A semismooth Newton algorithm for semidiscrete Optimal Control Problems with control constraints

Morten Vierling
joint work with Michael Hinze

UH
Universität Hamburg

Department Mathematik
morten.vierling@math.uni-hamburg.de

March 29, 2008
Overview

1. Introduction

2. Newton algorithm and semismoothness

3. Implementation

4. Numerical examples
Overview

1. Introduction

2. Newton algorithm and semismoothness

3. Implementation

4. Numerical examples
Overview

1. Introduction
2. Newton algorithm and semismoothness
3. Implementation
4. Numerical examples
Overview

1. Introduction
2. Newton algorithm and semismoothness
3. Implementation
4. Numerical examples
Aim:
Algorithm to approximate the solution of an optimal control problem.

- linear-quadratic
- tracking-type cost functional
- box constraints to the control

The Problem
\[
\min_{u, y \in L^2(Q)} \frac{1}{2} \| y - z \|_{L^2(Q)}^2 + \frac{\alpha}{2} \| u \|_{L^2(Q)}^2 \\
u. d. N.
\]
\[
Su = y
\]
\[
a \leq u \leq b \quad \text{a.e. in } Q
\]
**Approach**

1. Chose an appropriate discretization of the state equation ⇒ $S_d$.
3. Apply a Newton algorithm to the first order optimality condition.
4. Use an iterative solver for the occurring linear equations (for example a CG alg. for the linear operators involved are self adjoint)

---

**The semidiscrete Problem**

$$\min_{u,y \in L^2(Q)} \frac{1}{2} \| y - z \|_{L^2(Q)}^2 + \frac{\alpha}{2} \| u \|_{L^2(Q)}^2$$

u.d.N.

$S_d u = y$

$a_d \leq u \leq b_d$ a.e. in $Q$
Approach

1. Chose an appropriate discretization of the state equation $\Rightarrow S_d$.
3. Apply a Newton algorithm to the first order optimality condition.
4. Use an iterative solver for the occurring linear equations (for example a CG alg. for the linear operators involved are self adjoint)

The semidiscrete Problem

$$\min_{u, y \in L^2(Q)} \frac{1}{2} \|y - z\|_{L^2(Q)}^2 + \frac{\alpha}{2} \|u\|_{L^2(Q)}^2$$

u.d.N.

$S_d u = y$

$a_d \leq u \leq b_d$ a.e. in $Q$
1 Introduction

2 Newton algorithm and semismoothness

3 Implementation

4 numerical examples
Why using the semidiscretization? I

The strength of the semidiscrete approach shows in the following theorem.

**Theorem ($L^2$-convergence)**

Let $\bar{u}$ and $\bar{u}_d$ be the solutions of the continuous and the discrete problem. Then there holds

$$\alpha \| \bar{u}_d - \bar{u} \|_{L^2(Q)} \leq \| S - S_d \| \left( \| (S\bar{u} - z) \|_{L^2(Q)} + \| S_d \| \| \bar{u} \|_{L^2(Q)} \right) + \alpha + \| S_d \|^2 \left( \| a - a_d \|_{L^2(Q)} + \| b - b_d \|_{L^2(Q)} \right) + \alpha + \| S_d \|^2 \left( \| a - a_d \|_{L^2(Q)} + \| b - b_d \|_{L^2(Q)} \right)$$

The numerical results confirm this result.

*Remark:* The assertions may be weakened a lot. Normwise convergence of the operators $S$ and $S_d$ which usually requires (at least) a smooth boundary is not necessary.

In the case of a parabolic Discontinuous Galerkin discretization $S_d$ (piecewise constant in time, piecewise linear and globally continuous in space) one has the following convergence result.
Why using the semidiscretization? II

Theorem ($L^\infty$-convergence (parabolical case, Dirichlet boundary cond.))

