Research Memorandum

February 26, 2018

Alternating Direction Method of Multipliers (ADMM) for Frictional Contact [*]

Vincent Acary [†] , Yoshihiro Kanno [‡] , Nicolas Molina [§]

Abstract

Abstract: In this report, we review the quadratic programming (QP) over second-order cones formulation of the discrete frictional contact problem that arises in space and time discretized mechanical systems with unilateral contact and three-dimensional Coulomb's friction. Thanks to this formulation, various numerical methods emerge naturally for solving the problem [1]. Here we propose the application of the Alternating Direction Method of Multipliers (ADMM) with various improvements proposed in the literature. This numerical technique is compared over a large set of test examples using performance profiles.

Keywords: Multibody systems, nonsmooth mechanics, unilateral constraints, Coulomb's friction, impact, numerical methods, ADMM.

[*] `https://github.com/molinavergara24/inria-admm`

[†] INRIA, 655, Avenue de l'Europe, 38334 Saint Ismier Cedex, France. E-mail: `vincent.acary@inria.fr`.

[‡] Laboratory for Future Interdisciplinary Research of Science and Technology, Institute of Innovative Research, Tokyo Institute of Technology, Nagatsuta 4259, Yokohama 226-8503, Japan. E-mail: `kanno.y.af@m.titech.ac.jp`.

[§] Department of Mechanical and Metallurgical Engieering, Escuela de Ingenieria, Pontificia Universidad Catolica de Chile, Vicuna Mackenna 4860, Santiago, Chile. E-mail: `namolina@uc.cl`.

Contents

1 Introduction

More than thirty years after the pioneering work of several authors (see [1] for references) on numerically solving mechanical problems with contact and friction, there are still active research on this subject in the computational mechanics and applied mathematics communities. This can be explained by the fact that problems from mechanical systems with unilateral contact and Coulomb's friction are difficult to numerically solve and the mathematical results of convergence of the numerical algorithms are rare and most of these require rather strong assumptions. In this report, we want to give some insights of the advantages and weaknesses of the ADMM numerical scheme by comparing it on the large sets of examples coming from the simulation of a wide range of mechanical systems.

The ADMM is an invaluable element of the modern optimization toolbox. ADMM decomposes complex optimization problems into sequences of simpler subproblems, often solvable in closed form; its simplicity, flexibility, and broad applicability, make ADMM a state-of-the-art solver in machine learning, signal processing, and many other areas [2].

1.1 Problem statement

In this section, following [3, 4] we formulate an abstract, algebraic finite-dimensional frictional contact problem. This problem is modeled as a complementarity problem over second-cones, and the properties of the latter are discussed. Let $n_c \in \mathbb{N}$ be the number of contact points and $n \in \mathbb{N}$ the number of degrees of freedom of a 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 $H \in \mathbb{R}^{n\times m}$ 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

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

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 and $K^*_{e,\mu}$ its dual cone, i.e.,

$$K_{e,\mu} = \prod_{\alpha=1\ldots n_c} K^\alpha_{e,\mu} = \prod_{\alpha=1\ldots n_c} \{(x_1, \boldsymbol{x}_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|\boldsymbol{x}_2\| \leq \mu x_1\} \tag{2}$$

$$K^*_{e,\mu} = \prod_{\alpha=1\ldots n_c} K^{\alpha *}_{e,\mu} = \prod_{\alpha=1\ldots n_c} \{(x_1, \boldsymbol{x}_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|\boldsymbol{x}_2\| \leq \frac{1}{\mu} x_1\}. \tag{3}$$

The function $\Phi : \mathbb{R}^m \to \mathbb{R}^m$ is a nonsmooth function defined as

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

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

2 ADMM method

2.1 Fundamentals

In this section, following [2] we explain the alternating direction method of multipliers (ADMM) for convex optimization.

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex functions. Consider the following convex optimization problem in variables $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$:

$$\begin{aligned} &\text{Minimize} && f(\boldsymbol{x}) + g(\boldsymbol{y}) && \text{(5a)} \\ &\text{subject to} && A\boldsymbol{x} + B\boldsymbol{y} = \boldsymbol{c}, && \text{(5b)} \end{aligned}$$

where $A \in \mathbb{R}^{l \times n}$ and $B \in \mathbb{R}^{l \times m}$ are constant matrices, and $\boldsymbol{c} \in \mathbb{R}^l$ is a constant vector.

