A globalized semi-smooth Newton method for variational discretization of control constrained elliptic optimal control problems

Michael Hinze*& Morten Vierling†

March 17, 2010

Abstract: When combining the numerical concept of variational discretization introduced in [5, 6] and semi-smooth Newton methods for the numerical solution of pde constrained optimization with control constraints [3, 11] special emphasis has to be taken on the implementation, convergence and globalization of the numerical algorithm. In the present work we stretch all these issues. In particular we prove fast local convergence of the algorithm and propose a globalization strategy which is applicable in many practically relevant mathematical settings. We illustrate our analytical and algorithmical findings by numerical experiments.

Mathematics Subject Classification (2010): 49J20, 49K20, 49M15

Keywords: Variational discretization, semi-smooth Newton method, primal-dual active set strategy, Elliptic optimal control problem, control constraints, error estimates.

1 Introduction and mathematical setting

We are interested in the numerical treatment of the following control problem

$$(\mathbb{P}) \quad \begin{cases} \min_{(y,u)\in Y\times U_{ad}} J(y,u) := \frac{1}{2}\|y-z\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|^2_U \\ \text{s.t.} \\ \begin{array}{rcll} -\Delta y & = & Bu & \text{in } \Omega, \\ y & = & 0 & \text{on } \partial\Omega\,. \end{array} \end{cases} \tag{1.1}$$

Here, $\Omega \subset \mathbb{R}^d\,(d \geq 1)$ denotes an open, bounded sufficiently smooth (polyhedral) domain. Given some Hilbert space U and some closed, convex admissible set $U_{ad} \subset U$ for the controls and a linear, continuous control operator $B : U \to H^{-1}(\Omega)$, the states live in $Y := H^1_0(\Omega)$. Let us note that also additional state constraints could be included into our problem setting, as is done in [1] and [2], and also more general (linear) elliptic or parabolic state equations may be considered. However, all structural issues discussed in the present work are induced by the control constraints, hence to keep the exposition as simple as possible state constraints are not considered here.

Typical configurations of $\mathbb{P}$ are

*Schwerpunkt Optimierung und Approximation, Universität Hamburg, Bundesstraße 55, 20146 Hamburg, Germany.

†Schwerpunkt Optimierung und Approximation, Universität Hamburg, Bundesstraße 55, 20146 Hamburg, Germany.

Examples.

(i) $U := \mathbb{R}^m$, $Y = H^1_0(\Omega)$, $B : \mathbb{R}^m \to H^{-1}(\Omega)$, $Bu := \sum_{j=1}^m u_j F_j$, $F_j \in H^{-1}(\Omega)$, $U_{\text{ad}} := \{v \in \mathbb{R}^m; a_j \le v_j \le b_j\}$, $a, b \in \mathbb{R}^m$, $a < b$.

(ii) $U := L^2(\Omega)$, $Y = H^1_0(\Omega)$, $B = \imath : L^2(\Omega) \to H^{-1}(\Omega)$, $\imath$ being the canonical injection, $U_{\text{ad}} := \{v \in L^2(\Omega); a \le v \le b\}$, $a, b \in L^\infty(\Omega)$, $a < b$.

Problem $\mathbb{P}$ admits a unique solution $(y, u) \in Y \times U_{\text{ad}}$, and can equivalently be rewritten as the optimization problem

$$\min_{u \in U_{\text{ad}}} \hat{J}(u) \tag{1.2}$$

for the reduced functional $\hat{J}(u) := J(y(u), u) \equiv J(SBu, u)$ over the set U_{ad}, where $S : Y^* \to Y$ denotes the (continuous) solution operator associated with $-\Delta$ and Dirichlet boundary conditions. We further know that the first order necessary (and here also sufficient) optimality conditions take the form

$$\langle \hat{J}'(u), v - u\rangle_{U^*,U} \ge 0 \text{ for all } v \in U_{\text{ad}} \tag{1.3}$$

where $\hat{J}'(u) = \alpha(u, \cdot)_U + B^*S^*(SBu - z) \equiv \alpha(u, \cdot)_U + B^*p$, with $p := S^*(SBu - z)$ denoting the adjoint variable. Since Y is reflexive, the function p in our setting satisfies

$$\begin{aligned} -\Delta p &= y - z && \text{in } \Omega, \\ p &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{1.4}$$

For the numerical treatment of problem (1.1) it is convenient to rewrite (1.3) for $\sigma > 0$ arbitrary in form of the following non–smooth operator equation;

$$u = P_{U_{\text{ad}}}\left(u - \sigma\nabla\hat{J}(u)\right) \overset{\sigma=1/\alpha}{\equiv} P_{U_{\text{ad}}}\left(-\frac{1}{\alpha}R^{-1}B^*p\right),$$

with the Riesz isomorphism $R : U \to U^*$, the gradient $\nabla\hat{J}(u) = R^{-1}\hat{J}'(u)$ and $P_{U_{\text{ad}}}$ denoting the orthogonal projector onto U_{ad}.

2 Finite element discretization

To discretize ($\mathbb{P}$) we concentrate on Finite Element approaches and make the following assumptions.

