptive method in the following step-by-step presentation.

ALGORITHM

Start: Inizialization

Step 1: collect the snapshots in $[0, T]$

Step 2: divide $[0, T]$ according to $\mathcal{E}(\ell)$

For $i=0$ to $N-1$

Do

Step 3: apply SVD to get the POD basis in each sub-interval $[t_i, t_{i+1}]$

Step 4: discretize the space of controls

Step 5: project the dynamics onto the (reduced) POD space

Step 6: select the intervals for the POD reduced variables

Step 7: solve the corresponding HJB in the reduced space
for the interval $[t_i, t_{i+1}]$

Step 8: go back to the original coordinate space

End

Test 3: controlled advection-diffusion equation The advection-diffusion equation needs a different method. We can not use the same \hat{y} we had in the parabolic case, mainly because in Riccati's equation the control is free and is not bounded, on the contrary when we solve an HJB we have to discretize the space of controls. We modified the problem in order to deal with bang-bang controls. We get \hat{y} in (33) just plugging in the control $\hat{u} \equiv 0$. We have considered the control space corresponding only to three values in $[-1, 1]$, then $U = \{-1, 0, 1\}$. We first have tried to get a controlled solution, without any adaptive method and, as expected, we obtained a bad approximation (see Figure 10). From Figure 10 it is clear that POD with

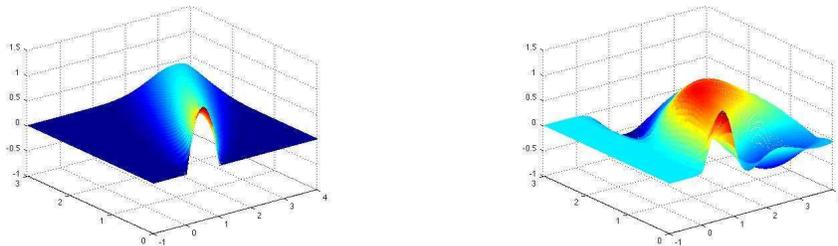


Figure 10: Test 3: Solution \hat{y} (left), approximate solution with POD (4 basis functions) (right).

four basis functions is not able to catch the behavior of the dynamics, so we have applied our adaptive method.

We have consider: $T = 3, \Delta x = 0.1, \Delta t = 0.008, a = -1, b = 4, R = 0.01$. According to our algorithm, the time interval $[0, 3]$ was divided into $[0, 0.744] \cup [0.744, 1.496] \cup [1.496, 3]$. As we can see our last interval is bigger than the others, this is due to the diffusion term (see Figure 11). The L^2 -error is 0.0761, and the computation of the optimal solution via HJB has required about six hours of CPU time. In Figure 4 we compare the exact solution with the numerical solution based on a POD representation. Note that, in this case, the choice of only 4 basis functions for the whole interval $[0, T]$ gives a very poor result due to the presence of the advection term. Looking at Figure 5 one can see the improvement of our adaptive technique which takes always 4 basis functions in each sub-interval.

In order to check the quality of our approximation we have computed the numerical residual, defined as:

$$\mathcal{R}(y) = \|y_s(x, s) - \varepsilon y_{xx}(x, s) + cy_x(x, s) - u(s)\|.$$

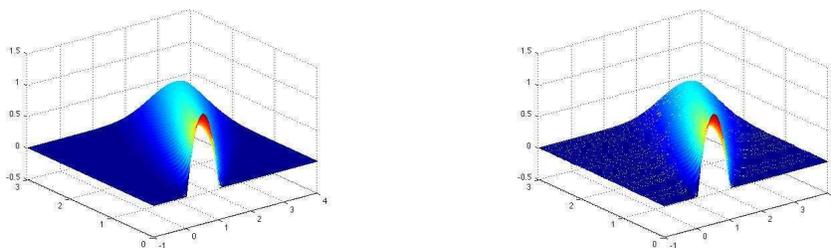


Figure 11: Test 3: Solution for $\hat{u} \equiv 0$ (left), approximate optimal solution (right).