The augmented Lagrangian of problem (5) is defined as

$$L_\rho(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{w}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \boldsymbol{w}^\top (A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{c}) + \frac{\rho}{2} \|A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{c}\|^2, \tag{6}$$

where $\rho > 0$ is the penalty parameter, and $\boldsymbol{w} \in \mathbb{R}^l$ is the Lagrange multiplier (or the dual variable). ADMM consists of the iterations

$$\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{w}^k), \tag{7}$$

$$\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{w}^k), \tag{8}$$

$$\boldsymbol{w}^{k+1} := \boldsymbol{w}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}). \tag{9}$$

For reference, the algorithm called the method of multipliers (or the augmented Lagrangian method) updates the variables as follows:

$$(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}) := \arg\min_{\boldsymbol{x}, \boldsymbol{y}} L_\rho(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{w}^k) \tag{10a}$$

$$\boldsymbol{w}^{k+1} := \boldsymbol{w}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}). \tag{10b}$$

Namely, the method of multipliers updates the primal variables $\boldsymbol{x}$ and $\boldsymbol{y}$ simultaneously. In contrast, ADMM updates $\boldsymbol{x}$ and $\boldsymbol{y}$ sequentially.

ADMM is often described in a slightly different form. Define $\boldsymbol{z}$ by $\boldsymbol{z} = \boldsymbol{w}/\rho$. Then the augmented Lagrangian in (6) is reduced to

$$L_\rho(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \frac{\rho}{2} \|A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{c} + \boldsymbol{z}\|^2 - \frac{\rho}{2} \|\boldsymbol{z}\|^2. \tag{11}$$

Using L_ρ in (11), we can express ADMM as follows:

$$\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k), \tag{12}$$

$$\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k), \tag{13}$$

$$\boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}. \tag{14}$$

The expression in (12), (13), and (14) is called the ADMM in the scaled form, and $\boldsymbol{z}$ is called the scaled dual variable.

2.2 Residuals

The necessary and sufficient optimality condiction for the ADMM problem (5) are primal and dual feasibility, therefore, one common way to measure how well the iterates satisfy the KKT conditions is to define the primal and dual residuals:

$$\begin{aligned} \boldsymbol{r}^{k+1} &:= A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c} \\ \boldsymbol{s}^{k+1} &:= \rho A^T B \left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right) \end{aligned} \tag{15}$$

2.3 Stop criterion

The residuals of the optimality conditions can be related to an approximated bound on the objective suboptimalty of the current point

$$\begin{aligned} \|\boldsymbol{r}^k\| &\leq \epsilon^{pri} \\ \|\boldsymbol{s}^k\| &\leq \epsilon^{dual} \end{aligned} \tag{16}$$

where $\epsilon^{pri} > 0$ and $\epsilon^{dual} > 0$ are feasibility tolerances for the primal and dual residuals. These tolerances can be choosen using an absolute and relative criterion, such as

$$\begin{aligned} \epsilon^{pri} &= \sqrt{l}\epsilon^{abs} + \epsilon^{rel} \max\{\|A\boldsymbol{x}^{k+1}\|, \|B\boldsymbol{y}^{k+1}\|, \|\boldsymbol{c}\|\} \\ \epsilon^{dual} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|A^T \rho \boldsymbol{z}^{k+1}\| \end{aligned} \tag{17}$$

where $\epsilon^{abs} > 0$ and $\epsilon^{dual} > 0$ is an absolute and relative tolerance, respectively. A reasonable value for the absolute and relative tolerance might be 10^{-6} and 10^{-3}, respectively. This values typically depend on the scale of the data size.

3 ADMM formulation of problem (1)

3.1 Coulomb friction problem via convex optimization

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 [4], a parametric convex optimization formulation is presented for the dynamical Coulomb friction problem. The problem reads

$$\begin{aligned} &\underset{\boldsymbol{v},\tilde{\boldsymbol{u}}}{\text{Minimize}} && \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} && \text{(18a)}\\ &\text{subject to} && \tilde{\boldsymbol{u}} = H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}), && \text{(18b)}\\ & && \tilde{\boldsymbol{u}} \in K^*_{e,\mu}, && \text{(18c)}\end{aligned}$$

where $\boldsymbol{s} \in \mathbb{R}^m$ is a parameter vector. Let $\delta_{K^*_{e,\mu}} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ denote the indicator function of $K^*_{e,\mu}$, i.e.,

$$\delta_{K^*_{e,\mu}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in K^*_{e,\mu}, \\ +\infty & \text{otherwise.} \end{cases} \tag{19}$$

Then problem (18) is equivalently rewritten as follows:

$$\begin{aligned} &\underset{\boldsymbol{v},\tilde{\boldsymbol{u}}}{\text{Minimize}} && \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) && \text{(20a)}\\ &\text{subject to} && \tilde{\boldsymbol{u}} = H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}) && \text{(20b)}\end{aligned}$$

where $\Phi(\boldsymbol{s}) = [\mu^\alpha \|E_t \boldsymbol{s}^\alpha\| \hat{e}_t, \alpha = 1...n_c]^\top$. Therefore, in the notation of (5): $f(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v}, g(\tilde{\boldsymbol{u}}) = \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}), A = H, B = -I$ and $\boldsymbol{c} = -\boldsymbol{w} - \Phi(\boldsymbol{s})$.