Assumption 2.1.
$\Omega \subset \mathbb{R}^d$ denotes a polyhedral domain, $\bar{\Omega} = \cup_{j=1}^{nt}\bar{T}_j$ with admissible quasi-uniform sequences of partitions $\{T_j\}_{j=1}^{nt}$ of Ω, i.e. with $h_{nt} := \max_j \operatorname{diam} T_j$ and $\sigma_{nt} := \min_j\{\sup \operatorname{diam} K; K \subseteq T_j\}$ there holds $c \le \frac{h_{nt}}{\sigma_{nt}} \le C$ uniformly in nt with positive constants $0 < c \le C < \infty$ independent of nt. We abbreviate $\mathcal{T}_h := \{T_j\}_{j=1}^{nt}$.

For $k \in \mathbb{N}$ we set

$$W_h := \{v \in C^0(\bar{\Omega}); v_{|T_j} \in \mathbb{P}_k(T_j) \text{ for all } 1 \le j \le nt\} =: \langle\phi_1, \ldots, \phi_{ng}\rangle, \text{ and}$$
$$Y_h := \{v \in W_h, v_{|\partial\Omega} = 0\} =: \langle\phi_1, \ldots, \phi_n\rangle \subseteq Y,$$

with some $0 < n < ng$. The resulting Ansatz for y_h then is of the form $y_h = \sum_{i=1}^{n} y_i \phi_i$. Now we approximate problem ($\mathbb{P}$) by its variational discretization (compare [6])

$$
(\mathbb{P}_h) \quad \begin{cases} \min_{(y_h,u)\in Y_h\times U_{\mathrm{ad}}} J(y_h,u) := \frac{1}{2}\|y_h - z\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|^2_U \\ \text{s.t.} \\ \quad a(y_h, v_h) \quad = \quad \langle Bu, v_h\rangle_{Y^*,Y} \quad \text{for all } v_h \in Y_h, \end{cases} \tag{2.1}
$$

where $a(y,v) := \int_\Omega \nabla y \nabla v dx$ denotes the bilinear form associated with $-\Delta$. Problem ($\mathbb{P}_h$) admits a unique solution $(y_h, u) \in Y_h \times U_{\mathrm{ad}}$ and, as above, can equivalently be rewritten as the optimization problem

$$
\min_{u\in U_{\mathrm{ad}}} \hat{J}_h(u) \tag{2.2}
$$

for the discrete reduced functional $\hat{J}_h(u) := J(y_h(u), u) \equiv J(S_h Bu, u)$ over the set U_{ad}, where $S_h : Y^* \to Y_h \subset Y$ denotes the solution operator associated with the finite element discretization of $-\Delta$. The first order necessary (and here also sufficient) optimality conditions take the form

$$
\langle \hat{J}'_h(u_h), v - u_h\rangle_{U^*,U} \geq 0 \text{ for all } v \in U_{\mathrm{ad}} \tag{2.3}
$$

where $\hat{J}'_h(v) = \alpha(v,\cdot)_U + B^* S_h^*(S_h Bv - z) \equiv \alpha(v,\cdot)_U + B^* p_h$, with $p_h := S_h^*(S_h Bv - z)$ denoting the adjoint variable. The function p_h in our setting satisfies

$$
a(v_h, p_h) = \langle y_h - z, v_h\rangle_{Y^*,Y} \text{ for all } v_h \in Y_h. \tag{2.4}
$$

Analogously to (1.3), for $\sigma > 0$ arbitrary, we have

$$
u_h = P_{U_{\mathrm{ad}}}\left(u_h - \sigma\nabla \hat{J}_h(u_h)\right) \overset{\sigma=1/\alpha}{\equiv} P_{U_{\mathrm{ad}}}\left(-\frac{1}{\alpha}R^{-1}B^* p_h\right). \tag{2.5}
$$

Remark 2.2. Problem (2.1) is still infinite–dimensional in that the control space is not discretized. This is reflected through the appearance of the projector $P_{U_{\mathrm{ad}}}$ in (2.5). The numerical challenge now consists in designing numerical solution algorithms for problem (2.1) which are implementable, and which reflect the infinite–dimensional structure of the *discrete* problem (2.1) [5, 6].

Next let us investigate the error $\|u - u_h\|_U$ between the solutions u of (1.2) and u_h of (2.2), compare [7].

Theorem 2.3. *Let u denote the unique solution of* (1.2)*, and u_h the unique solution of* (2.2)*. Then there holds*

$$
\alpha\|u - u_h\|^2_U + \frac{1}{2}\|y(u) - y_h\|^2 \leq \langle B^*(p(u) - \tilde{p}_h(u)), u_h - u\rangle_{U*,U} + \frac{1}{2}\|y(u) - y_h(u)\|^2_{L^2(\Omega)}, \tag{2.6}
$$

where $\tilde{p}_h(u) := S_h^*(SBu - z)$, $y_h(u) := S_h Bu$, and $y(u) := SBu$.

