STRATEGIES FOR TIME-DEPENDENT PDE CONTROL WITH INEQUALITY CONSTRAINTS USING AN INTEGRATED MODELING AND SIMULATION ENVIRONMENT.

(AUTHORS VERSION. THE ORIGINAL PUBLICATION IS AVAILABLE AT WWW.SPRINGERLINK.COM.)

IRA NEITZEL*, UWE PRÜFERT★ AND THOMAS SLAWIG♦

Abstract. In [17] we have shown how time-dependent optimal control for partial differential equations can be realized in a modern high-level modeling and simulation package. In this article we extend our approach to (state) constrained problems. "Pure" state constraints in a function space setting lead to non-regular Lagrange multipliers (if they exist), i.e. the Lagrange multipliers are in general Borel measures. This will be overcome by different regularization techniques.

To implement inequality constraints, active set methods and barrier methods are widely in use. We show how these techniques can be realized in a modeling and simulation package.

We implement a projection method based on active sets as well as a barrier method and a Moreau Yosida regularization, and compare these methods by a program that optimizes the discrete version of the given problem.

1. Introduction

In this paper we show how time-dependent optimal control problems (OCP) subject to point-wise state constraints can be solved using an equation based integrated modeling and simulation environment (IMSE) like e.g. Comsol Multiphysics, Pdetool, or OpenFoam.

We extend the approach from [17] where we considered time-dependent OCPs without inequality constraints. There we focused on two possible methods to deal with reverse time directions appearing in the optimality systems: A somehow "classical" iterative approach based on subsequently solving the forward state and the backward adjoint equation and an alternative approach transforming the complete optimality system into a coupled elliptic system by interpreting time as an additional space dimension. In this paper we focus on the latter approach.

Throughout this paper we consider optimal control problems

$$\begin{aligned}(1)\qquad \min j(y,u) \;&=\; \frac{\theta_\Omega}{2}\|y(T)-y_\Omega\|^2_{L^2(\Omega)}+\frac{\theta_Q}{2}\|y-y_Q\|^2_{L^2(Q)}\\ &\quad+\frac{\kappa_\Sigma}{2}\|u_\Sigma\|^2_{L^2(\Sigma)}+\frac{\kappa_Q}{2}\|u_Q\|^2_{L^2(Q)}\end{aligned}$$

Key words and phrases. Optimal control, parabolic PDEs, inequality constraints, integrated modeling and simulation environments.

*Research supported by the DFG Schwerpunktprogramm SPP 1253, ★Research supported by the DFG Research Center Matheon, ♦Research supported by the DFG Cluster of Excellence The Future Ocean and the DFG Schwerpunktprogramm SPP 1253.

subject to the parabolic PDE

$$
\begin{aligned}
y_t - \nabla \cdot (\nabla y) &= \beta_Q u_Q + f && \text{in } Q \\
\vec{n} \cdot \nabla y + \alpha y &= \beta_\Sigma u_\Sigma + g && \text{on } \Sigma \\
y(0) &= y_0 && \text{in } \Omega
\end{aligned} \tag{2}
$$

and to the (lower) unilateral constraints[1]

$$
y_a \leq \tau y + \lambda u_Q \quad \text{a.e. in } Q. \tag{3}
$$

In the following, we refer to the parabolic PDE (2) as the state equation. We consider only the case $\tau \geq 0, \lambda \geq 0$ and $\tau + \lambda > 0$. The setting $\tau > 0, \lambda > 0$ implements mixed control-state constraints. For this class of problems, the existence of Lagrange multipliers is often shown by Slater point arguments. Therefore, the state is considered in the space of continuous functions. Due to parabolic regularity theory this can only be expected in the case of one dimensional distributed controls unless higher regularity of controls can be assumed. Even then, caused by the fact that for "pure" state constraints the associated Lagrange multipliers are i.g. Borel measures, regularization techniques can improve the behavior of numerical algorithms, cf. [13], [8], or [1]. For constraints given on the same domain as the control we can use the well investigated Lavrentiev-type regularization, see e.g. the works [14] or [19]. In the case of boundary control and constraints given on the space-time domain the Lavrentiev regularization cannot be applied. Here, some different regularization concepts have been developed, examples can be found in [24] and [9] for elliptic problems and [18] for parabolic PDE control. Alternatively, for distributed as well as for boundary controlled problems, the Moreau-Yosida regularization suggested in [8] may be applied.

The structure and underlying theory of the optimal control problem should be kept in mind when considering appropriate discretization schemes. In most FEM packages and IMSEs an adequate implementation of regular measures is not possible. Of course, one can try to approximate measures by finite elements or discretize them by point-wise evaluations [5], but our aim is to implement optimality conditions (and solve optimal control problems) without additional programming effort. Therefore, measures should be avoided, and we will mainly consider regularized problem formulations in our experiments, that offer the additional benefit that Lagrange multipliers exist without any further restrictions.

In this paper, we investigate some methods to handle state constraints.

First, we use the regularization suggested in [18] in the parabolic boundary controlled case and the classical Lavrentiev-type regularization as discussed in [14] in the case of distributed control. The optimality system can in this case be implemented via a projection formula by the max-function.

Secondly, we test some barrier methods to eliminate the point-wise state constraints. In [21], it is shown that under certain assumptions barrier methods do not need any additional regularization if their order is sufficiently high and the solutions of the state equations are sufficiently smooth. In this sense, barrier methods regularize the problem "by default". The integration of a path-following algorithm into the framework established in [17] needs only minor changes in comparison to the solution of problems without inequality constraints.

Another possible way to overcome the lack of regularity of Lagrange multipliers is the Moreau-Yosida approximation of the Lagrange multipliers, cf. [8], which we also apply to some examples.

[1] In this paper, we only consider unilateral constraints of lower type $y_a \leq \beta y + \gamma u$. The theory for upper constraints $\beta y + \gamma u \leq y_b$ is completely analogous.

This paper is organized as follows: In Section 2 we specify the optimality conditions and quote some results concerning the existence and uniqueness of a minimizer for the given class of problems.

In the following Section 3 we introduce regularization techniques to avoid measures as Lagrange multipliers and ensure the existence of multipliers. In Section 4 we describe different methods to handle inequality constraints algorithmically, followed by the implementation of these algorithms in Section 5 using Comsol Multiphysics as a concrete example for an IMSE.

In Section 6 some examples illustrate the properties of our approach.

2. Optimality conditions for problems with inequality constraints

Assumption 2.1. Let $\Omega \subset \mathbb{R}^N$, $N \in \mathbb{N}$, be a bounded domain with $C^{0,1}$-boundary $\partial\Omega$ if $N \geq 2$, and a bounded interval in $\mathbb{R}$ if $N = 1$. Moreover, for $Q = \Omega \times (0,T)$, $\Sigma = \partial\Omega \times (0,T)$ let functions $f \in L^2(Q)$, $g \in L^2(\Sigma)$, $y_0 \in C(\bar{\Omega})$, $y_Q \in L^2(Q)$, and $y_\Omega \in L^2(\Omega)$ be given. Further, we have real numbers α, β_Q, β_Σ, and non-negative numbers θ_Q, θ_Ω, κ_Q, and κ_Σ. The constraint is a function in $C(\bar{Q})$ with $y_0(x) > y_a(x,0)$.

Note, that the following assertions hold valid if we replace the $-\nabla\cdot\nabla$-operator in (2) by a more general elliptic operator $-\nabla \cdot A\nabla y + a_0 y$, where $a_0 \in L^\infty(Q)$, and $A = (a_{ij}(x))$, $i,j = 1,...,N$ is a symmetric matrix with $a_{ij} \in C^{1,\gamma}(\Omega)$, $\gamma \in (0,1)$, that satisfies the following condition of uniform ellipticity: There is an $m > 0$ such that

$$\xi^\top A(x)\xi \quad \geq \quad m|\xi|^2 \quad \text{ for all } \xi \in \mathbb{R}^N \text{ and all } x \in \bar{\Omega}.$$

In the following we give a brief survey on known results concerning existence and regularity of solutions of parabolic PDEs as well as on the existence of an optimal solution of the control problem. Further, we specify the necessary optimality conditions for the considered cases.

Definition 2.2 We define the solution space for the state equation (2) by

$$W(0,T) = \{y \in L^2(0,T;H^1(\Omega)) \mid y_t \in L^2(0,T,H^1(\Omega)^*)\}.$$

Theorem 2.3. For any triple $(\beta_Q u_Q + f, \beta_\Sigma u_\Sigma + g, y_0) \in L^2(Q)\times L^2(\Sigma)\times L^2(\Omega)$ the initial value problem (2) admits a unique solution $y \in W(0,T)$. There exists a $c > 0$ such that

$$\|y\|_{W(0,T)} \leq c\left(\|\beta_Q u_Q + f\|_{L^2(Q)} + \|\beta_\Sigma u_\Sigma + g\|_{L^2(\Sigma)} + \|y_0\|_{L^2(\Omega)}\right)$$

holds true.

For the proof we refer to [26], [11], or [12].

