Lukáš Pospíšil
An optimal algorithm with Barzilai-Borwein steplength and superrelaxation for QPQC problem


Persistent URL: http://dml.cz/dmlcz/702721

Terms of use:
© Institute of Mathematics AS CR, 2013

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these Terms of use.
AN OPTIMAL ALGORITHM WITH BARZILAI-BORWEIN STEPLENGTH AND SUPERRELAXATION FOR QPQC PROBLEM

Lukáš Pospíšil
FEECS VSB-Technical University of Ostrava
17. listopadu 15, CZ-70833 Ostrava, Czech Republic
lukas.pospisil@vsb.cz

Abstract

We propose a modification of MPGP algorithm for solving minimizing problem of strictly convex quadratic function subject to separable spherical constraints. This active set based algorithm explores the faces by the conjugate gradients and changes the active sets and active variables by the gradient projection with the Barzilai-Borwein steplength. We show how to use the algorithm for the solution of separable and equality constraints. The power of our modification is demonstrated on the solution of a contact problem with Tresca friction.

1. Motivation

Let us consider simple contact problem with given friction. The block of homogeneous material has prescribed zero displacements on boundary $\Gamma_D$ and imposed traction $\mathbf{F}$ on $\Gamma_F$. The part $\Gamma_C$ denotes the part of boundary that may get into contact with rigid obstacle. The block is attracted to obstacle by gravity force $\mathbf{F}_G$.

Figure 1: Contact problem with rigid obstacle and given friction.
We solve discretized form of the problem using FEM. This technique leads to optimizing problem (see [3])

\[
\bar{u} := \min_{u \in \Omega} (f(u) + j_h(u)), \quad f(u) := \frac{1}{2} u^T K u - f^T u, \quad j_h(u) := \sum_{i=1}^{m_c} \psi_i \|T_i u\|, \quad (1)
\]

where \(N \in \mathbb{N}\) is number of used nodes and \(n = 3N\) is number of variables, \(u \in \mathbb{R}^n\) is a vector of unknown displacements, \(\Omega := \{u \in \mathbb{R}^n : u_z \geq -c\}\) is set of feasible \(u\), \(c \in \mathbb{R}_0^+\) is a distance between body and rigid obstacle, \(f : \mathbb{R}^n \rightarrow \mathbb{R}\) denotes function of total potential energy, \(K \in \mathbb{R}^{n,n}\) is a symmetric-positive definite stiffness matrix, \(f \in \mathbb{R}^n\) is vector of internal forces resulting from the stresses imposed on the structure during a displacement, \(j_h : \mathbb{R}^n \rightarrow \mathbb{R}\) is numerical integration of functional describing the friction forces in the weak formulation of the problem, \(T_i \in \mathbb{R}^{2,n}\) are formed by appropriately placed multiples of the unit tangential vectors in such way that the jump of tangential displacement due to displacement \(u\) is given by \(T_i u\), \(\psi_i \in \mathbb{R}\) is slip bound associated with \(T_i\).

At first denote \(m_c \leq N\) as number of FEM nodes in \(\Gamma_C\).

Our problem has simple geometry, so we can simply choose \(n := [0, 0, -1]\) as normal vector and \(t_1 := [1, 0, 0], t_2 := [0, 1, 0]\) as tangential vectors for every FEM node in \(\Gamma_C\).

![Figure 2: Normal and tangential vectors on \(\Gamma_C\).](image)

So for every contact node (\(i\)-th node from \(\Gamma_C\)) is \(T_i \in \mathbb{R}^{2,n}\) given by sparse matrix with 1 in first row on appropriate \(x\)-coordinate of \(i\)-th node and in second row on appropriate \(y\)-coordinate of \(i\)-th node. Then we assume that \(T := [T_1^T, \ldots, T_{m_c}^T]^T\) is the full rank matrix.

In our problem with Dirichlet conditions, \(f\) is strictly convex quadratic function (i.e. quadratic function with symmetric positive-definite matrix \(K\)), so in next reductions, we can use standard inversion \(K^{-1}\).

