Optimization Methods for Dirichlet Control Problems

Mariano Mateos

Dpto. de Matemáticas, Universidad de Oviedo, Campus de Gijón, 33203 Gijón, Spain

ARTICLE HISTORY
Compiled October 19, 2018

ABSTRACT
We discuss several optimization procedures to solve finite element approximations of linear-quadratic Dirichlet optimal control problems governed by an elliptic partial differential equation posed on a 2D or 3D Lipschitz domain. The control is discretized explicitely using continuous piecewise linear approximations. Unconstrained, control-constrained, state-constrained and control-and-state constrained problems are analyzed. A preconditioned conjugate method for a reduced problem in the control variable is proposed to solve the unconstrained problem, whereas semismooth Newton methods are discussed for the solution of constrained problems. State constraints are treated via a Moreau-Yosida penalization. Convergence is studied for both the continuous problems and the finite dimensional approximations. In the finite dimensional case, we are able to show convergence of the optimization procedures even in the absence of Tikhonov regularization parameter. Computational aspects are also treated and several numerical examples are included to illustrate the theoretical results.

KEYWORDS
Dirichlet optimal control, discretization, constrained optimization, preconditionate conjugate gradient, semismooth Newton methods

AMS CLASSIFICATION
49J20, 49M05, 49M15, 65N30

1. Introduction

Only in the last ten years it has been possible to develop a systematic study of Dirichlet optimal control problems governed by elliptic equations, which started with the seminal paper [1]. In that work a 2D control-constrained problem governed by a semilinear equation posed on a convex polygonal domain is studied. Several other works have been published about numerical error estimates; see [2] or [3, Ch 2.1] for a variational approach to control-constrained problems posed on smooth 2D or 3D domains, [4] for a superconvergence result on the state approximation for unconstrained 2D problems, [5] for control-and-state constrained problems posed on convex polygonal domains. In [6] the authors study the control-constrained problem in a 2D smooth convex domain taking into account the problems derived by the boundary approximation and in [7] an apparent paradox between [2] and [6] is explained. The regularity of the solution in

Email: mmateos@uniovi.es.

possibly nonconvex polygonal plane domains is studied in [8]; see also the introduction of that paper for further references about related problems. In the recent publication [9], error estimates for a Dirichlet control problem governed by a parabolic equation are obtained. In this work, the spatial discretization of the control is studied in both the cases of continuous piecewise linear finite elements and variational approach.

Just before that, several works dealing with efficient optimization methods for control or state constrained problems had appeared; in [10–16] the semismooth Newton method is thoroughly studied. Since all these papers about optimization are previous to the papers about the numerical analysis of Dirichlet control problems, very little or no reference is made in them to their applicability to the problems we are going to deal with. Only in [16] two different Dirichlet control problems with more regular solutions than the ones we treat here are studied in the infinite-dimensional case. Let us also mention that in [17] the Dirichlet boundary condition is replaced by a Robin penalization. For this kind of penalization the methods developed in the aforementioned references are directly applicable. In [18], the semismooth Newton method is studied in the context of the variational approach of the control. Although the authors only exemplify their results through distributed and Robin boundary control, a combination of their results and some of the results we present in Section 4 can be applied to Dirichlet control. See Remark 3 below.

In this work we describe optimization methods for Dirichlet control problems in both the infinite and finite dimensional cases. Convergence proofs, examples and practical implementation details are discussed for all the algorithms through the paper.

In Section 2 we state the problem and prove that it is well posed in Lipschitz domains; see Lemma 2.1. So far, only smooth, convex, polygonal or polyhedral domains had been studied.

Next, we discretize the problem. As is usual in control problems, we have three choices to discretize the control: the first option is not to discretize the control, using a variational discretization as introduced in [19] for distributed problems; as a second option we may use piecewise constant functions; and finally we can use continuous piecewise linear functions.

The choice is not trivial because first order optimality conditions for Dirichlet control problems involve the normal derivative of the adjoint state.

If we discretize directly the optimality system using e.g. continuous piecewise linear finite elements, we would obtain two different approximations of the control: the trace of the discrete state would be continuous piecewise linear and the normal derivative of the discrete adjoint state would be piecewise constant. In [20, Ch. 3.2.7.3] and in [3, Example 2.1.11] this difficulty is solved using a mixed formulation of the state equation which is discretized with the lowest order Raviart-Thomas element. In [21] a convergence proof for this kind of discretization is given. In this way, both the trace of the discrete state and the normal derivative of the discrete adjoint state are piecewise constant functions, so the identification of both with the discrete control is meaningful. The discretization of the control in the presence of control constraints is carried out in these papers using a variational discretization. This kind of approximation may be convenient when the gradient of the state is the variable of interest, since this quantity is treated as one of the variables of the problem.

Another approach, in finite dimension, is to use the variational discrete normal derivative of the discrete adjoint state introduced in [1], which is a continuous piecewise linear function. Doing so, both the trace of the discrete state and the normal derivative of the discrete adjoint state are continuous piecewise linear functions, so the identification of both with the discrete control is meaningful. Following this idea, in [2] the control is not discretized for the control-constrained case, but a variational approach is followed.

In this work we investigate optimization methods for the case of discretizing the control explicitely using continuous piecewise linear functions.

The end of Section 2 is devoted to describe with some detail some computational aspects that will be important in the rest of the work.

We next provide in Section 3 an efficient method to solve the unconstrained problem. We propose in Subsection 3.2 a preconditioned conjugate gradient method (pcg in the rest of the work) for a reduced problem in the spirit of [22, Section 5]. We are able to prove convergence of the conjugate gradient even for the case where the Tikhonov regularization parameter ν vanishes.

In sections 4, 5 and 6 we study the convergence of the semismooth Newton method for the constrained problems and write practical algorithms for the solution of the finite dimensional approximations. In Section 4 we deal with the control-constrained problem. In many of the aforementioned references about the semismooth Newton method for control-constrained problems the authors study convergence of an abstract problem or an infinite dimensional problem. For distributed and Neumann control problems the results are immediately applicable to the finite dimensional approximations because the controls can be discretized using piecewise constant functions, and therefore the variational inequality arising from first order necessary optimality conditions can be written in an element-wise form. The same idea applies when we are dealing with a variational approach as proposed in [2, 3, 19] or a mixed formulation, as studied in [3, 20, 21]. When we use continuous piecewise linear elements,the variational inequality cannot be written in a point-wise or elementwise form; see (36d) and Remark 4. We include the analysis of Newton methods for both the original problem and the discretized one. Local convergence results are provided in Theorems 4.2, 4.3 and 4.5.

In Section 5 we study the state-constrained problem using a Moreau-Yosida penalization. Since there are no control constraints, the analysis of the semismooth Newton method for the infinite dimensional problem is applicable to the finite dimensional one, so we do not need to repeat it. We prove that the finite-element approximation of the penalized problems converge to the solution of the penalized problem. This result cannot be deduced straightforward from the ones in the literature since the penalized functional is not of class C^2. A continuation strategy as proposed in [16] is developed. Finally, in Section 6 we discuss the problem with both control and state constraints.

It is well known that the main difficulty with Dirichlet control problems is the low regularity of the solutions. This regularity and related error estimates are, so far, well established in 2D polygonal domains [1, 5, 8] and 2D or 3D smooth domains [2] but there is not, up to our best knowledge, a general study in 3D polyhedral domains. Although the main focus of this work is on optimization methods, we also study the regularity of the solution and error estimates of the approximations in one example case in a polyhedron; see Example 3.6.

2. Statement of the problem and first properties

Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or $d = 3$, be a bounded domain with Lipschitz boundary Γ and in this domain consider a target state $y_\Omega \in L^2(\Omega)$. Consider also the continuous linear operator $S : L^2(\Gamma) \to L^2(\Omega)$ such that $y = Su$ if and only if y is the solution in the transposition sense (see Definition 2.2 below) of

$$-\Delta y = 0 \text{ in } \Omega,\ y = u \text{ on } \Gamma. \tag{1}$$

Let $\omega \subset \Omega$ be a domain such that $\bar{\omega} \subset \Omega$ and define two pairs of functions $\alpha,\ \beta \in C(\Gamma)$ and $a,\ b \in C(\bar{\omega})$ such that $\alpha(x) < \beta(x)$ for all $x \in \Gamma$ and $a(x) < b(x)$ for all $x \in \bar{\omega}$. For some fixed regularization parameter $\nu \geq 0$, define

$$J(u) = \frac{1}{2}\|Su - y_\Omega\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Gamma)}$$

and consider the sets

$$U_{\alpha,\beta} = \{u \in L^2(\Gamma) :\ \alpha(x) \leq u(x) \leq \beta(x) \text{ for a.e. } x \in \Gamma\},$$

and

$$K_{a,b} = \{y \in L^2(\Omega) \cap C(\bar{\omega}) :\ a(x) \leq y(x) \leq \beta(x) \text{ for a.e. } x \in \bar{\omega}\}.$$

In this work we will study optimization procedures for the following four Dirichlet control problems:

$$(P^{\mathrm{U}}) \min_{u \in L^2(\Gamma)} J(u), \quad (P^{\mathrm{C}}) \min_{u \in U_{\alpha,\beta}} J(u), \quad (P^{\mathrm{S}}) \min_{Su \in K_{a,b}} J(u), \quad (P^{\mathrm{CS}}) \min_{\substack{u \in U_{\alpha,\beta} \\ Su \in K_{a,b}}} J(u),$$

namely the unconstrained, control-constrained, state-constrained and control-and-state constrained problems.

Remark 1. Almost all the literature related to these problems is written using the state equation (1). There would be no problem in taking into account an equation of the form

$$Ay := -\sum_{i=1}^{d} \partial_i(a_{i,j}\partial_j y) + a_0 y = F \text{ in } \Omega,\ y = u + G \text{ on } \Gamma$$

with regular enough F, G, $a_0 \geq 0$ and $a_{i,j} = a_{j,i}$ satysfing an uniform ellipticity condition.

Let us state precisely what we mean by solution in the transposition sense. Consider the space

$$\Phi = \{\phi : \phi \in H^1_0(\Omega) \text{ and } \Delta\phi \in L^2(\Omega)\}.$$

This is a Banach space with the graph norm $\|\phi\|_\Phi = \|\phi\|_{H^1(\Omega)} + \|\Delta\phi\|_{L^2(\Omega)}$. Further, the functions in this space satisfy $\partial_n\phi \in L^2(\Gamma)$. This is known to be true for smooth domains; convex domains, see [17, Lemma A.1]; plane polygonal domains, see [8, Corollary 2.3]; or polyhedral domains, see [23, Theorem 2.6.7] and the usual trace theorem. We have not been able to find a proof of this fact for general Lipschitz domains. In [24, Theorem B.2] the regularity $\phi \in H^{3/2}(\Omega)$ is proved. Nevertheless, as the authors notice in page 165 of this reference, the trace theorem is not valid neither in $H^{3/2}(\Omega)$ nor in $H^{1/2}(\Omega)$, so we cannot deduce immediately that $\partial_n\phi \in L^2(\Gamma)$. The results in [24, 25] imply that the usual trace result can be extended to harmonic functions in the limit cases. We show next how to take advantage of this to prove that $\partial_n\phi \in L^2(\Gamma)$ in Lipschitz domains. Regarding the analysis of semismooth Newton methods –see Lemma 4.1 below– we also prove $L^q(\Gamma)$ regularity for some $q > 2$. Sobolev spaces on the boundary are defined following [26, Definition 1.3.3.2]; cf. also the equivalent definition of $W^{1,q}(\Gamma)$ given before Theorem 5.6 in [24]. With this definition, it is clear that $H^t(\Gamma) = \mathrm{tr}H^{1/2+t}(\Omega)$ for $0 < t < 1$; see [26, Theorem 1.5.12].

Lemma 2.1. *Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or $d = 3$, be a bounded domain with Lipschitz boundary Γ and consider $\phi \in \Phi$. Then, there exists $q_0 > 2$ depending on the domain such that, for $2 \le q < q_0$, we have $\partial_n\phi \in L^q(\Gamma)$ and*

$$\|\partial_n\phi\|_{L^q(\Gamma)} \le C\|\Delta\phi\|_{L^2(\Omega)}. \tag{2}$$

If, further, Ω is smooth or convex or polygonal or polyhedral, then there exists $t > 0$ such that $\partial_n\phi \in H^t(\Gamma)$ and

$$\|\partial_n\phi\|_{H^t(\Gamma)} \le C\|\Delta\phi\|_{L^2(\Omega)}. \tag{3}$$

Proof. Denote $z = -\Delta\phi$, extend z by zero to $\mathbb{R}^d$, consider the Newtonian potential centered at the origin $N(x)$ and define $w = z * N$, the convolution product of z and N. Then $w \in H^2(\Omega)$ and $\nabla w \in H^1(\Omega)^d$, so it is clear that $\mathrm{Tr}(\nabla w) \in H^{1/2}(\Gamma)^d \hookrightarrow L^q(\Gamma)^d$ for all $q < \infty$ if $d = 2$ and all $q \le 4$ if $d = 3$, since the dimension of Γ is $d - 1$. This implies that:

(a) $\partial_n w = \nabla w \cdot n \in L^q(\Gamma)$ because the boundary Γ is Lipschitz and therefore it has a unit normal vector defined almost everywhere; and there exists $C > 0$ such that

$$\|\partial_n w\|_{L^q(\Gamma)} \le C\|\Delta\phi\|_{L^2(\Omega)}; \tag{4}$$

(b) $g = \mathrm{Tr}(w) \in W^{1,q}(\Gamma)$ due precisely to the definition of $W^{1,q}(\Gamma)$, and there exists $C > 0$ such that

$$\|g\|_{W^{1,q}(\Gamma)} \le C\|\Delta\phi\|_{L^2(\Omega)}. \tag{5}$$

Define now $v \in H^1(\Omega)$ the unique solution of $-\Delta v = 0$ in Ω, $v = g$ on Γ.

Following [27], for $s \in \Gamma$ and $\alpha > 0$, we define the nontangential cone $T_\alpha(s) = \{x \in \Omega,\ |x - s| < (1 + \alpha)\mathrm{dist}(x, \Gamma)\}$, and the nontangential maximal function of some function w as $M(w)(s) = \sup\{|w(x)| :\ x \in T_1(s)\}$. Existence of nontangential limit means that $\lim_{x\to s,\, x\in T_\alpha(s)} w(x)$ exists and is finite for all $\alpha > 0$.

Using [24, Theorem 5.6], we have that there exists $q_0 > 2$ such that if $2 \le q < q_0$, then the nontangential maximal function of the gradient of v satisfies $\mathrm{M}(\nabla v) \in L^q(\Gamma)$ and there exists $C > 0$ such that

$$\|\mathrm{M}(\nabla v)\|_{L^q(\Gamma)} \le C\|g\|_{W^{1,q}(\Gamma)}. \tag{6}$$

As is pointed out in [27, p. 438], this implies that ∇v has nontangential limit a.e. on Γ and we can define the normal derivative of v at a point $s \in \Gamma$ as the nontangential limit as $x \to s$ of $\nabla v(x) \cdot n(s)$. This, together with inequalities (6) and (5) imply that $\partial_n v \in L^q(\Omega)$ and

$$\|\partial_n v\|_{L^q(\Gamma)} \leq C\|\Delta\phi\|_{L^2(\Omega)}. \tag{7}$$

So we have that $\phi = v - w$ and we can define in a natural way $\partial_n\phi = \partial_n v - \partial_n w \in L^q(\Gamma)$. The estimate (2) follows from (4) and (7).

For smooth, convex, polygonal or polyhedral domains, the second result follows from the regularity $\phi \in H^{3/2+t}(\Omega)$ and the usual trace theorem for $\nabla\phi$. See [17, Lemma A.1] for convex domains, [8, Corollary 2.3] for plane polygonal domains and [23, Theorem 2.6.7] for polyhedral domains. □

Definition 2.2. We will say that $y \in L^2(\Omega)$ is the solution in the transposition sense of (1) if

$$(y, -\Delta\phi)_\Omega = -(u, \partial_n\phi)_\Gamma \text{ for all } \phi \in \Phi. \tag{8}$$

Here and in the rest of the work $(\cdot,\cdot)_X$ stands for the standard inner product in $L^2(X)$.

The adjoint operator of S is $S^* : L^2(\Omega) \to L^2(\Gamma)$ defined by $S^*z = -\partial_n\phi$, where ϕ is the unique weak solution of

$$-\Delta\phi = z \text{ in } \Omega,\ \ \phi = 0 \text{ on } \Gamma. \tag{9}$$

We can write now

$$J(u) \;=\; \frac{1}{2}(Su - y_\Omega, Su - y_\Omega)_\Omega + \frac{\nu}{2}(u,u)_\Gamma = \frac{1}{2}(S^*Su + \nu u, u)_\Gamma - (S^*y_\Omega, u)_\Gamma + c_\Omega$$

where $c_\Omega = 0.5\|y_\Omega\|^2_{L^2(\Omega)}$ is a constant.

From this expression, we can easily compute the derivative of J at a point $u \in L^2(\Gamma)$ in the direction $v \in L^2(\Gamma)$:

$$J'(u)v = (S^*Su + \nu u, v)_\Gamma - (S^*y_\Omega, v)_\Gamma.$$

For later use, we will define now for every $u \in L^2(\Gamma)$, $y_u = Su \in H^{1/2}(\Gamma)$ and $\varphi_u \in H^1_0(\Omega)$ the unique solution of

$$-\Delta\varphi_u = y_u - y_\Omega \text{ in } \Omega,\ \ \varphi_u = 0 \text{ on } \Gamma.$$

Discretization.