Theorem 2.4. Let Ω be a bounded domain with $C^{1,1}$-boundary. Further let $(\beta_Q u_Q + f, \beta_\Sigma u_\Sigma + g, y_0) \in L^p(Q)\times L^q(\Sigma)\times L^\infty(\Omega)$ be given. For every $p > N/2+1$ and $s > N+1$, the weak solution of (2) belongs to $L^\infty(Q) \cap C([\delta,T]\times\bar{\Omega})$ for all $\delta > 0$. There is a constant c not depending on (u_Q, u_Σ, f, g), such that

$$\|y\|_{L^\infty(Q)} \leq c\|\beta_Q u_Q + f\|_{L^p(Q)} + \|\beta_\Sigma u_\Sigma + g\|_{L^q(\Sigma)} + \|y_0\|_{L^\infty(\Omega)}$$

holds true. If $y_0 \in C(\bar{\Omega})$, then $y \in C(\bar{Q})$.

For a proof we refer to [20, Prop. 3.3] or [4]

Depending on the choice of the parameters κ_Q, κ_Σ, β_Q, and β_Σ the following meaningful constellations are possible:

(1) distributed control: Let $\kappa_\Sigma = 0$, $\beta_\Sigma = 0$, $\kappa_Q > 0$, $\beta_Q \neq 0$, then $u := u_Q$, $U := L^2(Q)$
(2) boundary control: Let $\kappa_Q = 0$, $\beta_Q = 0$, $\kappa_\Sigma > 0$, $\beta_\Sigma \neq 0$, then $u := u_\Sigma$, $U := L^2(\Sigma)$

(3) boundary and distributed control: Let $\kappa_Q > 0$, $\beta_Q \neq 0$, $\kappa_\Sigma > 0$, $\beta_\Sigma \neq 0$, then $u = (u_Q, u_\Sigma)$. $U := L^2(Q) \times L^2(\Sigma)$,

where U denotes the space of controls.

In our numerical experiments we will consider either distributed or boundary controlled problems.

Theorem 2.5. Let Assumption 2.1. hold. For each of the cases 1–3, Problem (1)–(3) has a unique solution $u^* \in U$ with associated state $y^* \in W(0,T)$.

For a proof we refer to [23].

In order to formulate first order necessary optimality conditions, we rely on the continuity of the optimal state. In contrast to elliptic PDEs here the necessity of continuous states is a stronger restriction. In the case of boundary control, we have only $\beta_\Sigma u_\Sigma + g \in L^2(\Sigma)$, so that the assumption $q > N+1$ of Theorem 2.5 is not fulfilled for $q = 2$. In the case of distributed control, for space dimension $N > 1$ the assumption $p > N/2+1$ is not fulfilled for $p = 2$. In both cases, we do not have the necessary regularity of the state y. To obtain optimality conditions or Lagrange multipliers at all, we have to claim that the optimal state belongs to $C(\bar{Q})$, hence we rely on the following assumption.

Assumption 2.6. Let "pure" state constraints be given. Then we demand: Either $N = 1$ or the control is sufficiently regular, i.e. $u_Q \in L^p$ and $u_\Sigma \in L^q$ with p and q sufficiently large.

Later we will see that regularization techniques will help to avoid this regularity problem.

2.1. **Control constraints.** Setting $\tau = 0$ and $\lambda = 1$, the mixed control-state constraint (3) becomes a control constraint $y_a \leq u$.

Theorem 2.7. Let u^* be the optimal solution to problem (1)–(3) with associated optimal state y^*. The adjoint state p is the solution of the adjoint equation

$$\begin{aligned} -p_t - \nabla\cdot(\nabla p) &= \theta_Q(y^* - y_Q) && \text{in } Q \\ \vec{n}\cdot\nabla p + \alpha p &= 0 && \text{on } \Sigma \\ p(T) &= \theta_\Omega(y^*(T) - y_\Omega) && \text{in } \Omega. \end{aligned}$$

Further, u^* satisfies the projection formula

$$u^* = \begin{cases} \max\{y_a, -\frac{\beta_Q}{\kappa_Q} p\} \text{ in } Q \text{ if } \kappa_Q > 0 \\ \\ \max\{y_a, -\frac{\beta_\Sigma}{\kappa_\Sigma} p\} \text{ on } \Sigma \text{ if } \kappa_\Sigma > 0. \end{cases}$$

The proof of this theorem can be found e.g. in the monograph [23]. The numerical treatment of control constrained problems is widely discussed in the literature, cf. [6], [2], [10], so we abstain from giving examples.

Although we do not specifically deal with purely control constrained problems, these optimality conditions are the basis of the derivation of optimality systems in the case of Lavrentiev-type regularized problems. cf. [15].

2.2. **Pure state constraints.** Setting $\lambda = 0$ and $\tau = 1$, the constraint becomes a pure state constraint $y_a \leq y$.

Theorem 2.8. Let u^* be the optimal solution of problem (1)–(3) that fulfills the Assumption 2.6. with associated optimal state $y^* \in C(\bar{Q})$. Then there exist an adjoint state p and a regular Borel measure $\mu = \mu_Q + \mu_\Sigma + \mu_\Omega$, $\mu \in B(\bar{\Omega} \times (0,T])$, that together with u^* and y^* fulfill the adjoint equation

$$\begin{aligned} -p_t - \nabla\cdot(\nabla p) &= \theta_Q(y^* - y_Q) - \mu_Q && \text{in } Q \\ \vec{n}\cdot\nabla p + \alpha p &= -\mu_\Sigma && \text{on } \Sigma \\ p(T) &= \theta_\Omega(y^*(T) - y_\Omega) - \mu_\Omega && \text{in } \Omega, \end{aligned}$$

the gradient equation

$$0 \;=\; \begin{cases} \kappa_Q u^* + \beta_Q p \text{ in } Q \text{ if } \kappa_Q > 0 \\ \kappa_\Sigma u^* + \beta_\Sigma p \text{ on } \Sigma \text{ if } \kappa_\Sigma > 0 \end{cases}$$

and the complementary slackness conditions

$$\begin{aligned} \iint_{\bar{Q}} (y^* - y_a)\, d\mu(x,t) &= 0 \\ \mu &\geq 0 \\ y^*(x,t) - y_a(x,t) &\geq 0 \quad \forall\, (x,t) \in \bar{Q}. \end{aligned}$$

For a proof we refer to e.g. [20, 4]. Note, that due to Assumption 2.1 we have $\mu|_{\bar{\Omega}\times\{0\}} = 0$ since $y_0(x) > y_a(x,0)$. Note again, that without assumptions on the dimension N or the regularity of the optimal control, optimality conditions cannot generally be shown. This can be overcome for instance by the regularization techniques presented in the following sections. We will also see that then the regularity of the multipliers can be improved so that the use of standard discretization schemes in an IMSE is justified.

3. Regularization of state constraints

3.1. **Mixed control-state constraint.** Let $\lambda > 0$ and $\tau > 0$ be given. This mixed control-state constraint can be seen as model-given or in the case of $\tau \gg \lambda > 0$ as regularization of pure state constraints by perturbation of the state constraint by a small quantity of the control. This technique is known under the term *Lavrentiev-type regularization*, cf. [14]. Without loss of generality, we scale the constraints such that $\tau = 1$, $\lambda > 0$.

Theorem 3.1. Let u^* be the optimal solution of problem (1)–(3) with associated optimal state $y^* \in W(0,T)$. Then there exist an adjoint state $p \in W(0,T)$ and a Lagrange multiplier $\mu = \mu_Q \in L^2(Q)$ such that the adjoint equation

$$\begin{aligned} -p_t - \nabla\cdot(\nabla p) &= \theta_Q(y^* - y_Q) - \mu_Q && \text{in } Q \\ \vec{n}\cdot\nabla p + \alpha p &= \theta_\Sigma(y^* - y_\Sigma) && \text{on } \Sigma \\ p(T) &= \theta_\Omega(y^*(T) - y_\Omega) && \text{in } \Omega, \end{aligned}$$

the gradient equation

$$0 \;=\; \begin{cases} \kappa_Q u^* + \beta_Q p \text{ in } Q \text{ if } \kappa_Q > 0 \\ \kappa_\Sigma u^* + \beta_\Sigma p \text{ on } \Sigma \text{ if } \kappa_\Sigma > 0, \end{cases}$$

and the complementary slackness conditions

$$\begin{aligned} \iint_{\bar{Q}} (y^* + \lambda u^* - y_a)\, \mu_Q \, dxdt &= 0 \\ \mu_Q &\geq 0 \text{ a.e. in } Q \\ y^* + \lambda u^* - y_a &\geq 0 \text{ a.e. in } Q \end{aligned}$$

are satisfied.

For a proof we refer to [15]. Note, that no additional assumptions on the control or dimensions are necessary.