We can express the non-differentiable term \(j_h\) in (1) by (see [7])

\[
j_h(u) = \sum_{i=1}^{m_c} \max_{\|\tau_i\| \leq \psi_i} \tau_i^T T_i u, \quad (2)
\]

where \(\tau_i \in \mathbb{R}^2\) are regulation variables.
2. Saddle point problem equivalency and dual formulation

At first, we denote function and vector
\[
\tilde{L}(u, \tau) := f(u) + \tau^T T u, \quad \tau := [\tau_1^T, \ldots, \tau_{mc}^T]^T. \tag{3}
\]

Then the conditions \(|\tau_i| \leq \psi_i\) can be written in form
\[
\sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, \quad i = 1, \ldots, mc, \tag{4}
\]
where \(\tau_j\) is \(j\)-th component of \(\tau\).

Now we can simplify the notation, we denote set of feasible \(\tau\) as
\[
\Lambda_\tau := \left\{\sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \ldots, mc\right\}. \tag{5}
\]

After substituting (2) into (1) and using (3),(4) we get
\[
\min_{u \in \Omega} (f(u) + j_h(u)) = \min_{u \in \Omega} \left( f(u) + \sum_{i=1}^{mc} \max_{\|\tau_i\| \leq \psi_i} \tau_i^T T_i u \right) = \min_{u \in \Omega} \sup_{\tau \in \Lambda_\tau} \tilde{L}(u, \tau). \tag{6}
\]

If we consider \(\tilde{L}(u, \tau)\) as Lagrange function and \(\tau\) as vector of Lagrange multipliers (in notation (3)), we can use the classical duality theorem (see [4]) to reformulate problem (6) and get
\[
\min_{u \in \Omega} \sup_{\tau \in \Lambda_\tau} \tilde{L}(u, \tau) = \max_{\tau \in \Lambda_\tau} \min_{u \in \Omega} \tilde{L}(u, \tau). \tag{7}
\]

Now we can include condition \(u \in \Omega\) by creating new Lagrange multipliers.
\[
\max_{\tau \in \Lambda_\tau, u \in \Omega_C} \min_{u \in \Omega_C} \tilde{L}(u, \tau) = \max_{\tau \in \Lambda_\tau, \lambda \geq 0} \min_{u \in \mathbb{R}^n} \left( \tilde{L}(u, \tau) + \lambda^T (Bu - c) \right), \tag{8}
\]
where matrix \(B \in \mathbb{R}^{mc \times n}\) and vector \(c \in \mathbb{R}^{mc}\) are constructed in such way, that
\[
\{u \in \mathbb{R}^n : Bu \leq c\} = \Omega.
\]

Due to geometry in our problem we can construct \(B\) very simply. \(B\) is a sparse matrix with \(-1\) in every \(i\)-th row (which is corresponding to \(i\)-th node in \(\Gamma_C\)) on appropriate \(z\)-coordinate of \(i\)-th node (see former choice of normal vectors for nodes in \(\Gamma_C\)).

So problem (1) is equivalent to the saddle point problem
\[
(\tilde{u}, \tilde{\lambda}) := \arg \max_{\lambda \in \Lambda} \min_{u \in \mathbb{R}^n} L(u, \lambda), \tag{9}
\]
where
\[
L(u, \lambda) := f(u) + \lambda^T (\tilde{B}u - \tilde{c}) \tag{10}
\]

157
is Lagrange function, which includes both of friction and non-penetration conditions, and

\[ \lambda := \begin{bmatrix} \tau \\ \lambda_C \end{bmatrix}, \quad \tilde{B} := \begin{bmatrix} T \\ B \end{bmatrix}, \quad \tilde{c} := \begin{bmatrix} o \\ c \end{bmatrix}, \]

\[ \Lambda := \{ \begin{pmatrix} \tau, \lambda_C \end{pmatrix} \in \mathbb{R}^{3mc} : \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \ldots, mc, \lambda_C \geq o \}. \]