3.2 Iteration

Problem (20) is a minimization problem of a convex function under linear equality constraints. Its augmented Lagrangian (in the scaled form) is formulated as

$$L_\rho(\boldsymbol{v};\tilde{\boldsymbol{u}};\boldsymbol{\zeta}) = \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta} - \tilde{\boldsymbol{u}}\|^2 - \frac{\rho}{2}\|\boldsymbol{\zeta}\|^2, \tag{21}$$

where ρ is the penalty parameter, and $\boldsymbol{\zeta} \in \mathbb{R}^m$ is the scaled dual variable (the scaled Lagrange multiplier). The ADMM solving problem (20) consists of iterating the updates (see (12), (13), and (14))

$$\boldsymbol{v}^{k+1} := \arg\min_{\boldsymbol{v}} L_\rho(\boldsymbol{v};\tilde{\boldsymbol{u}}^k;\boldsymbol{\zeta}^k), \tag{22}$$

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} L_\rho(\boldsymbol{v}^{k+1};\tilde{\boldsymbol{u}};\boldsymbol{\zeta}^k), \tag{23}$$

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H\boldsymbol{v}^{k+1} - \tilde{\boldsymbol{u}}^{k+1} + \Phi(\boldsymbol{s}). \tag{24}$$

The first step in (22) corresponds to unconstrained minimization of a convex quadratic function. It follows from the stationarity condition of this convex quadratic function that $\boldsymbol{v}^{k+1}$ is the solution to the following system of linear equations:

$$\left[M + \rho H^\top H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho H^\top(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{s}) - \boldsymbol{\zeta}^k). \tag{25}$$

The second step in (23) can be computed independently for each $\alpha = 1...n_c$ as

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta}^k - \tilde{\boldsymbol{u}}\|^2. \tag{26}$$

(26) can be rewritten by using the projection onto the second-order cone as

$$\tilde{\boldsymbol{u}}^{k+1} := \Pi_{K^*_{e,\mu}}(H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta}^k) \tag{27}$$

It is worthy to note that for (25) we carry out a Cholesky or LU factorization and (27) can be solved explicitly by using the formulan given in the next section.

3.3 Projection onto dual second-order cone

For vector $\boldsymbol{x} = (x_1, \boldsymbol{x}_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, its spectral factorization with respect to $K^*_{e,\mu}$ by means of Jordan algebra (in a general SOCC-function) is defined by [5]

$$\boldsymbol{x} = \lambda_1\boldsymbol{u}^1 + \lambda_2\boldsymbol{u}^2. \tag{28}$$

Here, λ_1, $\lambda_2 \in \mathbb{R}$ are the spectral values given by

$$\lambda_i = x_1 + (-1)^i \mu^{(-1)^{i+1}}\|\boldsymbol{x}_2\|, \tag{29}$$

and $\boldsymbol{u}^1$, $\boldsymbol{u}^2 \in \mathbb{R}^n$ are the spectral vectors given by

$$\boldsymbol{u}^i = \begin{cases} \dfrac{\mu^2}{1+\mu^2}\begin{bmatrix} \mu^{2(i-2)} \\ (-1)^i \frac{1}{\mu}\boldsymbol{x}_2/\|\boldsymbol{x}_2\| \end{bmatrix} & \text{if } \boldsymbol{x}_2 \neq \boldsymbol{0}, \\ \dfrac{\mu^2}{1+\mu^2}\begin{bmatrix} \mu^{2(i-2)} \\ (-1)^i \frac{1}{\mu}\boldsymbol{\omega} \end{bmatrix} & \text{if } \boldsymbol{x}_2 = \boldsymbol{0}, \end{cases} \tag{30}$$

with $\boldsymbol{\omega} \in \mathbb{R}^{n-1}$ satisfying $\|\boldsymbol{\omega}\| = 1$.
For $\boldsymbol{x} \in \mathbb{R}^n$, let $\Pi_{K^*_{e,\mu}}(\boldsymbol{x}) \in \mathbb{R}^n$ denote the projection of $\boldsymbol{x}$ onto $K^*_{e,\mu}$, i.e.,

$$\Pi_{K^*_{e,\mu}}(\boldsymbol{x}) = \arg\min\{\|\boldsymbol{x}' - \boldsymbol{x}\| \mid \boldsymbol{x}' \in K^*_{e,\mu}\}. \tag{31}$$