3.2. **Problem case: state constraints in the domain and boundary control.** One standard problem belonging to this class is the following:

$$\min j(y,u) = \frac{\theta_Q}{2}\|y-y_Q\|^2_{L^2(Q)} + \frac{\kappa_\Sigma}{2}\|u_\Sigma\|^2_{L^2(\Sigma)} \tag{4}$$

subject to the boundary controlled PDE

$$\begin{aligned} y_t - \nabla\cdot(\nabla y) &= 0 && \text{in } Q\\ \vec{n}\cdot\nabla y + \alpha y &= \alpha u_\Sigma && \text{on } \Sigma\\ y(0) &= y_0 && \text{in } \Omega \end{aligned} \tag{5}$$

and to the point-wise constraints in the interior of the space-time domain

$$y_a \le y \quad \text{a.e. in } Q \tag{6}$$

In that case we cannot generally expect existence of a Lagrange multiplier, not even in the space of regular Borel measures. The standard Lavrentiev regularization cannot be applied because the control u is not defined in Q. Here, the new approach in [24] (for an elliptic PDE) or [18] (for a parabolic PDE) will help to overcome this problem.

The source-term representation

The idea behind this regularization is to introduce an auxiliary distributed control v which is coupled with u by an adjoint equation, $u = S^*v$, where S denotes the solution operator of the state equation and S^* its adjoint. Here, w is the solution of the adjoint equation and u its trace, cf. [18].

We replace (4)–(6) by the problem

$$\min j(y,w,v) = \frac{\theta_Q}{2}\|y-y_Q\|^2_{L^2(Q)} + \frac{\kappa_\Sigma}{2}\|\alpha w\|^2_{L^2(\Sigma)} + \frac{\epsilon}{2}\|v\|^2_{L^2(Q)} \tag{7}$$

subject to the state equation

$$\begin{aligned} y_t - \nabla\cdot(\nabla y) &= 0 && \text{in } Q\\ \vec{n}\cdot\nabla y + \alpha y &= \alpha^2 w && \text{on } \Sigma\\ y(0) &= y_0 && \text{in } \Omega, \end{aligned} \tag{8}$$

to the additional equation

$$\begin{aligned} -w_t - \nabla\cdot(\nabla w) &= v && \text{in } Q\\ \vec{n}\cdot\nabla w + \alpha w &= 0 && \text{on } \Sigma\\ w(T) &= 0 && \text{in } \Omega \end{aligned} \tag{9}$$

and to the state constraints with modified Lavrentiev-type regularization

$$y_a \le y + \lambda v \quad \text{a.e. in } Q. \tag{10}$$

In [18] convergence for vanishing Lavrentiev parameters is shown if ϵ is chosen according to $\epsilon = c_0\lambda^{1+c_1}$, $c_0 > 0$ and $0 \le c_1 < 1$.

Theorem 3.2. Let (y^*, v^*, w^*) be the optimal solution of (7)–(10). Then there exist adjoint states $p, q \in W(0,T)$, and a Lagrange multiplier $\mu_Q \in L^2(Q)$ such that the state and the adjoint equation

$$\begin{aligned} y_t - \nabla\cdot(\nabla y^*) &= 0 && \text{in } Q\\ \vec{n}\cdot\nabla y^* + \alpha y^* &= \alpha^2 w^*_\Sigma && \text{on } \Sigma\\ y^*(0) &= y_0 && \text{in } \Omega \end{aligned}$$

$$\begin{aligned} -p_t - \nabla\cdot(\nabla p) &= \theta_Q\,(y^* - y_Q) - \mu_Q && \text{in } Q\\ \vec{n}\cdot\nabla p + \alpha p &= 0 && \text{on } \Sigma\\ p(T) &= 0 && \text{in } \Omega, \end{aligned} \tag{11}$$

the control and the second adjoint equation

$$\begin{aligned} -w_t - \nabla \cdot (\nabla w^*) &= -\frac{1}{\epsilon} q + \frac{\lambda}{\epsilon} \mu_Q \quad \text{in } Q \\ \vec{n} \cdot \nabla w^* + \alpha w^* &= 0 \qquad\qquad\quad \text{on } \Sigma \\ w^*(T) &= 0 \qquad\qquad\quad \text{in } \Omega \end{aligned}$$

$$\begin{aligned} q_t - \nabla \cdot (\nabla q) &= 0 \qquad\qquad\qquad \text{in } Q \\ \vec{n} \cdot \nabla q + \alpha q &= \alpha^2 (\kappa_\Sigma w^*_\Sigma + p) \quad \text{on } \Sigma \\ q(0) &= 0 \qquad\qquad\qquad \text{in } \Omega, \end{aligned} \tag{12}$$

the gradient equation

$$\epsilon v^* + q - \lambda \mu_Q = 0 \quad \text{a.e. in } Q, \tag{13}$$

and the complementary slackness conditions

$$\begin{aligned} \iint_Q (y^* + \lambda v - y_a)\, \mu_Q \, dxdt &= 0 \\ \mu_Q &\geq 0 \quad \text{a.e. in } Q \\ y^* + \lambda v^* - y_a &\geq 0 \quad \text{a.e. in } Q \end{aligned} \tag{14}$$

hold.

The proof is given in [18]. Here, the Lagrange multiplier is a regular L^2-function without additional assumptions. Note, that the equations (9) and (12) are of the same type as the state equation (8) and the adjoint equation (11).

3.3. **Moreau-Yosida regularization.** The Moreau Yosida regularization, cf. e.g. [8], replaces the state constraint by a modified objective functional. We consider only the case with state constraints given on the space-time domain Q. The problem reads now

$$\begin{aligned} \min j(y,u) = \; & \frac{\theta_\Omega}{2} \|y(T) - y_\Omega\|^2_{L^2(\Omega)} + \frac{\theta_Q}{2} \|y - y_Q\|^2_{L^2(Q)} \\ & + \frac{\kappa_Q}{2} \|u\|^2_{L^2(Q)} + \frac{\kappa_\Sigma}{2} \|u\|^2_{L^2(\Sigma)} \\ & + \frac{1}{2\gamma} \iint_Q (\gamma (y_a - y) + \bar{\mu}_Q)^2_+ \, dxdt \end{aligned} \tag{15}$$

subject to the state equation (2). This method is referred as penalization, the last integral in (15) is called a penalty term.

Here, $\bar{\mu}_Q$ is an arbitrary function that belongs to $L^2(Q)$ and $\gamma \in \mathbb{R}$, $\gamma \gg 1$ is a regularization parameter. We obtain the Moreau-Yosida regularized multiplier

$$\mu_Q = \max\{0, \bar{\mu}_Q + \gamma (y_a - y)\},$$

which is an object from $L^2(Q)$. Note that this regularization approach works without additional PDEs even for boundary control problems.

4. Algorithms to handle state constraints

4.1. **The projection method.** The projection method replaces the complementary slackness conditions by a projection. This can be viewed as an implementation of the well known active set strategy as a semi-smooth newton method, cf. [7]. In this sense it is possible to show that the complementary slackness conditions (14) are equivalent to

$$\mu_Q = \max\{0, \mu_Q + c\,(y_a - \lambda v^* - y^*)\},$$

for an arbitrarily chosen fixed $c > 0$. Equation (13) yields $\mu_Q = \frac{1}{\lambda}(\epsilon v^* + q)$ in the boundary-controlled case. Choosing $c = \frac{\epsilon}{\lambda^2} > 0$ as in [24], we obtain a projection formula for the Lagrange multiplier

$$(16) \qquad \mu_Q \;=\; \max\left\{0, \frac{1}{\lambda}q + \frac{\epsilon}{\lambda^2}(y_a - y^*)\right\} \quad \text{a.e. in } Q.$$

Now, we have to solve an optimality system consisting of the PDEs (8), (11), (9), and (12), and the projection equation (16). In Section 5.1 we present some details of the implementation of this method in a COMSOL MULTIPHYSICS-script. Note, that for distributed control the approach leads to

$$(17) \qquad \mu_Q = \max\left\{0, \frac{1}{\lambda}p + \frac{\kappa}{\lambda^2}(y_a - y^*)\right\} \quad \text{a.e. in } Q.$$

4.2. **The Moreau Yosida regularization by Ito and Kunisch.** From Section 3.3 we obtain the optimality system

$$\begin{aligned}
y_t - \nabla\cdot(\nabla y) &= \beta_Q u_Q + f \quad \text{in } Q\\
\vec{n}\cdot\nabla y + \alpha y &= \beta_\Sigma u_\Sigma + g \quad \text{on } \Sigma\\
y(0) &= y_0 \quad \text{in } \Omega
\end{aligned}$$

$$\begin{aligned}
-p_t - \nabla\cdot(\nabla p) &= \theta_Q(y - y_Q) - \mu_Q \quad \text{in } Q\\
\vec{n}\cdot\nabla y + \alpha y &= 0 \quad \text{on } \Sigma\\
y(T) &= \theta_\Omega(y(T) - y_\Omega) \quad \text{in } \Omega\\
\mu_Q = \max\{0, \gamma(y_a - y)\} &\quad \text{in } Q
\end{aligned}$$

$$0 \;=\; \begin{cases} \kappa_Q u^* + \beta_Q p \text{ in } Q \text{ if } \kappa_Q > 0\\ \kappa_\Sigma u^* + \beta_\Sigma p \text{ on } \Sigma \text{ if } \kappa_\Sigma > 0\end{cases}$$

where we choose $\bar{\mu}_Q \equiv 0$. For a proof of the validity of the optimality system and for further details see [8] for the elliptic case. We implement this method by a semi-smooth Newton method in Section 5.1.