Proof. Since (2.2) is an optimization problem defined on all of U_{ad}, the unique solution u of (1.2) is an admissible test function in (2.3). Let us emphasize, that this is different for approaches, where the control space is discretized explictly. In this case we may only expect that u_h is an admissible test function for the continuous problem (if ever). So let us test (1.3) with u_h, and (2.3) with u, and then add the resulting variational inequalities. This leads to

$$
\langle \alpha(u - u_h) + B^*S^*(SBu - z) - B^*S_h^*(S_h Bu_h - z), u_h - u\rangle_{U^*,U} \geq 0.
$$

This inequality is equivalent to

$$\alpha\|u-u_h\|_U^2 \leq \langle B^*(p(u)-\tilde{p}_h(u)) + B^*(\tilde{p}_h(u)-p_h(u_h)), u_h-u\rangle_{U^*,U}.$$

Let us investigate the second addend on the right hand side of this inequality. By definition of the adjoint variables there holds

$$\begin{aligned}\langle B^*(\tilde{p}_h(u)-p_h(u_h), u_h-u\rangle_{U^*,U} &= \langle \tilde{p}_h(u)-p_h(u_h), B(u_h-u)\rangle_{Y,Y^*} = \\ &= a(y_h - y_h(u), \tilde{p}_h(u)-p_h(u_h)) = \int_\Omega (y_h(u_h)-y_h(u))(y(u)-y_h(u_h))dx = \\ &= -\|y_h-y\|_{L^2(\Omega)}^2 + \int_\Omega (y-y_h)(y-y_h(u))dx \leq -\frac{1}{2}\|y_h-y\|_{L^2(\Omega)}^2 + \frac{1}{2}\|y-y_h(u)\|_{L^2(\Omega)}^2\end{aligned}$$

so that the claim of the theorem follows. ∎

Theorem 2.6 tells us that an error estimate for $\|u-u_h\|_U$ is at hand, if

- an error estimate for $\|R^{-1}B^*(p(u)-\tilde{p}_h(u)\|_U$ is available, and
- an error estimate for $\|y(u)-y_h(u)\|_{L^2(\Omega)}$ is available.

Remark 2.4. The error $\|u-u_h\|_U$ between the solution u of problem (1.2) and u_h of (2.2) is completely determined by the approximation properties of the discrete solution operators S_h and S_h^*.

3 Semi-smooth Newton algorithm

In the following we restrict our considerations to the practically relevant case of Example (ii) given in Section 1, i.e. we set $U = L^2(\Omega)$, $Y = H_0^1(\Omega)$, $U_{ad} = \{v \in L^2(\Omega);\ a \leq v \leq b\}$. For simplicity, we assume $a, b \in \mathbb{R}$, $a < b$. Also the control operator is the injection $\imath : L^2(\Omega) \to Y^*$, hence the adjoint $B^* = \imath^*$ is the injection from Y into $L^2(\Omega)$. Below, the operators B, B^* and R are omitted for notational convenience. Problem $(\mathbb{P})$ and its variational discretization $(\mathbb{P}_h)$ can now be expressed by means of

$$G(v) := v - P_{[a,b]}\left(-\frac{1}{\alpha}p(y(v))\right), \text{ and } G_h(v) := v - P_{[a_h,b_h]}\left(-\frac{1}{\alpha}p_h(y_h(v))\right), \tag{3.1}$$

where $P_{[a,b]}$ is the pointwise projection onto the interval $[a, b]$, and for given $v \in L^2(\Omega)$ the functions p, p_h are defined through (1.4) and (2.4), respectively. As discussed in the previous sections,

$$G(u),\ G_h(u_h) = 0 \text{ in } L^2(\Omega). \tag{3.2}$$

These equations will be shown to be amenable to semi–smooth Newton methods as proposed in [3] and [11]. We begin with formulating

Algorithm 3.1. (Semi–smooth Newton algorithm for (3.2))

```
    Start with $v \in L^2(\Omega)$ given. Do until convergence
    Choose $M \in \partial G_h(v)$.
    Solve $M\delta v = -G_h(v)$, $v := v + \delta v$.
```

If we choose Jacobians $M \in \partial G_h(v)$ with $\|M^{-1}\|$ uniformly bounded throughout the iteration, and at the solution u_h the function G_h is ∂G_h-semismooth of order μ, this algorithm is locally superconvergent of order $1 + \mu$ as in [11]. Although Algorithm 3.1 works on the infinite dimensional space $L^2(\Omega)$, it is possible to implement it numerically, as is shown subsequently.

3.1 Semismoothness

To apply the Newton algorithm, we need to confirm that the discretized operator G_h is indeed semismooth. To establish this fact we rewrite G_h in the form

$$G_h(u) = u - P_{[a,b]}\big(F_h(u)\big) \quad \text{with} \quad F_h(u) = -\frac{1}{\alpha}\big(S_h^*(S_h u - z)\big)$$

and apply ([11], Theorem 5.2), with $P_{[a,b]} : \mathbb{R} \to \mathbb{R}$ taking the role of ψ. Note that $F_h : L^2 \to L^q$, $q > 2$ is a smoothing-operator as in [11]. We only mention that

- the operator F_h is differentiable with constant derivative for any $q \geq 1$. In fact, for sufficiently smooth domains Ω, the operators S_h and S_h^* map $L^2(\Omega)$ continuously into $H^2(\Omega)$, which is continuously embedded in $L^q(\Omega)$ for any $q \in [1, \infty]$.
- $P_{[a,b]} : \mathbb{R} \to \mathbb{R}$ is $\partial P_{[a,b]}$-semismooth of order 1, with

$$\partial P_{[a,b]}(x) = \begin{cases} 0 & \text{if } x \notin \text{[a,b]} \\ 1 & \text{if } x \in \text{(a,b)} \\ \text{[0,1]} & \text{if } x = a \text{ or } x = b \end{cases} . \tag{3.3}$$

- for piecewise linear elements the semismooth complementarity condition (5.3) in ([11], theorem 5.2) holds automatically with $\gamma = 1$.

