Pontifical Catholic University of Chile
Engineering School
Innovation and Research Department
The Brigde - Inria Program

Accelerated Gradient Descent Method for the Frictional Contact Problem

Vicent Acary [1], Denise Cariaga[2]

Abstract

In this report we review the Discrete Frictional Contact Problem using a quadratic programming over second-order cones formulation in space and time discretized systems with unilateral contact and three dimensional Coulomb's friction. Many numerical methods are available in literature using this formulation [1]. However, we propose the application of Accelerated Projected Gradient Descent Method (APGD) with some improvements suggested by different authors. This numerical technique is compared over a large set of test examples using performance profiles.

Key words: Multibody Systems, Nonsmooth Mechanics, Unilateral Constraints, Coulomb's Friction, Accelerated Projected Gradient Descent Method.

1 Introduction

After more than 30 years of the pioneering work of several authors (see [1] references) on numerically solving mechanical problems with contact and friction, there are still active research activities on this subject in the computational mechanics community. The reason is that problems from mechanical systems with unilateral contact and Coulomb friction are numerically difficult to solve and the convergence of the numerical methods are rare and require strong assumptions. In this report we want to give some insights of the accelerated projected descent method by comparing it on the large sets of examples coming from the simulation of a wide range of mechanical systems.
The Accelerated Projected Gradient Descent Method is a powerful yet simple optimization technique to solve complex optimization problems such as this one with quadratic objective function and constrains.

1.1 Problem Statement

We formulate the abstract algebraic finite-dimensional contact problem [1, 2, 3]. This problem is modeled as a complementarity problem over second-order cones. Let $n_c \in \mathbb{N}$ be the number of contact points and

[1]INRIA, 655, Avenue de l'Europe, 38334 Saint Ismier Cedex, France. E-mail: vincent.acary@inria.fr.
[2]Institute for Mathematical and Computational Engineering, Pontifical Catholic University of Chile, 4860 Vicuna Mackenna, 7810000 Santiago, Chile. E-mail: dccariaga@uc.cl

$n \in \mathbb{N}$ the number of degrees of freedom of the discrete mechanical system. The problem data are a positive definite matrix $M \in \mathbb{R}^{n\times n}$, a vector $f \in \mathbb{R}^n$, a matrix $H \in \mathbb{R}^{m\times n}$ with $m = 3n_c$, a vector $w \in \mathbb{R}^m$ and a vector of coefficients of friction $\mu \in \mathbb{R}^{n_c}$. The unknowns are two vectors $v \in \mathbb{R}^n$, a velocity-like vector and $r \in \mathbb{R}^m$, a contact reaction or impulse, solution to

$$\begin{cases} Mv + f = H^\top r & u := Hv + w \\ K^*_{e,\mu} \ni \tilde{u} \perp r \in K_{e,\mu} & \tilde{u} := u + \Phi(u) \end{cases} \tag{1}$$

where the set $K_{e,\mu} \subseteq \mathbb{R}^m$ is the Cartesian product of Couloumb's friction second-order cone at each contact and $K^*_{e,\mu}$ it's dual cone, i.e.,

$$K_{e,\mu} = \prod_{\alpha=1\ldots n_c} K^\alpha_{e,\mu} = \prod_{\alpha=1\ldots n_c} \{(x_1, x_2) \in \mathbb{R}\times\mathbb{R}^{n-1} \mid \|x_2\| \le \mu x_1\} \tag{2}$$

$$K^*_{e,\mu} = \prod_{\alpha=1\ldots n_c} K^{\alpha *}_{e,\mu} = \prod_{\alpha=1\ldots n_c} \{(x_1, x_2) \in \mathbb{R}\times\mathbb{R}^{n-1} \mid \|x_2\| \le \frac{1}{\mu} x_1\} \tag{3}$$

The function $\Phi : \Re^m \to \Re^m$ is a nonsmooth function defined as

$$\Phi(u) = [\mu^\alpha \|E_t u^\alpha\| \hat{e}_n, \alpha = 1\ldots n_c]^\top \tag{4}$$

where $\hat{e}_t = [1, 0, 0]^\top$ is the normal component vector and $E_t = (0, 1, 1) \in \mathbb{R}^{3\times 3}$ the linear transformation that maps to the tangential part of $\tilde{u}$.