To discretize the problems, consider $\{\mathcal{T}_h\}_h$ a regular family of triangulations of $\bar\Omega$. To simplify the notation, we will suppose that Γ is polygonal or polyhedral. Related to the mesh, we will call N the number of nodes and $\{x_j\}_{j=1}^N$ the nodes of the mesh. We define the sets of interior indexes, boundary indexes and indexes in $\bar\omega$ as $\mathbb{I} = \{j : x_j \in \Omega\}$, $\mathbb{B} = \{j : x_j \in \Gamma\}$ and $\mathbb{J} = \{j : x_j \in \bar\omega\}$. For the discretization of the state and the adjoint state we use the space of linear finite elements $Y_h \subset H^1(\Omega)$,

$$Y_h = \{y_h \in C(\bar\Omega) \colon y_h \in P^1(T)\ \forall T \in \mathcal{T}_h\}.$$

As usual, we will abbreviate $Y_{h0} = Y_h \cap H_0^1(\Omega)$. For the control we use the space U_h of continuous piecewise linear functions on Γ

$$U_h = \{u_h \in C(\Gamma) \colon u_h \in P^1(T \cap \Gamma)\ \forall T \in \mathcal{T}_h\}.$$

Notice that the elements of U_h are the traces of the elements of Y_h. We will denote $I_h : C(\bar{\Omega}) \to Y_h$ or $I_h : C(\Gamma) \to U_h$ the interpolation operator related to these spaces and $\Pi_h : L^2(\Omega) \to Y_h$ or $\Pi_h : L^2(\Gamma) \to U_h$ the projection onto this spaces in the L^2 sense. For all $y \in L^2(\Omega)$ and all $u \in L^2(\Gamma)$:

$$(\Pi_h y, y_h)_\Omega = (y, y_h)_\Omega\ \forall y_h \in Y_h \text{ and } (\Pi_h u, u_h)_\Gamma = (u, u_h)_\Gamma\ \forall u_h \in U_h.$$

We discretize the state equation following the work by Berggren [28]: define $S_h : L^2(\Gamma) \to L^2(\Omega)$ such that for $u \in L^2(\Gamma)$, $y_h = S_h u$ if and only if $y_h \in Y_h$ is the unique solution of

$$a(y_h, z_h) = 0\ \forall z_h \in Y_{h0},\ (y_h, v_h)_\Gamma = (u, v_h)_\Gamma\ \forall v_h \in U_h, \tag{10}$$

where $a(\cdot,\cdot)$ is the bilinear form associated to the operator in the PDE. This definition of S_h corresponds to performing a L^2-projection of u data to U_h and prescribing the result as boundary data for the discrete state. In the case of the Laplace operator $a(y, z) = \int_\Omega \nabla^T y \nabla z dx$. The discrete functional is thus defined as

$$J_h(u) = \frac{1}{2}(S_h u - y_\Omega, S_h u - y_\Omega)_\Omega + \frac{\nu}{2}(u, u)_\Gamma.$$

We define now

$$U^h_{\alpha,\beta} = \{u_h \in U_h :\ \alpha(x_j) \leq u_h(x_j) \leq \beta(x_j)\ \forall j \in \mathbb{B}\},$$

and

$$K^h_{a,b} = \{y_h \in Y_h :\ a(x_j) \leq y_h(x_j) \leq b(x_j)\ \forall j \in \mathbb{J}\}.$$

The discrete problems read thus as

$$(P_h^{\mathrm{U}}) \min_{u_h \in U_h} J_h(u_h),\ (P_h^{\mathrm{C}}) \min_{u \in U^h_{\alpha,\beta}} J_h(u_h),\ (P_h^{\mathrm{S}}) \min_{S_h u_h \in K^h_{a,b}} J_h(u_h),\ (P_h^{\mathrm{CS}}) \min_{\substack{u_h \in U^h_{\alpha,\beta} \\ S_h u_h \in K^h_{a,b}}} J_h(u_h).$$

The adjoint operator of S_h is given by the discrete variational normal derivative. See [1]. $S_h^* : L^2(\Omega) \to L^2(\Gamma)$ and $w_h = S_h^* y$ if $w_h \in U_h$ satisfies

$$(w_h, z_h)_\Gamma = (y, z_h)_\Omega - a(z_h, \phi_h) \text{ for all } z_h \in Y_h, \tag{11}$$

where $\phi_h \in Y_{h0}$ is the unique solution of

$$a(z_h, \phi_h) = (y, z_h)_\Omega \text{ for all } z_h \in Y_{h0}. \tag{12}$$

It is customary to write $S_h^* y = -\partial_n^h \phi_h$. For $u \in L^2(\Gamma)$ and $y \in L^2(\Omega)$, we have

$$(S_h u, y)_\Omega = (u, S_h^* y)_\Gamma.$$

We can then write

$$J_h(u) = \frac{1}{2}(S_h^* S_h u + \nu u, u)_\Gamma - (S_h^* y_\Omega, u)_\Gamma + c_\Omega.$$

The computation of the derivative of J_h at a point $u \in L^2(\Gamma)$ in the direction $v \in L^2(\Gamma)$ is then obtained with

$$J_h'(u)v = (S_h^* S_h u + \nu u, v)_\Gamma - (S_h^* y_\Omega, v)_\Gamma.$$

Since our final objective is to optimize in U_h, let us see in more detail how to make the computations in this space.

Consider $\{e_j\}_1^N$ the nodal basis in Y_h, where N is the dimension of Y_h and satisfies $N = N_\mathbb{I} + N_\mathbb{B}$, the latter being respectively the number of interior and boundary nodes. With an abuse of notation, we will also denote e_j the restriction of e_j to Γ. Define the usual finite element stress, mass and boundary mass matrices by

$$\mathcal{K}_{i,j} = a(e_i, e_j),\ \mathcal{M}_{i,j} = (e_i, e_j)_\Omega,\ \mathcal{B}_{i,j} = (e_i, e_j)_\Gamma \text{ for } 1 \le i, j \le N.$$

We will also use $\mathcal{I} \in \mathbb{R}^{N\times N}$ for the identity matrix, $\mathcal{O} \in \mathbb{R}^{N\times N}$ for the zero matrix and usually refer to submatrices as $\mathcal{K}_{\mathbb{I},\mathbb{I}}$ or $\mathcal{K}_{\mathbb{I},:}$ defined by taking the rows or columns designed by the sets of indexes in the subscripts, the semicolon meaning “all the indexes”. For instace, the usual boundary mass matrix is $\mathcal{B}_{\mathbb{B},\mathbb{B}}$.

Given $u_h = \sum_{j\in\mathbb{B}} u_j e_j$, we denote $\boldsymbol{u} \in \mathbb{R}^{N_\mathbb{B}\times 1}$ the vector $(u_1, \ldots, u_{N_\mathbb{B}})^T$ and for $y_h = \sum_{j=1}^N y_j e_j$ we denote $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ the vector $(y_1, \ldots, y_N)^T$. Using (10), we have that $y_h = S_h u_h$ iff

$$\begin{bmatrix} \mathcal{K}_{\mathbb{I},\mathbb{I}} & \mathcal{K}_{\mathbb{I},\mathbb{B}} \\ \mathcal{O}_{\mathbb{B},\mathbb{I}} & \mathcal{B}_{\mathbb{B},\mathbb{B}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_\mathbb{I} \\ \boldsymbol{y}_\mathbb{B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} \end{bmatrix}. \tag{13}$$

Since $\mathcal{B}_{\mathbb{B},\mathbb{B}}$ is nonsingular, we can write this as

$$\begin{array}{rcr} \mathcal{K}_{\mathbb{I},\mathbb{I}}\boldsymbol{y}_\mathbb{I} & = & -\mathcal{K}_{\mathbb{I},\mathbb{B}}\boldsymbol{u}, \\ \boldsymbol{y}_\mathbb{B} & = & \boldsymbol{u}. \end{array} \tag{14}$$

If we define $\mathcal{S} \in \mathbb{R}^{N_\mathbb{B}\times N}$ as

$$\mathcal{S} = \begin{bmatrix} \mathcal{K}_{\mathbb{I},\mathbb{I}} & \mathcal{K}_{\mathbb{I},\mathbb{B}} \\ \mathcal{O}_{\mathbb{B},\mathbb{I}} & \mathcal{I}_{\mathbb{B},\mathbb{B}} \end{bmatrix}^{-1} \mathcal{I}_{:,\mathbb{B}}$$

we have that $y_h = S_h u_h$ if and only if $\boldsymbol{y} = \mathcal{S}\boldsymbol{u}$.

Given $y_h \in Y_h$, let $\phi_h = \sum_{j\in\mathbb{I}} \phi_j e_j$ be the solution of (12) for $y = y_h$. Denoting $\boldsymbol{\phi} \in \mathbb{R}^{N_\mathbb{I}\times 1}$ the corresponding vector, it can be computed as the solution of

$$\mathcal{K}_{\mathbb{I},\mathbb{I}}\boldsymbol{\phi} = \mathcal{M}_{\mathbb{I},:}\boldsymbol{y}. \tag{15}$$

Then we could compute $\boldsymbol{w} \in \mathbb{R}^{N_{\mathbb{B}}\times 1}$, the vector whose components are the coefficients of the (minus) discrete normal derivative $-\partial_n^h \phi_h = S_h^* y_h = w_h = \sum_{j\in\mathbb{B}} w_j e_j$ solving the system

$$\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{w} = \mathcal{M}_{\mathbb{B},:}\boldsymbol{y} - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi}. \tag{16}$$

Formally, we can also write that $\boldsymbol{w} = \mathcal{B}_{\mathbb{B},\mathbb{B}}^{-1}\mathcal{S}^T\mathcal{M}\boldsymbol{y}$. To finish this section we also define the matrix $\mathcal{A} \in \mathbb{R}^{N_{\mathbb{B}},\times N_{\mathbb{B}}}$ and the vector $\boldsymbol{f} \in \mathbb{R}^{N_{\mathbb{B}}\times 1}$ by

$$\mathcal{A}_{i,j} = (S_h^* S_h e_i + \nu e_i, e_j)_\Gamma \text{ and } f_i = (S_h^* y_\Omega, e_i)_\Gamma.$$

We have that

$$J_h(u_h) = \frac{1}{2}\boldsymbol{u}^T\mathcal{A}\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u} + c_\Omega. \tag{17}$$

We also note that

$$\mathcal{A} = \mathcal{S}^T\mathcal{M}\mathcal{S} + \nu\mathcal{B}_{\mathbb{B}\mathbb{B}}. \tag{18}$$

To compute the vector $\boldsymbol{f}$, we consider the projection of y_Ω onto Y_h in the $L^2(\Omega)$ sense, $y_{\Omega,h} = \Pi_h y_\Omega$ and denote $\boldsymbol{y}_\Omega \in \mathbb{R}^{N\times 1}$, the vector whose j-th component is $y_{\Omega,h}(x_j)$, and

$$\boldsymbol{f} = \mathcal{S}^T\mathcal{M}\boldsymbol{y}_\Omega. \tag{19}$$

So for $u_h, v_h \in U_h$, the latter represented by the vector $\boldsymbol{v}$, we can compute

$$J'(u_h)v_h = \boldsymbol{v}^T\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{w} + \nu\boldsymbol{v}^T\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} - \boldsymbol{v}^T\boldsymbol{f}.$$

Notice that applying the equality (16), the explicit computation of $\boldsymbol{w}$ is not necessary, and we can write

$$J'(u_h)v_h = \boldsymbol{v}^T(\mathcal{M}_{\mathbb{B},:}\boldsymbol{y} - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} - \boldsymbol{f}).$$

3. Unconstrained problem

Problem (P^{U}) has a unique solution $\bar{u} \in L^2(\Gamma)$ that satisfies $J'(\bar{u}) = 0$. For every $0 < h < h_0$, problem (P_h^{U}) has also a unique solution $\bar{u}_h$ that satisfies $J_h'(\bar{u}_h) = 0$. The problems being convex, these conditions are also sufficient. Moreover it is known that $\bar{u}_h \to \bar{u}$ strongly in $L^2(\Gamma)$ and also error estimates are available in terms of the mesh size h if $\nu > 0$ and the domain is either 2D and polygonal or smooth. See [1, 2, 4–6, 29].

Let us comment two different approaches for the solution of the FE approximation of (P^{U}).

3.1. Solve the optimality system for the state, the adjoint state and the control.

We write first order optimality conditions for (P_h^{U}). There exists some $h_0 > 0$ such that for every $0 < h < h_0$, there exist unique $\bar{u}_h \in U_h$, $\bar{y}_h \in Y_h$ and $\bar{\varphi}_h \in Y_{h0}$ satisfying the optimality system:

$$\begin{aligned} a(\bar{y}_h, z_h) &= 0 \text{ for all } z_h \in Y_{h0}, && \text{(20a)}\\ \bar{y}_h &\equiv \bar{u}_h \text{ on } \Gamma, && \text{(20b)}\\ a(z_h, \bar{\varphi}_h) &= (\bar{y}_h - y_\Omega, z_h)_\Omega \text{ for all } z_h \in Y_{h0}, && \text{(20c)}\\ \nu\bar{u}_h &\equiv \partial_n^h \bar{\varphi}_h \text{ on } \Gamma. && \text{(20d)} \end{aligned}$$

Taking into account the definition of discrete variational normal derivative and relations (14)–(16), we can write this optimality system as

$$\begin{bmatrix} \mathcal{K}_{\mathbb{I},\mathbb{I}} & \mathcal{O}_{\mathbb{I},\mathbb{B}} & \mathcal{K}_{\mathbb{I},\mathbb{B}} & \mathcal{O}_{\mathbb{I},\mathbb{I}} \\ \mathcal{O}_{\mathbb{B},\mathbb{I}} & \mathcal{I}_{\mathbb{B},\mathbb{B}} & -\mathcal{I}_{\mathbb{B},\mathbb{B}} & \mathcal{O}_{\mathbb{B},\mathbb{I}} \\ -\mathcal{M}_{\mathbb{I},\mathbb{I}} & -\mathcal{M}_{\mathbb{I},\mathbb{B}} & \mathcal{O}_{\mathbb{I},\mathbb{B}} & \mathcal{K}_{\mathbb{I},\mathbb{I}} \\ \mathcal{M}_{\mathbb{B},\mathbb{I}} & \mathcal{M}_{\mathbb{B},\mathbb{B}} & \nu\mathcal{B}_{\mathbb{B},\mathbb{B}} & -\mathcal{K}_{\mathbb{B},\mathbb{I}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_{\mathbb{I}} \\ \boldsymbol{y}_{\mathbb{B}} \\ \boldsymbol{u} \\ \boldsymbol{\varphi}_{\mathbb{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\mathbb{I}} \\ \mathbf{0}_{\mathbb{B}} \\ -\mathcal{M}_{\mathbb{I},:}\boldsymbol{y}_\Omega \\ \mathcal{M}_{\mathbb{B},:}\boldsymbol{y}_\Omega \end{bmatrix}.$$

We may eliminate $\boldsymbol{u}$ with the boundary condition, and reorder the equations to get the linear system

$$\begin{bmatrix} \mathcal{M} + \nu\mathcal{B} & -\mathcal{K}_{:,\mathbb{I}} \\ -\mathcal{K}_{\mathbb{I},:} & \mathcal{O}_{\mathbb{I},\mathbb{I}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\varphi}_{\mathbb{I}} \end{bmatrix} = \begin{bmatrix} \mathcal{M}\boldsymbol{y}_\Omega \\ \mathbf{0} \end{bmatrix}. \tag{21}$$

Notice that system (21) is solvable even for $\nu = 0$ (it is equivalent to (22), and $\mathcal{A}$ is symmetric and positive definite for $\nu \geq 0$; see Lemma 3.1 below.). Solving this equation would completely solve the problem for the unconstrained case. When the discretization is very fine, the number of unknowns may make the solution of the system by direct methods too difficult. The preconditioned conjugate gradient method for this kind of linear systems is studied by Schöberl and Zulehner in [30] and Herzog and Sachs in [31]. A preconditioner can be built using matrices representing scalars products in Y_h and Y_{h0}. Following Algorithm 1 in the last-mentioned reference, at each iteration three linear systems must be solved: two of size N and one of size $N_{\mathbb{I}}$. In [31, Algorithm 3], the systems are solved using a multigrid method.

Reduced problems in the adjoint state variable and related pcg have also been studied in [32]. Nevertheless, the structure of the matrix we have in (21) is different to the one treated in that reference and application of their results is not straightforward.

3.2. Use an iterative method to solve a reduced problem in the control variable.

Let us see another way of solving (P_h^{U}). Using (17), first order optimality conditions can also be written as

$$\mathcal{A}\boldsymbol{u} = \boldsymbol{f}. \tag{22}$$

Lemma 3.1. *The matrix $\mathcal{A}$ is symmetric and positive definite for all $\nu \geq 0$ and there exists $C > 0$ independent of h and ν such that its condition number is bounded by*

$$\kappa(\mathcal{A}) \leq \frac{C\frac{\lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}})}{\lambda_1(\mathcal{M})}\lambda_N(\mathcal{M}) + \nu\lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}})}{\lambda_1(\mathcal{M}) + \nu\lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}})}, \tag{23}$$

where $0 < \lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}}) < \lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}})$ and $0 < \lambda_1(\mathcal{M}) < \lambda_N(\mathcal{M})$ are respectively the smallest and greatest eigenvalues of the matrices $\mathcal{B}_{\mathbb{B},\mathbb{B}}$ and $\mathcal{M}$.

Proof. It is clear from (18) that $\mathcal{A}$ is symmetric. The mass matrices $\mathcal{M}$ and $\mathcal{B}_{\mathbb{B},\mathbb{B}}$ are symmetric and positive definite and therefore $\lambda_1(\mathcal{M}) > 0$ and $\lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}}) > 0$. Since the boundary components of $\mathcal{S}\boldsymbol{u}$ are exactly $\boldsymbol{u}$, we have that $\|\mathcal{S}\boldsymbol{u}\|_{\mathbb{R}^N} \geq \|\boldsymbol{u}\|_{\mathbb{R}^{N_{\mathbb{B}}}}$ and hence

$$\begin{aligned} \boldsymbol{u}^T\mathcal{A}\boldsymbol{u} &= \boldsymbol{u}^T(\mathcal{S}^T\mathcal{M}\mathcal{S} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}})\boldsymbol{u} = (\mathcal{S}\boldsymbol{u})^T\mathcal{M}(\mathcal{S}\boldsymbol{u}) + \nu\boldsymbol{u}^T\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} \\ &\geq (\lambda_1(\mathcal{M}) + \nu\lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}}))\|\boldsymbol{u}\|^2_{\mathbb{R}^{N_{\mathbb{B}}}} \end{aligned} \tag{24}$$

so $\mathcal{A}$ is positive definite.

From [4, Equation (3.8)], we know that there exists some $C > 0$ such that $\|S_h u_h\|_{L^2(\Omega)} \leq \sqrt{C}\|u_h\|_{L^2(\Gamma)}$. Since

$$\|S_h u_h\|^2_{L^2(\Omega)} = (\mathcal{S}\boldsymbol{u})^T\mathcal{M}(\mathcal{S}\boldsymbol{u}) \geq \lambda_1(\mathcal{M})\|\mathcal{S}\boldsymbol{u}\|^2_{\mathbb{R}^N}$$

and

$$\|u_h\|^2_{L^2(\Gamma)} = \boldsymbol{u}\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} \leq \lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}})\|\boldsymbol{u}\|^2_{\mathbb{R}^{N_{\mathbb{B}}}},$$

we obtain

$$\boldsymbol{u}^T\mathcal{A}\boldsymbol{u} = (\mathcal{S}\boldsymbol{u})^T\mathcal{M}(\mathcal{S}\boldsymbol{u}) + \nu\boldsymbol{u}^T\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u} \leq (C\frac{\lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}})}{\lambda_1(\mathcal{M})}\lambda_N(\mathcal{M}) + \nu\lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}}))\|\boldsymbol{u}\|^2_{\mathbb{R}^{N_{\mathbb{B}}}}.$$

The bound for the condition number follows directly from the last inequality and (24). □

For a graded family of meshes with grading parameter $0 < \mu \leq 1$ (see [33]), we usually define $h = N^{-1/d}$, N being the number of nodes of the mesh and d the dimension of Ω. The case $\mu = 1$ corresponds to that of a quasi-uniform mesh family.