This can be computed explicitly as [6]

$$\Pi_{K^*_{e,\mu}}(\boldsymbol{x}) = \max\{0, \lambda_1\}\boldsymbol{u}^1 + \max\{0, \lambda_2\}\boldsymbol{u}^2. \tag{32}$$

Therefore the projection of $\boldsymbol{x}$ onto $K^*_{e,\mu}$ could be written as follows

$$\Pi_{K^*_{e,\mu}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } -\boldsymbol{x} \in K_{e,\mu} \to \lambda_i \leq 0 \\ \boldsymbol{x} & \text{if } \boldsymbol{x} \in K^*_{e,\mu} \to \lambda_i \geq 0 \\ \dfrac{\mu^2\left(x_1 + \frac{1}{\mu}\|\boldsymbol{x}_2\|\right)}{1+\mu^2}\begin{bmatrix} 1 \\ \frac{1}{\mu}\boldsymbol{x}_2/\|\boldsymbol{x}_2\| \end{bmatrix} & \text{if } -\boldsymbol{x} \notin K_{e,\mu} \wedge \boldsymbol{x} \notin K^*_{e,\mu} \to \lambda_1 < 0 \wedge \lambda_2 > 0, \end{cases} \tag{33}$$

4 Penalty parameter for QP

The ADMM technique is surprisingly robust to poorly selected algorithm parameters: under mild conditions, the method is guaranteed to convergence for all positive values of ρ [2, 7, 8]. However, a key ingredient in the efficiency and the convergence time of ADMM is the choise of the sequence $\{\rho_k\}$. A sensible work has been done in the literature mainly motivated by the rate of convergence. This section presents the most popular approach for choosing the sequence $\{\rho_k\}$ for QPs.

4.1 Optimal penalty parameter selection for QP

In this section we present two main results that have been proposed for QPs in the initial guess of the parameter and we define another two ones:

4.1.1 Ghadimi

The $\rho^\star$ that minimizes the rate of convergence is

$$\rho^\star := \left(\sqrt{\lambda_{min}(HM^{-1}H^\top)\lambda_{max}(HM^{-1}H^\top)}\right)^{-1} \tag{34}$$

If the constraint matrix H is not full rank, the smallest non-zero eigenvalue is taken, which still works as a heuristic to reduce the rate of convergence [9].

4.1.2 Di Cairano

The $\rho^\star$ that minimizes the rate of convergence is

$$\rho^\star := \sqrt{\lambda_{min}(M)\lambda_{max}(M)} \tag{35}$$

In this case, the nullity of H equal to zero is avoided [10].

4.1.3 Acary

In this report is proposed the following value, where the penalty parameter could be seen as the factor that balance the matrix involved in the the first term of (25)

$$\rho^\star := \frac{\|M\|_1}{\|H\|_1} \tag{36}$$

4.1.4 Normal

Finally, a less expensive and standard method is to set

$$\rho^\star := 1 \tag{37}$$

4.2 Varying penalty parameter

4.2.1 He

He et al. [11] argue that adaptively choosing the penalty parameter to balance the residuals is a reasonable heuristic for minimising the distance from convergence: increasing ρ strengthens the penalty term, yielding smaller primal residuals but larger dual ones; conversely, decreassing ρ leads to larger primal and smaller dual residuals. Therefore, this heuristic is implemeted so that ρ keeps both residuals of similar magnitude

$$\rho^{k+1} := \begin{cases} \tau\rho^k & \text{if } \|\boldsymbol{r}^k\| > \varphi\|\boldsymbol{s}^k\|, \\ \tau^{-1}\rho^k & \text{if } \|\boldsymbol{s}^k\| > \varphi\|\boldsymbol{r}^k\|, \\ \rho^k & \text{otherwise,} \end{cases} \tag{38}$$

where $\tau > 1$ and $\varphi > 1$. This method appears to becoming quite popular in the literature (see [12] for references).