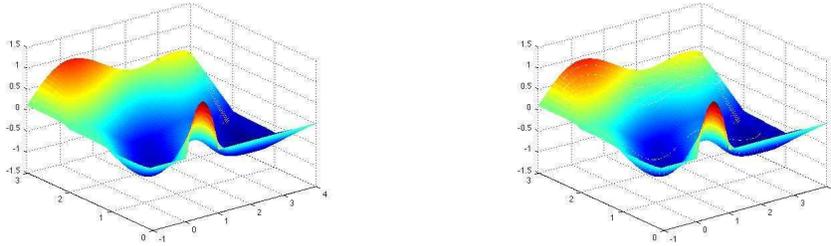


Figure 12: Test 4: Solution for \hat{u} (left), approximate optimal solution (right).

The residual for the solution of the control problem computed without our adaptive technique is 1.1, whereas the residual for the adaptive method is $2 \cdot 10^{-2}$. As expected from the pictures, there is a big difference between these two value.

Test 4: controlled advection-diffusion equation In this test we take a different \hat{y} , namely the solution of (32) corresponding to the control

$$\hat{u}(t) = \begin{cases} -1 & 0 \leq t < 1 \\ 0 & 1 \leq t < 2 \\ 1 & 2 \leq t \leq 3. \end{cases}$$

We want to emphasize we can obtain nice results when the space of controls has few element. The parameters were the same used in Test 3. The L^2 -error is 0.09, and the time was the same we had in Test 3. In Figure 12 we can see our approximation. In Figure 6 one can see that the adaptive technique can also deal with discontinuous controls.

In this test, the residual for the solution of the control problem without our adaptive technique is 2, whereas the residual for the adaptive method is $3 \cdot 10^{-2}$. Again, the residual shows the higher accuracy of the adaptive routine.

5 Conclusions

We presented some recent results concerning the numerical approximation of shape optimization problems and optimal control problems governed by evolutive partial differential equations. In particular, with respect to shape optimization problems, we introduced and discussed two novel techniques, namely a fully geometric multigrid approach and an adaptive sequential quadratic programming algorithm. Several numerical experiments assessed the efficacy of the proposed strategies. Concerning the optimal control of evolutive problems, we detailed how a reasonable coupling between POD and HJB equation can produce feedback controls for infinite dimensional problem. For advection dominated equations that simple idea has to be implemented in a clever way to be successful. In particular, the application of an adaptive technique is crucial to obtain accurate approximations with a low number of POD basis functions. This is still an essential requirement when dealing with the Dynamic Programming approach, which suffers from the curse-of-dimensionality although recent de-

velopments in the methods used for HJB equations will allow to increase this bound in the next future (for example by applying patchy techniques, see [9]).

Another important point is the discretization of the control space. In our examples, the number of optimal control is rather limited and this will be enough for problems which have a bang-bang structure for optimal controls. In general, we will need also an approximation of the control space via reduced basis methods. This point as well as a more detailed analysis of the procedure outlined in this paper will be addressed in our future work.