Corollary 3.2. *If $\mathcal{T}_h$ is a family of graded meshes with grading parameter $0 < \mu \leq 1$, then there exists $C > 0$ independent of h and ν such that*

$$\kappa(\mathcal{A}) \leq Ch^{(2d-1)\left(1-\frac{1}{\mu}\right)}\frac{1 + \nu h^{\left(1-\frac{1}{\mu}\right)d}}{h^{\frac{1}{\mu}} + \nu}. \tag{25}$$

In particular, for a quasi-uniform family, there exists $C > 0$ such that

$$\kappa(\mathcal{A}) \leq C\frac{1+\nu}{\nu + h}. \tag{26}$$

Proof. from [34] and [33], we have that there exists constants $0 < C_1 < C_2$

$$\lambda_{N_{\mathbb{B}}}(\mathcal{B}_{\mathbb{B},\mathbb{B}}) \leq C_2 h^{d-1},\ \lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}}) \geq C_1 h^{\frac{d-1}{\mu}},\ \lambda_N(\mathcal{M}) \leq C_2 h^d,\ \lambda_1(\mathcal{M}) \geq C_1 h^{\frac{d}{\mu}}$$

Estimate (25) follows then from (23). Estimate (26) follows from (25) for $\mu = 1$. □

The explicit computation of the matrix $\mathcal{A}$ is out of the question, since it requires the solution of $2N_{\mathbb{B}}$ systems of size $N_{\mathbb{I}}$. Much better performance can be achieved using an iterative method which only requires the computation of $\boldsymbol{d} = \mathcal{A}\boldsymbol{u}$. This is shown in Algorithm 1.

Algorithm 1: Function $\boldsymbol{d} = \mathcal{A}\boldsymbol{u}$

```
solve (14) for $\boldsymbol{y}$
solve (15) for $\boldsymbol{\phi}$
set $\boldsymbol{d} = \mathcal{M}_{\mathbb{B},:}\boldsymbol{y} - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u}$
```

Preconditioned conjugate gradient methods can be carried out at the cost of one evaluation of $\mathcal{A}\boldsymbol{u}$ per iteration. The computation of $\boldsymbol{f}$ can be done in a similar way, but of course it only must be computed once; see Algorithm 2.

Algorithm 2: Computation of $\boldsymbol{f}$

```
define $\boldsymbol{y}_\Omega$ by $y_j = y_{\Omega,h}(x_j)$ for $j = 1 : N$
set $\boldsymbol{y} = \boldsymbol{y}_\Omega$ in (15) and solve for $\boldsymbol{\phi}$
set $\boldsymbol{f} = \mathcal{M}_{\mathbb{B},:}\boldsymbol{y}_\Omega - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi}$
```

To finish this section, let us say a word about a preconditioner to solve (22). In practice, matrices like $\mathcal{B}_{\mathbb{B},\mathbb{B}}$ or $\mathcal{P} = \mathcal{M}_{\mathbb{B},\mathbb{B}} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}$ make acceptable preconditioners. This is specially important when using graded meshes; see Example 3.5 below. Notice that at each pcg iterate, we only need to solve two linear systems, each of size $N_{\mathbb{I}}$, to compute $\mathcal{A}\boldsymbol{u}$, plus another one of size $N_{\mathbb{B}}$ to compute $\mathcal{P}^{-1}\boldsymbol{r}$.

Example 3.3. All the code for all the examples has been made by the author using Matlab R2015b and has been run on a desktop PC with Windows 10 Pro 64bits with 16GB RAM DIMM 1333Mhz on an Intel Core i7 CP 870@2.93Ghz.

In this example we compare the different execution times, t_D and t_P, that we obtain when we solve the optimality system using a direct solver for equation (21) or using an iterative method, the preconditioned conjugate gradient method in our case, to solve the reduced problem (22). We have used respectively Matlab builtin commands `mldivide` and `pcg`.

We use the example in [5], where Ω the interior of the convex hull of the points $(-0.5, -0.5)$, $(0.5, -0.5)$, $(0.5, 0)$, $(0, 0.5)$, $(-0.5, 0.5)$; $y_\Omega \equiv 1$ and $\nu = 1$.

A rough initial mesh is built with Matlab PDEToolbox (version 2.1) commands `initmesh` and `'hmax'` set to 0.2 and subsequent nested meshes are obtained by regular diadic refinement using `refinemesh`. We use our own code to assemble the matrices $\mathcal{K}$, $\mathcal{M}$ and $\mathcal{B}$.

For the pcg we use as initial guess the null vector, a relative tolerance of 1E-10 and the preconditioner $\mathcal{P} = \mathcal{M}_{\mathbb{B},\mathbb{B}} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}$. At each iteration we have to solve the linear systems (14) and (15). To do this, we first obtain the Cholesky factors of $\mathcal{K}_{\mathbb{I},\mathbb{I}}$ so that at each iteration we only have to solve triangular systems. The interior mesh nodes were ordered using a symmetric approximate minimum degree permutation in order to minimize the number of nonzero elements in the Cholesky factors of $\mathcal{K}_{\mathbb{I},\mathbb{I}}$ using Matlab command `symamd`. An analogous reordering of the boundary nodes is done

Table 1. Execution times for the 2D problem in Example 3.3. $\nu = 1$.

h	N	$N_{\mathbb{I}}$	$N_{\mathbb{B}}$	t_D (s)	t_P (s)	pcg	$J(\bar{u}_h)$
0.2×2^{-4}	10657	10337	320	0.5	0.2	7	0.3470689275
0.2×2^{-5}	42305	41665	640	4.1	1.0	7	0.3471023330
0.2×2^{-6}	168577	167297	1280	46.1	5.9	7	0.3471129922
0.2×2^{-7}	673025	670465	2560	453.0	31.0	7	0.3471163823
0.2×2^{-8}	2689537	2684417	5120	∞	177.5	7	0.3471174585

to optimize the sparsity pattern of the Cholesky factors of the preconditioner. This reorderings and factorizations take more or less the same cputime than each single iteration of the pcg. This time is included in t_P.

For reference and possible double-check, we also report on the optimal value of the functional. We have marked with ∞ the experiments in which we have run out of RAM.

Example 3.4. In this example we show that the number of pcg iterations for each fixed h is independent of the value of $\nu \geq 0$ for all $\nu < \nu_0$. We take the same domain and meshes as in the 2D problem in Example 3.3 and set $y_\Omega(x) = |x|^2$, so that $J(\bar{u}) > 0$ even for $\nu = 0$. The results are summarized in Table 2.

Table 2. Dependence of the number of pcg iterations w.r.t. the regularization parameter in Example 3.4.

$h \setminus \nu$	1E4	1E2	1	1E-2	1E-4	1E-6	0
0.2×2^{-5}	2	3	7	26	49	49	49
0.2×2^{-6}	2	3	7	28	63	65	65
0.2×2^{-7}	2	3	7	28	81	86	86

This example and the previous one show that choosing to use a direct or an iterative method depends strongly on the mesh size and the number of iterations required for the pcg to converge to a prescribed accuracy. The computation time for pcg is proportional to the number of iterates needed. Comparing Tables 1 and 2 shows that for $h > 0$ not very small and $\nu > 0$ small enough, the direct method will be faster than the iterative one. For instance, for $h = 0.2 \times 2^{-5}$ the direct method will be faster for $\nu < 1\text{E} - 2$; for $h = 0.2 \times 2^{-6}$, the critical ν would be slightly greater than $1\text{E} - 4$; and for smaller h it seems preferable to use pcg.

Example 3.5. In this example we show the effect of the use of $\mathcal{P} = \mathcal{M}_{\mathbb{B},\mathbb{B}} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}$ as a preconditioner to solve (22). In this case $\nu = 0.01$, $\Omega = \{x = re^{i\theta} \in \mathbb{R}^2 : r < 1,\ 0 < \theta < 11\pi/12\}$, and we use a mesh family graded at the origin with parameter $\mu = 13/33 \approx 0.4$; see [33]. In Table 3 we compare the number of iterations used to reach the prescribed accuracy of 1E−10 without and with preconditioner. This example shows very clearly the effectiveness of this strategy of preconditioning.

Table 3. Number of conjugate gradient iterations without and with preconditioner in Example 3.5. Graded mesh and $\nu = 0.01$.

N	$N_{\mathbb{I}}$	$N_{\mathbb{B}}$	No $\mathcal{P}$	$\mathcal{P}$	$J(\bar{u}_h)$
289	225	64	46	26	0.0228920077
1089	961	128	103	24	0.0230219261
4225	3969	256	214	24	0.0231321072
16641	16129	512	421	24	0.0231982941
66049	65025	1024	733	26	0.0232287479

Example 3.6. A 3D example in a polyhedron. Up to our best knowledge, there is not a general theory about regularity of the solutions or approximation results for 3D Dirichlet control problems posed on polyhedral domains. In [2], the authors study a semi-discrete or variational approach to 3D problems on regular domains. Although the semi-discrete approach coincides with the full approach for unconstrained problems, we cannot profit their results since the regularity of the solutions in a smooth domain is (see [2, Lemma 2.1]) much higher than the one we may obtain in polyhedral domains.

For this example we will take Ω the unit cube $(-0.5, 0.5)^3$, $y_\Omega \equiv 1$ and $\nu = 1$. First, we obtain a regularity result for the solution of our problem.

Proposition 3.7. *Let $\Omega \subset \mathbb{R}^3$ be the interior of a rectangular parallelepiped and $y_\Omega \in L^p(\Omega)$ for some $3 < p < +\infty$. Consider $\bar{u}$ the solution of problem (P^{U}). Then, $\bar{u} \in W^{1-1/p}(\Gamma)$, $\bar{y} \in W^{1,p}(\Omega)$ and $\bar{\varphi} \in W^{2,p}(\Omega)$. Moreover, $\bar{u} \equiv 0$ on the edges and corners of Γ.*

Proof. Since $\bar{u} \in L^2(\Gamma)$, using transposition and interpolation it is clear that $y \in H^{1/2}(\Omega)$. Classical Sobolev embedding in 3D leads to $y \in L^3(\Omega)$. Since Ω is a parallelepiped and y_Ω is regular enough, [35, Theorem 1] states that $\bar{\varphi} \in W^{2,3}(\Omega)$, and hence $\partial_n\bar{\varphi} \in \Pi_i W^{2/3,3}(\Gamma_i)$, where Γ_i, $i = 1 : 6$, are the faces of Γ. This does not imply immediately that $\bar{u}$ belongs to $W^{2/3,3}(\Gamma)$ because $2/3 \cdot 3 = 2$, which is the topological dimension of Γ, and some integral compatibility condition should be imposed on the edges; but for all $q < 3$, it is true that $\bar{u} \in W^{1-1/q,q}(\Gamma)$, and therefore $\bar{y} \in W^{1,q}(\Omega)$. Using again Sobolev embeddings, we have that $\bar{y} \in L^s(\Omega)$ for all $s < +\infty$ and hence $\bar{y} - y_\Omega \in L^p(\Omega)$. Applying once again [35, Theorem 1], we have that $\bar{\varphi} \in W^{2,p}(\Omega)$.

Now we have that $\partial_n\bar{\varphi} \in \Pi_i W^{1-1/p,p}(\Gamma_i)$ and we can prove that if we define $\partial_n\bar{\varphi} = 0$ on the edges of Ω, then we obtain a continuous function. To do this, we use a similar argument to the one used in [7, Section 4] for 2D problems. Since $p > 3$, $\bar{\varphi} \in C^1(\bar{\Omega})$. Consider two faces A and B with a common edge AB and let τ_X^1, τ_X^2 be two linearly independent vectors tangent to face X ($X = A$ or $X = B$) such that $\tau_A^1 = \tau_B^1$ is also tangent to the edge AB. Since $\bar{\varphi} = 0$ on Γ, we have that for every $x \in A$, $\nabla\bar{\varphi}(x) \cdot \tau_A^1 = \nabla\bar{\varphi}(x) \cdot \tau_A^2 = 0$ and for every $x \in B$, $\nabla\bar{\varphi}(x) \cdot \tau_B^1 = \nabla\bar{\varphi}(x) \cdot \tau_B^2 = 0$. So for $x \in AB$, we have that $\nabla\bar{\varphi}(x) = 0$ and therefore $\nabla\bar{\varphi}(x) \cdot n$ can be seen as a continuous function if we set its value to 0 on the edges, despite the jump discontinuities of the normal vector n.

So $\bar{u}$ is continuous and hence $\bar{u} \in W^{1-1/p,p}(\Gamma)$. The regularity of the optimal state follows from the trace theorem; see e.g. [36]. □

Using this regularity, an error estimate can be proved for the example problem.

Proposition 3.8. *Let Ω be a rectangular parallelepiped and $y_\Omega \in L^p(\Omega)$ for some $3 < p < +\infty$. Consider $\bar{u}$ be the solution of problem (P^{U}) and $\bar{u}_h$ be the solution of (P_h^{U}). Then there exists some $h_0 > 0$ and $C > 0$ such that for all $0 < h < h_0$*

$$\|\bar{u} - \bar{u}_h\|_{L^2(\Gamma)} \leq Ch^{1-1/p}.$$

The proof follows the same guidelines as those of [1] or [5, Section 6.1] and thus will be omitted. An interesting remark is that for 2D problems, we can deduce uniform convergence for the controls using the $L^2(\Gamma)$ estimate and an inverse inequality, since the boundary is 1D. This does not work for 3D problems with the error estimate at hand, since now Γ is 2D.

To solve the problem we have used a regular mesh of identical cubes of size h, each

Table 4. Execution times for the 3D problem in Example 3.3. $\nu = 1$.

h	N	$N_{\mathbb{I}}$	$N_{\mathbb{B}}$	t_D (s)	t_P (s)	pcg	$J(\bar u_h)$
2^{-4}	4913	3375	1538	0.7	0.2	6	0.4142332683
2^{-5}	35937	29791	6146	123.2s	5.2	6	0.4159847757
2^{-6}	274625	250047	24578	∞	183.7	6	0.4164610762

of them divided into 6 tetrahedra according to the Kuhn triangulation of the cube (see e.g. [37]). Up to our best knowledge, the current version of the Matlab PDEToolbox (version 2.1) computes an approximation of $\mathcal{B}$ using the barycenter formula. Although this does not affect the order of convergence for the FEM, this matrix plays a central role in Dirichlet control problems (for instance, $\mathcal{B}_{\mathbb{B},\mathbb{B}}$ is not singular, but the barycenter approximation may be singular), so we have computed it in an exact way (using the mid-sides formula). Mesh data, computation times and optimal values are displayed in Table 4.

4. Control constrained problems

Problem (P^{C}) has a unique solution $\bar u \in L^2(\Gamma)$ that satisfies $J'(\bar u)(u - \bar u) \geq 0$ for all $u \in U_{\alpha,\beta}$. For every $0 < h < h_0$, problem (P^{C}_h) has also a unique solution $\bar u_h$ that satisfies $J'_h(\bar u_h)(u_h - \bar u_h) \geq 0$ for all $u_h \in U^h_{\alpha,\beta}$. The problems being convex, these conditions are also sufficient. Moreover it is known that $\bar u_h \to \bar u$ strongly in $L^2(\Gamma)$ and also error estimates are available in terms of the mesh size h when $\nu > 0$ and the domain is convex in 2D. See [1, 6]. In [2] the smooth (2D and 3D) case is treated using variational approach, whose optimization process is different from the one we are presenting in the work at hand: in [3, Ch. 2.1], the problem is solved using a fixed point method, convergent for ν large enough; see [18] and Remark 3 below for the convergence of the semismooth Newton method.

4.1. Continuous problem

We can formulate first order optimality conditions as: there exist unique $\bar u \in L^2(\Gamma)$, $\bar y \in H^{1/2}(\Omega)$ and $\bar\varphi \in H^1_0(\Omega)$ such that

$$
\begin{aligned}
-\Delta \bar y &= 0 && \text{in } \Omega,\ \bar y = \bar u \text{ on } \Gamma, && \text{(27a)}\\
-\Delta \bar\varphi &= \bar y - y_\Omega && \text{in } \Omega,\ \bar\varphi = 0 \text{ on } \Gamma, && \text{(27b)}\\
(-\partial_n \bar\varphi + \nu \bar u, u - \bar u) &\geq 0 && \text{for all } u \in U_{\alpha,\beta}. && \text{(27c)}
\end{aligned}
$$

We first describe a semismooth Newton method to solve this optimality system and prove a convergence result for it (see Theorem 4.2). Next, we will reformulate the optimality system in terms of the Lagrange multipliers related to the constraints. We will see that this approach is better suited to the discrete problem, but has the drawback that the Newton method related to it is not semismooth; nonetheless, we will prove a convergence result for it; cf. Theorem 4.3. To facilitate the notation, we will skip the lower bound α, and will work only with the constraint $u \leq \beta$.

The variational inequality (27c) is a projection in $L^2(\Gamma)$. In this case, it is equivalent to a pointwise projection:

$$\nu\bar{u}(x) = \min\{\nu\beta(x), \partial_n\bar{\varphi}(x)\} \text{ on } \Gamma. \tag{27d}$$

In order to analyze the semismooth Newton method to solve the optimality system, we define $G : L^2(\Gamma) \to L^2(\Gamma)$ by

$$G(u) = \nu u - \min\{\nu\beta, -S^*(Su - y_\Omega)\}.$$

Thus, solving the optimality system is equivalent to solving

$$G(u) = 0. \tag{28}$$

Given $u \in L^2(\Gamma)$, we define the sets of active and free points related to u as

$$\Gamma_A(u) = \{x \in \Gamma : \ -S^*(Su - y_\Omega) > \nu\beta\},\ \Gamma_F(u) = \Gamma \setminus \Gamma_A(u).$$

Abusing notation and when this does not lead to confusion, we will often drop the u and only write Γ_A and Γ_F. (Although it is customary to use the word "inactive", we have preferred to use "free" since we have already used the letter I for the identity matrix $\mathcal{I}$ and the interior nodes $\mathbb{I}$.) χ_B will denote the characteristic function of a set B.