Assume $\Omega \subset \mathbb{R}^2$ to be a convex polyhedral domain in $\mathbb{R}^2$ and let

$\partial_t (S^* S \tilde{u}), \partial_t (S^* z) \in L^\infty ([0, T], L^\infty (\Omega))$ hold as well as

$S^* S \tilde{u}, S^* z \in L^\infty ([0, T], H^2 (\Omega))$. Then there holds for the solutions $\tilde{u}_d$ and $\tilde{u}$

$$
\| \tilde{u}_d - \tilde{u} \|_\infty \leq C (| \log k | + 1)^{\frac{1}{2}} \left( k (\| \partial_t (S^* z) \|_\infty + \| \partial_t (S^* S \tilde{u}) \|_\infty) \right) \ldots
$$

$$
+ h^2 (\| S^* z \|_{L^\infty ([0, T], H^2 (\Omega))} + \| S^* S \tilde{u} \|_{L^\infty ([0, T], H^2 (\Omega))}) \right) \ldots
$$

$$
+ C (| \log h | + 1)^{\frac{1}{2}} (h^2 + k) \| \tilde{u} \|_{L^2 ([0, T], L^2 (\Omega))} .
$$
Necessary conditions

The optimum $\bar{u}_d$ of the semidiscrete problem fulfills the following necessary condition.

$$\langle S_d^* S_d \bar{u}_d - z + \alpha \bar{u}_d, u - \bar{u}_d \rangle_{L^2(Q)} \geq 0, \quad \forall u : a \leq u \leq b.$$ 

Since the feasible set is convex and the $L^2$-projection onto it equals the pointwise projection this is equivalent to

$$\bar{u}_d = P_{[a,b]} \left( \bar{u}_d - c (\alpha \bar{u}_d + S_d^* \left( S_d \bar{u}_d - z \right)) \right)$$

for any $c > 0$. The choice $c = \frac{1}{\alpha}$ yields some insight into the structure of $\bar{u}_d$.

$$\bar{u}_d = P_{[a,b]} \left( - \frac{1}{\alpha} S_d^* \left( S_d \bar{u}_d - z \right) \right)$$
Necessary conditions

The optimum \( \bar{u}_d \) of the semidiscrete problem fulfills the following necessary condition.

\[
\langle S_d^* S_d \bar{u}_d - z + \alpha \bar{u}_d, u - \bar{u}_d \rangle_{L^2(Q)} \geq 0, \quad \forall u : a \leq u \leq b.
\]

Since the feasible set is convex and the \( L^2 \)-projection onto it equals the pointwise projection this is equivalent to

\[
\bar{u}_d = P_{[a,b]} (\bar{u}_d - c (\alpha \bar{u}_d + S_d^* (S_d \bar{u}_d - z)))
\]

for any \( c > 0 \). The choice \( c = \frac{1}{\alpha} \) yields some insight into the structure of \( \bar{u}_d \).

\[
\bar{u}_d = P_{[a,b]} \left( - \frac{1}{\alpha} S_d^* (S_d \bar{u}_d - z) \right)
\]

structure of \( \bar{u}_d \).
Solving the optimality system

**Fixed point iteration**

For $\alpha > \|S_d\|$ the solution $\bar{u}_d$ can be computed by simple fixed point iteration.

**Newton algorithm**

Given that

$$G_d(u) = u - P_{[a_d,b_d]} \left( - \frac{1}{\alpha} S_d^*(S_d u - z) \right)$$

is semismooth, a Newton algorithm will converge at least locally superlinearly.
Semismoothness I

- We assume a finite element space of piecewise linear globally continuous functions over a quasiuniform Triangulation $T_h$ with maximal diameter $h$. (in space)
- In the case of a linear parabolic solution operator $S$ over a parabolic cylinder $Q_T = \Omega \times [0, T]$ the Ansatz functions are piecewise constant in time.

**Corollary from a theorem of M. Ulbrich**

The operator $G_d : L^2(Q) \to L^2(Q)$

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

is semismooth of order $\delta$ für $0 < \delta < \frac{1}{2}$, provided that $S_d^* (S_d u - z)$ maps continuously into $L^p(Q)$ for $1 < p < \infty$. 
Semismoothness II

- An elliptic solution operator $S : L^2(\Omega) \rightarrow L^2(\Omega)$ will map continuously into $H^2(\Omega)$, given smooth enough data. Given a little more smoothness $S^*(Su - z)$ then maps into $H^4(\Omega)$. For $\Omega \subset \mathbb{R}^n$, $n \leq 3$ the space $H^2(\Omega)$ is continuously embedded in $C^0(\Omega)$.