4.3. **The barrier method.** Barrier methods replace the inequality constraints by adding an arbitrary barrier (or penalty) term to the objective functional.

Definition 4.1.
For all $q \geq 1$ and $\nu > 0$ the function $g(z;\nu;q) : \mathbb{R}_+ \to \mathbb{R}\cup\{+\infty\}$ defined by

$$g(y;\nu;q) := \begin{cases} -\nu\ln(y - y_a) & : q = 1\\ \dfrac{\nu^q}{(q-1)(y-y_a)^{q-1}} & : q > 1\end{cases}$$

is called barrier function of order q. The barrier functional is defined by

$$b(y;\nu,q) := \int_Q g(y;\nu,q)\,dxdt.$$

We can now redefine our basic problems without any constraints. Let $g_i(z_i;\nu;q)$ be barrier functions to the given inequality constraints. Then we eliminate all constraints by defining the new problem

$$(18) \qquad \min f(y,u) = j(y,u) + b(y;\nu,q)$$

subject to (2) and (3). The presence of (3) is necessary because:

- if $q = 1$, the barrier function is not defined if $y - y_a < 0$ on a subset of Ω with measure greater zero.

- if $q > 1$, g has finite values on infeasible points, hence it is not a barrier function.

Another way to handle this problem is introducing an indicator function with respect to the feasible set, cf. [21].

First we observe that every pair (y, u) that holds $f(y, u) < \infty$ is feasible with respect to the inequality constraints. The functional f is coercive, convex, and lower semi-continuous. For $\nu \to 0$ this optimal control problem is equivalent to the problem (1) with pure state constraints.

In our numerical tests we will use logarithmic barrier functionals. This choice is based on [21, Lemma 6.1.] and the following remarks. By the assumed regularity of the optimal state [21, Lemma 6.1.] provides the strict feasibility of the solution (y_ν, u_ν) of (18). Then the objective is directional differentiable at (y_ν, u_ν) and the following theorem holds:

Theorem 4.2. For fixed $\nu > 0$ let u_ν be the optimal solution of Problem (18) with associated optimal state y_ν. Then there exists an adjoint state p such that the first order optimality conditions

$$\begin{aligned} -p_t - \nabla \cdot (\nabla p) &= \theta_Q(y_\nu - y_Q) - \frac{\nu}{y_\nu - y_a} && \text{in } Q \\ \vec{n} \cdot \nabla p + \alpha p &= 0 && \text{on } \Sigma \\ p(T) &= \theta_\Omega(y_\nu(T) - y_\Omega) && \text{in } \Omega, \end{aligned}$$

and

$$0 = \begin{cases} \kappa_Q u_\nu + \beta_Q p & \text{in } Q \text{ if } \kappa_Q > 0 \\ \kappa_\Sigma u_\nu + \beta_\Sigma p & \text{on } \Sigma \text{ if } \kappa_\Sigma > 0 \end{cases}$$

hold. We use the last theorem to implement the path-following Algorithm 1.

Algorithm 1 Path-following

Choose ν_0, $0 < \sigma < 1$, and $\delta > 0$.
Set $k = 0$.
Choose some initial values for y_k, u_k p_k
while $\nu_k > eps$
solve the optimality system
by Newton's method with initial value
(y_k, u_k, p_k) up to an accuracy δ.
set $\nu_{k+1} = \nu_k \cdot \sigma$,
set $k = k + 1$;
end

Remark 4.3. For fixed ν_k [21, Theorem 4.1] provides a unique solution. To find that solution, we have to solve the optimality conditions given by Theorem 4.2. In the spirit of [17], we want to do this by using an IMSE to solve the optimality conditions, i.e. a system of non-linear coupled PDEs, by Newton's method. Unfortunately, we have no information concerning the convergence radius of Newton's method, so we cannot ensure the convergence of the path-following method. However, if we find a solution for some ν_0, we can decrease the path parameter ν by setting $\nu_k = \sigma \nu_{k-1}$. In the worst case, this σ will be almost one. For further details we refer to [25, 19], and [21].

4.4. **Pros and cons of regularization.** The Lavrentiev-type regularization discussed in Section 3.1, the regularization for boundary controlled problems with distributed state constraints from Section 3.2, as well as the Moreau-Yosida regularization lead to well-posed problems in numerous ways.

- First of all, only functions of at least L^2-regularity are involved in the optimality systems. Therefore, we do not need special techniques to handle measures in discrete systems.
- This may lead to an improvement of the numerical treatment, i.e. the Lagrange multipliers can be discretized without additional effort.
- Moreover, Lagrange multipliers exist for arbitrary spatial dimensions without additional assumptions on the optimal control.

On the other hand, some disadvantages have to be mentioned:

- Feasible solutions with respect to the mixed control state constraints $y_a \leq y + \lambda u$ are in general infeasible with respect to the original state constraint $y_a \leq y$. This behavior is known under the term "relaxation of the state constraints".
- For boundary controlled systems, the source-term representation leads to a system that contains two additional PDEs, hence the numerical effort grows significantly. The Moreau-Yosida regularization avoids this disadvantage.
- The choice of parameters is difficult for all methods. One has to find a balance between good smoothing of the Lagrange multiplier, an acceptable violation of feasibility, and an improvement of the properties of the numerical problems, e.g. the condition numbers of matrices, the number of iterations, etc. For the Lavrentiev-type regularization, the results in [13] suggest that the regularization parameter λ should be chosen between 10^{-5} and 10^{-3}. Different choices for parameters for the source-term representation and for the Moreau-Yosida regularization have been considered in [18] for specific examples. The Robin-parameter α is chosen as suggested in [22], the Tikhonov parameter κ is a fixed problem constant. We point out that in our numerical experiments we do not study convergence with respect to regularization parameters numerically. Instead we use fixed parameters.

5. Implementing the optimality systems

In this section we show how the optimality systems collected in the last section can be implemented using an IMSE. We use the (commercial) program Comsol Multiphysics. Our choice is mainly based on the fact that Comsol Multiphysics provides both, a graphical user interface (GUI) and a scripting language (Comsol script), which is very similar to Matlab. If required, it should be easy to translate our programs into e.g. OpenFoam C++ codes.

Although the methods considered in the previous sections are applicable to a wide range of problems, we will carry out our numerical tests on three specific cases: A distributed control problem with observation of the state in the whole space-time domain Q, a distributed control problem with observation at final time T, and a boundary control problem with observation in Q.

Therefore we describe the codes for these specific problems with specifically given data.

5.1. Distributed control.

We begin by showing how distributed control problems can be solved.

Projection method

For a Lavrentiev-type regularized state constrained OCP with distributed control and $\beta_Q = \theta_Q = 1,\ \beta_\Sigma = \theta_\Omega = 0,\ \kappa_\Sigma = 0$, we obtain the optimality system

$$
\begin{aligned}
y_t - \nabla\cdot(\nabla y^*) &= u + f && \text{in } Q\\
\vec{n}\cdot\nabla y^* + \alpha y^* &= g && \text{on } \Sigma\\
y^*(0) &= y_0 && \text{in } \Omega
\end{aligned}
$$

$$
\begin{aligned}
-p_t - \nabla\cdot(\nabla p) &= y^* - y_Q - \mu_Q && \text{in } Q\\
\vec{n}\cdot\nabla p + \alpha p &= 0 && \text{on } \Sigma\\
p(T) &= 0 && \text{in } \Omega
\end{aligned}
$$

$$
\kappa_Q u^* + p - \lambda\mu_Q = 0 \quad \text{a.e. in } Q \tag{19}
$$

$$
\begin{aligned}
\iint\limits_Q (y^* + \lambda u - y_a)\,\mu_Q\,dxdt &= 0\\
\mu_Q &\geq 0 \quad \text{a.e. in } Q\\
y^* + \lambda u^* - y_a &\geq 0 \quad \text{a.e. in } Q.
\end{aligned} \tag{20}
$$

We solve this system using the projection method described in Section 4.1 as one elliptic boundary value problem by interpreting the time as an additional space dimension, for details cf. [17]. Note, that we eliminate the control by $u = -\frac{1}{\kappa_Q}p + \frac{\lambda}{\kappa_Q}\mu_Q$. Also, (19) and (20) are equivalent to (17), cf. Section 4.1.

COMSOL script (as well as most other IMSEs) is based mainly on one data structure here called `fem` which is an object of class structure. It contains all necessary information about the geometry of the domain, the coefficients of the PDE, etc. If one has completely defined the `fem` structure, the problem can be solved by one call of a PDE solver, e.g. by calling the solver for non-linear problems, `femnlin`. The COMSOL script in Listing 1 solves a Lavrentiev-type regularized state constrained problem by the projection method.

We now briefly explain the script line by line since it will serve as the basis for the following scripts. All codes of this section can also be found on the web-page
www.math.tu-berlin.de/Strategies-for-time-dependent-PDE-control.html

Remark 5.1.
In the following we will explain the code for Example 1, cf. Section 6. This means in particular that not all functions from the general problem definition appear in the programs. For instance, in our second example we have $g = 0$ in contrast to the problem explained here. To avoid unnecessary function definitions we disregard all functions and parameters that are identically zero.

