Lukáš Pospíšil

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

In: Jan Chleboun and Karel Segeth and Jakub Šístek and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 3-8, 2012. Institute of Mathematics AS CR, Prague, 2013. pp. 155–161.

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

Programs and Algorithms of Numerical Matematics 16
J. Chleboun, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2013

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 Γ_D and imposed traction $\boldsymbol{F}$ on Γ_F. The part Γ_C denotes the part of boundary that may get into contact with rigid obstacle. The block is attracted to obstacle by gravity force $\boldsymbol{F}_G$.

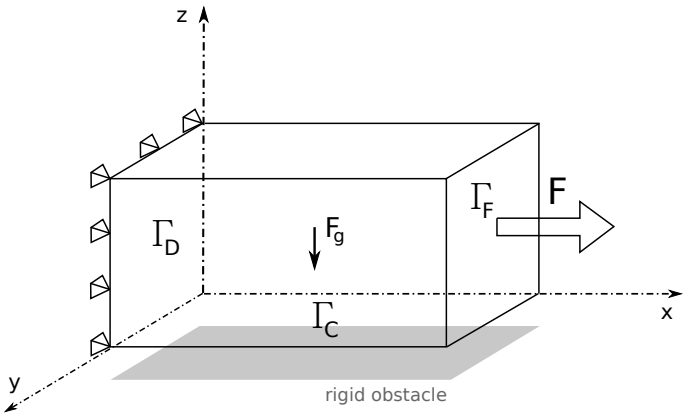

Figure 1: Contact problem with rigid obstacle and given friction.

We solve discretized form of the problem using FEM. This technique leads to optimizing problem (see [3])

$$\bar{u} := \min_{\boldsymbol{u}\in\Omega} \left(f(\boldsymbol{u}) + j_h(\boldsymbol{u})\right), \quad f(\boldsymbol{u}) := \frac{1}{2}\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}^T\boldsymbol{u}, \quad j_h(\boldsymbol{u}) := \sum_{i=1}^{m_c} \psi_i \|\boldsymbol{T}_i\boldsymbol{u}\|, \quad (1)$$

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

At first denote $m_c \leq N$ as number of FEM nodes in Γ_C.
Our problem has simple geometry, so we can simply choose $\boldsymbol{n} := [0, 0, -1]$ as normal vector and $\boldsymbol{t}_1 := [1, 0, 0], \boldsymbol{t}_2 := [0, 1, 0]$ as tangential vectors for every FEM node in Γ_C.

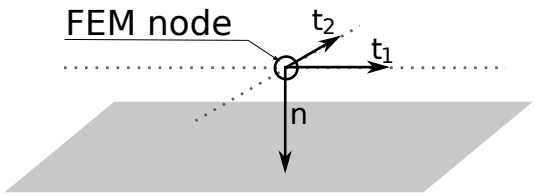

Figure 2: Normal and tangential vectors on Γ_C.

So for every contact node (i-th node from Γ_C) is $\boldsymbol{T}_i \in \mathbb{R}^{2,n}$ 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 $\boldsymbol{T} := \left[\boldsymbol{T}_1^T, \ldots, \boldsymbol{T}_{m_c}^T\right]^T$ 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 $\boldsymbol{K}$), so in next eductions, we can use standard inversion $\boldsymbol{K}^{-1}$.
We can express the non-differentiable term j_h in (1) by (see [7])

$$j_h(\boldsymbol{u}) = \sum_{i=1}^{m_c} \max_{\|\tau_i\|\leq\psi_i} \boldsymbol{\tau}_i^T\boldsymbol{T}_i\boldsymbol{u}, \quad (2)$$

where $\boldsymbol{\tau}_i \in \mathbb{R}^2$ are regulation variables.

2. Saddle point problem equivalency and dual formulation

At first, we denote function and vector

$$\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}) := f(\boldsymbol{u}) + \boldsymbol{\tau}^T\boldsymbol{T}\boldsymbol{u}, \quad \boldsymbol{\tau} := [\boldsymbol{\tau}_1^T,\ldots,\boldsymbol{\tau}_{m_c}^T]^T. \tag{3}$$

Then the conditions $\|\boldsymbol{\tau}_i\| \le \psi_i$ can by written in form

$$\sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \le \psi_i, \quad i = 1,\ldots,m_c, \tag{4}$$

where τ_j is j-th component of $\boldsymbol{\tau}$.
Now we can simplify the notation, we denote set of feasible $\boldsymbol{\tau}$ as

$$\Lambda_\tau := \left\{ \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \le \psi_i, i = 1,\ldots,m_c \right\}. \tag{5}$$