4.2.2 Wohlberg

The behavior of ADMM under scaling is addresed by Wohlberg [12]. For problems involving physical quantities, for example, scaling corresponds to choices of the units in which the functional value and solution are expressed. The heuristic implemented by He et al. is appropriately scaled to mantain invariance of the algorithm as:

$$\rho^{k+1} := \begin{cases} \tau\rho^k & \text{if } \|\boldsymbol{r}^k_{rel}\| > \xi\varphi\|\boldsymbol{s}^k_{rel}\|, \\ \tau^{-1}\rho^k & \text{if } \|\boldsymbol{s}^k_{rel}\| > \xi\varphi\|\boldsymbol{r}^k_{rel}\|, \\ \rho^k & \text{otherwise,} \end{cases} \tag{39}$$

where $\xi \in \mathbb{R}^+$ and the relative primal and dual residuals are defined by

$$\begin{aligned} \boldsymbol{r}^{k+1}_{rel} &:= \frac{A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}}{\max\{\|A\boldsymbol{x}^{k+1}\|, \|B\boldsymbol{y}^{k+1}\|, \|\boldsymbol{c}\|\}} \\ \boldsymbol{s}^{k+1}_{rel} &:= \frac{\rho A^T B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right)}{\|A^T\boldsymbol{w}^{k+1}\|} = \frac{A^T B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right)}{\|A^T\boldsymbol{\zeta}^{k+1}\|} \end{aligned} \tag{40}$$

This approach avoids the need for explicit compensation for problem scaling when the formulation is modified. The stopping critera in (16) is invariant to problem scaling when $\epsilon^{abs} = 0$.

The fixed multiplier τ is a potential weakness of the penalty update policies: if τ is small, then a large number of iterations may be equired to reach appropiate ρ value if ρ^0 is poorly chosen; on the other hand, if τ is large, the corrections to ρ may be too large when ρ is close to the optimal value. A straightforward solution is to adapt τ at each iteration

$$\tau^{k+1} := \begin{cases} \sqrt{\xi^{-1}\dfrac{\|\boldsymbol{r}^k_{rel}\|}{\|\boldsymbol{s}^k_{rel}\|}} & \text{if } 1 \le \sqrt{\xi^{-1}\frac{\|\boldsymbol{r}^k_{rel}\|}{\|\boldsymbol{s}^k_{rel}\|}} < \tau_{max}, \\ \sqrt{\xi\dfrac{\|\boldsymbol{s}^k_{rel}\|}{\|\boldsymbol{r}^k_{rel}\|}} & \text{if } \tau^{-1}_{max} < \sqrt{\xi^{-1}\frac{\|\boldsymbol{r}^k_{rel}\|}{\|\boldsymbol{s}^k_{rel}\|}} < 1, \\ \tau_{max} & \text{otherwise,} \end{cases} \tag{41}$$

where τ_{max} provides a bound on τ, i.e., the convergence results in [11] still hold for this extension.

4.2.3 Spectral

Xu et al. [13] propose to automate and speed up ADMM by using stepsize selection rules adapted from the gradient descent literature, namely the Barzilai-Borwein ‘spectral’ method, based in the dual of the ADMM in (5).

The spectral stepsize estimation requires the curvature parameters α and β. They are estimated based on the results from iteration k and an older iteration $k_0 < k$

$$\boldsymbol{z}_S^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^k - \boldsymbol{c}\right) \tag{42}$$

$$\Delta\hat{w}_k := \rho^k \boldsymbol{z}_S^{k+1} - \rho^{k_0}\boldsymbol{z}_S^{k_0} \tag{43}$$

$$\Delta\hat{F}_k := A\left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^{k_0}\right) \tag{44}$$

$$\Delta\hat{G}_k := B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^{k_0}\right) \tag{45}$$

so that the curvatures are estimated via least squares yielding

$$\begin{aligned} \hat{\alpha}_k^{SD} &:= \frac{\langle \Delta\hat{w}_k, \Delta\hat{w}_k\rangle}{\langle \Delta\hat{F}_k, \Delta\hat{w}_k\rangle} \quad \hat{\beta}_k^{SD} := \frac{\langle \Delta\hat{w}_k, \Delta\hat{w}_k\rangle}{\langle \Delta\hat{G}_k, \Delta\hat{w}_k\rangle} \\ \hat{\alpha}_k^{MG} &:= \frac{\langle \Delta\hat{F}_k, \Delta\hat{w}_k\rangle}{\langle \Delta\hat{F}_k, \Delta\hat{F}_k\rangle} \quad \hat{\beta}_k^{MG} := \frac{\langle \Delta\hat{G}_k, \Delta\hat{w}_k\rangle}{\langle \Delta\hat{G}_k, \Delta\hat{G}_k\rangle} \end{aligned} \tag{46}$$