First, we define the geometry $Q = (0, \pi)^2$ (line 1). To obtain a structured mesh we initialize the mesh by `meshinit(fem,'hmax',inf)`, line 2. It will be refined by applying `meshrefine` five times, cf. lines 3–5.

For readability, we change the names of the independent variables to `x` and `time` instead of the defaults `x1` and `x2` (line 6). We will describe the PDE in divergence form (`fem.form = 'general'`, cf. line 7).

The unknowns are renamed into y, p, mu, (defaults would be u1 u2 etc.), (line 8). We discretize y and p by quadratic finite elements, whereas we use linear finite elements for `mu`, line 9.

In line 10, we set the Robin parameter `alpha`, the Lavrentiev parameter `lambda`, and the Tikhonov parameter `kappa`. To formulate the PDE in an easy way we specify the inline functions, `y_a, y_d, g,` and `y0` containing the code that describes the lower bound y_a, the desired state y_Q, the function $g(t)$, and the initial value y_0 (lines 11–26). Next, we assign the definition of the inlines to the `fem.function` structure (line 27). In fact, this could also be done directly, e.g. by the assigments

```
fem.functions{1}.type = 'inline';
fem.functions{1}.name = 'y_a(x,time)';
fem.functions{1}.expr = 'min(-sin(x)*sin(time),-0.5)'
```

Now we describe the PDE system given in divergence form $\nabla \cdot \Gamma = F$, $\Gamma = (\Gamma_{l,k})_{l=1\ldots NU, k=1,\ldots NE}$ where NE is the number of partial differential equations contained in the system . Note, that here $NU = N+1$ and the operator ∇ is defined by $\left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_N} \, \frac{\partial}{\partial t}\right)$. Further, F is a column vector of length NE.

Then, the boundary conditions read

$$
\begin{aligned}
-\vec{n} \cdot Ga_l &= G_l + \sum_{m=1}^{M} \frac{\partial R_m}{\partial y_l} \mu_m \\
R_m &= 0.
\end{aligned}
$$

The notation `fem.equ.ga` and `fem.equ.f` in Listing 1 refers to Γ and F, respectively, and `fem.bnd.r` and `fem.bnd.g` correspond to G_l and R_m when defining the PDEs and the boundary conditions.

In lines 28–29 we implement the PDEs. We define Γ as cell array. Formal multiplication of Γ by ∇ yields the Laplacian of y and p. Note, that the derivative with respect to `time` is shifted into F. To complete the definition of the PDEs we define the RHS F including the derivative with respect to `time`, `ytime` and `ptime`. `ytime` as well as `yx` and `px` are predefined operators. This is a major feature of an IMSE.

In lines 30–32 we incorporate the boundary conditions. Note, that on Σ we have the same boundary conditions for all PDEs, which is why we collect them in boundary group number two, cf. line 30. On the (temporal) boundaries at `time=0` and `time=pi` we have different conditions that we define as different Dirichlet boundary conditions in group one (first column in the definition of `fem.bnd.r` and `fem.bnd.g`) and in group three (third column in the definition of `fem.bnd.r` and `fem.bnd.g`).

In case of true Robin boundary conditions

$$
\vec{n} \cdot (\nabla y) + \alpha y = g
$$

`fem.bnd` changes to `g(time)-alpha*y`.

At last, we have to build the structure `fem.xmesh` that contains all necessary data to solve the PDEs (line 33). To obtain an initial solution for the adaptive nonlinear solver `adaption` we call the linear solver `femlin, cf.` line 34.

We solve the problem by one call of the adaptive solver `adaption` supplying the solution of `femlin` as an initial solution (line 35). Finally, in line 36 we visualize the solution `y`.

```
1  fem.geom = rect2(0,pi,0,pi);
2  fem.mesh = meshinit(fem,'hmax',inf);
3  for k = 1:5,
4      fem.mesh = meshrefine(fem,'Rmethod','regular');
5  end
6  fem.sdim={'x' 'time'};
7  fem.form = 'general';
8  fem.dim={'y' 'p' 'mu'};
9  fem.shape=[2 2 1];
10 fem.const={'alpha''0''kappa''1e-3''lambda''1e-3'};
11 fcns{1}.type='inline'; fcns{1}.name='y_a(x,time)';
12 fcns{1}.expr='min(-sin(x)*sin(time),-0.5)';
13 fcns{2}.type='inline'; fcns{2}.name='y0(x)';
14 fcns{2}.expr='0';
15 fcns{3}.type='inline'; fcns{3}.name='g(time)';
16 fcns{3}.expr='sin(time)';
17 fcns{4}.type='inline'; fcns{4}.name='yd(x,time)';
18 fcns{4}.expr='-(1+pi-time)*cos(x)-max(0,sin(x)*sin(time)-0.5)-sin(x)*
       sin(time)';
19 fcns{5}.type='inline'; fcns{5}.name='ud(x,time)';
20 fcns{5}.expr='-sin(x)*(cos(time)+sin(time))+1000*(pi-time)*cos(x)';
21 fcns{6}.type='inline'; fcns{6}.name='y_ex(x,time)';
22 fcns{6}.expr='-sin(x)*sin(time)';
23 fcns{7}.type='inline'; fcns{7}.name='u_ex(x,time)';
24 fcns{7}.expr='-sin(x)*(cos(time)+sin(time))';
25 fcns{8}.type='inline'; fcns{8}.name='p_ex(x,time)';
26 fcns{8}.expr='(pi-time)*cos(x)';
27 fem.functions=fcns;
28 fem.equ.ga={ { {'-yx''0'}{'-px''0'}{'0''0'} } };
29 fem.equ.f={{'-ytime-1/kappa*p+ud(x,time)+lambda/kappa*mu''ptime+y-yd(x
       ,time)-mu''lambda*mu-max(0,1*p-kappa*ud(x,time)+kappa/lambda*(y_a(
       x,time)-y))'}};
30 fem.bnd.ind = [1 2 3 2];
31 fem.bnd.r = { {'y-y0(x)''0''mu'};{'0''0''mu'};{'0''p''mu'} };
32 fem.bnd.g = { {'0''0''0'};{'-alpha*y+g(time)''-alpha*p
       ''0'};{'0''0''0'} };
33 fem.xmesh = meshextend(fem);
34 fem.sol = femlin(fem);
35 fem = adaption(fem,'out','fem','init',fem.sol, 'ngen',1,'Maxiter
       ',500,'Hnlin','off','Ntol',1e-6 );
36 postplot(fem,'tridata','y','triz','y','title','y')
```

Listing 1

A distributed control problem with observation in Ω at time T can be realized by the same code if the definition of the adjoint equation is changed as follows:

```
fem.eqn.f ={ {'-ytime-1/lambda*p' ...
              'ptime-mu'...
              'mu-max(0,1/lambda*p+kappa/lambda^2*(y_a(x,time)-y))'
                  }};
fem.bnd.r = { {'y-y0(x)' '0' 'mu'};
              {'0' '0' 'mu'};
              {'0' 'p-y+y_d(x,time)' 'mu'} };
```

Remark 5.2.

In the definition of `fem.eqn.f` we use the maximum function to implement the projection. This is not a direct call of the function max, but a symbolic definition. COMSOL MULTIPHYSICS evaluates these lines as symbolic expressions and they are also derived symbolically. COMSOL MULTIPHYSICS handles the non-differentiable `max` function by approximating it by a smoothed version of `max`. On the other hand, a direct implementation of a smooth maximum function is possible, for example by using the identity

$$\begin{aligned} \max(x_1, x_2) &= \frac{x_1 + x_2 + |x_1 - x_2|}{2} \\ &= \frac{x_1 + x_2 + \operatorname{sign}(x_1 - x_2) \cdot (x_1 - x_2)}{2} \end{aligned}$$

and replacing `sign` by a smoothed signum function `flsmsign`. The definition of `fem.eqn.f` reads now equivalently

```
fem.equ.f = { {'-ytime-1/kappa*p'...
             'ptime-mu'...
             'mu-0.5*((1/lambda*p+kappa/lambda^2*(y_a(x,time)-y))
             +flsmsign(1/lambda*p+kappa/lambda^2*(y_a(x,time)-y),0.001)
             *(1/lambda*p+kappa/lambda^2*(y_a(x,time)-y)))' }};
```

Moreau-Yosida regularization
For implementing the Moreau Yosida regularization for distributed control, cf. Sections 3.3 and 4.2 , we have to adapt `fem.equ.f`:

```
fem.equ.f = { {'-ytime-alpha^2/kappa*p'...
                   'ptime-mu+y-y_d(x,time)'...
                   '-mu+max(0,gamma*(ya(x,time)-y))' } };
```

and the definition of the boundary conditions in `fem.bnd.r` and `fem.bnd.g`:

```
fem.bnd.r = { {'y-y0(x,time)' '0' '0'};
              {'0' '0' '0'};
              {'0' 'p' '0'}}
fem.bnd.g = { {'0' '0' '0'};
              {'0' '0' '0'};
              {'0' '0' '0'} };
```

Barrier method
In Theorem 3.3. we obtain: If a control u_ν is optimal, and y_ν is the associated optimal state then the state equation

$$\begin{aligned} (y_\nu)_t - \nabla \cdot (\nabla y_\nu) &= (u_\nu)_Q + f \quad \text{in } Q \\ \vec{n} \cdot \nabla y_\nu + \alpha y_\nu &= g \qquad\qquad \text{on } \Sigma \\ y_\nu(0) &= y_0 \qquad\qquad \text{in } \Omega \end{aligned}$$

the adjoint equation

$$\begin{aligned} -p_t - \nabla \cdot (\nabla p) &= (y_\nu - y_Q) - \frac{\nu}{y_\nu - y_a} \quad \text{in } Q \\ \vec{n} \cdot \nabla p + \alpha p &= 0 \qquad\qquad\qquad\quad \text{on } \Sigma \\ p(T) &= 0 \qquad\qquad\qquad\quad \text{in } \Omega, \end{aligned}$$

and the gradient equation

$$p + \kappa_Q u_\nu = 0 \quad \text{a.e. in } Q$$

are satisfied.