2 Research Methodology

2.1 Accelerated Projected Gradient Descent Method (APGD)

Using the formulation described by Gordon and Tibshirani [5] for the

$$\underset{x\in\mathbb{R}^n}{\text{Minimize}} \qquad f(x) \tag{5}$$

with $f(x)$ convex and differentiable, the accelerated projected gradient descent method works as:

- Choose any initial $x_0 = x_{-1} \in \mathbb{R}^n$
- Repeat for $k = 1, 2, 3, \ldots$ and $\mathbb{P}_K$ the projection over the space K.

$$y = x_{k-1} + \frac{k-2}{k+1}(x_{k-1} - x_{k-2}) \tag{6a}$$

$$x_k = \mathbb{P}_K[y - \rho_k \nabla f(y)] \tag{6b}$$

- Stop with the metric [4], $\epsilon > 0$ and $g_d = 10^{-6}$:

$$\left\| \frac{1}{m g_d}(x_k - \mathbb{P}_K[x_k - g_d \nabla f(x_k)]) \right\|_2 < \epsilon \tag{7}$$

2.2 APGD formulaction of problem (1)

Since the problem (1) is nonsmooth and nonconvex, the use of an associated optimization problem is interesting from the numerical point of view if we want to improve the robustness and the stability of the numerical methods. In vincent1, a parametric convex optimization formulation is presented for the dynamical Coulomb friction problem. Using a direct substitution of $v = M^{-1}H^\top r - M^{-1}f$ in $u = Hv + w$. We

have $u = HM^{-1}H^\top r - HM^{-1}f + w = Wr + q$ with $W = HM^{-1}H^\top$ and $q = -HM^{-1}f + w$.
The problem reads

$$\underset{r}{\text{Minimize}} \quad \frac{1}{2} r^\top W r + r^\top (q + s) \tag{8a}$$

$$\text{subject to} \quad r \in K_{e,\mu} \tag{8b}$$

with $s = \Phi(u)$ the fixed value of $\Phi(u)$ in this iteration.

So the APGD formulation for the Frictional Contact Problem with $\bar{q} = q + s$, is:

$$y = r_{k-1} + \frac{k-2}{k+1}(r_{k-1} - r_{k-2}) \tag{9a}$$

$$r_k = \mathbb{P}_{K_{e,\mu}}[y - \rho_k(Wy + \bar{q})] \tag{9b}$$

and stop metric defined in [4] with $\epsilon = 10^{-3}$ and $g_d = 10^{-6}$:

$$\left\| \frac{1}{m g_d}(r_k - \mathbb{P}_{K_{e,\mu}}[r_k - g_d(Wr_k + \bar{q})]) \right\|_2 < \epsilon \tag{10}$$

Algorithm 1 APGD

Require: Data of (1), $error = \infty$, $\epsilon > 0$, $Iter_{\max} > 0$, ρ_k, s and $r_{-1} = r_0 = r_1 = 0$
1: $k \leftarrow 1$
2: **while** $error > \epsilon$ and $k < Iter_{\max}$ **do**
3: Update ρ_k if needed (see Algorithm 2)
4: $y = r_{k-1} + \frac{k-2}{k+1}(r_{k-1} - r_{k-2})$
5: $r_k = \mathbb{P}_{K_{e,\mu}}[y - \rho_k(Wy + q + s)]$
6: $error = \left\| \frac{1}{m g_d}(r_k - \mathbb{P}_{K_{e,\mu}}[r_k - g_d(Wr_k + \bar{q})]) \right\|_2$
7: $k \leftarrow k + 1$
8: **end while**
9: $r \leftarrow r_k$

2.3 Projection onto second-order cone

The projection onto second-order cone is described in [6, 1] using Jordan algebra (in a general SOCC-function) For $\boldsymbol{x} \in \mathbb{R}^n$, let $\mathbb{P}_{K_{e,\mu}}(\boldsymbol{x}) \in \mathbb{R}^n$ denote the projection of $\boldsymbol{x}$ onto $K_{e,\mu}$, i.e.,

$$\mathbb{P}_{K_{e,\mu}}(\boldsymbol{x}) = \{\|\boldsymbol{x}' - \boldsymbol{x}\| \mid \boldsymbol{x}' \in K_{e,\mu}\}. \tag{11}$$