Now we are going to solve problem (9) using dual formulation, dual function and KKT conditions (again can be found in [4]). At first we induce first Karush-Kuhn-Tucker condition (the minimizer \( \bar{u} \) of function \( L(u,) \) satisfy state of stationary point - we put part of gradient of \( L \) corresponding to derivation with respect to components of \( u \) equal to zero)

\[ \nabla_u L(u, \lambda) = Ku - f + \tilde{B}^T \lambda = o \quad \Rightarrow \quad \bar{u} = K^{-1} \left( f - \tilde{B}^T \lambda \right) \] (11)

and induct this into Lagrange function (10) and make some simplifications. We get

\[ L(\bar{u}, \lambda) = L(K^{-1} \left( f - \tilde{B}^T \lambda \right), \lambda) = -\frac{1}{2} \lambda^T \tilde{B} K^{-1} T^T \lambda + \lambda^T \tilde{B} K^{-1} f - \frac{1}{2} f^T K^{-1} f. \]

We get function of only one variable \( \lambda \). Our task is to find maximizer (see saddle-point problem (9)), so we can omit the constant term and change signs. Then \( \tilde{\lambda} \) solves minimization problem

\[ \tilde{\lambda} = \arg \min_{x \in \Lambda} \Theta(x), \quad \Theta(x) := \frac{1}{2} x^T A x - x^T b, \] (12)

where we denoted

\[ A := \tilde{B} K^{-1} \tilde{B}^T, \quad b := \tilde{B} K^{-1} f. \]

After solving minimizing problem (12), the corresponding solution \( \bar{u} \) of primary problem (1) can be evaluated using (11).

Obviously \( A \in \mathbb{R}^{3mc,3mc} \) is symmetric positive-definite matrix and problem (12) is the problem of minimizing strictly convex quadratic functions with separable quadratic constraints (QPQC) combined with bound constraints.

3. MPGP and projected Barzilai-Borwein algorithm

Now we are ready to introduce Modified proportioning with gradient projections algorithm (MPGP) (also included in [4, 3]), which convergence for QPQC was analysed in [5]. This active-set based algorithm solves problem on a free set using Conjugate gradient (CG) method (eventually do only halfstep) and finalize optimizing process on active set using gradient projection method with constant step-size.

Our modification lies in replacement of constant step-size in projection step by step-size used in recently developed Spectral Projected Gradient Method (SPG, see [2]). This method is based on projected version of Barzilai-Borwein algorithm.
(see [1]) combined with additional modified GLL line-search (see [6]). This additional line-search does not affect our algorithm, because it usually evokes leaving the border of feasible set, i.e. in our case it evokes extension of free set and restart CG method. So we use only first spectral projected step.

1: Choose \( x_0 \in \Omega, \alpha \in (0, 2\|A\|^{-1}), \delta \in (0, 1/2) \)
2: \( \alpha_{bb} := \alpha \)
3: \( k := 0 \)
4: while \( \|x_k - P(x_k - g_k)\| \geq \epsilon \|b\| \) do
5: \( \text{if } 2 \delta g_k^T g_k^P \leq \|\varphi(x_k)\|^2 \text{ then} \)
6: CG step or CG halfstep.
7: make CG step to solve problem on free set.
8: if this step means leaving \( \Omega \), do only a half-step and restart CG.
9: \( k := k + 1 \)
10: \( \text{else} \)
11: Barzilai-Borwein gradient projection step.
12: \( x_{k+1} := P(x_k - \alpha_{bb} g_k) \)
13: \( s := x_{k+1} - x_k \)
14: \( \alpha_{bb} := s^T s / s^T A s \)
15: restart CG
16: \( k := k + 1 \)
17: \( \text{end if} \)
18: \( \text{end while} \)

In our algorithm we use these notations

\[
g_k := Ax_k - b, \quad \hat{g}_k := \frac{1}{\alpha} (x_k - P(x_k - \alpha g_k)), \quad \hat{g}_k^P := \varphi(x_k) + \beta(x_k),
\]

\( \varphi \) and \( \beta \) are free gradient and chopped gradient defined in [4].

4. Numerical experiments

In our numerical experiment, we choose steel brick \((E = 2.10^5, \mu = 0.35, \rho = 7.85.10^{-2})\) and force \( F = 5.10^3 \).

For generating discretized problem we used MatSol library (see [8]).