Again, we can replace the control by the adjoint p using the gradient equation $\kappa_Q u_\nu + p = 0$ a.e. in Q. We introduce an auxiliary unknown as an approximation on the Lagrange multiplier $\mu_\nu = \dfrac{\nu^q}{(y_\nu - y_a)^q}$. Now we have the additional relations $\mu_\nu (y_\nu - y_a)^q = \nu^q$, $\mu_\nu > 0$, and $(y_\nu - y_a)^q > 0$ a.e. in Q. In the following excerpts of the COMSOL MULTIPHYSICS-script we show how this system is implemented using a complementary function of Fischer-Burmeister type:

$$F_{FB}(y, \mu) := y - y_a + \mu - \sqrt{(y - y_a)^2 + \mu^2 + 2\nu}.$$

```
fem.equ.ga = { { {'-yx'  '0'}
                 {'-px'  '0'}
                 {'0'    '0'} } };
fem.equ.f = { {'-ytime-1/kappa*p' ...
               'ptime-mu+y-y_d(x,time)'...
               '(y-y_a(x,time))+mu-sqrt(mu^2+...
               (y-y_a(x,time))^2+2*nu)'} };
% boundaries: 1:t=0,2:x=pi,3:t=5,4:x=0
fem.bnd.ind = [1 2 3 2];
% boundary conditions:
fem.bnd.r = { {'y-y0(x)'  '0'  'mu'};
              {'0'        '0'  'mu'};
              {'0'        'p'  'mu'} };
fem.bnd.g = { {'0' '0' '0'};
              {'0' '0' '0'};
              {'0' '0' '0'} };
```

The implementation of the heart of the program, the path following loop, is given by

```
fem = adaption(fem,'ngen',1,'Maxiter',50,'Hnlin','on');
nu0=1e-1;
while nu0>0.1e-8,
  nu0=nu0*0.5;
  fem.const{4} =  num2str(nu0);
  fem.xmesh = meshextend(fem);
  fem   = femnlin(fem,'init',fem.sol,...
                'out','fem','Hnlin','off','Maxiter',50);
end;
```

We choose $eps = 10^{-8}$ and $\sigma = 1/2$.

5.2. **Boundary controlled problem. Projection method**

For the solution of the boundary controlled problem with state constraints in the whole domain we have observed the optimality system in Section 3.2., Theorem 3.2. The projection method replaces (13)–(14) by the projection formula

$$\mu_Q \quad = \quad \max\left\{0, \frac{1}{\lambda} q + \frac{\epsilon}{\lambda^2}\left(y_a - y^*\right)\right\} \qquad \text{a.e. in } Q.$$

In order to implement this example, it is sufficient to replace the corresponding lines in Listing 1 by the following ones. Note, that we choose $\alpha = \beta_\Sigma$.

The definition of the unknowns and parameters:

```
fem.dim = {'y' 'w' 'p' 'q' 'mu'};
fem.shape = [2 2 2 2 1];
% parameters:
fem.const = {'alpha' '1e+3'  'kappa'  '1e-3'...
             'lambda' '1e-6' 'epsilon' '1e-9'};
```

The definition of the PDEs:

```
% coefficients + rhd side:
fem.globalexpr = {'v' '-1/epsilon*q+lambda/epsilon*mu'};
fem.equ.ga = {{{'-yx' '0'}
               {'-wx' '0'}
               {'-px' '0'}
               {'-qx' '0'}
               {'0'    '0'}}};
fem.equ.f = { {'-ytime'
               'wtime+v'
               'ptime+y-mu'
               '-qtime'...
               'mu-max(0,1/lambda*q+epsilon/lambda^2...
                *(ya(x,time)-y))' } };
```

The definition of the boundary conditions:

```
fem.bnd.r = {{'y' '0' '0' 'q' '0'};
             {'0' '0' '0' '0' '0'};
             {'0' 'w' 'p' '0' '0'} };
fem.bnd.g = {{'0' '0' '0' '0' '0'};
             {'alpha^2*w-alpha*y' '-alpha*w' '-alpha*p'...
              '-alpha*q+nu*alpha^2*w+alpha^2*p' 0};
             {'0' '0' '0' '0' '0'}};
```

The main difference to Listing 1 is caused by the fact that we have five unknowns `y`, `p`, `w`, `q`, `mu` instead of just `y`, `p`, `mu`. Hence, five PDEs with appropriate boundary conditions have to be implemented. To improve readability we choose to introduce an expression for the control `v`, derived from the gradient equation, in `fem.globalexpr`. This allows the use of `v` in the PDE.

Moreau Yosida regularization

Analogously to the other examples, we use the gradient equation to replace the control u_Σ by $-\frac{\alpha}{\kappa_\Sigma}p$. Altogether, we have to change only four lines of code. First, we have to define some parameters $\alpha = 10$, $\kappa = 10^{-3}$, and $\gamma = 10^3$, where γ is the Moreau-Yosida regularization parameter.

```
% parameters:
fem.const = {'alpha' '1e+3' 'kappa' '1e-3' 'gamma' '1e+3'};
```

The definition of the right-hand-side reads now

```
fem.equ.f = { {'-ytime'
               'ptime+y-mu'
               '-mu+max(0,gamma*(ya(x,time)-y))'} };
```

Finally, we have to write the boundary conditions as

```
fem.bnd.r = { {'y' '0' 'mu'};
              {'0' '0' 'mu-max(0,gamma*(ya(x,time)-y))'};
              {'0' 'p' '0'}}
fem.bnd.g = { {'0' '0' '0'};
              {'-alpha*y-alpha^2/kappa*p' '-alpha*p' '0'};
              {'0' '0' '0'} };
```

Barrier method

Again, very few changes have to be made.

Definition of the PDE:

```
fem.equ.ga = { { {'-yx' '0'}
                 {'-px' '0'}
                 {'0'    '0'}} };
```

```
fem.equ.f = { {'-ytime'...
              'ptime+y-mu'...
              '(y-y_a(x,time))+mu...
             -sqrt(mu^2+(y-y_a(x,time))2+2*nu^2)'} };
```

The implementation of the boundary conditions on Γ reads:

```
fem.bnd.r = {{'y' '0' 'mu'};
             {'0' '0' 'mu'};
             {'0' 'p' 'mu'} };
fem.bnd.g = {{'0' '0' '0'};
             {'-alpha/kappa*p-alpha*y' '-alpha*p' '0'};
             {'0' '0' '0'} };
```

The path-following loop from the distributed case completes the program.

6. Numerical tests

6.1. **An example with given exact solution.** Our first example is a distributed control problem with observation in the whole space-time domain and known analytic solution in order to demonstrate the level of accuracy to be expected by the different solution methods. It is given by

$$\min \frac{1}{2}\|y - y_Q\|^2_{L^2(Q)} + \frac{\kappa_Q}{2}\|u - u_d\|^2_{L^2(Q)}$$

subject to (2) with $\beta_Q = 1$ and $\beta_\Sigma = 0$, and the pointwise state constraint in (3) with $\tau = 1$ and $\lambda = 0$. Due to the linearity of the state equation, the presence of the control shift u_d is covered by our theory. The only difference appears in the gradient equation, which now reads

$$\kappa_Q(u - u_d) + p = 0.$$

Accordingly, all projection formulas change such that u_d is incorporated appropriately. Similarly to the example previously considered in [18] we set

$$\begin{aligned}
u_d &= -\sin(x)\,(\cos(t) + \sin(t)) + \frac{\pi - 1}{\kappa}\cos(x),\\
y_Q &= -(1 + \pi - t)\cos(x) - \max\{\sin(x)\sin(t) - 0.5, 0\} - \sin(x)\sin(t),\\
y_a &= \min\{-\sin(x)\sin(t), -0.5\}.
\end{aligned}$$

Further, we have $y_0 = 0$, $\alpha = 0$, $g = \sin(t)$, $f \equiv 0$, and $\kappa_Q = 10^{-3}$. The space-time domain is given as $Q = (0, \pi)^2$. It can easily be verified that

$$y^* = -\sin(x)\sin(t), \quad u^* = -\sin(x)(\sin(t) + \cos(t))$$

together with

$$p^* = (\pi - t)\cos(x), \quad \mu = \max(\sin(x)\sin(t) - 0.5, 0)$$

solve the optimal control problem. We apply Lavrentiev-type regularization with $\lambda = 10^{-3}$, Moreau-Yosida-regularization with $\gamma = 10^2$, and also solve the problem with the barrier method without additional regularization, setting the stopping parameter $eps = 10^{-5}$. In all our computations we use quadratic finite elements for state and adjoint equations and linear elements for the Lagrange multipliers. We choose a mesh generated by applying the refinement function `meshrefine` five times to an initial mesh created by a call of meshinit with `hmax` set to `inf`. We initialize the adaptive solver with the solution obtained by `femlin`. In `adaption` we set the number of new grid generations `ngen` to one. Moreover, when applying the projection method and the Moreau Yosida regularization, we use the option `hnlin` ("highly nonlinear problem"), which results in a smaller damping factor in Newton's method.