- An elliptic solution operator $S^*_d : L^2(\Omega) \rightarrow L^2(\Omega)$ will at least map $L^2$ continuously into $H^1(\Omega)$. The embedding $H^1(\Omega) \hookrightarrow L^p(\Omega)$ holds for $1 < p < \infty$ if $n = 2$ and for $1 < p \leq 6$. For $n = 3$ we obtain semismoothness of order $0 < \delta < \frac{1}{3}$.

- A parabolic solution operator $S^*_d : L^2(Q_T) \rightarrow L^2(Q_T)$ maps $L^2(Q_T)$ continuously into $L^\infty([0, T], H^1(\Omega))$ which for $n = 2$ lies in any $L^\infty([0, T], L^p(\Omega))$ for $1 < p < \infty$. For $n = 3$ at least $L^\infty([0, T], H^1(\Omega)) \hookrightarrow L^\infty([0, T], L^6(\Omega))$ and we obtain semismoothness of order $0 < \delta < \frac{1}{3}$. 


The Newton step for $G_d(u) = 0$ reads

$$\left( I + \frac{g_i}{\alpha} S_d^* S_d \right) (s_i) = -u_i + P_{[a,b]} \left( -\frac{1}{\alpha} (S_d^* (S_d u_i - z)) \right)$$

where $g_i$ is the characteristic function of the inactive set in the $i$-th iteration.

- $(I + \frac{g_i}{\alpha} S_d^* S_d)$ invertible?
- Practicability?
- Solver for the linear equation?
Practicability

Dampened Newton Algorithm: \( u_{i+1} = u_i + \beta_i s_i \).

**Problem**

\[
\begin{align*}
u_{i+1} &= u_i + \beta_i s_i \\
&= (1 - \beta_i u_i) + \beta_i P_{[a,b]} \left( -\frac{1}{\alpha} \left( S_d^*(S_d u_i - z) \right) \right) - \beta_i \frac{g_i}{\alpha} S_d^* S_d(s_i)
\end{align*}
\]

\( \Rightarrow \) Not implementable for \( \beta \neq 1 \). BUT: The next iterate \( u_{i+1} \) depends only on \( S_d^* S_d u_i \).

**Way out**

Projection of \( u_{i+1} \) to prevent accumulation of jumps. For example by \( L^2 \)-projecting \( u_{i+1} \) onto the current FE space that contains \( P_{[a,b]} \left( -\frac{1}{\alpha} \left( S_d^*(S_d u_{i+1} - z) \right) \right) \) as well as the base FE space. Or by applying \( S_d^* S_d \) to the above equation.
Invertability

The undampened Newton step can be written as

\[
(I + \frac{g_i}{\alpha} S_d^* S_d) u_{i+1} = P_{[a,b]} \left( - \frac{1}{\alpha} (S_d^* (S_d u_i - z)) \right) + \frac{g_i}{\alpha} S_d^* S_d u_i
\]

- Invertability of \((I + \frac{g_i}{\alpha} S_d^* S_d)\) not obvious yet.
- Not self-adjoint.

Way out

\[
\left( I + \frac{g_i}{\alpha} S_d^* S_d \right) g_i u_{i+1} = P_{[a,b]} \left( - \frac{1}{\alpha} (S_d^* (S_d u_i - z)) \right) + \frac{g_i}{\alpha} S_d^* S_d u_i \ldots
\]

\[
- \left( I + \frac{g_i}{\alpha} S_d^* S_d \right) (1 - g_i) u_{i+1}
\]
Self-adjointness

- On the active set $u_{i+1}$ is already known.
- On the inactive set there holds

\[ (I + \frac{g_i}{\alpha} S_d^* S_d) g_i u_{i+1} = \frac{g_i}{\alpha} S_d^* z - \frac{g_i}{\alpha} S_d^* S_d (1 - g_i) u_{i+1} \]

A CG routine can be applied!
We also get an estimate for the inverse

\[ \left\| (I + \frac{1}{\alpha} g S_d^* S_d)^{-1} \right\| \leq 1 + \frac{1}{\alpha} \left\| S_d \right\|^2 \]

which together with the semismoothness is sufficient for local superlinear convergence.
Convergence of the Newton algorithm

Global convergence

For $\alpha > \frac{4}{3} \|S_d\|^2$ the iteration converges independently of the initial guess $u_0$.