Thus we are in the position to apply Theorem 5.2 from [11] with, in local notation, $\alpha = 1$ and $q_0 > r = 2$ and $\gamma = 1$ and obtain

Theorem 3.2. *The function G_h defined in (3.1) is ∂G_h-semismooth of order $\mu < \frac{1}{3}$. There holds*

$$\partial G_h(v)w = w + \frac{1}{\alpha}\partial P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v))\right) \cdot \left(S_h^* S_h w\right) ,$$

where the application of the differential $\partial P_{[a,b]}$ and the multiplication by $S_h^* S_h w$ are pointwise operations a.e. in Ω.

Remark 3.3. In [4] the mesh independence of the superlinear convergence is stated. Recent results from [12] indicate semismoothness of G of order $\frac{1}{2}$ as well as mesh independent q-superlinear convergence of the Newton algorithm of order $\frac{3}{2}$, if for example the modulus of the slope of $-\frac{1}{\alpha}p(y(\bar{u}))$ is bounded away from zero on the border of the active set, and if the mesh parameter h is reduced appropriately.

3.2 Newton-Algorithm

In order to implement Algorithm 3.1, we have to choose $M \in \partial G_h(v)$. The set-valued function $\partial P_{[a,b]}\big(-\frac{1}{\alpha}p_h(y_h(v))\big)$ contains the characteristic function $\chi_{\mathcal{I}(v)}$ of the inactive set

$$\mathcal{I}(v) = \big\{\omega \in \Omega \,\big|\, \big(-\frac{1}{\alpha}p_h(y_h(v))\big)(\omega) \in (a,b)\big\} .$$

By χ^v we will denote synonymously the characteristic function

$$\chi_{\mathcal{I}(v)} = \begin{cases} 1 \text{ on } \mathcal{I}(v) \\ 0 \text{ everywhere else} \end{cases}$$

as well as the self-adjoint endomorphism in $L^2(\Omega)$ given by the pointwise multiplication with $\chi_{\mathcal{I}(v)}$. With $M = \chi^v$ the Newton-step in Algorithm 3.1 takes the form

$$\Big(I + \frac{1}{\alpha}\chi^v S_h^* S_h\Big)\delta v = -v + P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v)))\right) . \tag{3.4}$$

To obtain an impression of the structure of the next iterate $v^+ = v + \delta v$ we rewrite (3.4) as

$$v^+ = P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v))\right) - \frac{1}{\alpha}\chi^v S_h^* S_h \delta v\,.$$

Since the range of S_h^* is Y_h, the first addend is continuous and piecewise polynomial (of degree k) on a refinement $\mathcal{K}_h$ of $\mathcal{T}_h$. The partition $\mathcal{K}_h$ is obtained from $\mathcal{T}_h$ by inserting nodes and edges along the boundary between the inactive set $\mathcal{I}(v)$ and the according active set, and in general contains simplices of higher order than $\mathcal{T}_h$. The inserted edges are level sets of polynomials of order $\leq k$ since we assume $a, b \in \mathbb{R}$.

The second addend, involving the cut-off function χ^v, is also piecewise polynomial of degree k on $\mathcal{K}_h$ but may jump along the edges not contained in $\mathcal{T}_h$.

Finally v^+ lies in the following finite dimensional subspace of $L^2(\Omega)$

$$Y_h^+ = \left\{\chi^v\varphi_1 + (1-\chi^v)\varphi_2 \,\middle|\, \varphi_1, \varphi_2 \in Y_h\right\} = \operatorname{span}\left(\{\phi_j\chi^v\}_{j=1}^n, \{\phi_j(1-\chi^v)\}_{j=1}^n\right) .$$

The iterates generated by the Newton-algorithm can be represented exactly with about constant effort, since the number of inserted nodes varies only mildly from step to step, once the algorithm begins to converge. Furthermore the number of inserted nodes is bounded, see [5],[6].