The final mesh obtained by the projection method is shown in Figure 6.1, we observe very similar meshes for the other methods. Table 1 lists the L^2-errors for state, adjoint state and control for the different methods. Note that these errors consist of discretization as well as regularization error, where applicable.

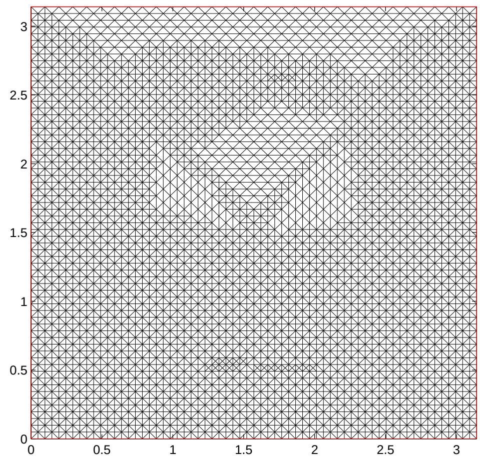

FIGURE 1. Refined mesh to our first example.
Inner-circle diameter of the mesh is between hmin=0.0102 and hmax=0.0204.

	$\|y_h - y^*\|$	$\|u_h - u^*\|$	$\|p_h - p\|$	$j_h(y_h, u_h)$
projection method	1.718e-3	1.558e-2	5.297e-4	8.136e3
moreau-yosida approx.	5.253e-3	2.840e-2	2.840e-5	8.136e3
barrier method	1.687e-4	9.568e-3	9.568e-6	8.136e3

TABLE 1. Errors in y, u, and p measured in the L^2-norm, and values of j_h.

	$\|u_h - u^*\|$			
	N=3	N=4	N=5	N=6
projection method	0.6405	0.0975	0.0185	0.0142
moreau-yosida approx.	0.6386	0.0987	0.0301	0.0276
barrier method	0.6416	0.0946	0.0144	0.0035

TABLE 2. Errors in u measured in the L^2-norm, depending on the number of mesh refinements.

6.2. **Distributed control with state constraints.** Our second example is taken from [19], Example 7.2. There is no given analytic solution so we can only compare numerically computed solutions. The problem is given by

$$\min j(y,u) := \frac{1}{2}\|y(T)\|^2_{L^2(\Omega)} + \frac{\kappa}{2}\|u\|^2_{L^2(Q)}$$

subject to

$$\begin{aligned} y_t - \Delta y &= u && \text{in } Q, \\ \partial_n y + 10y &= 0 && \text{on } \Sigma, \\ y(0) &= y_0 && \text{in } \Omega, \end{aligned}$$

and to the state constraint

$$y \geq y_a := \min\{-100(t(t-1) + x(x-1)) - 49.0, 0.5\} \qquad \text{a.e. in } Q.$$

We choose $\Omega = (0,1) \subset \mathbb{R}^1$, $T = 1$. Further, let $y_0 = \sin(\pi x)$ be given and set $\kappa = 10^{-3}$. Obviously, this problem fits into our general setting with $\theta_\Omega = 1$, $\theta_\Sigma = 0$, $\kappa_Q = \kappa$, $\kappa_\Sigma = 0$, $\beta_Q = 1$, $\beta_\Sigma = 0$, as well as $\alpha = 10$.

For comparison of the solutions, we use the quadratic programming solver from the MOSEK[2] package to compute a reference solution. MOSEK offers an interface to MATLAB's `quadprog` function (from the optimization toolbox). For that, we have to formulate our problem as a discrete optimization problem of the form

$$\min\ z^\top \left(\frac{1}{2}H\right) z + b^\top z$$

subject to

$$\begin{aligned} A_{eq} z &= b_{eq} \\ A_{in} z &\leq b_{in}. \end{aligned}$$

In contrast to `quadprog` from MATLAB's optimization toolbox, MOSEK can handle sparse matrices so that the limitation of the number of unknowns is lifted.

Results

We compare the solution of all methods with a solution computed by the function `quadprog` provided by the package MOSEK. Note, that the solution computed by `quadprog` belongs to the unregularized discrete problem formulation, hence measures do not appear in the optimality system of the problem.

In this example, we refine the initial grid once. The option `hnlin on` is used for all methods. In Table 1 we present the values of j computed by the barrier method, the projection method, and the Moreau Yosida regularization. For comparison, we give results computed by `quadprog`. In the `quadprog`-solver we choose the time step-size as half of the space mesh-size. We set $\lambda = 10^{-3}$, $\gamma = 10^3$, and $eps = 10^{-5}$.

Barrier method	Projection method	Moreau Yosida regularization	quadprog
$j_h(y_h, u_h)$	$j_h(y_h, u_h)$	$j_h(y_h, u_h)$	$j_h(y_h, u_h)$
0.0012427	0.0012430	0.0012703	0.0012899

TABLE 3. Values of j_h for the different solution methods.

6.3. **A boundary controlled problem.** Our third example was taken from [3].

Let $\Omega = [0,1]$ and the time interval $I = (0,5)$ be given. The problem formulation reads as follows:

$$\min\ j(y,u) = \frac{1}{2}\|y\|^2_{L^2(Q)} + \frac{\kappa}{2}\|u\|^2_{L^2(\Sigma)}$$

[2] MOSEK uses an interior point solver. It is an implementation of the homogeneous and self-dual algorithm. For details see the MOSEK manual [16] and the referred literature there. Incontrast to our approch, MOSEK solves a discrete optimization problem ("first discretize, then optimize approach"). See the discussion of the different approaches e.g. in [3].

subject to

$$
(21) \qquad \begin{aligned} y_t - \Delta y &= 0 && \text{in } Q \\ y &= u && \text{on } \Sigma \\ y(0) &= 0 && \text{in } \Omega \end{aligned}
$$

and the point-wise state constraints

$$
y_a \leq y \quad \text{a.e. in } Q.
$$

The state constraint is given by $y_a(x,t) = \sin(x)(\sin(\pi t/5)) - 0.7$, and κ is set to 10^{-3}.

The Dirichlet-boundary condition is difficult in two ways: Neither can it be handled by finite element methods in the usual way, since the usual technique to substitute the boundary integral of the outer normal derivative of the state by the boundary conditions is not applicable, nor are optimality conditions derived easily, cf. [4].

A possible way to overcome this problem is to approximate the Dirichlet-boundary conditions by Robin boundary conditions: For some $\alpha \gg 1$ arbitrarily chosen, we replace (21) by $\vec{n} \cdot \nabla y + \alpha y = \alpha u$ on Σ. [3] We use Robin boundary conditions for a correct finite element implementation of the state equation as well as for a correct implementation of the adjoint equation. We choose $\alpha = 10^3$. In the following, we assume that there is a continuous optimal state which ensures the existence of optimality conditions wherever we consider an unregularized problem.

Results

There is no analytically given solution for the Betts-Campbell problem, so we can only compare solutions computed by different numerical methods as in the last example. In the barrier method, we compute a solution for ν_0 on an adaptively refined mesh before we enter the path following loop and twice within the path-following loop. Obviously, the adaption process leads to very small mesh sizes near the boundaries at time t_0 and where the distance between the state y and the bound y_a is small, i.e. the bound y_a is almost active. In the projection method, the grid-adaption process is done in the same call of the function `adaption` with `ngen` set to two, that solves the complete problem. Figure 2 shows the adaptively refined meshes.

Figure 3 shows the control $u(\pi, t)$ computed by different numerical methods.

In Table 4 we present the values of j computed by the barrier method, the projection method and the Moreau Yosida regularization. For comparison, we give results computed by `quadprog`.

Barrier method.	Projection method	Moreau Yosida regularization	quadprog
$j_h(y_h, u_h)$	$j_h(y_h, u_h)$	$j_h(y_h, u_h)$	$j_h(y_h, u_h)$
0.2327	0.2261	0.2212	0.2346

TABLE 4. Values of j_h for the different solution methods.

7. Conclusion and outlook

As shown in [17] for the unconstrained case, IMSEs (here tested COMSOL MULTIPHYSICS) can be used also for solving optimal control problems with state constraints. Again, a knowledge about the optimality system for the given problem is

[3]Some IMSE like PDETOOL or COMSOL MULTIPHYSICS use this technique for solving Dirichlet boundary problems by default. In this way it is possible to implement the boundary conditions directly in COMSOL MULTIPHYSICS, where it will be corrected internally by using a Robin formulation with a well-chosen parameter α, cf. [22].