So the global convergence properties of the algorithm are (nearly) as good as those of the fixed point iteration.

Globalization

If inexact Amijo linesearch works for the canonicalMerit functional

$$\left\| u - P_{[a,b]} \left( -\frac{1}{\alpha} S_d^* \left( S_d u - z \right) \right) \right\|^2$$

it works also for the projection strategies proposed for damping.
1 Introduction

2 Newton algorithm and semismoothness

3 Implementation

4 numerical examples
Consider the problem

\[
\min_{u, y \in L^2(\Omega)} \| y - y_d \|_{L^2(\Omega)}^2 + \alpha \| u \|_{L^2(\Omega)}^2
\]

under the constraints

\[-\Delta y = u - r \quad \text{in } \Omega = [0, 1]^2, \quad y = 0 \quad \text{auf } \partial \Omega,
\]

\[a \leq u \leq b \quad \text{a.e.}
\]

with \( \alpha = 0.001 \) and parameters

\[r = P_{[a, b]}(2 \sin(\pi x) \sin(\pi y)) ,
\]

\[y_d = -4\pi^2 \alpha \sin(\pi x) \sin(\pi y)
\]

and bounds \( a = 0.3, b = 1 \).

The problem admits a unique solution \( \bar{u} = r \).
**Figure:** The first four Newton iterates for a random initial guess scaled by $10^6$.

<table>
<thead>
<tr>
<th>mesh param. $h$</th>
<th>$ERR$</th>
<th>$ERR_\infty$</th>
<th>$EOC$</th>
<th>$EOC_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>4.0728e-003</td>
<td>1.8773e-002</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.05</td>
<td>9.2078e-004</td>
<td>2.4076e-003</td>
<td>2.1451</td>
<td>2.9629</td>
</tr>
<tr>
<td>0.025</td>
<td>2.2090e-004</td>
<td>8.7739e-004</td>
<td>2.0594</td>
<td>1.4563</td>
</tr>
<tr>
<td>0.0125</td>
<td>5.6551e-005</td>
<td>2.7165e-004</td>
<td>1.9658</td>
<td>1.6915</td>
</tr>
<tr>
<td>0.00625</td>
<td>1.3913e-005</td>
<td>9.4819e-005</td>
<td>2.0232</td>
<td>1.5185</td>
</tr>
</tbody>
</table>

**Table:** $L^2$- and $L^\infty$-errors for different meshes and estimated order of convergence.
global convergence

On smooth domains $\Omega$ the mapping $u \mapsto y(u) \in L^2(\Omega)$ can be shown to be a continuous (in fact compact) operator with norm $2\pi^2$. The fixed point iteration for the equation

$$u = P_{[a,b]} \left( -\frac{1}{\alpha} S^*_h (S_h u - y_d) \right)$$

converges for $\|S_h\|^2/\alpha < 1$. Yet the operator norm $\|S_h\|$ converges to $2\pi^2$, so the iteration is convergent for $\alpha \gtrsim (4\pi^4)^{-1} \simeq 390^{-1}$. For the Newton algorithm a similar estimate holds.

**global convergence**

The undampened Newton algorithm converges globally if $\alpha > \frac{4}{3} \|S_h\|^2$. 
Figures: Undampened Newton steps for $\alpha = 10^{-5}$. The iterates bounce between upper and lower bound.

Figure: $\alpha = 10^{-5}$. The inexactly Amijo-dampened Newton algorithm terminates after 13 steps, when the jump along the border between active and inactive set is smaller than $10^{-12}$. 
better damping strategy

Instead of solving the fixed point equation

\[ G_d(u) = u - P_{[a_d, b_d]} \left( -\frac{1}{\alpha} S_d^* (S_d u - z) \right) = 0 \]

we may switch to the equivalent problem

\[ p - S_d^* S_d P_{[a_d, b_d]} \left( -\frac{1}{\alpha} p \right) + S_d^* z = 0 \]

and apply a semismooth Newton algorithm to it. (This is a finite dimensional problem!)

The construction of the algorithm remains nearly the same (Invertability, Selfadjointness...)