Since the Newton-increment δv may have jumps along the borders of both the new and the old active and inactive sets, it is advantageous to compute v^+ directly, because v^+ lies in Y_h^+. To achieve an equation for v^+ we add $G_h'(v)v$ on both sides of (3.4) to obtain

$$\Big(I + \frac{1}{\alpha}\chi^v S_h^* S_h\Big)v^+ = P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v)))\right) + \frac{1}{\alpha}\chi^v S_h^* S_h v\,, \tag{3.5}$$

and reformulate Algorithm 3.1 as

Algorithm 3.4 (Newton Algorithm)**.**

```
$v \in U$ given. Do until convergence
Solve (3.5) for $v^+$, $v := v^+$.
```

3.3 Computing the Newton-Step

Since v^+ defined by (3.5) is known on the active set $\mathcal{A}(v) := \Omega \setminus \mathcal{I}(V)$ it remains to compute v^+ on the inactive set. So we rewrite (3.5) in terms of the unknown $\chi^v v^+$ by splitting v^+ as

$$v^+ = (1-\chi^v)v^+ + \chi^v v^+$$

and obtain

$$\Big(I + \frac{1}{\alpha}\chi^v S_h^* S_h\Big)\chi^v v^+ = P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v)))\right) + \frac{1}{\alpha}\chi^v S_h^* S_h v - \Big(I + \frac{1}{\alpha}\chi^v S_h^* S_h\Big)(1-\chi^v)v^+ .$$

As $(1-\chi^v)v^+$ is already known, we can restrict the latter equation to the inactive set $\mathcal{I}(v)$

$$\Big(\chi^v + \frac{1}{\alpha}\chi^v S_h^* S_h \chi^v\Big)v^+ = \frac{1}{\alpha}\chi^v S_h^* z - \frac{1}{\alpha}\chi^v S_h^* S_h(1-\chi^v)v^+ . \tag{3.6}$$

On the left-hand side of (3.6) we have now a continuous, self-adjoint Operator on $L^2(\mathcal{I}^v)$, which is positive definite, because it is the restriction of the positive definite Operator $\big(I + \frac{1}{\alpha}\chi^v S_h^* S_h \chi^v\big)$ to $L^2(\mathcal{I}^v)$.
Hence we are in the position to apply a CG-algorithm to solve (3.6). Moreover under the assumption of the first iterate lying in

$$Y_h^+\big|_{\mathcal{I}^v} = \left\{\chi^v \varphi \,\middle|\, \varphi \in Y_h\right\} ,$$

as does the solution $\chi^v v^+$, the algorithm does not leave this space because of

$$\left(I + \frac{1}{\alpha}\chi^v S_h^* S_h \chi^v\right) Y_h^+\big|_{\mathcal{I}^v} \subset Y_h^+\big|_{\mathcal{I}^v}$$

and all CG-iterates lie in $Y_h^+\big|_{\mathcal{I}^v}$. These considerations lead to the following

Algorithm 3.5 (Solving (3.5))**.**

> Compute the active and inactive sets $\mathcal{A}^v$ and $\mathcal{I}^v$.
>
> $\forall q \in \mathcal{A}^v$ set
>
> $$v^+(q) = P_{[a,b]}\left(-\frac{1}{\alpha} p_h(y_h(v))(q)\right) .$$
>
> Solve
>
> $$\Big(I + \frac{1}{\alpha}\chi^v S_h^* S_h\Big)\chi^v v^+ = \frac{1}{\alpha}\chi^v S_h^* z - \frac{1}{\alpha}\chi^v S_h^* S_h (1-\chi^v) v^+$$
>
> for $\chi^v v^+$ by CG-iteration. By choosing a starting point in $Y_h^+\big|_{\mathcal{I}^v}$ one ensures that all iterates lie inside $Y_h^+\big|_{\mathcal{I}^v}$.
>
> $v^+ = (1-\chi^v)v^+ + \chi^v v^+$.

We note that the use of this procedure in Algorithm 3.4 coincides with the active set strategy proposed in [3].

3.4 Globalization

Globalization of Algorithm 3.4 may require a damping step of the form

$$v_\lambda^+ = v + \lambda(v^+ - v)$$

with some $\lambda > 0$. According to the considerations above, we have

$$v_\lambda^+ = (1-\lambda)v + \lambda\Big(P_{[a,b]}\left(-\frac{1}{\alpha}p_h(y_h(v))\right) - \frac{1}{\alpha}\chi^v S_h^* S_h \delta v\Big) .$$

Unless $\lambda = 1$ the effort of representing v_λ^+ will in general grow with every iteration of the algorithm, due to the jumps introduced in each step. This problem can be bypassed by focussing on the adjoint state $p_h(v)$ instead of the control v. In fact the function χ^v, now referred to as χ^p, and thus also Equation (3.5) do depend on v only indirectly through the adjoint $p = p_h(v) = S_h^*(S_h v - z)$

$$\Big(I + \frac{1}{\alpha}\chi^p S_h^* S_h\Big)v^+ = P_{[a,b]}\left(-\frac{1}{\alpha}p\right) + \frac{1}{\alpha}\chi^p(p + S_h^* z) . \tag{3.7}$$

Now in each iteration the next full-step iterate v^+ is computed from (3.7). If damping is necessary, one computes $p_\lambda^+ = p_h(v_\lambda^+)$ instead of v_λ^+. In our (linear) setting the adjoint state p_λ^+ simply is a convex combination of $p = p_h(v)$ and $p^+ = p_h(v^+)$

$$p_\lambda^+ = \lambda p^+ + (1-\lambda)p\,,$$

and unlike v_λ^+ the adjoint state p_λ^+ lies in the finite element space Y_h. Thus only a set of additional nodes according to the jumps of the most recent full-step iterate v^+ have to be managed, exactly as in the undamped case.