where SD stands for *steepest descent* and MG for *minimum gradient* method. A hybrid stepsize rule is proposed

$$\begin{aligned} \hat{\alpha}_k &:= \begin{cases} \hat{\alpha}_k^{SD} & \text{if } 2\hat{\alpha}_k^{MG} > \hat{\alpha}_k^{SD} \\ \hat{\alpha}_k^{MG} & \text{otherwise} \end{cases} \\ \hat{\beta}_k &:= \begin{cases} \hat{\beta}_k^{SD} & \text{if } 2\hat{\beta}_k^{MG} > \hat{\beta}_k^{SD} \\ \hat{\beta}_k^{MG} & \text{otherwise} \end{cases} \end{aligned} \tag{47}$$

On some iterations, the linear models underlying the spectral stepsize choice may be very inaccurate. Xu et al. propose to safeguard the method by assesing the quality of the curvature estimates, and only updating the stepsize if the curvature estimates satisfy a reliability criterion. To test the validity of this assumption, the correlation between the follow quatintites are measured:

$$\alpha_k^{cor} := \frac{\langle \Delta\hat{F}_k, \Delta\hat{w}_k\rangle}{\|\Delta\hat{F}_k\|\|\Delta\hat{w}_k\|} \wedge \beta_k^{cor} := \frac{\langle \Delta\hat{G}_k, \Delta\hat{w}_k\rangle}{\|\Delta\hat{G}_k\|\|\Delta\hat{w}_k\|} \tag{48}$$

Finally, the safeguard spectral adaptative penalty rule is

$$\rho_{k+1} := \begin{cases} \sqrt{\hat{\alpha}_k\hat{\beta}_k} & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \hat{\alpha}_k & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} \le \epsilon^{cor} \\ \hat{\beta}_k & \text{if } \alpha_k^{cor} \le \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \rho_k & \text{otherwise} \end{cases} \tag{49}$$

where ϵ^{cor} is a quality threshold for the curvature estimates.

5 ADMM improvements

5.1 Relaxed ADMM

A relaxed version of ADMM was proposed in [14]. This method applies Nesterov's type overrelaxation scheme to the updates of $\boldsymbol{y}^k$ and $\boldsymbol{z}^k$ which takes into account past iterations. It is shown that the method converges in the primal and dual residuals with rate $O(1/k^2)$.

$$\alpha_{k+1} := \frac{1}{2}\left(1 + \sqrt{1 + 4\alpha_k^2}\right) \tag{50a}$$

$$\hat{\boldsymbol{y}}^{k+1} := \boldsymbol{y}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k) \tag{50b}$$

$$\hat{\boldsymbol{z}}^{k+1} := \boldsymbol{z}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\boldsymbol{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k}\boldsymbol{z}^k). \tag{50c}$$

5.2 Relaxed + Restart ADMM

For weakly convex problems, we must enforce stability using a restart rule, which often makes the relaxed algorithm more efficient. It is also proposed in [14]. The restart rule relies on a combined resiual, which measures both the primal and dual error simultaneously:

$$e_k := \|\boldsymbol{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k}\hat{\boldsymbol{z}}^k\|^2 + \rho^k\|B(\boldsymbol{y}^{k+1} - \hat{\boldsymbol{y}}^k)\|^2 \tag{51}$$

if $e_k < \eta e_{k-1}$ the relaxed ADMM (50) is performed, otherwise, the restart rule states

$$\alpha_{k+1} := 1 \tag{52a}$$

$$\hat{\boldsymbol{y}}^{k+1} := \boldsymbol{y}^{k+1} \tag{52b}$$

$$\hat{\boldsymbol{z}}^{k+1} := \boldsymbol{z}^{k+1} \tag{52c}$$

$$e_k \leftarrow \frac{e_{k-1}}{\eta} \tag{52d}$$

6 ADMM algorithms

This section presents a summary of the algorithms (or solvers) that arise from section 4 and 5.

6.1 Constant penalty parameter

The relaxed ADMM involves a parameter $\eta \in (0,1)$. It is desirible to restart the method as infrequently as possible, it is recommended a value of η close to 1 [14]. In our case $\eta = 0.999$ was used.

Algorithm 1 ADMM.

```
Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4:     $\boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5:     Stop criterion in (16)
6: end for
```

Algorithm 2 Relaxed ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5:     Stop criterion in (16)
6:     Relaxation in (50)
7: end for
```

Algorithm 3 Relaxed + Restart ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\eta \approx 1$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5:     Stop criterion in (16)
6:     Relaxation + Restart in (50) (52)
7: end for
```

6.2 Varying penalty parameter - He

In Wang et al. the value $\tau = 2$ generally performs well in the ρ update [15].