Lemma 4.1. *The relation $G : L^2(\Gamma) \to L^2(\Gamma)$ is slantly differentiable in the sense stated in [15] and $\partial^{CL}G$ semismooth in the sense stated in [20, Ch. 2], where $\partial^{CL}G$ is Clarke's generalized differential [38].*

A slanting functional $M(u) \in \mathcal{L}(L^2(\Gamma), L^2(\Gamma))$ is given by

$$M(u)v = \nu v + \chi_{\Gamma_F(u)}S^*Sv$$

for all $v \in L^2(\Gamma)$.

Finally, if $\nu > 0$, $M(u)$ has an inverse uniformly bounded in $\mathcal{L}(L^2(\Gamma), L^2(\Gamma))$ for all $u \in L^2(\Gamma)$.

Proof. Using (2) in Lemma 2.1, we have that there exists some $q > 2$ such that $S^*S \in \mathcal{L}(L^2(\Gamma), L^q(\Gamma))$. The slant differentibility then follows directly from [15, Pr 4.1 b)] and the semismoothness from [20, Theorem 2.13]; see also [14].

The expression for the slanting functional follows from the slant derivative of the function $\min(0, \cdot)$ and the chain rule [20, Theorem 2.10(c)].

Given $z \in L^2(\Gamma)$, existence of $v \in L^2(\Gamma)$ satisfying $M(u)v = z$ is straightforward, since the equation can be reduced to a symmetric positive definite system on the inactive set:

$$M(u)v = z \iff \begin{cases} \nu v = z & \text{on } \Gamma_A(u), \\ \nu v = z - S^*Sv & \text{on } \Gamma_F(u). \end{cases}$$

This equations can be read as

$$\begin{aligned} -\Delta y = 0 \text{ in } \Omega, &\qquad y = v \text{ on } \Gamma,\\ -\Delta\varphi = y \text{ in } \Omega, &\qquad \varphi = 0 \text{ on } \Gamma,\\ \nu v = z \text{ on } \Gamma_A(u), &\qquad \nu v = z + \partial_n\varphi \text{ on } \Gamma_F(u). \end{aligned}$$

Let us define $w = \chi_{\Gamma_F(u)}v$. Taking into account the definition of S and S^*, we have that $\nu v = \nu w + z\chi_{\Gamma_A(u)}$ and

$$\begin{aligned} -\Delta y_w = 0 \text{ in } \Omega, &\qquad y_w = w \text{ on } \Gamma,\\ -\Delta\varphi_w = y_w \text{ in } \Omega, &\qquad \varphi_w = 0 \text{ on } \Gamma,\\ w = 0 \text{ on } \Gamma_A(u), &\qquad \nu w = z + \partial_n\varphi_w - S^*Sz\chi_{\Gamma_A(u)} \text{ on } \Gamma_F(u). \end{aligned}$$

On one hand, using the continuity of S^* (cf. (2)) and of S, we have

$$\|S^*Sz\chi_{\Gamma_A(u)}\|_{L^2(\Gamma)} \leq C\|z\|_{L^2(\Gamma_A(u))} \leq C\|z\|_{L^2(\Gamma)}. \tag{29}$$

On the other hand, using the definition of solution in the transposition sense, we have that

$$(y_w, y_w)_\Omega = -(w, \partial_n\varphi_w)_\Gamma = -\nu(w,w)_\Gamma + (w,z)_\Gamma - (w, S^*Sz\chi_{\Gamma_A(u)})_\Gamma.$$

So

$$\begin{aligned} \nu\|w\|^2_{L^2(\Gamma)} &= (w,z)_\Gamma - (w, S^*Sz\chi_{\Gamma_A(u)})_\Gamma - \|y_w\|^2_{L^2(\Omega)}\\ &\leq \|w\|_{L^2(\Gamma)}\|z\|_{L^2(\Gamma)} + \|w\|_{L^2(\Gamma)}\|S^*Sz\chi_{\Gamma_A(u)}\|_{L^2(\Gamma)}\\ &\leq C\|w\|_{L^2(\Gamma)}\|z\|_{L^2(\Gamma)}. \end{aligned}$$

And we get $\|w\|_{L^2(\Gamma)} \leq C/\nu\|z\|_{L^2(\Gamma)}$. Taking into account the definition of w and the condition for v on the active set, we get that $\|v\|_{L^2(\Gamma)} \leq C/\nu\|z\|_{L^2(\Gamma)}$, where C is independent of u, and hence $M(u)$ has a uniformly bounded inverse for each $\nu > 0$. □

Remark 2. Notice that in the infinite dimensional case, if $\nu = 0$ then $M(u)v = \chi_{\Gamma_F(u)}S^*Sv$. In practical cases, it is known (cf. (3) or [8]) that there exists $t > 0$ such that $S^*Sv \in H^t(\Gamma)$ which is compactly embedded in $L^2(\Gamma)$, and hence $M(u)$ does not have a bounded inverse.

Given a current iterate $u \in L^2(\Gamma)$, we may compute next iterate u^+ using a semismooth Newton method as follows:

$$M(u)(u^+ - u) = -G(u).$$

Writing this in detail leads to

$$\nu u^+ - \nu u + \chi_{\Gamma_F(u)}S^*S(u^+ - u) = -\nu u + \min\{\nu\beta, -S^*(Su - y_\Omega)\},$$

which means that, if $\nu > 0$,

$$\nu u^+ - \nu u = -\nu u + \nu\beta \Rightarrow u^+ = \beta \text{ on } \Gamma_A(u),$$

and

$$u^+ = -\frac{1}{\nu} S^*(Su^+ - y_\Omega) \text{ on } \Gamma_F(u).$$

This equations can be read as

$$\begin{aligned} -\Delta y^+ &= 0 \text{ on } \Omega, & y^+ &= u^+ \text{ on } \Gamma, && (30a)\\ -\Delta \varphi^+ &= y^+ - y_\Omega \text{ on } \Omega, & \varphi^+ &= 0 \text{ on } \Gamma, && (30b)\\ u^+ &= \beta \text{ on } \Gamma_A(u), & u^+ &= \frac{1}{\nu}\partial_n \varphi^+ \text{ on } \Gamma_F(u). && (30c) \end{aligned}$$

With all these considerations, we can write a semismooth Newton method to solve the optimality system in the infinite dimensional case.

Algorithm 3: Semismooth Newton method to solve (27a), (27b), (27d)

```
1 Initialize $k = 0$ and provide $u^0 \in L^2(\Gamma)$
2 Set $u = u^k$
3 Compute $\Gamma_A(u)$ and $\Gamma_F(u)$
4 Solve the optimality system (30a) (30b), (30c)
5 Set $u^{k+1} = u^+$
6 Set $k = k + 1$
7 Stop or go to 2
```

Theorem 4.2. *The semismooth Newton method described in Algorithm 3 converges q-superlinealy to $\bar{u}$ provided u^0 is close enough to $\bar{u}$ in the sense of $L^2(\Gamma)$.*

Proof. Once we have proved Lemma 4.1, this result is a direct consequence of [10, Theorem 3.4]; see also [15, Theorem 1.1] or [20, Theorem 2.12]. □

The discrete version of the variational inequality (27c) –see (36d) below– does not have a pointwise version analog to (27d). A more convenient approach to the continuous problem, from the point of view of the discretized problem, is obtained using Lagrange multipliers associated to the bound control constraints. The con of this approach is that we do not obtain a semismooth Newton method, in the sense that the involved functions are known not to be semismooth.

Remark 3. It must noticed that, when the variational discretization of the control is used, [2, 3, 18, 19], the discrete optimal control is obtained as the pointwise projection of the discrete optimal adjoint state as in (27d), so convergence of the semismooth Newton method for such a discretization would follow from Theorem 4.2 and Lemma 4.1 with S replaced by S_h.

To simplify the notation, we will reduce the exposition to the case of having only an upper bound $u \leq \beta$ on Γ.

Condition (27c) can be replaced by the following pair of equations: there exists also $\bar{\lambda} \in L^2(\Gamma)$ such that equation (27d) can be written as

$$\begin{aligned} \nu \bar{u}(x) &= \partial_n \bar{\varphi}(x) - \bar{\lambda}(x) \text{ on } \Gamma. && (31a)\\ \bar{u}(x) \leq \beta(x),\ \bar{\lambda}(x) \geq 0,\ \bar{\lambda}(x)(\bar{u}(x) - \beta(x)) &= 0 \text{ for a.e. } x \in \Gamma. && (31b) \end{aligned}$$

Condition (31b) can be written as

$$\lambda(x) - \max\{0, \lambda(x) + c(u(x) - \beta(x))\} = 0 \text{ for a.e. } x \in \Gamma, \tag{32}$$

which is true for any $c > 0$. It is known that the operator described in (32) neither is semismooth in $L^2(\Gamma)$ (following [20, Lemma 2.7]) nor has a slant derivative (as in [15, Pr 4.1]). Since $u \in L^2(\Gamma)$ appears inside the max operation, it is not clear that Newton's method applied to (27a), (27b), (32) converges. Define

$$F_c(u, \lambda) = \begin{pmatrix} \nu u + S^*Su - S^*y_\Omega + \lambda \\ \lambda - \max\{0, \lambda + cu - c\beta\} \end{pmatrix}$$

and for each pair control-multiplier $(u, \lambda) \in L^2(\Gamma) \times L^2(\Gamma)$, set

$$\Gamma_A(u, \lambda) = \{x \in \Gamma : \frac{1}{c}\lambda + u > \beta\},\ \Gamma_F(u, \lambda) = \Gamma \setminus \Gamma_A(u, \lambda).$$

We have that

$$\begin{pmatrix} \nu I + S^*S & I \\ -c\chi_{\Gamma_A(u,\lambda)} I & \chi_{\Gamma_F(u,\lambda)} I \end{pmatrix} \in \partial^{CL} F_c(u, \lambda).$$

So, given a current iterate (u, λ), a Newton-like iterate to obtain (u^+, λ^+) reads as

$$\nu u^+ = -S^*(Su^+ - y_\Omega) - \lambda^+, \tag{33a}$$

$$u^+ = \beta \text{ on } \Gamma_A(u, \lambda), \tag{33b}$$

$$\lambda^+ = 0 \text{ on } \Gamma_F(u, \lambda). \tag{33c}$$

It is remarkable that the parameter c only appears in this equations hidden in the definition of the active set.

Algorithm 4: Newton-like method to solve (27a), (27b), (32)

```
1 Initialize $k = 0$ and provide $(u^0, \lambda^0) \in L^2(\Gamma) \times L^2(\Gamma)$
2 Set $u = u^k$ and $\lambda = \lambda^k$
3 Compute $\Gamma_A(u, \lambda)$ and $\Gamma_F(u, \lambda)$
4 Compute $(u^+, \lambda^+)$ using (33a) (33b), (33c)
5 Set $u^{k+1} = u^+$ and $\lambda^{k+1} = \lambda^+$
6 Set $k = k + 1$
7 Stop or go to 2
```

Theorem 4.3. *If $c = \nu > 0$, the sequence u^k generated by the Newton-like method described in Algorithm 4 converges q-superlinearly to $\bar{u}$ in $L^2(\Gamma)$ provided u^0 is close enough to $\bar{u}$ in the sense of $L^2(\Gamma)$ and $\lambda^0 = S^*y_\Omega - (S^*Su^0 + \nu u^0)$.*

Proof. In this case we cannot apply directly the results in [10, Theorem 3.4]; see also [15, Theorem 1.1] or [20, Theorem 2.12] since $u \in L^2(\Gamma)$ appears inside the max operation, and it is known that the operator described in (32) is not semismooth (following [20, Lemma 2.7]) or has not a slant derivative (as in [15, Pr 4.1]).

We will follow instead the method of proof of [15, Theorem 4.1] and we will show that the sequence u^k generated by Algorithm 4 is exactly the same as the one generated by Algorithm 3, which we will call $\tilde{u}^k$.

Writing in detail (33a) we obtain

$$\begin{aligned} -\Delta y^+ &= 0 \text{ on } \Omega, & y^+ &= u^+ \text{ on } \Gamma, && (34a)\\ -\Delta\varphi^+ &= y^+ - y_\Omega \text{ in } \Omega, & \varphi^+ &= 0 \text{ on } \Gamma, && (34b)\\ && \nu u^+ &= \partial_n\varphi^+ - \lambda^+ \text{ on } \Gamma. && (34c) \end{aligned}$$

Using now (33b) and (33c) we obtain

$$\begin{aligned} -\Delta y^+ &= 0 \text{ on } \Omega, & y^+ &= u^+ \text{ on } \Gamma, && (35a)\\ -\Delta\varphi^+ &= y^+ - y_\Omega \text{ on } \Omega, & \varphi^+ &= 0 \text{ on } \Gamma, && (35b)\\ u^+ &= \beta \text{ on } \Gamma_A(u,\lambda), & u^+ &= \frac{1}{\nu}\partial_n\varphi^+ \text{ on } \Gamma_F(u,\lambda), && (35c)\\ \lambda^+ &= \partial_n\varphi^+ - \nu\beta \text{ on } \Gamma_A(u,\lambda), & \lambda^+ &= 0 \text{ on } \Gamma_F(u,\lambda). && (35d) \end{aligned}$$

Notice that (35a), (35b) and (35c) is exactly as (30a), (30b) and (30c), provided that $\Gamma_A(u,\lambda) = \Gamma_A(u)$.

To finish, let us prove by induction that $u^k = \tilde{u}^k$ and $\Gamma_A(u^k,\lambda^k) = \Gamma_A(\tilde{u}^k)$ for all $k \in \mathbb{N}\cup\{0\}$. For $k = 0$ it is clear from the definition of the active sets and the choice of λ^0 made in the assumption of this theorem. Suppose now $u^k = \tilde{u}^k$ and $\Gamma_A(u^k,\lambda^k) = \Gamma_A(\tilde{u}^k)$. It is clear that $u^{k+1} = \tilde{u}^{k+1}$. On the other hand, from (35c) and (35d) and the choice $c = \nu$ we have that $cu^{k+1} + \lambda^{k+1} = -S^*S(u^{k+1} - y_\Omega)$, and therefore $\Gamma_A(u^{k+1},\lambda^{k+1}) = \Gamma_A(\tilde{u}^{k+1})$.

The result, hence, follows from Theorem 4.2. □

We want to remark here that writting the Newton step as in (35a), (35b), (35c) and (35d) we obtain exactly the same kind of algorithm as the one described in [13]

4.2. Finite dimensional approximation

Let us focus now on the finite dimensional approximation. For every $0 < h < h_0$, there exist a unique solution $\bar{u}_h \in U_h$ of (P_h^C) and unique, $\bar{y}_h \in Y_h$ and $\bar{\varphi}_h \in Y_{h0}$ such that

$$\begin{aligned} a(\bar{y}_h, z_h) &= 0 \text{ for all } z_h \in Y_{h0}, && (36a)\\ \bar{y}_h &\equiv \bar{u}_h \text{ on } \Gamma, && (36b)\\ a(z_h, \bar{\varphi}_h) &= (\bar{y}_h - y_\Omega, z_h)_\Omega \text{ for all } z_h \in Y_{h0}, && (36c)\\ (-\partial_n^h\bar{\varphi}_h + \nu\bar{u}_h, u_h - \bar{u}_h) &\geq 0 \text{ for all } u_h \in U_{\alpha,\beta}^h. && (36d) \end{aligned}$$

This way of writting the optimality system is useful to obtain error estimates (cf. [1, 2, 5]). Nevertheless, we cannot deduce a pointwise projection formula for the optimal control from the variational inequality.

Remark 4. This difficulty is of course not exclusive of Dirichlet control problems. Think of this naive example. Let $\Gamma = [-1,1]$ and take a mesh with nodes $\{-1,0,1\}$. For every $u_h, v_h \in U_h$, we have that $(u_h, v_h)_\Gamma = \boldsymbol{u}^T\mathcal{B}\boldsymbol{v}$, where $\mathcal{B}$ is the mass matrix

$$\mathcal{B} = \frac{1}{6}\begin{pmatrix} 2 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 2 \end{pmatrix}.$$

Consider $\alpha = -\infty$, $\beta = 0$, and take , e.g., $\boldsymbol{w} = (-2, 1, 1)^T$. Then we have that

$$(-w_h+\bar{u}_h, u_h-\bar{u}_h) \geq 0 \, \forall u_h \in U^h_{\alpha,\beta} \implies \bar{u}_h = -1.5e_1+0e_2+0e_3 \implies \boldsymbol{u} = (-1.5, 0, 0)^T,$$

but

$$\min(\boldsymbol{w}, \boldsymbol{0}) = (-2, 0, 0)^T,$$

and we get different results with the $L^2(\Gamma)$ projection and the pointwise projection.

To circumvent this difficulty, we rewrite conditions (36a)–(36d) in order to use standard semismooth Newton method. Taking into account (17), we can write (P^{C}_h) as

$$\begin{array}{c} \min \frac{1}{2}\boldsymbol{u}^T\mathcal{A}\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u} \\ \text{subject to } \boldsymbol{\alpha} \leq \boldsymbol{u} \leq \boldsymbol{\beta}, \end{array} \tag{37}$$

where $\mathcal{A}$ and $\boldsymbol{f}$ are defined in (18) and (19) and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{N_{\mathbb{B}}\times 1}$ are the vectors whose j-component are respectively $\alpha(x_j)$ and $\beta(x_j)$ and the inequalities are understood componentwise. In order to simplify the notation, we will restrict ourselves again to the case without lower bound. The optimality system can thus be written in the following form: if $\boldsymbol{u}$ is the solution of problem (37), then there exists a unique $\boldsymbol{\lambda} \in \mathbb{R}^{N_{\mathbb{B}}\times 1}$ such that

$$\begin{array}{rcl} \mathcal{A}\boldsymbol{u} + \boldsymbol{\lambda} - \boldsymbol{f} & = & 0, \\ \boldsymbol{\lambda} \geq 0, \ \boldsymbol{u} \leq \boldsymbol{\beta}, \ \boldsymbol{\lambda}^T(\boldsymbol{u} - \boldsymbol{\beta}) & = & 0. \end{array} \tag{38}$$

To continue, we rewrite again the second condition in (38) to obtain

$$\begin{array}{rcl} \mathcal{A}\boldsymbol{u} + \boldsymbol{\lambda} - \boldsymbol{f} & = & 0, \\ \boldsymbol{\lambda} - \max(0, \boldsymbol{\lambda} + c(\boldsymbol{u} - \boldsymbol{\beta})) & = & 0. \end{array} \tag{39}$$

Notice that (39) looks like a discrete version of the optimality system formed by (27a), (27b) and (32). Acting in an analogous way as we did for the continuous problem, we define

$$F_{h,c}(\boldsymbol{u}, \boldsymbol{\lambda}) = \begin{pmatrix} \mathcal{A}\boldsymbol{u} + \boldsymbol{\lambda} - \boldsymbol{f} \\ \boldsymbol{\lambda} - \max(\boldsymbol{0}, \boldsymbol{\lambda} + c(\boldsymbol{u} - \boldsymbol{\beta})) \end{pmatrix}.$$

and for every pair $(\boldsymbol{u}, \boldsymbol{\lambda})$, we define the sets of active and free indexes as

$$\mathbb{A}(\boldsymbol{u}, \boldsymbol{\lambda}) = \{j : \lambda_j + c(u_j - \beta(x_j)) > 0\}, \ \mathbb{F}(\boldsymbol{u}, \boldsymbol{\lambda}) = \{j : \lambda_j + c(u_j - \beta(x_j)) \leq 0\}. \tag{40}$$

Abusing notation and when this does not lead to confusion, we will often just write $\mathbb{A}$ and $\mathbb{F}$. Notice that $\mathbb{A} \cup \mathbb{F} = \mathbb{B}$ for every possible pair control-multiplier.