Algorithm 3.6 (Dampened Newton-Algorithm)**.** $v \in U$ given.
Do until convergence

```
    Solve Equation (3.7) for $v^+$.
    Compute $p^+ = p_h(y_h(v^+))$.
    Choose the damping-parameter $\lambda$. (for example by Armijo line search)
    Set $p := p_\lambda^+ = \lambda p^+ + (1-\lambda)p$.
```

Algorithm 3.4 is identical to Algorithm 3.6 without damping ($\lambda = 1$).

Remark 3.7. The above algorithm is equivalent to a dampened Newton algorithm applied to the equation

$$p_h = S_h^* S_h P_{[a,b]}\left(-\frac{1}{\alpha}p_h\right) - S_h^* z\,, \qquad u := P_{[a,b]}\left(-\frac{1}{\alpha}p_h\right)\,.$$

3.5 Global Convergence of the undamped Newton Algorithm

Since orthogonal projections are non-expansive, it is not difficult to see that the fixed-point equation for problem $(\mathbb{P}_h)$

$$u_h = P_{[a,b]}\left(-\frac{1}{\alpha}S_h^*(S_h u_h - z)\right)$$

can be solved by simple fixed-point iteration that converges globally for $\alpha > \|S_h\|^2_{L^2(\Omega),L^2(\Omega)}$, see [5, 6]. A similar global convergence result holds for the undamped Newton Algorithm 3.4

Lemma 3.8. *For sufficiently small $h > 0$, the Newton algorithm 3.4 converges globally if $\alpha > \frac{4}{3}\|S\|^2$.*

Proof. See [13]. ∎

4 Numerical examples

We end this paper by illustrating our theoretical findings by numerical examples. We apply the globalized Algorithm 3.6 with the following Armijo line search strategy. The merit function

$$MF_h(p) = \left\| p - S_h^* S_h P_{[a,b]}\left(-\frac{1}{\alpha}p\right) + S_h^* z \right\|^2_{L^2(\Omega)}\,,$$

is chosen to govern the step size.

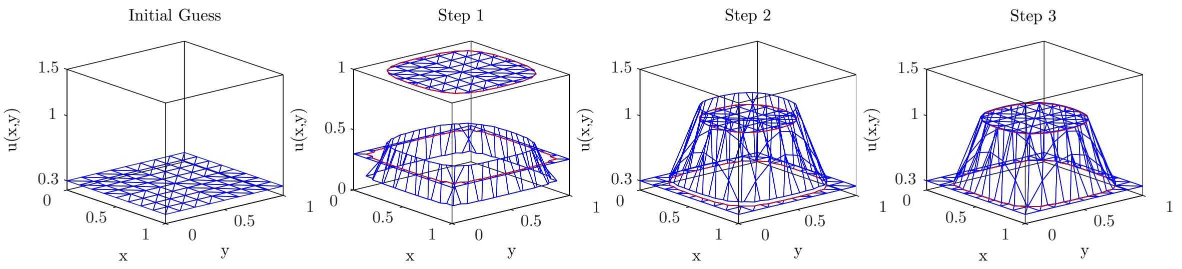

Figure 1: The first four Newton-iterates for Example 4.2 (Dirichlet) with parameter $\alpha = 0.001$

Algorithm 4.1 (Armijo)**.** Start with $\lambda = 1$. If

$$MF_h(p_\lambda^+) \leq MF_h(p) + 0.7\,\underbrace{\langle MF_h'(p), p_\lambda^+ - p\rangle}_{\leq 0}, \tag{4.1}$$

accept p_λ^+. If not, redefine $\lambda := 0.02\,\lambda$ and test (4.1) again.

As stopping criterion we use $\|P_{[a,b]}(-\frac{1}{\alpha}p_\lambda^+) - \bar{u}_h\|_{L^2(\Omega)} < 10^{-11}$ in Algorithm 3.6, using the a posteriori bound for admissible $v \in U_{ad}$

$$\|v - \bar{u}_h\|_{L^2(\Omega)} \leq \frac{1}{\alpha}\|\zeta\|_{L^2(\Omega)}\,, \qquad \zeta(\omega) = \begin{cases} [\alpha v + p_h(v)]_- & \text{if } v(\omega) = a \\ [\alpha v + p_h(v)]_+ & \text{if } v(\omega) = b \\ \alpha v + p_h(v) & \text{if } a < v(\omega) < b \end{cases} \,,$$

presented in [8] and [10]. Clearly, this estimate applies to $v = P_{[a,b]}(-\frac{1}{\alpha}p_\lambda^+)$, whereas in general it does not hold for the iterates v^+ generated by Algorithm 3.4 or 3.6 that need not lie in U_{ad}.
For the first two examples, Algorithm 3.6 reduces to Algorithm 3.4, i.e. the algorithm proceeds in full Newton steps ($\lambda = 1$), thus reflecting the global convergence property from Lemma 3.8. The third example involves a small parameter $\alpha = 10^{-7}$ and the undampened Algorithm 3.4 would not converge in this case.