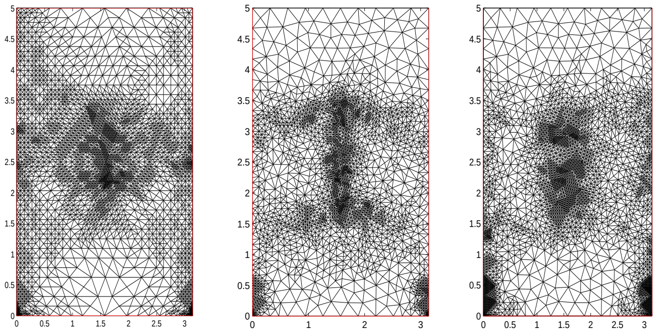

FIGURE 2. Adaptive meshes: barrier method (left), projection method (center), and Moreau Yosida regularization (right). All meshes computed by COMSOL MULTIPHYSICS' `adaption` method with two new grid generations.

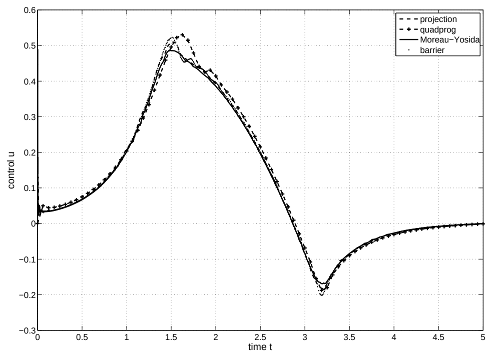

FIGURE 3. Solutions of the Betts-Campbell problem. Controls u_h computed by the Moreau Yosida regularization (solid line), by the projection method (dashed), by the MOSEK solver `quadprog` (dash-cross), and by the barrier method (dotted). An analytic solution is not known. Obviously, the Betts-Campbell problem is symmetric so we show only the control on the boundary with $x = \pi$.

necessary. From the theoretical point of view, the handling of the state constraints and their associated Lagrange multipliers is the most difficult problem. To guarantee the existence of Lagrange multipliers and to avoid measures in the optimality systems, we apply different regularization techniques in order to justify the use of standard FEM discretizations. To handle (state)-constraints algorithmically, three approaches are considered.

Via Lavrentiev-type regularization and source-term representation we arrived at a projection formula for the Lagrange multiplier which leads to an interpretation of the active set strategy as a (semi-smooth) Newton method. The resulting algorithm solved the problem by calls of the solvers `adaption` or `femnlin`, respectively.

Next, we applied a Moreau-Yosida regularization method which can be interpreted as a penalization method and easily implemented with the help of the maximum function. We point out that we used fixed regularization parameters since we do not study the convergence behavior of these methods. It is beyond the scope of this paper to analyze the choice of regularization parameters in more detail. Instead, our main point was to show the implementability of the regularization techniques in IMSEs.

At last, we implemented a barrier method by adding a barrier term to the objective functional. The resulting algorithm is a classical (iterative) path following method. Here, in every step a call of a PDE solver is necessary. On the other hand, the regularity of barrier methods permits to pass additional regularization if the order of the rational barrier function is high enough and the solution of the PDEs are sufficiently smooth, cf. [21, Lemma 6.1].

We first tested our methods on an example with known exact solution. Then, we applied them to model problems where the solutions are not given. To confirm our results, we computed reference-solutions by a well-proven program. All methods produced similar results in a variety up to ten percent. The difference in the results are a combination of discretization and regularization effects. Therefore we cannot directly compare the controls defined on different grids, but we may expect that a well chosen combination of discretization parameters and parameters inherent to the solution technique such as Lavrentiev parameter, the penalty parameter, or the path parameter in the interior point method, results in a closer approximation of the "real" solution.

All together, IMSEs have the capability of solving optimal control problems with inequality constraints and offer an easily implementable alternative for solving such problems if optimality conditions in PDE formulation are known.

Website

WWW.math.tu-berlin.de/Strategies-for-time-dependent-PDE-control.html

References

[1] M. Bergounioux, M. Haddou, M. Hintermüller, and K. Kunisch. A comparison of a Moreau-Yosida-based active set strategy and interior point methods for constrained optimal control problems. *SIAM J. Optimization*, 11:495–521, 2000.

[2] M. Bergounioux, K. Ito, and K. Kunisch. Primal-dual strategy for constrained optimal control problems. *SIAM J. Control and Optimization*, 37:1176–1194, 1999.

[3] J. T. Betts and S. L. Campbell. Discretize then Optimize. In D. R. Ferguson and T. J. Peters, editors, *Mathematics in Industry: Challenges and Frontiers A Process View: Practice and Theory*. SIAM Publications, Philadelphia, 2005.

[4] E. Casas. Pontryagin's principle for state-constrained boundary control problems of semilinear parabolic equations. *SIAM J. Control and Optimization*, 35:1297–1327, 1997.

[5] K. Deckelnick and M. Hinze. Convergence of a finite element approximation to a state constrained elliptic control problem. *SIAM J. Numer. Anal.*, 45:1937–1953, 2007.

[6] W. A. Gruver and E. W. Sachs. *Algorithmic Methods in Optimal Control*. Pitman, London, 1980.

[7] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM J. Optim.*, 13:865–888, 2003.

[8] K. Ito and K. Kunisch. Semi-smooth Newton methods for state-constrained optimal control problems. *Systems and Control Letters*, 50:221–228, 2003.

[9] K. Krumbiegel and A. Rösch. A virtual control concept for state constrained optimal control problems. *Computational Optimization and Applications. Online first. DOI 10.1007/s10589-007-9130-0*, 2007. Computational Optimization and Applications.

[10] K. Kunisch and A. Rösch. Primal-Dual Active Set Strategy for a General Class of Constrained Optimal Control Problems. *SIAM J. on Optimization*, 13:321–334, 2002.

[11] O. A. Ladyzhenskaya, V. A. Solonnikov, and N. N. Ural'ceva. *Linear and Quasilinear Equations of Parabolic Type*. American Math. Society, Providence, R.I., 1968.

[12] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, Berlin, 1971.

[13] C. Meyer, U. Prüfert, and F. Tröltzsch. On two numerical methods for state-constrained elliptic control problems. *Optimization Methods and Software*, 22(6):871–899, 2007.

[14] C. Meyer, A. Rösch, and F. Tröltzsch. Optimal control of PDEs with regularized pointwise state constraints. *Computational Optimization and Applications*, 33:209–228, 2006.

[15] C. Meyer and F. Tröltzsch. On an elliptic optimal control problem with pointwise mixed control-state constraints. In A. Seeger, editor, *Recent Advances in Optimization. Proceedings of the 12th French-German-Spanish Conference on Optimization held in Avignon, September 20-24, 2004*, Lectures Notes in Economics and Mathematical Systems. Springer-Verlag, 2005.

[16] MOSEK ApS. *The MOSEK optimization tools manual. Version 5.0 (Revision 60). http://www.mosek.com*, 2007.

[17] I. Neitzel, U. Prüfert, and T. Slawig. Strategies for time-dependent PDE control using an integrated modeling and simulation environment. part one: problems without inequality constraints. Technical Report 408, Matheon, Berlin, 2007.

[18] I. Neitzel and F. Tröltzsch. On regularization methods for the numerical solution of parabolic control problems with pointwise state constraints. Technical report, SPP 1253, 2007. to appear in ESAIM: COCV.

[19] U. Prüfert and F. Tröltzsch. An interior point method for a parabolic optimal control problem with regularized pointwise state constraints. *ZAMM*, 87(8–9):564–589, 2007.

[20] J.-P. Raymond and H. Zidani. Pontryagin's principle for state-constrained control problems governed by parabolic equations with unbounded controls. *SIAM J. Control and Optimization*, 36:1853–1879, 1998.

[21] A. Schiela. Barrier Methods for Optimal Control Problems with State Constraints. ZIB-Report 07-07, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2007.

[22] The MathWorks. *Partial Differential Equation Toolbox User's Guide*. The Math Works Inc., 1995.

[23] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen*. Vieweg, Wiesbaden, 2005.

[24] F. Tröltzsch and I. Yousept. A regularization method for the numerical solution of elliptic boundary control problems with pointwise state constraints. *Computational Optimization and Applications. Online first. DOI 10.1007/s10589-007-9114-0*, 2008.

[25] M. Ulbrich and S. Ulbrich. Primal-dual interior point methods for PDE-constrained optimization. *Mathematical Programming. Online first.DOI 10.1007/s10107-007-0168-7*, 2007.

[26] J. Wloka. *Partielle Differentialgleichungen*. Teubner-Verlag, Leipzig, 1982.

Address: Ira Neitzel, Uwe Prüfert: TU Berlin - Fakultät II, Institut für Mathematik, Strasse des 17. Juni 136, 10623 Berlin, Germany. Thomas Slawig: Christian-Albrechts-Universität zu Kiel, Technische Fakultät, Christian-Albrechts-Platz 4, 24118 Kiel, Germany