Lemma 4.4. *The function $F_{h,c}$ is slantly differentiable for every $c > 0$. A slanting function for $F_{h,c}$ is*

$$M_{h,c}(\boldsymbol{u}, \boldsymbol{\lambda}) = \begin{pmatrix} \mathcal{A} & \mathcal{I}_{\mathbb{B},\mathbb{B}} \\ -c\mathcal{I}_{\mathbb{B},\mathbb{A}}\mathcal{I}_{\mathbb{A},\mathbb{B}} & \mathcal{I}_{\mathbb{B},\mathbb{F}}\mathcal{I}_{\mathbb{F},\mathbb{B}} \end{pmatrix}.$$

Finally, for all $\nu \geq 0$, the inverse of $M_{h,c}(\boldsymbol{u}, \boldsymbol{\lambda})$ is uniformly bounded w.r.t $(\boldsymbol{u}, \boldsymbol{\lambda})$.

Proof. In finite dimension, the function $\max\{\mathbf{0}, \cdot\}$ is semismooth due to Rademacher's theorem and $M_{h,c}$ is a slanting function; see e.g. [20, Example 2.4] or [15, Lemma 3.1].

Let us prove the uniform boundedness of the inverse. Given $\boldsymbol{z} \in \mathbb{R}^{N_{\mathbb{B}}}$ and $\boldsymbol{\eta} \in \mathbb{R}^{N_{\mathbb{B}}}$, we have that

$$M_{h,c}(\boldsymbol{u}, \boldsymbol{\lambda}) \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{\eta} \end{bmatrix} \iff \left\{ \begin{array}{rcl} \mathcal{A}\boldsymbol{v} + \boldsymbol{\mu} & = & \boldsymbol{z}, \\ -c\boldsymbol{v}_{\mathbb{A}} & = & \boldsymbol{\eta}_{\mathbb{A}}, \\ \boldsymbol{\mu}_{\mathbb{F}} & = & \boldsymbol{\eta}_{\mathbb{F}}. \end{array} \right.$$

We write the last equalities as

$$\begin{array}{rcl} \mathcal{A}\boldsymbol{v} + \begin{bmatrix} \boldsymbol{\mu}_{\mathbb{A}} \\ \mathbf{0}_{\mathbb{F}} \end{bmatrix} & = & \boldsymbol{z} - \begin{bmatrix} \mathbf{0}_{\mathbb{A}} \\ \boldsymbol{\eta}_{\mathbb{F}} \end{bmatrix} =: \boldsymbol{\zeta}, \\ -c\boldsymbol{v}_{\mathbb{A}} & = & \boldsymbol{\eta}_{\mathbb{A}}. \end{array}$$

This is the optimality system of the equality constrained optimization problem

$$\begin{array}{l} \boldsymbol{v} = \arg \min\limits_{\boldsymbol{w} \in \mathbb{R}^{N_{\mathbb{B}}}} \dfrac{1}{2} \boldsymbol{w}^T \mathcal{A} \boldsymbol{w} - \boldsymbol{\zeta}^T \boldsymbol{w}, \\ \text{subject to } \boldsymbol{w}_{\mathbb{A}} = -\boldsymbol{\eta}_{\mathbb{A}}/c. \end{array}$$

Writing $\boldsymbol{w} = \mathcal{I}_{\mathbb{B},\mathbb{F}} \boldsymbol{w}_{\mathbb{F}} + \mathcal{I}_{\mathbb{B},\mathbb{A}} \boldsymbol{w}_{\mathbb{A}}$ and taking into account the equality constraint, we have that $\boldsymbol{v}_{\mathbb{F}}$ is the solution of the following unconstrained optimization problem

$$\boldsymbol{v}_{\mathbb{F}} = \arg \min_{\boldsymbol{w}_{\mathbb{F}}} \frac{1}{2} \boldsymbol{w}_{\mathbb{F}}^T (\mathcal{I}_{\mathbb{F},\mathbb{B}} \mathcal{A} \mathcal{I}_{\mathbb{B},\mathbb{F}}) \boldsymbol{w}_{\mathbb{F}} - \big(\mathcal{I}_{\mathbb{F},\mathbb{B}}(\boldsymbol{\zeta} + \mathcal{A}\mathcal{I}_{\mathbb{B},\mathbb{A}} \boldsymbol{\eta}_{\mathbb{A}}/c)\big)^T \boldsymbol{w}_{\mathbb{F}}$$

and therefore $\boldsymbol{v}_{\mathbb{F}}$ is the solution of the following linear system

$$\mathcal{I}_{\mathbb{F},\mathbb{B}} \mathcal{A} \mathcal{I}_{\mathbb{B},\mathbb{F}} \boldsymbol{v}_{\mathbb{F}} = \mathcal{I}_{\mathbb{F},\mathbb{B}}(\boldsymbol{\zeta} + \mathcal{A}\mathcal{I}_{\mathbb{B},\mathbb{A}} \boldsymbol{\eta}_{\mathbb{A}}/c)$$

Since $\mathcal{A}$ is symmetric and positive definite, so is $\mathcal{I}_{\mathbb{F},\mathbb{B}} \mathcal{A} \mathcal{I}_{\mathbb{B},\mathbb{F}}$, and its smallest eigenvalue is bounded from below by $0 < \lambda_1(\mathcal{M}) + \nu\lambda_1(\mathcal{B}_{\mathbb{B},\mathbb{B}})$; see (24). Therefore, the previous system is solvable and there exists a constant $C > 0$, that may depend on h, ν and c, but is independent of $\boldsymbol{u}$, such that $\|\boldsymbol{v}_{\mathbb{F}}\| \leq C(\|\boldsymbol{z}\| + \|\boldsymbol{\eta}\|)$. From this it is straight to deduce that $\|\boldsymbol{v}\| + \|\boldsymbol{\mu}\| \leq C(\|\boldsymbol{z}\| + \|\boldsymbol{\eta}\|)$ and hence $M_{h,c}$ has a uniformly bounded inverse. □

With these considerations, given a current iterate $(\boldsymbol{u}, \boldsymbol{\lambda})$ with active and free index sets $\mathbb{A} = \mathbb{A}(\boldsymbol{u}, \boldsymbol{\lambda})$ and $\mathbb{F} = \mathbb{F}(\boldsymbol{u}, \boldsymbol{\lambda})$, we can compute the next step of Newton's method $(\boldsymbol{u}^+, \boldsymbol{\lambda}^+)$ solving

$$\begin{array}{c} \boldsymbol{u}^+ = \arg \min\limits_{\boldsymbol{u} \in \mathbb{R}^{N_{\mathbb{B}} \times 1}} \dfrac{1}{2} \boldsymbol{u}^T \mathcal{A} \boldsymbol{u} - \boldsymbol{f}^T \boldsymbol{u} \\ \text{subject to } \boldsymbol{u}_{\mathbb{A}} = \boldsymbol{\beta}_{\mathbb{A}}. \\ \text{Set } \boldsymbol{\lambda}^+ = \boldsymbol{f} - \mathcal{A}\boldsymbol{u}^+. \end{array}$$

At each iteration, this is equivalent to solving the following unconstrained optimization problem in the lower-dimensional space $\mathbb{R}^{N_{\mathbb{F}}\times 1}$:

$$\begin{aligned}
\boldsymbol{u}_{\mathbb{F}}^{+} &= \arg \min_{\boldsymbol{u}_{\mathbb{F}}\in\mathbb{R}^{N_{\mathbb{F}}\times 1}} \frac{1}{2}\boldsymbol{u}_{\mathbb{F}}^{T}(\mathcal{I}_{\mathbb{F},\mathbb{B}}\mathcal{A}\mathcal{I}_{\mathbb{B},\mathbb{F}})\boldsymbol{u}_{\mathbb{F}} - \big(\mathcal{I}_{\mathbb{F},\mathbb{B}}(\boldsymbol{f} - \mathcal{A}\mathcal{I}_{\mathbb{B},\mathbb{A}}\boldsymbol{\beta}_{\mathbb{A}})\big)^{T}\boldsymbol{u}_{\mathbb{F}} \\
\boldsymbol{u}_{\mathbb{A}}^{+} &= \boldsymbol{\beta}_{\mathbb{A}} \\
\boldsymbol{\lambda}^{+} &= \boldsymbol{f} - \mathcal{A}\boldsymbol{u}^{+}
\end{aligned} \tag{41}$$

Again the preconditioned gradient method works fine to solve this problem. A good preconditioner in practice is $\mathcal{P} = \mathcal{M}_{\mathbb{F},\mathbb{F}} + \nu\mathcal{B}_{\mathbb{F},\mathbb{F}}$.

Alternatively, taking into account the definition of $\mathcal{A}$ and $\boldsymbol{f}$ (see also Algorithms 1 and 2 respectively), we may write one step of the semismooth Newton algorithm as

$$\begin{array}{rcl}
\begin{bmatrix} \mathcal{M}+\nu\mathcal{B} & -\mathcal{K}_{:,\mathbb{I}} & \mathcal{I}_{:,\mathbb{A}} \\ -\mathcal{K}_{\mathbb{I},:} & \mathcal{O}_{\mathbb{I},\mathbb{I}} & \mathcal{O}_{\mathbb{I},\mathbb{A}} \\ \mathcal{I}_{\mathbb{A},:} & \mathcal{O}_{\mathbb{A},\mathbb{I}} & \mathcal{O}_{\mathbb{A},\mathbb{A}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^{+} \\ \boldsymbol{\varphi}_{\mathbb{I}}^{+} \\ \boldsymbol{\lambda}_{\mathbb{A}}^{+} \end{bmatrix} & = & \begin{bmatrix} \mathcal{M}\boldsymbol{y}_{\Omega} \\ \mathbf{0}_{\mathbb{I}} \\ \mathcal{I}_{\mathbb{A},\mathbb{B}}\boldsymbol{\beta} \end{bmatrix} \\
\boldsymbol{u}^{+} & = & \boldsymbol{y}_{\mathbb{B}}^{+} \\
\boldsymbol{\lambda}_{\mathbb{F}}^{+} & = & \mathbf{0}_{\mathbb{F}}
\end{array} \tag{42}$$

As for the unconstrained problem, direct methods for small size problems or preconditioned conjugate gradient techniques described in [31] can be applied to solve this system at each Newton step.

Remark 5. Notice that the only information from iteration k used to compute iteration $k+1$ is the set of active indexes. Therefore, when $\mathbb{A}(\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1}) = \mathbb{A}(\boldsymbol{u}_{k}, \boldsymbol{\lambda}_{k})$ we have reached an stationary point. This is usually the criterion used to stop the semismooth Newton method. Another consequence of this is that to initiate the algorithm, in principle we do not not need an initial guess $\boldsymbol{u}^0$ and $\boldsymbol{\lambda}^0$, but only an initial guess for the active set. We include nevertheless an initial guess for the control variable in Algorithm 5 because we use $\boldsymbol{u}_k$ as the initial guess for the pcg to obtain $\boldsymbol{u}_{k+1}$.

Algorithm 5: Semismooth Newton method for (P_h^{C})

```
1 Set $k = 0$ and initialize $\boldsymbol{u}_0$ and $\boldsymbol{\lambda}_0$; compute $\mathbb{A}_0 = \mathbb{A}(\boldsymbol{u}_0, \boldsymbol{\lambda}_0)$
2 Set $\mathbb{A} = \mathbb{A}_k$ and $\mathbb{F} = \mathbb{B} \setminus \mathbb{A}$
3 Compute $(\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1}) = (\boldsymbol{u}^+, \boldsymbol{\lambda}^+)$ using (41) [or (42)]
4 Set $\mathbb{A}_{k+1} = \mathbb{A}(\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1})$
5 Stop or set $k = k + 1$ and return to 2
```

Theorem 4.5. *The sequence $\boldsymbol{u}_k$ generated by Algorithm 5 converges q-superlinearly to $\boldsymbol{u}$, the solution of* (37)*, for every $c > 0$, provided $(\boldsymbol{u}_0, \boldsymbol{\lambda}_0)$ is close enough to $(\boldsymbol{u}, \boldsymbol{\lambda})$.*

This result is a direct consequence of Lemma 4.4 and [10, Theorem 3.4]; see also [15, Theorem 1.1] or [20, Theorem 2.12].

Example 4.6. We resume the 2D problem described in Example 3.3 and the 3D problem described in Example 3.6, with the upper constraint $\beta \equiv 0.16$. We test Algorithm 5.

Following the tip of Theorem 4.3, we have taken the parameter $c = \nu$ and $\boldsymbol{\lambda}_0 = \boldsymbol{f} - \mathcal{A}\boldsymbol{u}_0$. Nevertheless, as expected from Theorem 4.5, we have not been able to observe any problem for different values of c. The seed is set to $\boldsymbol{u}_0 = \mathbf{0}$.

To solve the optimality system of the unconstrained optimization problem at each Newton iterate, we have used the preconditioned conjugate gradient method for (41) with initial guess $\mathcal{I}_{\mathbb{F},\mathbb{B}}\boldsymbol{u}_k$. The use of the reduced problem is even more advisable in this case than it was in Examples 3.3 and 3.6, because the size of the system (42) is $N + N_{\mathbb{I}} + N_{\mathbb{A}}$, which in any case is greater or equal than the size of the system (21), which is $N + N_{\mathbb{I}}$.

In Tables 5 and 6 and we report on the number of Newton iterations for each mesh size as well as the total number of conjugate gradient iterations. For reference, we also report on the number of active nodes and the optimal solution of the discrete problem being approximated.

Table 5. Newton iterations and total pcg iterations for a 2D control constrained problem

h	$N_{\mathbb{B}}$	$N_{\mathbb{A}}$	Newton	pcg	$J(\bar{u}_h)$
0.2×2^{-4}	320	245	3	16	0.3537665421
0.2×2^{-5}	640	491	4	20	0.3537991309
0.2×2^{-6}	1280	977	4	19	0.3538099358
0.2×2^{-7}	2560	1951	3	16	0.3538134437
0.2×2^{-8}	5120	3905	4	19	0.3538145736

Table 6. Newton iterations and total pcg iterations for a 3D control constrained problem

h	$N_{\mathbb{B}}$	$N_{\mathbb{A}}$	Newton	pcg	$J(\bar{u}_h)$
2^{-2}	98	54	2	7	0.3935682160
2^{-3}	386	294	2	7	0.4007301110
2^{-4}	1538	894	3	16	0.4104264396
2^{-5}	6146	3210	3	15	0.4153200584
2^{-6}	24578	11958	4	19	0.4173850169

Choosing the initial guess for Algorithm 5. If possible, it is a good idea to select an initial point for the Newton method close to the solution. If we are dealing with the problem for some mesh size h, a good candidate for the initial iteration is $I_h\bar{u}_{h_{-1}}$, the solution in a coarser mesh with $h_{-1} \geq h$. This idea can be iterated with a mesh family with parameters $h_{-M}, \dots, h_0 = h$. The computation time should decrease for fine meshes provided that the interpolation can be carried out in an effective way. For instance, using nested meshes.

Example 4.7. Solving the last 2D problem in Table 5 takes a cputime of 620 seconds. Using a nested iteration, we solve the problem with 5120 boundary nodes in 436 seconds, with just 2 Newton iterates and 7 conjugate gradient iterates at the finest level. The times include mesh generations and matrix assembly. To generate the meshes and interpolate the solution, we have used MATLAB PDETOOLBOX command `refinemesh`.

In the 3D case (see Table 6), solving the problem for $h = 2^{-6}$ takes 681 seconds. Using a nested mesh strategy this time is reduced to 404 seconds with 2 Newton iterates and 9 pcg iterations at the finest level. In our example we have been able to make an efficient interpolation of the solution at the previous level just using MATLAB's `interp3`.

Example 4.8. Absence of Tikhonov parameter In the same 2D pentagonal domain, we take again $\beta \equiv 0.16$, but now $\alpha \equiv -1.2$, $\nu = 0$, $y_\Omega \equiv 1$ if $x_1 > 0.25$, and

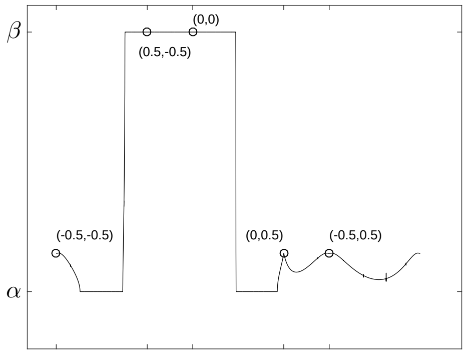

Figure 1. Discrete optimal control at the finest level for Example 4.8. The polygonal boundary has been stretched over a line and the vertexes have been marked with circles on the graph of $\bar{u}_h$.

$y_\Omega \equiv -1$ if $x_1 < 0.25$. We are able to solve the finite dimensional problem, but up to our best knowledge, there is no proof available about the convergence of the discrete optimal solutions to the continuous optimal solution.

To solve the problem we follow a nested mesh strategy. The number of pcg iterations per Newton iterate grow as h tends to zero, as is to be expected from Corollary 3.2; see Table 7.

The constraints have been chosen in such way that we seemingly find both a bang-bang part and a singular arc. We have sketched a plot of the discrete optimal control for $h = 0.2 \times 2^{-8}$ in Figure 1. The boundary has been stretched on a 1D line and its corners have been marked with circles on the graph of $\bar{u}_h$.