Therefore the projection of x onto $K_{e,\mu}$ could be written as follows

$$\mathbb{P}_{K_{e,\mu}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } -\boldsymbol{x} \in K^*_{e,\mu} \\ \boldsymbol{x} & \text{if } \boldsymbol{x} \in K_{e,\mu} \\ \dfrac{(x_1 + \mu\|\boldsymbol{x}_2\|)}{1+\mu^2} \begin{bmatrix} 1 \\ \mu \boldsymbol{x}_2 / \|\boldsymbol{x}_2\| \end{bmatrix} & \text{if } -\boldsymbol{x} \notin K^*_{e,\mu} \wedge \boldsymbol{x} \notin K_{e,\mu}, \end{cases} \tag{12}$$

2.4 Penalty Parameter ρ_k

A key ingredient in the efficiency and the convergence time of APGD is the choice of the sequence $\{\rho_k\}$. This section presents the most popular approach for choosing the sequence $\{\rho_k\}$ for quadratic problems QP, which is closely related to the formulation in (8).

2.4.1 Optimal Penalty Parameter

- Ghadimi [7]:

$$\rho^\star := \left(\sqrt{\lambda_{min}(W)\lambda_{max}(W)}\right)^{-1} \tag{13}$$

- Di Cairano [8]:

$$\rho^\star := \sqrt{\lambda_{min}(M)\lambda_{max}(M)} \tag{14}$$

- Acary:

$$\rho^\star := \frac{\|M\|_1}{\|H\|_1} \tag{15}$$

- Normal:

$$\rho^\star := 1 \tag{16}$$

- SmallerRho:

$$\rho^\star := \frac{2}{3} \tag{17}$$

- WRho:

$$\rho^\star := \|W\|^{-1} \tag{18}$$

- EigenWRho:

$$\rho^\star := (\lambda_{max}(W))^{-1} \tag{19}$$

2.4.2 Updating rule for ρ_k

It is described [1] a method to varying the penalty parameter in order to get a better convergence of the APGD.

- Ratio 1 in the iteration k:

$$ratio_1^k := \frac{\rho_k\|\nabla f(r_k) - \nabla f(\bar{r_k})\|}{\|r_k - \bar{r_k}\|} \tag{20}$$

- Ratio 2 in the iteration k:

$$ratio_2^k := \frac{\rho_k(r_k - \bar{r_k})^\top(\nabla f(r_k) - \nabla f(\bar{r_k}))}{\|r_k - \bar{r_k}\|^2} \tag{21}$$

Where $\bar{r_k} = \mathbb{P}_K[r_k - \rho_k\nabla f(r_k)]$

Algorithm 2 Updating rule for ρ_k

Require: $L \in (0,1)$, $0 < L_{\min} < L$, $w \in (0,1)$, r^k, and $\rho_{k-1} > 0$

$\rho_k \leftarrow \rho_{k-1}$
$\bar{r_k} = \mathbb{P}_K[r_k - \rho_k \nabla f(r_k)]$
Evaluate $ratio^k$ ((20) or (21))
while $ratio^k > L$ **do**
    $\rho_k \leftarrow w\rho_k$
    $\bar{r_k} = \mathbb{P}_K[r_k - \rho_k \nabla f(r_k)]$
    Evaluate $ratio^k$ ((20) or (21))
end while
if $ratio^k < L_{\min}$ **then**
    $\rho_k \leftarrow \frac{1}{w}\rho_k$
end if

2.5 Updating s

As we mentioned before, one way to solve the problem (1) with the new formulation (8) is by fixing the $s = \Phi(Wr + q)$ value and then solve it with the APGD method. Once you get the r result, the procedure it is repeated until the error is small.This is known as the Fixed Point method [1]. In [3] there is a proof of existence and uniqueness of the s external update.

The external error measurement is given for each contact $\alpha = 1...n_c$ by

$$\Big[\frac{\Phi(Wr_{k+1} + q) - \Phi(Wr_k + q)}{\Phi(Wr_k + q)}\Big]^{\alpha} \leq \epsilon^{ext} \tag{22}$$

where $\epsilon^{ext} = 10^{-3}$ is the external tolerance.