After substituing (2) into (1) and using (3),(4) we get

$$\min_{\boldsymbol{u}\in\Omega}\left(f(\boldsymbol{u}) + j_h(\boldsymbol{u})\right) = \min_{\boldsymbol{u}\in\Omega}\left(f(\boldsymbol{u}) + \sum_{i=1}^{m_c}\max_{\|\boldsymbol{\tau}_i\|\le\psi_i}\boldsymbol{\tau}_i^T\boldsymbol{T}_i\boldsymbol{u}\right) = \min_{\boldsymbol{u}\in\Omega}\sup_{\boldsymbol{\tau}\in\Lambda_\tau}\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}). \tag{6}$$

If we consider $\tilde{L}(\boldsymbol{u},\boldsymbol{\tau})$ as Lagrange function and $\boldsymbol{\tau}$ as vector of Lagrange multipliers (in notation (3)), we can use the classical duality theorem (see [4]) to reformulate problem (6) and get

$$\min_{\boldsymbol{u}\in\Omega}\sup_{\boldsymbol{\tau}\in\Lambda_\tau}\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}) = \max_{\boldsymbol{\tau}\in\Lambda_\tau}\min_{\boldsymbol{u}\in\Omega}\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}). \tag{7}$$

Now we can include condition $\boldsymbol{u}\in\Omega$ by creating new Lagrange multipliers.

$$\max_{\boldsymbol{\tau}\in\Lambda_\tau}\min_{\boldsymbol{u}\in\Omega_C}\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}) = \max_{\boldsymbol{\tau}\in\Lambda_\tau,\boldsymbol{\lambda}_C\ge 0}\min_{\boldsymbol{u}\in\mathbb{R}^n}\left(\tilde{L}(\boldsymbol{u},\boldsymbol{\tau}) + \boldsymbol{\lambda}_C^T(\boldsymbol{B}\boldsymbol{u}-\boldsymbol{c})\right), \tag{8}$$

where matrix $\boldsymbol{B}\in\mathbb{R}^{m_c,n}$ and vector $\boldsymbol{c}\in\mathbb{R}^{m_c}$ are constructed in such way, that

$$\{\boldsymbol{u}\in\mathbb{R}^n : \boldsymbol{B}\boldsymbol{u}\le\boldsymbol{c}\} = \Omega.$$

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

So problem (1) is equivalent to the saddle point problem

$$(\bar{\boldsymbol{u}},\bar{\boldsymbol{\lambda}}) := \arg\max_{\boldsymbol{\lambda}\in\Lambda}\min_{\boldsymbol{u}\in\mathbb{R}^n} L(\boldsymbol{u},\boldsymbol{\lambda}), \tag{9}$$

where

$$L(\boldsymbol{u},\boldsymbol{\lambda}) := f(\boldsymbol{u}) + \boldsymbol{\lambda}^T(\tilde{\boldsymbol{B}}\boldsymbol{u}-\tilde{\boldsymbol{c}}) \tag{10}$$

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

$$\boldsymbol{\lambda} := \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\lambda}_C \end{bmatrix}, \quad \tilde{\boldsymbol{B}} := \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{B} \end{bmatrix}, \quad \tilde{\boldsymbol{c}} := \begin{bmatrix} \boldsymbol{o} \\ \boldsymbol{c} \end{bmatrix},$$

$$\Lambda := \{[\boldsymbol{\tau}, \boldsymbol{\lambda}_C] \in \mathbb{R}^{3m_c} : \sqrt{\tau_{2i-1}^2 + \tau_{2i}^2} \leq \psi_i, i = 1, \ldots, m_c, \boldsymbol{\lambda}_C \geq \boldsymbol{o}\}.$$

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 $\bar{\boldsymbol{u}}$ of function $L(\boldsymbol{u}, .)$ satisfy state of stationary point - we put part of gradient of L corresponding to derivation with respect to components of $\boldsymbol{u}$ equal to zero)

$$\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{\lambda}) = \boldsymbol{K}\boldsymbol{u} - \boldsymbol{f} + \tilde{\boldsymbol{B}}^T\boldsymbol{\lambda} = \boldsymbol{o} \quad \Rightarrow \quad \bar{\boldsymbol{u}} = \boldsymbol{K}^{-1}\left(\boldsymbol{f} - \tilde{\boldsymbol{B}}^T\boldsymbol{\lambda}\right) \tag{11}$$

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

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

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

$$\bar{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{x}\in\Lambda} \Theta(\boldsymbol{x}), \quad \Theta(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}, \tag{12}$$

where we denoted

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

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

Obviously $\boldsymbol{A} \in \mathbb{R}^{3m_c,3m_c}$ 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.