Table 7. Convergence history for a bilaterally constrained 2D problem with $\nu = 0$.

h	Newton	pcg	$J(\bar{u}_h)$
0.2×2^{-0}	2	31	0.0694073016
0.2×2^{-1}	3	61	0.0849064798
0.2×2^{-2}	4	104	0.0920895196
0.2×2^{-3}	4	131	0.0971597067
0.2×2^{-4}	2	87	0.0998850162
0.2×2^{-5}	4	194	0.1011428500
0.2×2^{-6}	1	73	0.1017242162
0.2×2^{-7}	2	185	0.1020255084
0.2×2^{-8}	2	236	0.1021724081

5. State constraints

In the rest of the work, we will suppose that $d = 2$, Ω is convex and Γ is polygonal. According to [5], if problem (P^{S}) admits a feasible Slater point, then it has a unique solution $\bar{u} \in H^{1/2}(\Gamma)$ and there exist $\bar{y} \in H^1(\Omega) \cap C(\bar{\omega})$, $\bar{\varphi} \in W_0^{1,t}(\Omega)$ for all $t < 2$ and two nonnegative measures $\bar\mu^+,\ \bar\mu^- \in \mathcal{M}(\bar\omega)$ such that

$$-\Delta\bar y = 0 \text{ in } \Omega,\ \bar y = \bar u \text{ on } \Gamma, \tag{43a}$$

$$-\Delta\bar\varphi = \bar y - y_\Omega + \bar\mu^+ - \bar\mu^- \text{ in } \Omega,\ \bar\varphi = 0 \text{ on } \Gamma, \tag{43b}$$

$$\bar u(x) = \frac{1}{\nu}\partial_n\bar\varphi(x) \text{ on } \Gamma, \tag{43c}$$

$$\langle\bar\mu^+ - \bar\mu^-, y - \bar y\rangle \le 0\ \forall y \in K_{a,b}, \tag{43d}$$

and $\operatorname{supp}\bar\mu^+ \subset \{\bar y = b\}$, $\operatorname{supp}\bar\mu^- \subset \{\bar y = a\}$. In this case the adjoint state equation (43b) must be understood in the transposition sense, [5, Equation (4)], and $\langle\cdot,\cdot\rangle$ denotes the duality product between $\mathcal{M}(\bar\omega)$ and $C(\bar\omega)$.

As is pointed out in [12, page 197], or [15, page 878], the PDAS or semismooth Newton methods described in the previous section are not applicable to this problem, since the multiplier is a measure. In [39] a Moreau-Yosida regularization is proposed. This strategy is further investigated in [16]. Let us briefly describe this method. Again, to simplify the notation, we will do the exposition for the unilateral constraint $y \le b$ in $\bar\omega$.

Given a shift function $\mu^* \in L^q(\omega)$ for some $q > 2$ and a parameter $\gamma > 0$, we will solve the unconstrained problem

$$(Q^\gamma)\ \min_{u\in L^2(\Gamma)} J(u) + \frac{1}{2\gamma}\int_{\bar\omega} \max\{0, \mu^* + \gamma(Su - b)\}^2 dx.$$

This problem has a unique solution $u^\gamma \in H^{1/2}(\Gamma)$ for every $\gamma > 0$. Since the functional is of class C^1, first order optimality conditions can be derived directly from the work [1]: there exist unique $y^\gamma \in H^1(\Omega)$ and $\varphi^\gamma \in H_0^1(\Omega)$ such that

$$-\Delta y^\gamma = 0 \text{ in } \Omega,\ y^\gamma = u^\gamma \text{ on } \Gamma, \tag{44a}$$

$$-\Delta\varphi^\gamma = y^\gamma - y_\Omega + \chi_{\bar\omega}\max\{0, \mu^* + \gamma(y^\gamma - b)\} \text{ in } \Omega,\ \varphi^\gamma = 0 \text{ on } \Gamma, \tag{44b}$$

$$\nu u^\gamma = \partial_n\varphi^\gamma \text{ on } \Gamma. \tag{44c}$$

Theorem 5.1. *The semismooth Newton method to solve* (44a)*,* (44b)*,* (44c) *converges locally q-superlinearly.*

Proof. We may write the system (44a), (44b), (44c) as the equation $G(u) = 0$, where $G : L^2(\Gamma) \to L^2(\Gamma)$ is given by

$$G(u) = \nu u + S^*Su - S^*y_\Omega + S^*\big(\chi_{\bar\omega}\max\{0, \mu^* + \gamma(Su - b)\}\big).$$

Using the regularity results in [5, 8], together with the election of μ^* in $L^q(\omega)$ we have that $\mu^* + \gamma(Su - b) \in L^q(\Omega)$ for some $q > 2$, and hence G is semismooth in the sense stated in [20]; see Theorems 2.13 and 2.10(c) in the aforementioned reference.

Define now $\omega_A(u) = \{x \in \bar\omega :\ \mu^* + \gamma(Su - b) > 0\}$. A slant differential of $G(u)$ is given by the expression

$$M(u)v = \nu v + S^*Sv + \gamma S^*\chi_{\omega_A(u)}Sv.$$

Let us see that the inverse of $M(u)$ is uniformly bounded in $\mathcal{L}(L^2(\Gamma), L^2(\Gamma))$ for all

$u \in L^2(\Gamma)$. $M(u)$ is symmetric, continuous, and coercive on $L^2(\Gamma)$, existence and uniqueness of a solution $v \in L^2(\Gamma)$ of the equation $M(u)v = z$ for every $z \in L^2(\Gamma)$ follows from Lax-Milgram theorem. The equation $M(u)v = z$ can be read as

$$\begin{aligned} -\Delta y &= 0 \text{ in } \Omega, & y &= v \text{ on } \Gamma, \\ -\Delta\varphi &= y + \gamma\chi_{\omega_A(u)} y \text{ in } \Omega, & \varphi &= 0 \text{ on } \Gamma, \\ & & \nu v &= \partial_n\varphi + z \text{ on } \Gamma. \end{aligned}$$

Using the definition of solution in the transposition sense and the last equation of this system, we have that

$$(y, y + \gamma\chi_{\omega_A(u)} y)_\Omega = -(v, \partial_n\varphi) = -\nu(v, v)_\Gamma + \nu(v, z)_\Gamma$$

and hence

$$\nu\|v\|^2_{L^2(\Gamma)} = \nu(v, z)_\Gamma - \int_\Omega y^2 - \gamma\int_{\omega_A(u)} y^2 \leq \nu\|v\|_{L^2(\Gamma)}\|z\|_{L^2(\Gamma)}.$$

So we have that $\|v\|_{L^2(\Gamma)} \leq \|z\|_{L^2(\Gamma)}$ and therefore the inverse of $M(u)$ is uniformly bounded in $\mathcal{L}(L^2(\Gamma), L^2(\Gamma))$ for all $u \in L^2(\Gamma)$. □

When $\gamma \to +\infty$, $u^\gamma \to \bar{u}$; see [39, Theorem 3.1] or [16, Pr 2.1].

Let us turn now to the discrete problem (P_h^{S}). Consider the space $\mathcal{M}_h \subset \mathcal{M}(\bar{\omega})$ which is spanned by the Dirac measures corresponding to the nodes $\{x_j\}_{j\in\mathbb{J}}$, where $\mathbb{J} = \{j\colon x_j \in \bar{\omega}\}$. Following [5, Corollary 2], if the continuous problem (P^{S}) has a regular feasible Slater point, then (P_h^{S}) has a unique solution $\bar{u}_h$. Moreover there exist unique $\bar{y}_h \in K_{a,b}^h$, $\bar{\varphi}_h \in Y_{h,0}$ and two nonnegative measures $\bar{\mu}_h^+, \bar{\mu}_h^- \in \mathcal{M}_h$ such that

$$a(\bar{y}_h, z_h) = 0 \ \forall z_h \in Y_{h0}, \ \bar{y}_h = \bar{u}_h \text{ on } \Gamma, \tag{45a}$$

$$a(z_h, \bar{\varphi}_h) = (\bar{y}_h - y_\Omega, z_h) + \langle\bar{\mu}_h^+ - \bar{\mu}_h^-, z_h\rangle \ \forall z_h \in Y_{h0}, \tag{45b}$$

$$\langle\bar{\mu}_h^+ - \bar{\mu}_h^-, y_h - \bar{y}_h\rangle \leq 0 \ \forall y_h \in K_{a,b}^h, \tag{45c}$$

$$\nu\bar{u}_h = \partial_n^h\bar{\varphi}_h \text{ on } \Gamma, \tag{45d}$$

and $\operatorname{supp}\bar{\mu}_h^+ \subset \{x_j : \bar{y}(x_j) = b(x_j)\}$, $\operatorname{supp}\bar{\mu}_h^- \subset \{x_j : \bar{y}(x_j) = a(x_j)\}$.

Since we are dealing with a finite dimensional problem and the max function is known to be semismooth in finite dimension, we could think about applying directly a semismooth Newton method as described in [12]. Nevertheless, the other assumption fails here: the slant derivative may not have an inverse. Let us show this. Consider

$$F(\boldsymbol{u}, \boldsymbol{\mu}) = \begin{pmatrix} \mathcal{A}\boldsymbol{u} - \boldsymbol{f} + \mathcal{S}^T\boldsymbol{\mu} \\ \boldsymbol{\mu} - \max\{0, \boldsymbol{\mu} + \gamma(\mathcal{S}\boldsymbol{u} - \boldsymbol{b})\end{pmatrix}.$$

It is clear that the optimality system (45a)–(45d) is equivalent to the equation $F(\boldsymbol{u}, \boldsymbol{\mu}) = \boldsymbol{0}$. We define the sets of active and free nodes related to a pair control-multiplier as

$$\mathbb{A}(u_h, \boldsymbol{\mu}) = \{j \in \mathbb{J} : \mu_j + \gamma(S_h(u_h)(x_j) - b_j) > 0\}, \ \mathbb{F} = \mathbb{J} \setminus \mathbb{A}.$$

The slant derivative of $F(\boldsymbol{u},\boldsymbol{\mu})$ is given by

$$M(\boldsymbol{u},\boldsymbol{\mu}) = \left(\begin{array}{cc} \mathcal{A} & \mathcal{S}^T \\ -\gamma\mathcal{I}_{\mathbb{J},\mathbb{A}}\mathcal{I}_{\mathbb{A},:}\mathcal{S} & \mathcal{I}_{\mathbb{J},\mathbb{F}}\mathcal{I}_{\mathbb{F},\mathbb{J}} \end{array}\right).$$

Remark 6. The inverse of $M(\boldsymbol{u},\boldsymbol{\mu})$ may not exist for some $(\boldsymbol{u},\boldsymbol{\mu})$. Indeed, for any $(\boldsymbol{z},\boldsymbol{\delta}) \in \mathbb{R}^{N_\mathbb{B}} \times \mathbb{R}^{N_\mathbb{J}}$, we have that

$$M(\boldsymbol{u},\boldsymbol{\mu})\left(\begin{array}{c} \boldsymbol{v} \\ \boldsymbol{\eta} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{z} \\ \boldsymbol{\delta} \end{array}\right) \iff \left\{\begin{array}{rcl} \mathcal{A}\boldsymbol{v} + \mathcal{S}^T\boldsymbol{\eta} & = & \boldsymbol{z}, \\ -\gamma\mathcal{I}_{\mathbb{J},\mathbb{A}}\mathcal{I}_{\mathbb{A},:}\mathcal{S}\boldsymbol{v} + \mathcal{I}_{\mathbb{J},\mathbb{F}}\mathcal{I}_{\mathbb{F},\mathbb{J}}\eta & = & \boldsymbol{\delta}. \end{array}\right.$$

From the second equation we may deduce

$$\gamma\mathcal{I}_{\mathbb{A},:}\mathcal{S}\boldsymbol{v} = -\mathcal{I}_{\mathbb{A},\mathbb{J}}\boldsymbol{\delta}.$$

If $N_\mathbb{A} > N_\mathbb{B}$ we have a linear system with more equations than variables, and hence it will not be consistent for at least one value of $\boldsymbol{\delta}$ and M will not have an inverse.

Instead, we use a Moreau-Yosida penalization of (P_h^{S}). Again we will write only the case of unilateral upper constraint. For some shift function $\mu^* \in Y_h$ such that $\mu^*(x_j) = 0$ if $j \notin \mathbb{J}$, and a parameter $\gamma > 0$, a direct discretization of (Q^γ) could be

$$(\tilde{Q}_h^\gamma) \min_{u_h \in U_h} J_h(u_h) + \frac{1}{2\gamma}\int_{\bar{\omega}} \max\{0, \mu^* + \gamma(S_h u_h - b)\}^2 dx.$$

A practical way of computing the integral in the penalty term is to use the lumped mass matrix $\mathcal{L} \in \mathbb{R}^{N\times N}$. Since we are going to use it only for integrals in $\bar{\omega}$, we define it as

$$\mathcal{L}_{i,j} = 0 \text{ if } i \neq j,\ \mathcal{L}_{j,j} = 0 \text{ if } j \notin \mathbb{J},\ \mathcal{L}_{j,j} = \sum_{k=1}^{N} \mathcal{M}_{k,j}. \text{ if } j \in \mathbb{J}.$$

Therefore, we will be solving

$$(Q_h^\gamma) \min_{u_h \in U_h} J_h(u_h) + \frac{1}{2\gamma}\sum_{j\in\mathbb{J}} \mathcal{L}_{j,j} \max\{0, \mu^*(x_j) + \gamma(S_h u_h(x_j) - b(x_j))\}^2.$$

Problem (Q_h^γ) has a unique solution $u_h^\gamma \in U_h$ and there exist unique $y_h^\gamma \in Y_h$ and $\varphi_h^\gamma \in Y_{h0}$ such that

$$a(y_h^\gamma, z_h) = 0\ \forall z_h \in Y_{h0},\ y_h^\gamma = u_h^\gamma \text{ on } \Gamma, \tag{46a}$$

$$\begin{aligned} a(z_h, \varphi_h^\gamma) = {} & (y_h^\gamma - y_\Omega, z_h)_\Omega \\ & + \sum_{j\in\mathbb{J}} \mathcal{L}_{j,j} \max\{0, \mu^*(x_j) + \gamma(y_h^\gamma(x_j) - b(x_j))\} z_h(x_j)\ \forall z_h \in Y_{h0}, \end{aligned} \tag{46b}$$

$$\nu u_h^\gamma = \partial_n^h \varphi_h^\gamma \text{ on } \Gamma. \tag{46c}$$

The nonlinear system (46a)–(46c) can be solved using a semismooth Newton method. To this end, we will use the nonlinear operator $G_h : \mathbb{R}^{N_{\mathbb{B}}} \to \mathbb{R}^{N_{\mathbb{B}}}$ defined by

$$G_h(\boldsymbol{u}) = \mathcal{A}\boldsymbol{u} - \boldsymbol{f} + \mathcal{S}^T\mathcal{L}\max\{\boldsymbol{0}, \boldsymbol{\mu}^* + \gamma(\mathcal{S}\boldsymbol{u} - \boldsymbol{b})\}.$$

Thus, solving (46a)–(46c) is equivalent to solving $G_h(\boldsymbol{u}) = \boldsymbol{0}$.

For every $u_h \in U_h$ we define sets of active and free nodes as

$$\mathbb{A}_\omega(u_h, \gamma, h) = \{j \in \mathbb{J} : \mu^*_j + \gamma(S_h u_h(x_j) - b_j) > 0\},\ \mathbb{F}_\omega(u_h, \gamma, h) = \mathbb{I} \backslash \mathbb{A}_\omega(u_h, \gamma, h). \quad (47)$$

Abusing notation we will drop some or all of the arguments or will use the vector notation when this does not lead to confusion, e.g. $\mathbb{A}_\omega$, $\mathbb{A}_\omega(\boldsymbol{u})$. Notice that if $\mu^* \equiv 0$, then $\mathbb{A}_\omega(u_h, \gamma, h)$ is independent of γ. For the sake of notation, it is also convenient to define for every set $\mathbb{A}_\omega \subseteq \mathbb{J}$ the diagonal matrix $\mathcal{H}(\mathbb{A}_\omega) \in \mathbb{R}^{N\times N}$ such that

$$\mathcal{H}_{i,j} = \delta_{i,j} \left\{ \begin{array}{cl} 0 & \text{if } j \notin \mathbb{A}_\omega \\ \mathcal{L}_{j,j} & \text{if } j \in \mathbb{A}_\omega. \end{array} \right.$$

Abusing notation, we will often write $\mathcal{H}(\boldsymbol{u}) = \mathcal{H}(\mathbb{A}_\omega(\boldsymbol{u}))$ or even we will write $\mathcal{H}$ when this does not lead to confusion. The proof of the following result is as the corresponding one in infinite dimension.

Theorem 5.2. *$G_h(\boldsymbol{u})$ is slantly differentiable, a slant differential is given by*

$$M(\boldsymbol{u})\boldsymbol{v} = \mathcal{A}\boldsymbol{v} + \gamma\mathcal{S}^T\mathcal{H}(\boldsymbol{u})\mathcal{S}\boldsymbol{v}$$

and it has a uniformly bounded inverse w.r.t. $\boldsymbol{u}$.

Notice that using this $\mathcal{H}$ notation, we can write

$$G_h(\boldsymbol{u}) = \mathcal{A}\boldsymbol{u} - \boldsymbol{f} + \mathcal{S}^T\mathcal{H}(\boldsymbol{u})\big(\boldsymbol{\mu}^* + \gamma(\mathcal{S}\boldsymbol{u} - \boldsymbol{b})\big),$$

and therefore, for a given $\boldsymbol{u}$, and denoting $\mathcal{H} = \mathcal{H}(\boldsymbol{u})$, one Newton step reads like

$$\mathcal{A}\boldsymbol{u}^+ + \gamma\mathcal{S}^T\mathcal{H}\mathcal{S}\boldsymbol{u}^+ = \boldsymbol{f} + \mathcal{S}^T\mathcal{H}(\gamma\boldsymbol{b} - \boldsymbol{\mu}^*). \quad (48)$$

Let us comment that for the computation of $\boldsymbol{w} = \mathcal{S}^T\mathcal{H}\boldsymbol{y}$ for some $\boldsymbol{y} \in \mathbb{R}^N$, first we solve

$$\mathcal{K}_{\mathbb{I},\mathbb{I}}\boldsymbol{\phi}_{\mathrm{s}} = \mathcal{H}_{\mathbb{I},:}\boldsymbol{y}, \quad (49)$$

and, next, we have

$$\boldsymbol{w} = \mathcal{H}_{\mathbb{B},:}\boldsymbol{y} - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi}_{\mathrm{s}} = -\mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi}_{\mathrm{s}}$$

because $\mathcal{H}$ is diagonal and its nonzero components correspond to nodes that lie in $\bar{\omega}$, and are hence interior to Ω.