We require accuracy \( \epsilon = 10^{-4} \). We make two tests – in first we choose \( \psi = 900 \), in second \( \psi = 15.10^3 \).

For MPGP we used parameters \( \delta := 1/2, \alpha := 1.95/\|A\| \). For SPG were used parameters \( M := 1, \alpha_{\min} := 10^{-6}, \alpha_{\max} := 10^6, \gamma := 10^{-4}, \sigma_1 := 0.1, \sigma_2 := 0.9, \alpha_0 := 1 \).

In Tables 1 and 2, \( N \) is discretization parameter. Every edge of brick was divided into \( N \) intervals, so the number of all FEM nodes in model is given by \((N + 1)^3\).

Because the problem is computed in 3D, the number of primal variables is \( 3(N + 1)^3 \).

The number of FEM nodes in \( \Gamma_C \) is given by the number of nodes on bottom side
\[ \psi = 900 \]

<table>
<thead>
<tr>
<th>N</th>
<th>primal</th>
<th>dual</th>
<th>SPG</th>
<th>MPGP</th>
<th>MPGP-BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>375</td>
<td>75</td>
<td>36</td>
<td>9</td>
<td>44</td>
</tr>
<tr>
<td>6</td>
<td>1029</td>
<td>147</td>
<td>45</td>
<td>20</td>
<td>64</td>
</tr>
<tr>
<td>8</td>
<td>2187</td>
<td>243</td>
<td>27</td>
<td>12</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>3993</td>
<td>363</td>
<td>33</td>
<td>15</td>
<td>47</td>
</tr>
</tbody>
</table>

Table 1: Test with small radius.

\[ \psi = 15000 \]

<table>
<thead>
<tr>
<th>N</th>
<th>primal</th>
<th>dual</th>
<th>SPG</th>
<th>MPGP</th>
<th>MPGP-BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>375</td>
<td>75</td>
<td>1566</td>
<td>977</td>
<td>2542</td>
</tr>
<tr>
<td>6</td>
<td>1029</td>
<td>147</td>
<td>923</td>
<td>553</td>
<td>1475</td>
</tr>
<tr>
<td>8</td>
<td>2187</td>
<td>243</td>
<td>588</td>
<td>366</td>
<td>953</td>
</tr>
<tr>
<td>10</td>
<td>3993</td>
<td>363</td>
<td>1020</td>
<td>547</td>
<td>1566</td>
</tr>
</tbody>
</table>

Table 2: Test with larger radius.

of brick, i.e. \( m_c = (N + 1)^2 \). So the number of all Lagrange multipliers is given by \( 3m_c = 3(N + 1)^2 \). This number is a dimension of dual problem.

For SPG algorithm we counted outer iterations and denoted this number by \( \text{it} \). In the tables, one can find also number of all additional GLL-search iterations and a number of evaluations of cost function denoted by \( f(x) \). For MPGP and MPGP-BB we denoted the number of all iterations by \( \text{it} \) and we counted also each type of iterations.

These tables show typical performance properties of algorithms. If the radius of quadratic constraints is small (see Table 1), the type of the most of the iterations of MPGP and MPGP-BB is projection. Because MPGP-BB in projection uses similar rule for choosing step-size as SPG, the number of iterations of these two algorithms is similar. Choosing the constant step-size in MPGP is not so efficient.

If the radius of quadratic constraints is larger (see Table 2), MPGP and MPGP-BB are able to use more CG-iterations. That is the reason, why it is faster than non-monotone gradient descent method SPG.

5. Conclusions

Our numerical experiments predicate better performance of modified MPGP with BB step-size then original constant step-size for solving QPQC problems. But proof of convergence need be established, because the proof of convergence of original SPG in [2] is based on Armijo condition in GLL in additional line-search, but in our modification we did not use it.
Acknowledgements

This work was supported by grant GD103/09/H078 of Grant Agency of the Czech Republic. The author would like to express his thanks to Prof. RNDr. Zdeněk Dostál, DSc., Ing. Alexandros Markopoulos, Ph.D. and Ing. Tomáš Brzobohatý, Ph.D.

References