Choose $\boldsymbol{x}_0 \in \Omega, \alpha \in (0, 2\|\boldsymbol{A}\|^{-1}), \delta \in (0, 1/2\rangle$
$\alpha_{bb} := \alpha$
$k := 0$
while $\|\boldsymbol{x}_k - P(\boldsymbol{x}_k - \boldsymbol{g}_k)\| \geq \epsilon\|b\|$ **do**
    if $2\delta \boldsymbol{g}_k^T \boldsymbol{g}_k^P \leq \|\varphi(\boldsymbol{x}_k)\|^2$ **then**
        CG step or CG halfstep.
        make CG step to solve problem on free set.
        if this step means leaving Ω, do only a half-step and restart CG.
        $k := k + 1$
    else
        Barzilai-Borwein gradient projection step.
        $\boldsymbol{x}_{k+1} := P(\boldsymbol{x}_k - \alpha_{bb}\boldsymbol{g}_k)$
        $\boldsymbol{s} := \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
        $\alpha_{bb} := \boldsymbol{s}^T\boldsymbol{s}/\boldsymbol{s}^T\boldsymbol{A}\boldsymbol{s}$
        restart CG
        $k := k + 1$
    end if
end while

In our algorithm we use these notations

$$\boldsymbol{g}_k := \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}, \quad \tilde{\boldsymbol{g}}_k := \frac{1}{\alpha}\left(\boldsymbol{x}_k - P(\boldsymbol{x}_k - \alpha\boldsymbol{g}_k)\right), \quad \boldsymbol{g}_k^P := \varphi(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k),$$

φ and β 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 Γ_C is given by the number of nodes on bottom side

$\psi = 900$													
N	**primal**	**dual**	**SPG**			**MPGP**				**MPGP-BB**			
			it	GLL	f(x)	it	cg	half	proj	it	cg	half	proj
4	375	75	36	9	44	5176	0	1	5175	41	0	1	40
6	1029	147	45	20	64	2746	0	1	2745	57	0	1	56
8	2187	243	27	12	38	1236	0	1	1235	51	0	1	50
10	3993	363	33	15	47	661	0	1	660	40	0	1	39

Table 1: Test with small radius.

$\psi = 15000$													
N	**primal**	**dual**	**SPG**			**MPGP**				**MPGP-BB**			
			it	GLL	f(x)	it	cg	half	proj	it	cg	half	proj
4	375	75	1566	977	2542	43	33	9	1	43	33	9	1
6	1029	147	923	553	1475	48	29	18	1	48	29	18	1
8	2187	243	588	366	953	53	24	28	1	53	24	28	1
10	3993	363	1020	547	1566	101	40	46	15	73	27	40	6

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 *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 *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 descend method SPG.

5. Conclusions

Our numerical experiments predicate better performace 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

[1] Barzilai, J. and Borwein, J. M.: Two-point step size gradient methods. IMA Journal of Numerical Analysis **8** (1988), 141–148.

[2] Birgin, E., Martínez, J., and Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10** (2000), 1196–1211.

[3] Dostál, Z. et al.: A theoretically supported scalable tfeti algorithm for the solution of multibody 3d contact problems with friction. Computer Methods in Applied Mechanics and Engineering **205208** (2012), 110–120.

[4] Dostál, Z.: *Optimal quadratic programming algorithms: with applications to variational inequalities*. Springer Publishing Company, Incorporated, 2009, 1st edn.

[5] Dostál, Z. and Kozubek, T.: An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. Math. Program. **135** (2012), 195–220.

[6] Grippo, L., Lampariello, F., and Lucidi, S.: A nonmonotone line search technique for newtons method. SIAM Journal on Numerical Analysis **23** (1986), 707–716.

[7] Hlaváček, I., Haslinger, J., Nečas, J., and Lovíšek, J.: *Solution of variational inequalities in mechanics*. No. sv. 66 in Applied Mathematical Sciences Series, Springer Verlag, Berlin, 1988.

[8] Kozubek, T. et al.: Matsol – Matlab efficient solvers for problems in engineering. URL `http://www.am.vsb.cz/matsol`.