Again a preconditioned conjugate gradient method can be used to solve this system, provided an efficient way of computing $\boldsymbol{d} = (\mathcal{A} + \gamma\mathcal{S}^T\mathcal{H}\mathcal{S})\boldsymbol{v}$ and the second member of the system. Notice that in each of the algorithms 6 and 7 the computation of $\boldsymbol{\phi}$ can be done with just one system solve.

Algorithm 6: Function $\boldsymbol{d} = (\mathcal{A} + \gamma\mathcal{S}^T\mathcal{H}\mathcal{S})\boldsymbol{v}$

```
solve (14) for $\boldsymbol{y}$
solve (15) for $\boldsymbol{\phi}_\mathrm{r}$
solve (49) for $\boldsymbol{\phi}_\mathrm{s}$
set $\boldsymbol{\phi} = \boldsymbol{\phi}_\mathrm{r} + \gamma\boldsymbol{\phi}_\mathrm{s}$
set $\boldsymbol{d} = \mathcal{M}_{\mathbb{B},:}\boldsymbol{y} - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi} + \nu\mathcal{B}_{\mathbb{B},\mathbb{B}}\boldsymbol{u}$
```

Algorithm 7: Computation of $\boldsymbol{c} = \boldsymbol{f} + \mathcal{S}^T\mathcal{H}(\boldsymbol{u})(\gamma\boldsymbol{b} - \boldsymbol{\mu}^*)$

```
set $\boldsymbol{y} = \boldsymbol{y}_\Omega$; solve (15) for $\boldsymbol{\phi}_\mathrm{r}$
set $\boldsymbol{y} = \gamma\boldsymbol{b} - \boldsymbol{\mu}^*$; solve (49) for $\boldsymbol{\phi}_\mathrm{s}$
set $\boldsymbol{\phi} = \boldsymbol{\phi}_\mathrm{r} + \boldsymbol{\phi}_\mathrm{s}$
set $\boldsymbol{c} = \mathcal{M}_{\mathbb{B},:}\boldsymbol{y}_\Omega - \mathcal{K}_{\mathbb{B},\mathbb{I}}\boldsymbol{\phi}$
```

Alternatively, the solution to (48) can be obtained solving

$$\begin{aligned} \begin{bmatrix} \mathcal{M} + \nu\mathcal{B} + \gamma\mathcal{H} & -\mathcal{K}_{:,\mathbb{I}} \\ -\mathcal{K}_{\mathbb{I},:} & \mathcal{O}_{\mathbb{I},\mathbb{I}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^+ \\ \boldsymbol{\varphi}_\mathbb{I}^+ \end{bmatrix} &= \begin{bmatrix} \mathcal{M}\boldsymbol{y}_\Omega + \mathcal{H}(\gamma\boldsymbol{b} - \boldsymbol{\mu}^*)) \\ \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{u}^+ &= \boldsymbol{y}_\mathbb{B}^+. \end{aligned} \tag{50}$$

Finally, the semismooth Newton algorithm to solve our problem reads as:

Algorithm 8: Semismooth Newton method for (Q_h^γ)

```
Set $k = 0$ and initialize $\boldsymbol{u}_0$; compute $\mathbb{A}_{\omega,0} = \mathbb{A}_\omega(\boldsymbol{u}_0)$
Set $\mathbb{A}_\omega = \mathbb{A}_{\omega,k}$ and $\mathcal{H} = \mathcal{H}(\mathbb{A}_\omega)$
Compute $\boldsymbol{u}_{k+1} = \boldsymbol{u}^+$ using (48) [or (50)]
Set $\mathbb{A}_{\omega,k+1} = \mathbb{A}_\omega(\boldsymbol{u}_{k+1})$
Stop or set $k = k + 1$ and return to 2
```

Remark 7. As we noticed in Remark 5, the only information from iteration k used to compute iteration $k+1$ is the set of active indexes. Therefore, when $\mathbb{A}_\omega(\boldsymbol{u}_{k+1}) = \mathbb{A}_\omega(\boldsymbol{u}_k)$ we have reached an stationary point. This is usually the criterion used to stop the semismooth Newton method.

Nevertheless, as we will see below in the context of algorithms 9 and 10, the main use for the solution of (Q_h^γ) will be to provide an initial guess for the next step in those procedures, so it does not seem necessary to solve exactly (Q_h^γ) in general. Following [16], we may implement the following stopping criterion for Algorithm 8. After step 3, we compute an approximation of the multiplier $\boldsymbol{\mu}_{k+1} \in \mathbb{R}^N$ as

$$\mu_{k+1,j} = 0 \text{ if } j \notin \mathbb{A}_{\omega,k},\ \mu_{k+1,j} = \mu_j^* + \gamma(y_{k+1,j} - b_j) \text{ if } j \in \mathbb{A}_{\omega,k}$$

and we may stop the algorithm if

$$\|\mu_{k+1} - \max(\mu^* + \gamma(y_{k+1} - b), 0\|_{L^2(\Omega)} < \varepsilon_\lambda.$$

In practice, this quantity gives a good measure of the change in the active set between iterates as well as the unfeasibility combined with the penalization parameter γ.

We state the convergence result for the Newton method. It follows directly from Theorem 5.2

Theorem 5.3. *The semismooth Newton method to solve* (Q_h^γ) *described in Algorithm 8 converges locally q-superlinearly to* $\boldsymbol{u}$*, the solution of* $G_h(\boldsymbol{u}) = 0$*, provided* $\boldsymbol{u}_0$ *is close enough to* $\boldsymbol{u}$*..*

In all the examples below we have taken the shift $\mu^* = 0$.

Example 5.4. In general it is not a good idea to solve directly (Q_h^γ) for some γ big enough. The resulting intermediate problems are usually very ill conditioned.

We resume the 2D problem taken from [5] and described in Example 3.3. As in [5], we take ω the ball centered at $(-0.1, -0.1)$ with radius 0.2 and $b \equiv 0.15$. We use Algorithm 8 taking $\boldsymbol{u}_0 = \boldsymbol{0}$. The results are summarized in Table 8. We measure the unfeasibility of the state as the maximum constraint violation $\mathrm{mcv}(y_h) = \|\max(y_h - b, 0)\|_{L^\infty(\bar\omega)}$.

Table 8. Direct application of Algorithm 8 for $\gamma = 10^9$

h	γ	mcv y_h^γ	Newton	pcg	$J_h(u_h^\gamma)$
0.2×2^{-4}	1E+09	6.0E-07	19	470	0.3552476482
0.2×2^{-5}	1E+09	1.1E-06	19	468	0.3552689635
0.2×2^{-6}	1E+09	1.6E-06	21	467	0.3552757641
0.2×2^{-7}	1E+09	2.5E-06	21	459	0.3552778183
0.2×2^{-8}	1E+09	2.7E-06	21	451	0.3552784696

The first thing that can be observed is that γ cannot be too big w.r.t. the problem size. Although the algorithm should converge in finite time, a value too big for γ will make the active sets of the intermediate steps fluctuate, and in practice it may not stop. This is what happens in this case with meshes coarser than the first one exposed in the table.

Nevertheless, the computational effort can be better measured by the total number of conjugate gradient iterations made when we solve (48) in step 3 of Algorithm 8. Solving the solution in the finest mesh takes 8246 seconds.

In [16] a continuation strategy is proposed. This reduces considerably the computational effort. We will use subscripts for the iteration number of the semismooth Newton method described in Algorithm 8 and superscripts for the numbering of the iterates of the continuation strategy provided in Algorithm 9. For some sequence $\{\tau^n\}_{n\geq 1}$ such that $\tau^n > 1$ for all $n \geq 1$, we have

Algorithm 9: Continuation strategy to solve (Q_h^γ)

```
Set $n = 0$ and initialize $\gamma^0$ and $\boldsymbol{u}^0$
Compute $\boldsymbol{u}^{n+1}$, the solution of $(Q_h^{\gamma^n})$ using Algorithm 8 with seed $\boldsymbol{u}^n$
Set $\gamma^{n+1} = \tau^{n+1}\gamma^n$
Stop or set $n = n + 1$ and return to 2
```

There are still three important details to be explained about Algorithm 9: first and most important, the choice of the initial guess $\boldsymbol{u}^0$; next, the appropriate values for τ^n; and finally, a suitable stopping criterion.

To obtain a good initial guess, we will solve the problem in a coarser mesh with a smaller value of γ. Since we are going to deal with meshes of different sizes, we will use the notation $\mathbb{A}_\omega(u_h, \gamma, h)$ when needed. We also recall that I_h is the pointwise interpolation operator from $C(\bar\Gamma)$ onto U_h. Given (h, γ^0), we fix $M > 0$ and pick a finite sequence $(h_n, \gamma_n)_{n=0}^{M}$ such that h_n is decreasing, $h_M = h$, γ_n is increasing and $\gamma_M \leq \gamma^0$.

Algorithm 10: Nested concept to choose $\boldsymbol{u}^0$ in Algorithm 9

```
1 Set $n = 0$ and initialize $u_{h_0} \in U_{h_0}$
2 Solve $(Q_{h_n}^{\gamma_n})$ using Algorithm 8 with initial guess $u_{h_n}$
3 Set $u_{h_{n+1}} = I_{h_{n+1}} u_{h_n}^{\gamma_n} \in U_{h_{n+1}}$
4 Set $n = n + 1$
5 If $n < M$ goto 2
6 Return $u_{h_M} \in U_h$
```

Although Algorithm 10 is meaningful for non-nested meshes, the computational effort needed to make the interpolation in step 3 in non-nested meshes can be considerable, specially for 3D problems.

In [16] some criteria are given to choose $\tau^{n+1} > 1$. In practice, if τ^{n+1} is very small the algorithm would not advance; if it is very big, we would lose the advantage given by the continuation strategy.

In [16] the authors stop the continuation strategy if residuals related to the state, the adjoint state and the multiplier are smaller than a certain tolerance of order $O(h^2)$. In our case, since the problem is linear quadratic, the residuals related to the state and the adjoint state are zero (at least up to roundoff error). The residual related to the multiplier can be computed as

$$r_d = \sum_{j \in \mathbb{A}_\omega^{n+1}} \mathcal{L}_{j,j}(y_j^{n+1} - b_j).$$

As an alternative, a tolerance for the maximum constraint violation e_∞ can be used stop if $\text{mcv}(y_h^{\gamma_n}) \leq e_\infty$.

With all these considerations, we propose the following algorithm. Fix a mesh sequence such that $\{h_j\}_{j=0}^{M}$ is decreasing, $\gamma_0 > 0$, $C > 0$, $u_{h_0} \in U_{h_0}$, $n_{\max} \in \mathbb{N}$ and a sequence $\{\tau^n\}_{n=1}^{n_{\max}+1}$ such that $\tau^n > 1$ for all n.

Algorithm 11: Solving (P_h^{S})

```
1  Set $j = 0$, $h = h_j$, $u_{h,0} = u_{h_0}$
2  Set $n = 0$
3  Compute $u_h^{\gamma_n}$, the solution of $(Q_h^{\gamma_n})$, using Algorithm 8 with seed $u_{h,n}$
4  if ($r_d < Ch^2$ or $\mathrm{mcv}(y_h^{\gamma_n}) \le e_\infty$) and $j < M$ then
5      Set $j = j + 1$ and $h = h_j$                    /* Refine mesh */
6      Set $u_{h,n+1} = I_h u_{h_{j-1}}^{\gamma_n} \in U_h$        /* Write last output in the new mesh
        basis */
7  else
8      Set $u_{h,n+1} = u_h^{\gamma_n}$
9  end
10 Set $\gamma_{n+1} = \tau^{n+1}\gamma_n$                        /* Update $\gamma$ */
11 Set $n = n + 1$
12 if ( ($r_d < Ch^2$ or $\mathrm{mcv}(y_h^{\gamma_n}) \le e_\infty$) and $j == M$) or $n > n_{\max}$ then
13     Stop
14 else
15     Goto 3
16 end
```

Example 5.5. Now we apply Algorithm 11 for $\gamma_0 = 1$, $\tau^n = 10$ and a family of 9 nested meshes of sizes $h_j = 0.2 * 2^{-j}$, $j = 0 : 8$, the coarsest for h_0 with 52 nodes and the finest for h_8 with 2689537 nodes.

A summary of the results can be found in Table 9. The total computation time was 1570 seconds, which is a significant improvement compared with the 8246 seconds used by Algorithm 8 (see Example 5.4).

Table 9. Data for continuation strategy and nested choice of the initial point.

j	γ_n	r_d	Newton	pcg	$J_h(u_h^\gamma)$
0	1E+00	1.20E-02	2	12	0.3434276446
1	1E+01	7.24E-03	1	7	0.3464821594
1	1E+02	1.96E-03	1	7	0.3512477862
2	1E+03	2.35E-04	3	25	0.3544114381
3	1E+04	2.40E-05	4	37	0.3550780145
4	1E+05	2.39E-06	4	44	0.3552160750
5	1E+06	2.38E-07	5	59	0.3552595829
6	1E+07	2.37E-08	5	63	0.3552727851
7	1E+08	2.37E-09	5	54	0.3552770205
8	1E+09	2.37E-10	5	50	0.3552784696

Determination of (the support of) the Lagrange multipliers of (P^{S})**.** Comparison of the adjoint state equations for (P_h^{S}) and (Q_h^γ) leads to the approximation formula

$$\bar{\mu}_h^+ \approx \sum_{j\in\mathbb{A}_\omega(u_h^\gamma)} \mathcal{L}_{j,j}(\mu_j^* + \gamma(y_h^\gamma(x_j) - b_j))\delta_{x_j}.$$

Since $\bar{\omega} \subset \Omega$, it is very common that the Lagrange multipliers of the original problem are finite sums of Dirac measures centered at points on the boundary of ω, say $\bar{\mu}^+ = \sum_{k=1}^n \mu_k \delta_{X_k}$ for some $n > 0$ and $X_k \in \partial\omega$.

Increasing γ will have as a result a better approximation of the Lagrange multiplier and its support. Notice, nevertheless, this will done with great effort and the approximation of both the control, the state and the functional optimal value will not improve in a significant way.

Example 5.6. In Example 5.7 the number of active nodes for the last mesh is 748. All of them are on $\partial\omega$ but they are not isolated. We can see in Table 10 how the number of active nodes decreases as γ increases, with little variation of the functional or the state.

Table 10. Determination of the active set. Data for $h = 0.2 \times 2^{-8}$.

γ	$N_{\mathbb{A}_\omega}$	$\mathrm{mcv}(y_h^\gamma)$	$J(u_h^\gamma)$
1E+09	748	2.7E-06	0.3552784696
1E+10	230	9.1E-07	0.3552787761
1E+11	106	2.0E-07	0.3552788799
1E+12	46	4.1E-08	0.3552789024
1E+13	21	7.8E-09	0.3552789074
1E+14	11	1.9E-09	0.3552789083
1E+15	5	2.0E-10	0.3552789086

6. Control and state constraints

Again according to [5], if problem (P^{CS}) admits a feasible Slater point, then it has a unique solution $\bar u \in H^{1/2}(\Gamma)$ and there exist $\bar y \in H^1(\Omega) \cap C(\bar\omega)$, $\bar\varphi \in W_0^{1,t}(\Omega)$ for all $t < 2$ and two nonnegative measures $\bar\mu^+,\ \bar\mu^- \in \mathcal{M}(\bar\omega)$ such that

$$-\Delta \bar y = 0 \text{ in } \Omega,\ \bar y = \bar u \text{ on } \Gamma, \tag{51a}$$

$$-\Delta \bar\varphi = \bar y - y_\Omega + \bar\mu^+ - \bar\mu^- \text{ in } \Omega,\ \bar\varphi = 0 \text{ on } \Gamma, \tag{51b}$$

$$\bar u(x) = \min\{\beta(x), \max\{\alpha(x), \frac{1}{\nu}\partial_n\bar\varphi(x)\}\} \text{ on } \Gamma, \tag{51c}$$

$$\langle \bar\mu^+ - \bar\mu^-, y - \bar y\rangle \le 0\ \forall y \in K_{a,b} \tag{51d}$$

and $\operatorname{supp}\bar\mu^+ \subset \{\bar y = b\}$, $\operatorname{supp}\bar\mu^- \subset \{\bar y = a\}$. As we said in the previous section, a semismooth Newton strategy for this problem is meaningless, so instead we are going to deal with a Moreau-Yosida approximation. As we did in the previous sections, we will consider only unilateral constraints $u \le \beta$ on Γ and $y \le b$ in $\bar\omega$ to simplify the notation. For a shift function $\mu^* \in L^q(\omega)$ for some $q > 2$ and a parameter $\gamma > 0$, we consider the problem

$$(Q^{C,\gamma}) \min_{u \in U_{-\infty,\beta}} J(u) + \frac{1}{2\gamma}\int_{\bar\omega} \max\{0, \mu^* + \gamma(Su - b)\}^2 dx.$$

This problem has a unique solution $u^\gamma \in H^{1/2}(\Gamma)$. Moreover, there exist $y^\gamma \in H^1(\Omega)$, $\varphi^\gamma \in H^s(\Omega)$, $s > 3/2$, such that

$$-\Delta y^\gamma = 0 \text{ in } \Omega,\ y^\gamma = u^\gamma \text{ on } \Gamma, \tag{52a}$$

$$-\Delta \varphi^\gamma = y^\gamma - y_\Omega + \max\{0, \mu^* + \gamma(y^\gamma - b)\} \text{ in } \Omega,\ \varphi^\gamma = 0 \text{ on } \Gamma, \tag{52b}$$

$$(-\partial_n\varphi^\gamma + \nu u^\gamma, u - u^\gamma) \ge 0 \text{ for all } u \in U_{-\infty,\beta}. \tag{52c}$$

Define

$$G^\gamma(u) = \nu u - \min\big\{\nu\beta, -S^*Su + S^*y_\Omega - S^*\chi_{\bar\omega}\max\{0, \mu^* + \gamma(Su-b)\}\big\}.$$

It is clear that u^γ is the unique solution of $(Q^{c,\gamma})$ if and only if $G^\gamma(u^\gamma) = 0$. For some fixed shift function $\mu^* \in L^q(\omega)$, $q > 2$, consider the active sets

$$\Gamma_A(u,\gamma) = \{x \in \Gamma : \ -S^*Su + S^*y_\Omega - S^*\chi_\omega \max\{0, \mu^* + \gamma(Su-b) > \nu\beta\}$$

and

$$\omega_A(u,\gamma) = \{x \in \bar\omega : \ \mu^* + \gamma(Su-b) > 0\}.$$

A slant differential of $G^\gamma(u)$ is given by

$$M^\gamma(u)v = \nu v + \chi_{\Gamma_A(u,\gamma)}S^*(1 + \gamma\chi_{\omega_A(u,\gamma)})Sv.$$