References

- [1] A. Alla, M. Falcone, *An adaptive POD approximation method for the control of advection-diffusion equations*. to appear in K. Kunisch, K. Bredies, C. Clason and G. Von Winckel (eds.) , "Control and Optimization with PDE Constraints", International Series of Numerical Mathematics, Birkhäuser, Basel, 2013
- [2] A. Alla, M. Falcone, *A time adaptive POD method for optimal control problems*, to appear in the Proceedings of the 1st IFAC CPDE Workshop, Paris, September 2013.
- [3] P.F. Antonietti, A. Borzì, and M. Verani, *Multigrid shape optimization governed by elliptic PDEs*, accepted for publication on SIAM J. Control Optim.
- [4] M. Bardi, I. Capuzzo Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhauser, Basel, 1997.
- [5] A. Borzì, *On the convergence of the MG/OPT method*, PAMM, 5 (1) (2005), 735–736.
- [6] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2011.
- [7] F. Beux and A. Dervieux, *A hierarchical approach for shape optimization*, Engrg. Comput. 11 (1) (1994), 25–48.
- [8] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numer.* 10 (2001),1–102.
- [9] S. Cacace, E. Cristiani, M. Falcone, A. Picarelli. *A patchy dynamic programming scheme for a class of Hamilton-Jacobi-Bellman equations*, SIAM J. Sci. Comp., 34 (2012), 26252649.
- [10] E. Carlini, M. Falcone, R. Ferretti. *An efficient algorithm for Hamilton-Jacobi equations in high dimension*. Computing and Visualization in Science, Vol.7, No.1 (2004), 15-29.
- [11] J.-A. Désidéri, B.A. El Majd and A. Janka, *Nested and self-adaptive Bézier parameterizations for shape optimization*, J. Comput. Phys., 224 (1) (2007), 117–131.
- [12] J.-A. Désidéri, *Hierarchical optimum-shape algorithms using embedded Bezier parameterizations*. In: Y. Kuznetsov et al., Editors, Numerical Methods for Scientific Computing, Variational Problems and Applications, CIMNE, Barcelona (2003).

- [13] J.-A. Désidéri and A. Janka, *Multilevel shape parameterization for aerodynamic optimization application to drag and noise reduction of transonic/supersonic business jet*, in: E. Heikkola et al., (Ed.), European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004).
- [14] J.-A. Désidéri, *Two-level ideal algorithm for parametric shape optimization*, in: W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, Y. Vassilevski (Eds.), *Advances in Numerical Mathematics, Proceedings of International Conferences, Moscow (2005)*.
- [15] J.-A. Désidéri and A. Dervieux, *Hierarchical methods for shape optimization in aerodynamics - I: multilevel algorithms for parametric shape optimization*, in: J. Periaux and H. Deconinck, (Eds.), *Introduction to Optimization and Multidisciplinary Design, Lecture Series 2006-3*, Von Karman Institute for Fluid Dynamics Publish., 2006.
- [16] F. de Gournay, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, *SIAM J. Control Optim.* 45 (2006) 343-367.
- [17] M.C. Delfour and J.-P. Zolesio, *Shapes and geometries analysis, differential calculus, and optimization*, SIAM, 2011.
- [18] G. Dogan, P. Morin, R.H. Nochetto and M. Verani, *Discrete gradient flows for shape optimization and applications*, *Comput. Methods Appl. Mech. Engrg.* 196 (2007) 3898-3914.
- [19] M. Falcone. *Numerical solution of dynamic programming equations*, Appendix of the book M. Bardi, I. Capuzzo Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Birkhäuser, Boston, 1997, 471-504.
- [20] M. Falcone, R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, SIAM, book to appear
- [21] M. Falcone, T. Giorgi. *An approximation scheme for evolutive Hamilton-Jacobi equations*, in W.M. McEneaney, G. Yin and Q. Zhang (eds.), "Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming", Birkhäuser, 1999, 289-303.
- [22] A. Henrot and M. Pierre, *Variation et optimisation de formes*, volume 48 of *Mathématiques et Applications*. Springer, 2005.
- [23] K. Kunisch, S. Volkwein. *Control of Burgers' Equation by a Reduced Order Approach using Proper Orthogonal Decomposition*. *Journal of Optimization Theory and Applications*, 102 (1999), 345- 371.
- [24] K. Kunisch, S. Volkwein. *Galerkin proper orthogonal decomposition methods for parabolic problems* *Numer. Math.* 90 (2001), 117-148.
- [25] K. Kunisch, S. Volkwein. *Optimal snapshot location for computing POD basis functions* *ESAIM: M2AN* 44 (2010), 509-529.