<table>
<thead>
<tr>
<th>mesh param. ( h )</th>
<th>( ERR )</th>
<th>( ERR_\infty )</th>
<th>( EOC )</th>
<th>( EOC_\infty )</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>3.4920e-003</td>
<td>2.3977e-002</td>
<td>-</td>
<td>-</td>
<td>28</td>
</tr>
<tr>
<td>0.05</td>
<td>6.8772e-004</td>
<td>3.8290e-003</td>
<td>2.3442</td>
<td>2.6466</td>
<td>18</td>
</tr>
<tr>
<td>0.025</td>
<td>1.6901e-004</td>
<td>1.3261e-003</td>
<td>2.0247</td>
<td>1.5298</td>
<td>19</td>
</tr>
<tr>
<td>0.0125</td>
<td>4.0953e-005</td>
<td>3.2358e-004</td>
<td>2.0451</td>
<td>2.0350</td>
<td>18</td>
</tr>
<tr>
<td>0.00625</td>
<td>1.0065e-005</td>
<td>9.3818e-005</td>
<td>2.0247</td>
<td>1.7862</td>
<td>18</td>
</tr>
</tbody>
</table>

Table: Development of the error and number of Newton iterations for \( \alpha = 10^{-7} \).
Consider the problem

$$\min_{u, y \in L^2(Q_T)} \|y - y_d\|^2 + \alpha \|u\|^2$$

$$\partial_t y - \Delta y = u - r \text{ in } Q_T = [0, T] \times \Omega,$$

$$\partial_n y = 0 \text{ on } [0, T] \times \partial \Omega \text{ and } y|_{t=0} = 0$$

$$a \leq u \leq b \quad \text{a.e. in } Q_T$$

with $\alpha = 1$ and parameters $r, y_d \in L^2(Q_T)$ with

$$r = \max \left( -0.5, \min \left( 0.5, \sin(\pi t) \cos(\pi x) \cos(\pi y) \right) \right),$$

$$y_d = \pi \alpha \cos(\pi x) \cos(\pi y) \left( 2\pi \cos(\pi t) - \sin(\pi t) \right)$$

and bounds $a = -0.5, b = 0.5$.

The choice of parameters leads to a solution $\bar{u} = r$. 
### parabolic example

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0.1, 0.01)$</td>
<td>$4.0723e-003$</td>
<td>$2.9171e-002$</td>
<td>-</td>
</tr>
<tr>
<td>$(\frac{0.1}{\sqrt{2}}, \frac{0.1}{2})$</td>
<td>$1.9935e-003$</td>
<td>$1.5093e-002$</td>
<td>$2.0611$</td>
</tr>
<tr>
<td>$(0.05, \frac{0.1}{4})$</td>
<td>$9.9564e-004$</td>
<td>$7.6957e-003$</td>
<td>$2.0032$</td>
</tr>
<tr>
<td>$(\frac{0.05}{\sqrt{2}}, \frac{0.1}{8})$</td>
<td>$5.1607e-004$</td>
<td>$3.8858e-003$</td>
<td>$1.8961$</td>
</tr>
<tr>
<td>$(0.025, \frac{0.1}{16})$</td>
<td>$2.5036e-004$</td>
<td>$1.9531e-003$</td>
<td>$2.0871$</td>
</tr>
<tr>
<td>$(\frac{0.025}{\sqrt{2}}, \frac{0.1}{32})$</td>
<td>$1.2566e-004$</td>
<td>$9.7917e-004$</td>
<td>$1.9889$</td>
</tr>
</tbody>
</table>
Result

- Fully implementable algorithm.
- The Newton alg. converges locally superlinearly with exponent 1.5.
- Global convergence for $\alpha$ sufficiently large.
- Globalization by inexact linesearch for smaller $\alpha$.

Outlook

- Mesh independence? [2]
- Superconvergence to a post processed fully discretized approach? [1]
- Application to a moving surface problem? [3]
- Application as a post processing step?
- Nonlinear PDEs?
Result

- Fully implementable algorithm.
- The Newton alg. converges locally superlinearly with exponent 1.5.
- Global convergence for \( \alpha \) sufficiently large.
- Globalization by inexact linesearch for smaller \( \alpha \).

Outlook

- Mesh independence? [2]
- Superconvergence to a post processed fully discretized approach? [1]
- Application to a moving surface problem? [3]
- Application as a post processing step?
- Nonlinear PDEs?
THANK YOU.