Algorithm 4 ADMM.

```
Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Update of $\rho$ in (He)
7: end for
```

Algorithm 5 Relaxed ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Relaxation in (50)
7:     Update of $\rho$ in (He)
8: end for
```

Algorithm 6 Relaxed + Restart ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$
1: for $k = 0, 1, 2, \dots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Relaxation + Restart in (50) (52)
7:     Update of $\rho$ in (He)
8: end for
```

6.3 Varying penalty parameter - Wohlberg

In He et al. the value $\tau_{max} = 100$ generally performs well in the ρ update [12].

Algorithm 7 ADMM.

```
Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, $\xi = 1$, $\varphi = 10$, $\tau_{max} = 100$, and $\rho > 0$
1: for $k = 0, 1, 2, \ldots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Update of $\rho$ in (Wohlberg)
7: end for
```

Algorithm 8 Relaxed ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\xi = 1$, $\varphi = 10$, $\tau_{max} = 100$, and $\rho > 0$
1: for $k = 0, 1, 2, \ldots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Relaxation in (50)
7:     Update of $\rho$ in (Wohlberg)
8: end for
```

Algorithm 9 Relaxed + Restart ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $\xi = 1$, $\varphi = 10$, $\tau_{max} = 100$, and $\rho > 0$
1: for $k = 0, 1, 2, \ldots$ do
2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
5:     Stop criterion in (16)
6:     Relaxation + Restart in (50) (52)
7:     Update of $\rho$ in (Wohlberg)
8: end for
```

6.4 Varying penalty parameter - Spectral

Xu et al. suggest only updating the stepsize every T_f iterations. Safeguarding threshold $\epsilon^{cor} = 0.2$ and $T_f = 2$ generally perform well [13].

Algorithm 10 ADMM.

```
Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$
 1: for $k = 0, 1, 2, \dots$ do
 2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
 3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
 4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
 5:     Stop criterion in (16)
 6:     if $mod(k, T_f) = 1$ then
 7:         Update of $\rho$ in (Spectral)
 8:     else
 9:         $\rho^{k+1} \leftarrow \rho^k$
10:     end if
11: end for
```

Algorithm 11 Relaxed ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$
 1: for $k = 0, 1, 2, \dots$ do
 2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
 3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
 4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
 5:     Stop criterion in (16)
 6:     Relaxation in (50)
 7:     if $mod(k, T_f) = 1$ then
 8:         Update of $\rho$ in (Spectral)
 9:     else
10:         $\rho^{k+1} \leftarrow \rho^k$
11:     end if
12: end for
```

Algorithm 12 Relaxed + Restart ADMM.