- [26] K. Kunisch, S. Volkwein, L. Xie. *HJB-POD Based Feedback Design for the Optimal Control of Evolution Problems*. SIAM J. on Applied Dynamical Systems, 4 (2004), 701-722.
- [27] K. Kunisch, L. Xie. *POD-Based Feedback Control of Burgers Equation by Solving the Evolutionary HJB Equation*, Computers and Mathematics with Applications. **49** (2005), 1113-1126.
- [28] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, volume 15 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), 2008.
- [29] R. M. Lewis and S. G. Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM Journal on Scientific Computing, 26 (6) (2005), 1811–1837.
- [30] J.L. Lions, *Optimal control of systems governed by partial differential equations*, Springer, Berlin, 1971.
- [31] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*, Oxford University Press, 2001.
- [32] P. Morin, R.H. Nochetto, S. Pauletti and M. Verani, *Adaptive finite element method for shape optimization*, ESAIM: Control, Optimisation and Calculus of Variations, ESAIM: Control, Optimisation and Calculus of Variations, 18 (4) (2012), 1122-1149
- [33] S. G. Nash, *A multigrid approach to discretized optimization problems*, Optimization Methods and Software, 14 (1 & 2) (2000), 99–116.
- [34] P. Neittanmaki P., D. Tiba, *Optimal Control of Nonlinear Parabolic Systems: Theory, Algorithms and Applications*, Dekker, New York, 1994.
- [35] A. T. Patera, G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2006.
- [36] O. Pironneau, *Optimal shape design for elliptic systems*. Springer Series in Computational Physics. Springer-Verlag, New York, 1984.
- [37] O. Pironneau, On optimum profiles in Stokes flow, *J. Fluid Mech.* 59 (1973), 117–128.
- [38] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization*, Shape Sensitivity Analysis, Springer Verlag, Berlin, 1992.
- [39] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and applications*, AMS 2010.
- [40] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
- [41] M. Vallejos and A. Borzì, *Multigrid optimization methods for linear and bilinear elliptic optimal control problems*, Computing, 82 (2008), 31-52.

- [42] J.C. Vassberg, and A. Jameson, *Aerodynamic Shape Optimization*, Part I & II, Von Karman Institute, Brussels, Belgium, 2006.
- [43] S. Volkwein, *Model Reduction using Proper Orthogonal Decomposition*, 2011
www.math.uni-konstanz.de/numerik/personen/volkwein/index.php

MOX Technical Reports, last issues

Dipartimento di Matematica “F. Brioschi”,
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 22/2013** FALCONE, M.; VERANI, M.
Recent Results in Shape Optimization and Optimal Control for PDEs
- 20/2013** AZZIMONTI, L.; NOBILE, F.; SANGALLI, L.M.; SECCHI, P.
Mixed Finite Elements for spatial regression with PDE penalization
- 19/2013** AZZIMONTI, L.; SANGALLI, L.M.; SECCHI, P.; DOMANIN, M.; NOBILE, F.
Blood flow velocity field estimation via spatial regression with PDE penalization
- 21/2013** PEROTTO, S.; VENEZIANI, A.
Coupled model and grid adaptivity in hierarchical reduction of elliptic problems
- 18/2013** DISCACCIATI, M.; GERVASIO, P.; QUARTERONI, A.
Interface Control Domain Decomposition (ICDD) Methods for Coupled Diffusion and Advection-Diffusion Problems
- 17/2013** CHEN, P.; QUARTERONI, A.
Accurate and efficient evaluation of failure probability for partial differential equations with random input data
- 16/2013** FAGGIANO, E. ; LORENZI, T. ; QUARTERONI, A.
Metal Artifact Reduction in Computed Tomography Images by Variational inpainting Methods
- 15/2013** ANTONIETTI, P.F.; GIANI, S.; HOUSTON, P.
Domain Decomposition Preconditioners for Discontinuous Galerkin Methods for Elliptic Problems on Complicated Domains
- 14/2013** GIANNI ARIOLI, FILIPPO GAZZOLA
A new mathematical explanation of the Tacoma Narrows Bridge collapse
- 13/2013** PINI, A.; VANTINI, S.
The Interval Testing Procedure: Inference for Functional Data Controlling the Family Wise Error Rate on Intervals.