Example 4.2 (Dirichlet)**.** We consider problem ($\mathbb{P}$) in (1.1) with controls $u \in L^2(\Omega)$ on the unit square $\Omega = (0,1)^2$ with $a \equiv 0.3$ and $b \equiv 1$. Further we set

$$z = -4\pi^2\alpha\sin(\pi x)\sin(\pi y) + (S \circ \imath)r\ , \quad \text{where } r = \min\big(1, \max\big(0.3, 2\sin(\pi x)\sin(\pi y)\big)\big)\,.$$

The choice of parameters implies a unique solution $\bar{u} = r$ to the continuous problem ($\mathbb{P}$).

Throughout this section, solutions to the state equation are approximated by continuous, piecewise linear finite elements on a quasiuniform triangulation $\mathcal{T}_h$ with maximal edge length $h > 0$. The meshes are generated through regular refinement starting from the coarsest mesh.

Problem ($\mathbb{P}_h$) admits a unique solution $\bar{u}_h$. In the setting of the present example we have

$$\|\bar{u}_h - \bar{u}\|_{L^2(\Omega)} = O(h^2)$$

and

$$\|\bar{u}_h - \bar{u}\|_{L^\infty(\Omega)} = O(|\log(h)|^{\frac{1}{2}}h^2)$$

for domains $\Omega \subset \mathbb{R}^2$, see [5]. Both convergence rates are confirmed in Table 1, where the L^2- and the L^∞-errors for Example 4.2 are presented, together with the corresponding experimental orders of convergence

$$EOC_i = \frac{\ln ERR(h_{i-1}) - \ln ERR(h_i)}{\ln(h_{i-1}) - \ln(h_i)}.$$

mesh param. h	ERR	ERR_∞	EOC	EOC_∞	Iterations	$\|\zeta\|/\alpha$
$\sqrt{2}/16$	2.5865e-03	1.2370e-02	1.95	1.79	4	2.16e-15
$\sqrt{2}/32$	6.5043e-04	3.2484e-03	1.99	1.93	4	2.08e-15
$\sqrt{2}/64$	1.6090e-04	8.1167e-04	2.02	2.00	4	2.03e-15
$\sqrt{2}/128$	4.0844e-05	2.1056e-04	1.98	1.95	4	1.99e-15
$\sqrt{2}/256$	1.0025e-05	5.3806e-05	2.03	1.97	4	1.69e-15
$\sqrt{2}/512$	2.5318e-06	1.3486e-05	1.99	2.00	4	1.95e-15

Table 1: L^2- and L^∞-error development for Example 4.2 (Dirichlet)

Lemma 3.8 ensures global convergence of the undamped Algorithm 3.4 only for $\alpha > 1/(3\pi^4) \simeq 0.0034$, but it is still observed for $\alpha = 0.001$.
The algorithm is initialized with $v_0 \equiv 0.3$. The resulting number of Newton steps as well as the value of $\|\zeta\|/\alpha$ for the computed solution are also given in Table 1.
Figure 1 shows the Newton iterates. Active and inactive sets are very well distinguishable, and jumps along their frontiers can be observed.

Example 4.3 (Neumann)**.** We next consider an elliptic problem with Neumann boundary conditions

$$\begin{aligned} -\Delta y + y &= u \qquad \text{in } \Omega\ , \\ \partial_n y &= 0 \qquad \text{on } \partial\Omega\ , \end{aligned}$$

on $\Omega = (0,1)^2$, with a similar discrete setting as in the previous example. It then is clear, how $(\mathbb{P})$ and $(\mathbb{P}_h)$ have to be understood. We set $\alpha = 1$ and choose

$$z = -2(2\pi^2+1)\alpha\cos(\pi x)\cos(\pi y) + (S\circ\imath)r\ , \ \text{ with } r = \min\big(1, \max\big(-1, 2\cos(\pi x)\cos(\pi y)\big)\big)$$

and bounds $a \equiv -1$ and $b \equiv 1$. The optimal control to the continuous problem is $\bar{u} = r$.

For $\alpha = 1$ the undamped iteration still converges globally, although the solution operator has norm $\|S\| = 1$ as an endomorphism in $L^2(\Omega)$. The predicted convergence properties and the stopping criterion are the same as above; Algorithm 3.6 is initialized with $v_0 \equiv -1$. The first four steps of the iteration are displayed in Figure 2 and the behaviour of the approximation error between the exact and the semidiscrete solution, as well as the number of iterations and the final value of $\|\zeta\|/\alpha$, is shown in Table 2.

mesh param. h	ERR	ERR_∞	EOC	EOC_∞	Iterations	$\|\zeta\|/\alpha$
$\sqrt{2}/16$	3.9866e-03	1.1218e-02	1.94	1.74	3	1.81e-12
$\sqrt{2}/32$	1.0025e-03	3.2332e-03	1.99	1.79	3	2.31e-12
$\sqrt{2}/64$	2.5188e-04	8.4398e-04	1.99	1.94	3	9.74e-13
$\sqrt{2}/128$	6.2936e-05	2.1856e-04	2.00	1.95	3	9.37e-13
$\sqrt{2}/256$	1.5740e-05	5.5223e-05	2.00	1.99	3	8.91e-13
$\sqrt{2}/512$	3.9346e-6	1.3928e-05	2.00	2.00	3	8.86e-13

Table 2: Development of the error in Example 4.3 (Neumann)