Theorem 6.1. *G^γ is slantly differentiable, $M^\gamma(u)$ is a slant differential of G^γ and for every fixed $\nu > 0$, $M^\gamma(u)$ has an inverse in $\mathcal{L}(L^2(\Gamma), L^2(\Gamma))$ uniformly bounded for all $u \in L^2(\Gamma)$. The semismooth Newton method $M^\gamma(u^+ - u) = -G^\gamma(u)$ converges q-superlinearly.*

Proof. The proof follows the same lines as those of Lemma 4.1 and Theorems 4.2 and 5.1 □

As we did for the pure control-constrained case, to deal with a problem better suited to the finite dimensional case, we write the optimality condition (52c) with the help of a Lagrange multiplier. There exists $\lambda^\gamma \in L^2(\Gamma)$ such that, for any $c > 0$,

$$\nu u^\gamma = \partial_n\varphi^\gamma - \lambda^\gamma \text{ on } \Gamma, \tag{53a}$$

$$\lambda^\gamma = \max\{0, \lambda^\gamma + c(u^\gamma - \beta)\} \text{ on } \Gamma. \tag{53b}$$

Define now

$$F_c^\gamma(u,\lambda) = \begin{pmatrix} \nu u + S^*Su - S^*y_\Omega + S^*\chi_{\bar\omega}\max\{0, \mu^* + \gamma(Su-b)\} + \lambda \\ \lambda - \max\{0, \lambda + cu - c\beta\} \end{pmatrix}.$$

We have that

$$\begin{pmatrix} \nu I + S^*(1 + \gamma\chi_{\omega_A(u,\gamma)})S & I \\ c\chi_{\Gamma_{\mathbb{A}(u,\lambda)}} I & \chi_{\Gamma_{\mathbb{F}(u,\lambda)}} I \end{pmatrix} \in \partial^{CL} F_c^\gamma(u,\lambda)$$

and hence a Newton-like method to solve $F_c^\gamma(u,\lambda) = 0$ is given by

$$\begin{array}{rcl} \nu u^+ & = & -S^*(1 + \gamma\chi_{\omega_{\mathbb{A}(u,\gamma)}})Su^+ - y_\Omega) - \lambda^+, \\ u^+ & = & \beta \text{ on } \Gamma_{\mathbb{A}(u,\lambda)}, \\ \lambda^+ & = & 0 \text{ on } \Gamma_{\mathbb{F}(u,\lambda)}. \end{array}$$

Although F_c^γ is known not to be slantly differentiable, it can be proved as in Theorem 4.3 that the sequence generated by the above described Newton-like method to solve

$F_c^\gamma(u,\lambda)=0$ is the same than the one generated by the semismooth Newton method to solve $G^\gamma(u)=0$ provided we take the same initial guess u_0, $c=\nu$ and $\lambda_0 = S^*(y_\Omega - (1+\gamma\chi_{\omega_{\mathbb{A}(u_0,\gamma)}})Su_0)-\nu u_0$.

Let us turn now to the finite dimensional problem. We can write the approximation of problem $(Q_h^{C,\gamma})$ as a constrained optimization problem in $\mathbb{R}^{N_{\mathbb{B}},1}$:

$$(Q_h^{C,\gamma})\left\{\begin{array}{l} \min \frac{1}{2}\boldsymbol{u}^T\mathcal{A}\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u} + \frac{1}{2\gamma}\max\{\boldsymbol{0},\boldsymbol{\mu}^*+\gamma(\mathcal{S}\boldsymbol{u}-\boldsymbol{b})\}^T\mathcal{L}\max\{\boldsymbol{0},\boldsymbol{\mu}^*+\gamma(\mathcal{S}\boldsymbol{u}-\boldsymbol{b})\} \\ \text{subject to } \boldsymbol{u}\le\boldsymbol{\beta}. \end{array}\right.$$

Existence and uniqueness of solution follows immediately from the coerciveness of the discrete penalized functional. Since this discrete functional is also of class C^1, first order optimality conditions are standard and read as

$$\begin{array}{rcl} \mathcal{A}\boldsymbol{u} + \mathcal{S}^T\mathcal{L}\max\{0,\boldsymbol{\mu}^*+\gamma(\mathcal{S}\boldsymbol{u}-\boldsymbol{b})\} + \boldsymbol{\lambda} & = & \boldsymbol{f}, \\ \boldsymbol{\lambda} - \max(0,\boldsymbol{\lambda}+c(\boldsymbol{u}-\boldsymbol{\beta})) & = & 0. \end{array} \tag{54}$$

Using the definitions (40) and (47) for the active sets of indexes $\mathbb{A}(\boldsymbol{u},\boldsymbol{\lambda})$ and $\mathbb{A}_\omega(\boldsymbol{u},\gamma,h)$ and the matrix $\mathcal{H}$ related to $\mathbb{A}_\omega$, we have that one step of Newton's method can be written as

$$\begin{array}{rcl} \boldsymbol{u}_{\mathbb{F}}^+ & = & \arg\min\limits_{\boldsymbol{u}_{\mathbb{F}}} \dfrac{1}{2}\boldsymbol{u}_{\mathbb{F}}^T(\mathcal{I}_{\mathbb{F},\mathbb{B}}(\mathcal{A}+\gamma\mathcal{S}^T\mathcal{H}\mathcal{S})\mathcal{I}_{\mathbb{B},\mathbb{F}})\boldsymbol{u}_{\mathbb{F}} \\ & & \qquad - \left(\mathcal{I}_{\mathbb{F},\mathbb{B}}(\,\boldsymbol{f} - \mathcal{A}\mathcal{I}_{\mathbb{B},\mathbb{A}}\boldsymbol{\beta}_{\mathbb{A}} - \mathcal{S}^T\mathcal{H}(\boldsymbol{\mu}^*+\gamma(\mathcal{S}\mathcal{I}_{\mathbb{B},\mathbb{A}}\boldsymbol{\beta}_{\mathbb{A}}-\boldsymbol{b}))\,)\right)^T\boldsymbol{u}_{\mathbb{F}} \\ \boldsymbol{u}_{\mathbb{A}}^+ & = & \boldsymbol{\beta}_{\mathbb{A}} \\ \boldsymbol{\lambda}^+ & = & \boldsymbol{f} - \mathcal{A}\boldsymbol{u}^+ - \mathcal{S}^T\mathcal{H}(\boldsymbol{\mu}^*+\gamma(\mathcal{S}\boldsymbol{u}^+-\boldsymbol{b})) \end{array} \tag{55}$$

or, alternatively, as

$$\begin{array}{c} \left[\begin{array}{ccc} \mathcal{M}+\nu\mathcal{B}+\gamma\mathcal{H} & -\mathcal{K}_{:,\mathbb{I}} & \mathcal{I}_{:,\mathbb{A}} \\ -\mathcal{K}_{\mathbb{I},:} & \mathcal{O}_{\mathbb{I},\mathbb{I}} & \mathcal{O}_{\mathbb{I},\mathbb{A}} \\ \mathcal{I}_{\mathbb{B},:} & \mathcal{O}_{\mathbb{B},\mathbb{I}} & \mathcal{O}_{\mathbb{B},\mathbb{A}} \end{array}\right] \left[\begin{array}{c} \boldsymbol{y}^+ \\ \boldsymbol{\varphi}_{\mathbb{I}}^+ \\ \boldsymbol{\lambda}^+ \end{array}\right] = \left[\begin{array}{c} \mathcal{M}\boldsymbol{y}_\Omega + \mathcal{H}(\gamma\boldsymbol{b}-\boldsymbol{\mu}^*) \\ \boldsymbol{0} \\ \mathcal{I}_{\mathbb{A},\mathbb{B}}\boldsymbol{\beta} \end{array}\right] \\ \begin{array}{rcl} \boldsymbol{u}^+ & = & \boldsymbol{y}_{\mathbb{B}}^+ \\ \boldsymbol{\lambda}_{\mathbb{F}}^+ & = & \boldsymbol{0}. \end{array} \end{array} \tag{56}$$

An adaptation of algorithms 5 and 8 to solve $(Q_h^{C,\gamma})$ is straightforward, and so is an adaptation of Algorithm 11 to use a continuation strategy together with a nested mesh strategy.

Example 6.2. We repeat Example 5.5 adding the control constraint $u\le 0.16$. We obtain the results summarized in Table 11. The total computation time was 2894 seconds. Although this may seem a lot of time, we have to remember that we are solving an optimal control problem with pointwise constraints in both the control and the state using a mesh with almost 2.7E+06 nodes and 5.4E+06 elements to test the algorithm.

Table 11. Convergence history for the solution of (P^{CS}).

j	γ_n	r_d	Newton	pcg	$J_h(u_h^\gamma)$
0	1E+00	9.42E-04	2	10	0.3527536758
1	1E+01	1.12E-03	2	10	0.3530056222
2	1E+02	9.03E-04	3	15	0.3535235000
3	1E+03	2.26E-04	4	33	0.3546079241
4	1E+04	2.38E-05	4	40	0.3551408902
5	1E+05	2.36E-06	6	59	0.3552411683
6	1E+06	2.34E-07	6	65	0.3552759281
7	1E+07	2.33E-08	8	94	0.3552874384
8	1E+08	2.32E-09	9	104	0.3552912421

Acknowledgments

The author would like the anonymous referees for their careful reading of the original manuscript and their suggestions and corrections.

Funding

The author was partially supported by the Spanish Ministerio de Ciencia e Innovación under Projects MTM2014-57531-P and MTM2017-83185-P.

References

[1] Casas E, Raymond JP. Error estimates for the numerical approximation of Dirichlet boundary control for semilinear elliptic equations. SIAM J Control Optim. 2006; 45(5):1586–1611. Available from: `http://dx.doi.org/10.1137/050626600`.

[2] Deckelnick K, Günther A, Hinze M. Finite element approximation of Dirichlet boundary control for elliptic PDEs on two- and three-dimensional curved domains. SIAM J Control Optim. 2009;48(4):2798–2819. Available from: `http://dx.doi.org/10.1137/080735369`.

[3] Günther A. Structure exploiting Galerkin schemes for optimal control of PDEs with constraints on the involved variables [dissertation]. Universität Hamburg; 2010. Available from: `http://ediss.sub.uni-hamburg.de/volltexte/2010/4926/pdf/PhDThesis_Guenther.pdf`.

[4] May S, Rannacher R, Vexler B. Error analysis for a finite element approximation of elliptic Dirichlet boundary control problems. SIAM J Control Optim. 2013;51(3):2585–2611. Available from: `http://dx.doi.org/10.1137/080735734`.

[5] Mateos M, Neitzel I. Dirichlet control of elliptic state constrained problems. Comput Optim Appl. 2016;63(3):825–853. Available from: `http://dx.doi.org/10.1007/s10589-015-9784-y`.

[6] Casas E, Sokolowski J. Approximation of boundary control problems on curved domains. SIAM J Control Optim. 2010;48(6):3746–3780. Available from: `http://dx.doi.org/10.1137/090761550`.

[7] Casas E, Günther A, Mateos M. A paradox in the approximation of Dirichlet control problems in curved domains. SIAM J Control Optim. 2011;49(5):1998–2007. Available from: `http://dx.doi.org/10.1137/100794882`.

[8] Apel T, Mateos M, Pfefferer J, Rösch A. On the regularity of the solutions of Dirichlet optimal control problems in polygonal domains. SIAM J Control Optim. 2015;53(6):3620–3641. Available from: `http://dx.doi.org/10.1137/140994186`.

[9] Gong W, Hinze M, Zhou Z. Finite element method and a priori error estimates for

Dirichlet boundary control problems governed by parabolic PDEs. J Sci Comput. 2016; 66(3):941–967. Available from: `http://dx.doi.org/10.1007/s10915-015-0051-2`.

[10] Chen X, Nashed Z, Qi L. Smoothing methods and semismooth methods for nondifferentiable operator equations. SIAM J Numer Anal. 2000;38(4):1200–1216 (electronic). Available from: `http://dx.doi.org/10.1137/S0036142999356719`.

[11] Bergounioux M, Ito K, Kunisch K. Primal-dual strategy for constrained optimal control problems. SIAM J Control Optim. 1999;37(4):1176–1194 (electronic). Available from: `http://dx.doi.org/10.1137/S0363012997328609`.

[12] Bergounioux Mt, Kunisch K. Primal-dual strategy for state-constrained optimal control problems. Comput Optim Appl. 2002;22(2):193–224. Available from: `http://dx.doi.org/10.1023/A:1015489608037`.

[13] Kunisch K, Rösch A. Primal-dual active set strategy for a general class of constrained optimal control problems. SIAM J Optim. 2002;13(2):321–334. Available from: `http://dx.doi.org/10.1137/S1052623499358008`.

[14] Ulbrich M. Semismooth Newton methods for operator equations in function spaces. SIAM J Optim. 2002;13(3):805–842 (2003). Available from: `http://dx.doi.org/10.1137/S1052623400371569`.

[15] Hintermüller M, Ito K, Kunisch K. The primal-dual active set strategy as a semismooth Newton method. SIAM J Optim. 2003;13(3):865–888 (2003). Available from: `http://dx.doi.org/10.1137/S1052623401383558`.

[16] Hintermüller M, Kunisch K. Feasible and noninterior path-following in constrained minimization with low multiplier regularity. SIAM J Control Optim. 2006;45(4):1198–1221. Available from: `http://dx.doi.org/10.1137/050637480`.

[17] Casas E, Mateos M, Raymond JP. Penalization of Dirichlet optimal control problems. ESAIM Control Optim Calc Var. 2009;15(4):782–809. Available from: `http://dx.doi.org/10.1051/cocv:2008049`.

[18] Hinze M, Vierling M. The semi-smooth Newton method for variationally discretized control constrained elliptic optimal control problems; implementation, convergence and globalization. Optim Methods Softw. 2012;27(6):933–950. Available from: `http://dx.doi.org/10.1080/10556788.2012.676046`.

[19] Hinze M. A variational discretization concept in control constrained optimization: the linear-quadratic case. Comput Optim Appl. 2005;30(1):45–61. Available from: `http://dx.doi.org/10.1007/s10589-005-4559-5`.

[20] Hinze M, Pinnau R, Ulbrich M, Ulbrich S.. Optimization with PDE constraints. (Mathematical Modelling: Theory and Applications; Vol. 23). New York: Springer; 2009.

[21] Gong W, Yan N. Mixed finite element method for Dirichlet boundary control problem governed by elliptic PDEs. SIAM J Control Optim. 2011;49(3):984–1014. Available from: `http://dx.doi.org/10.1137/100795632`.

[22] Casas E, Mateos M, Tröltzsch F. Error estimates for the numerical approximation of boundary semilinear elliptic control problems. Comput Optim Appl. 2005;31(2):193–219. Available from: `http://dx.doi.org/10.1007/s10589-005-2180-2`.

[23] Grisvard P. Singularities in boundary value problems. (Recherches en Mathématiques Appliquées [Research in Applied Mathematics]; Vol. 22). Masson, Paris; Springer-Verlag, Berlin; 1992.

[24] Jerison D, Kenig C. The inhomogeneous Dirichlet problem in Lipschitz domains. J Funct Anal. 1995;130(1):161–219. Available from: `http://dx.doi.org/10.1006/jfan.1995.1067`.

[25] Jerison DS, Kenig CE. The Neumann problem on Lipschitz domains. Bull Amer Math Soc (NS). 1981;4(2):203–207. Available from: `http://dx.doi.org/10.1090/S0273-0979-1981-14884-9`.

[26] Grisvard P. Elliptic problems in nonsmooth domains. (Monographs and Studies in Mathematics; Vol. 24). Boston, MA: Pitman (Advanced Publishing Program); 1985.

[27] Dahlberg BEJ, Kenig CE. Hardy spaces and the Neumann problem in L^p for Laplace's equation in Lipschitz domains. Ann of Math (2). 1987;125(3):437–465. Available from:

http://dx.doi.org/10.2307/1971407.

[28] Berggren M. Approximations of very weak solutions to boundary-value problems. SIAM J Numer Anal. 2004;42(2):860–877 (electronic). Available from: http://dx.doi.org/10.1137/S0036142903382048.

[29] Apel T, Mateos M, Pfefferer J, Rosch A. Error estimates for Dirichlet control problems in polygonal domains: quasi-uniform meshes. Math. Control Relat. Fields. 2008;8(1):217–245. Available from: http://dx.doi.org/10.3934/mcrf.2018010

[30] Schöberl J, Zulehner W. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. SIAM J Matrix Anal Appl. 2007;29(3):752–773 (electronic). Available from: http://dx.doi.org/10.1137/060660977.

[31] Herzog R, Sachs E. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J Matrix Anal Appl. 2010;31(5):2291–2317. Available from: http://dx.doi.org/10.1137/090779127.

[32] Schiela A, Ulbrich S. Operator preconditioning for a class of inequality constrained optimal control problems. SIAM J Optim. 2014;24(1):435–466. Available from: http://dx.doi.org/10.1137/120877532.

[33] Apel T, Sändig AM, Whiteman JR. Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains. Math Methods Appl Sci. 1996;19(1):63–85.

[34] Kamenski L, Huang W, Xu H. Conditioning of finite element equations with arbitrary anisotropic meshes. Math Comp. 2014;83(289):2187–2211. Available from: http://dx.doi.org/10.1090/S0025-5718-2014-02822-6.

[35] Dauge M. Problèmes mixtes pour le laplacien dans des domaines polyédraux courbes. C R Acad Sci Paris Sér I Math. 1989;309(8):553–558.

[36] Nečas J. Direct methods in the theory of elliptic equations. Corrected 2nd ed. Springer Berlin Heidelberg; 2012. Monographs and Studies in Mathematics.

[37] Plaza A. The eight-tetrahedra longest-edge partition and Kuhn triangulations. Comput Math Appl. 2007;54(3):427–433. Available from: http://dx.doi.org/10.1016/j.camwa.2007.01.023.

[38] Clarke F. Optimization and nonsmooth analysis. Toronto: John Wiley & Sons; 1983.

[39] Ito K, Kunisch K. Semi-smooth Newton methods for state-constrained optimal control problems. Systems Control Lett. 2003;50(3):221–228. Available from: http://dx.doi.org/10.1016/S0167-6911(03)00156-7.