Algorithm 3 Updating s

Require: r, $s_0 = 0 \in \mathbb{R}^m$, $Iter_{\max} > 0$ and $\epsilon^{ext} = 10^{-3}$

$i \leftarrow 0$
while no (22) and $i < Iter_{\max}$ **do**
    if $i = 0$ **then**
        $s = s_0$
    else
        $s = \Phi(Wr + q)$
    end if
    Solve APGD to get r
    $i \leftarrow i + 1$
end while

3 Results and Discussion

3.1 Performance Profiles

This concept was introduced in [9] for benchmarking optimization solvers on a large set of problems. Following [1], for a set P of n_p problems, and a set S of n_s solvers, we define a performance criterion for a solver s, a problem p and a required precision tol by

$$t_{p,s} = \text{computing time required for } s \text{ to solve } p \text{ at precision } tol \tag{23}$$

A performance ratio over all the solvers is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}, s \in S\}} \geq 1 \tag{24}$$

For $\tau > 1$, we define a distribution function ρ_s for the performance ratio for a solvers as

$$\rho_s(\tau) = \frac{1}{n_p} card\{p \in P, r_{p,s} \leq \tau\} \leq 1 \tag{25}$$

This distribution computes the number of problems p that are solved with a performance ratio below a given threshold τ. In other words, $\rho_s(\tau)$ represents the probability that the solver s has a performance ratio not larger than a factor τ of the best solver. The higher ρ_s is, the better the method is. In the sequel, the term performance profile denotes a graph of the functions $\rho_s(\tau)$, $\tau > 1$. The computational time is used to measure performance in the algorithms.

In this section we perform three simulations for APGD, APGD with varying ρ_k and ratio 1, and APGD with varying ρ_k and ratio 2 to compare them and choose the proper method for future analysis. As recommended in [1], we took $L = 0.9$, $L_{\min} = 0.3$ and $w = 2/3$. Also, $\epsilon = 10^{-3}$ and $Iter_{\max} = 200$. The set used have 58 problems.

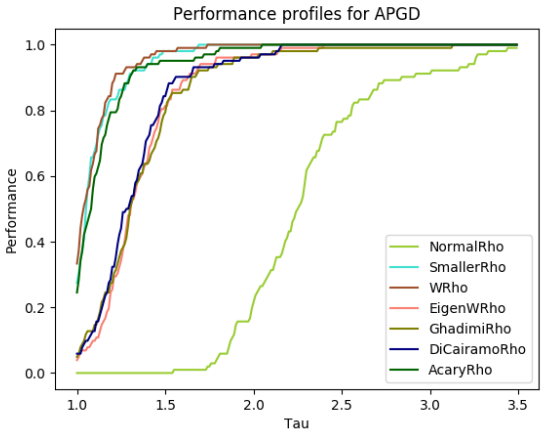
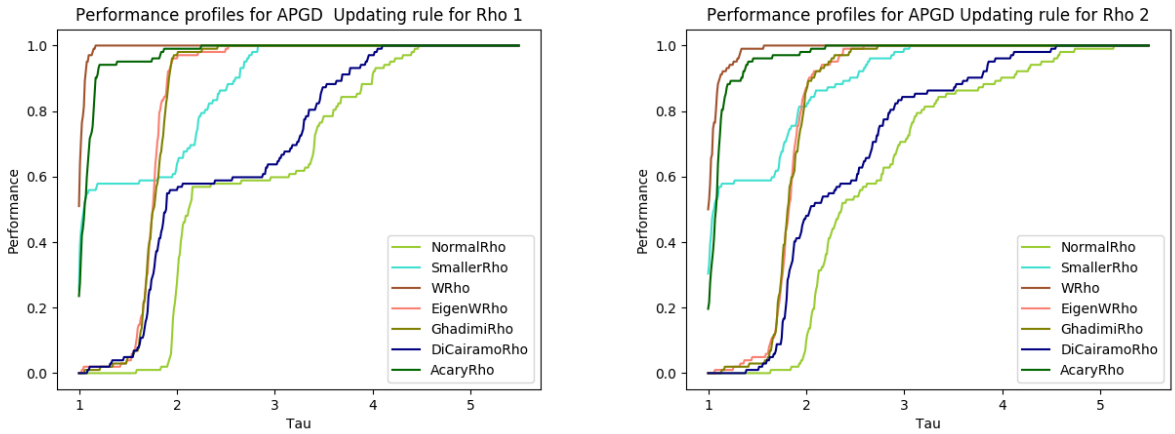


Figure 1: Performance profiles for different ρ_k in the APGD method

In the Figure 1, we can distinguish three groups of ρ values in terms of their performance. It is clear that the best are Acary, SmallerRho and WRho. On the other side, the worst performance was for Normal. Running this method took a total of 262 seconds to solve.