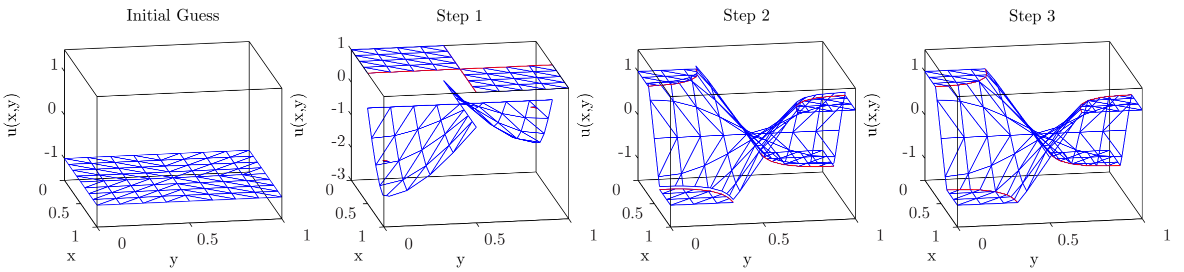

Figure 2: The first steps of the Newton-algorithm for Example 4.3 (Neumann) with $\alpha = 1$.

Algorithm 3.4 has also been implemented successfully for parabolic control problems with a discontinuous Galerkin approximation of the states, as well as for elliptic control problems with Lavrentiev-regularized state constraints, see [13].

To demonstrate Algorithm 3.6 with damping we again consider Example 4.2, this time with $\alpha = 10^{-7}$, again using the same stopping criterion as in the previous examples.

Table 3 shows errors, the number of iterations and the maximal number of Armijo damping steps performed for different mesh parameters h with given smoothing parameter $\alpha = 10^{-7}$. To compare the number of iterations we choose a common initial guess $u_0 \equiv 1$. The number of iterations appears to be independent of h, while the amount of damping necessary seems to decrease with falling h.

Acknowledgements

The first author gratefully acknowledges the support of the DFG Priority Program 1253 entitled Optimization With Partial Differential Equations. We also thank Andreas Günther for some fruitful discussions.

References

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

mesh param. h	ERR	ERR_∞	EOC	EOC_∞	Iterations	max#Armijo
$\sqrt{2}/2$	1.1230e-01	3.0654e-01	-	-	11	14
$\sqrt{2}/4$	3.8398e-02	1.4857e-01	1.55	1.04	22	11
$\sqrt{2}/8$	9.8000e-03	4.4963e-02	1.97	1.72	16	10
$\sqrt{2}/16$	1.7134e-03	1.2316e-02	2.52	1.87	18	10
$\sqrt{2}/32$	4.0973e-04	2.8473e-03	2.06	2.11	30	9
$\sqrt{2}/64$	8.2719e-05	6.2580e-04	2.31	2.19	15	7
$\sqrt{2}/128$	2.0605e-05	1.4410e-04	2.01	2.12	15	6
$\sqrt{2}/256$	4.7280e-06	4.6075e-05	2.12	1.65	15	6
$\sqrt{2}/512$	1.1720e-06	1.0363e-05	2.01	2.15	15	6

Table 3: Development of the error in Example 4.2 (Dirichlet) for $\alpha = 10^{-7}$.

[2] K. Deckelnick and M. Hinze.: *A finite element approximation to elliptic control problems in the presence of control and state constraints*, Preprint HBAM2007-01, Hamburger Beiträge zur Angewandten Mathematik, Universität Hamburg (2007).

[3] M. Hintermüller, K. Ito and K. Kunisch *The primal-dual active set method as a semismooth Newton method.* SIAM J. Control and Optim. **13** (3), 865–888 (2003).

[4] M. Hintermüller and M. Ulbrich.: *A mesh-independence result for semismooth Newton methods* Mathematical Programming **101**, 151–184 (2004).

[5] M. Hinze: *A generalized discretization concept for optimal control problems with control constraints*, Preprint MATH-NM-02-2003, Institut für Numerische Mathematik, Technische Universität Dresden (2003).

[6] M. Hinze. *A variational discretization concept in control constrained optimization: the linear-quadratic case*, J. Computational Optimization and Applications **30** (1), 45–61 (2005).

[7] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE constraints.* Mathematical Modelling: Theory and Applications 23. Dordrecht: Springer (2009).

[8] K. Krumbiegel and A. Rösch. *A new stopping criterion for iterative solvers for control constrained optimal control problems*, Archives of Control Sciences **18** (1), 17–42 (2008).

[9] C. Meyer and A. Rösch. *Superconvergence properties of optimal control problems.* SIAM J. Control Optim. **43** (3), 970–985 (2004).

[10] F. Tröltzsch and S. Volkwein: *POD a-posteriori error estimates for linear-quadratic optimal control problems*, Computational Optimization and Applications, Online First (2008).

[11] M. Ulbrich: *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim. **13**, 805–841 (2003).

[12] M. Ulbrich:*A new mesh-Independence result for semismooth Newton methods*, Oberwolfach Report **4/2009**, 78–81 (2009).

[13] M. Vierling: *Ein semiglattes Newtonverfahren für semidiskretisierte steuerungsbeschränkte Optimalsteuerungsprobleme*, Diplomarbeit, Universität Hamburg (2007).