Figure 2 shows us the APGD method updating ρ_k value. We implemented this method in order to verify if the speed of APGD was faster when modifying ρ_k. The sub-figure 2a uses the ratio 1 defined in (20). On the other side, the sub-figure 2b uses the ratio 2 defined in (21). As expected, some of the ρ_k improved, while others did not. However, it is important to notice that Acary and WRho reached performance 1 with $\tau \in (1, 1.5)$ approximately, compared to a $\tau \in (1.5, 2)$ in the simple APGD. Particularly, the WRho in the Upgrade ρ_k APGD method with ratio1 seems to be the one that is more suitable for the data set used in the Discrete Frictional Contact Problem. Running this methods took a total of 304 and 308 seconds to solve respectively.



(a) Ratio 1

(b) Ratio 2

Figure 2: Performance profiles for different ρ_k in the varying APGD method

4 Conclusions

Throughout this research we review the APGD method over the Discrete Frictional Contact Problem. In order to solve the problem with this method it is necessary to fix the value of a nonsmooth function $\Phi(u)$. This strategy slows down the computing time. However, we tried accelerating the Projected Gradient Descent Method and varying the Penalty Parameter ρ_k for different initial ρ_k values.

Regarding the last point, we could see that the best performance of this method was with the Acary and WRho rhos using the APGD with upgrading ρ_k and ratio 1 defined in (20). On the other hand, there was a significative worsen in the Normal and Di Cairano rhos when upgrading the ρ_k compared with the fixed rho APGD method.

In terms of computational time, solving APGD with and upgraded ρ_k increased by 18% the total time compared to the fixed ρ_k method. So it would be interesting to analyze in the future if the extra time is worth knowing that the performance of the method with some initial ρ_k, such as Acary and WRho, increases significantly.

To continue this research, the following steps should include a convergence analysis of the s values when they are updated. Also, compare the performance and computational time of the Projected Gradient Descent and the APGD formulated here.

References

[1] Acary, V., Brémond, M. & Huber, O. (2017). *On solving contact problems with Coulomb friction: formulations and numerical comparisons*. Research Report RR-9118. Grenoble: INRIA.

[2] Acary, V. & Cadoux, F. (2011). *Applications of an Existence Result for the Coulomb Friction Problem*. Pages 45–66. , Berlin: Springer Berlin Heidelberg.

[3] Acary, V., Cadoux, F., Lemaréchal, C. & Malick, J. (2011) *A formulation of thelinear discrete coulomb friction problem via convex optimization*. Pages 155–175. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik.

[4] Mazhar, H., Heyn, T., Negrut, D. & Tasora, A. (2015). *Using Nesterov's Method to Accelerate Multibody Dynamics with Friction and Contact*. ACM Trans. Graph. 34, 3, Article 32.

[5] Gordon, G. & Tibshirani, R. (2012). *Lecture 9: Review of Generalized Gradient Descent* Pittsburgh: Carnegie Mellon University Department of Statistics. Available in: `https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725_Lecture9.pdf`

[6] Hayashi, S., Yamashita, N. & Fukushima, M. (2005). *A combined smoothing and regularization method for monotone second-order cone complementarity problems*. SIAM Journal on Optimization 15, 2. Pages 593-615.

[7] Ghadimi, E., Teixeira, A., Shames, I. & Johansson, M. (2015). *Optimal parameterselection for the alternating direction method of multipliers (admm): quadratic problems*. IEEE Transactions on Automatic Control 60,3. Pages 644–658.

[8] Raghunathan, A. & Di Cairano, S. (2014). *Alternating direction method of multipliersfor strictly convex quadratic programs: Optimal parameter selection*. American Control Conference (ACC).Pages 4324–4329.

[9] Dolan, E. & Moré, J. (2002). *Benchmarking optimization software with performance profiles*. Mathematical Programming 91,2. Pages 201–213.