```
Require: $\boldsymbol{y}^0 = \hat{\boldsymbol{y}}^0$, $\boldsymbol{z}^0 = \hat{\boldsymbol{z}}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$
 1: for $k = 0, 1, 2, \ldots$ do
 2:     $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
 3:     $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
 4:     $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
 5:     Stop criterion in (16)
 6:     Relaxation + Restart in (50) (52)
 7:     if $mod(k, T_f) = 1$ then
 8:         Update of $\rho$ in (Spectral)
 9:     else
10:         $\rho^{k+1} \leftarrow \rho^k$
11:     end if
12: end for
```

7 Comparison

7.1 Measuring errors

Wohlberg in [12] states under his framework that the ADMM method is invariant when $\epsilon^{abs} = 0$. Xu et al. in [13] also use this value which helps to be less sensitive to scaling. However, in our case we set in every algorithm $\epsilon^{abs} = 10^{-6}$, which is a low value (therefore it aims to be less sensitive as possible to scaling) and also enhances the stop criterion if the data size is too small.

7.2 Performance profiles

In this section, following [1] we explain the concept of performance profiles that was introduced in [16] for benchmarking optimization solvers on a large set of problems. 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

$$t_{p,s} = \text{computing time required for } s \text{ to solve } p \text{ at precision } tol \tag{53}$$

A performance ratio over all the solvers is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}, s \in S\}} \geq 1 \tag{54}$$

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

$$\rho_s(\tau) = \frac{1}{n_p} card\{p \in P, r_{p,s} \leq \tau\} \leq 1 \tag{55}$$

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. It is worth noting that $\rho_s(1)$ represents the probability that the solver s beats the other solvers, and $\rho_s(\tau)$ characterizes the robustness of the method for large values of τ. To summarize: 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.

7.3 Nomenclature for the algorithms

The Table 1 shows the notation used in the results:

Name	Algorithm	Additional informations
cp-N	1	-
cp-R	2	-
cp-RR	3	-
vp-N-He	4	-
vp-R-He	5	-
vp-RR-He	6	-
vp-N-Wohlberg	7	-
vp-R-Wohlberg	8	-
vp-RR-Wohlberg	9	-
vp-N-Spectral	10	-
vp-R-Spectral	11	Relaxed terms used in ρ update
vp-RR-Spectral	12	Relaxed terms used in ρ update

Table 1: Nomenclature for the algorithms.

7.4 Results

In this section, we perform a comparison of the algorithms by family of optimal penalty parameter. The goal is to study the influence of the various parameters and possible strategies on the performance of the solvers.

7.4.1 Ghadimi

Figure 1 shows that most of the algorithms are robust but slow. They are able to solve all the problems but they require a lot of time. Wohlberg and Spectral ρ-updates depict the worst behavior. Furthermore, the difference bewteen the constant and varying penalty parameter is only comparable in He ρ-update.

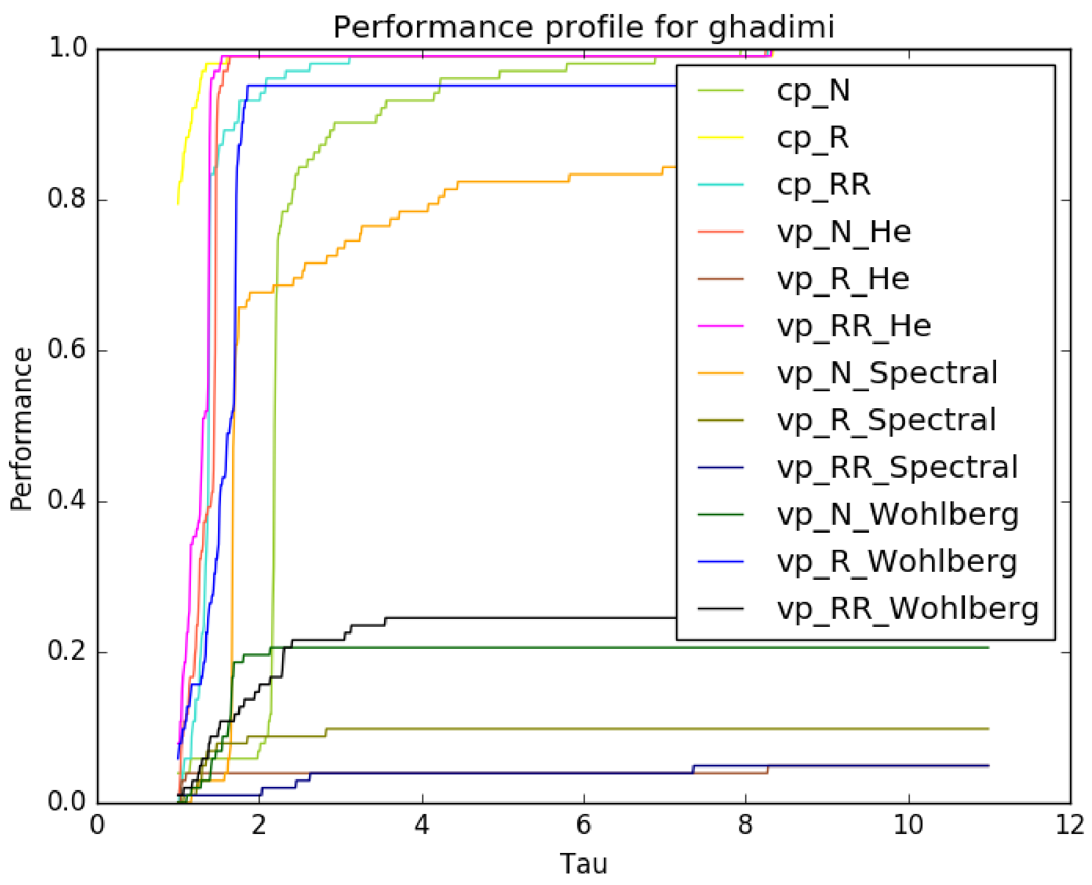

Figure 1: Comparison of algorithms for Ghadimi optimal penalty parameter.

7.4.2 Di Cairano

Figure 2 shows that half of the algorithms suffer of robustness. Wohlberg and Spectral ρ-updates depict the worst behavior. Furthermore, the difference bewteen the constant and varying penalty parameter is highly comparable in He ρ-update.

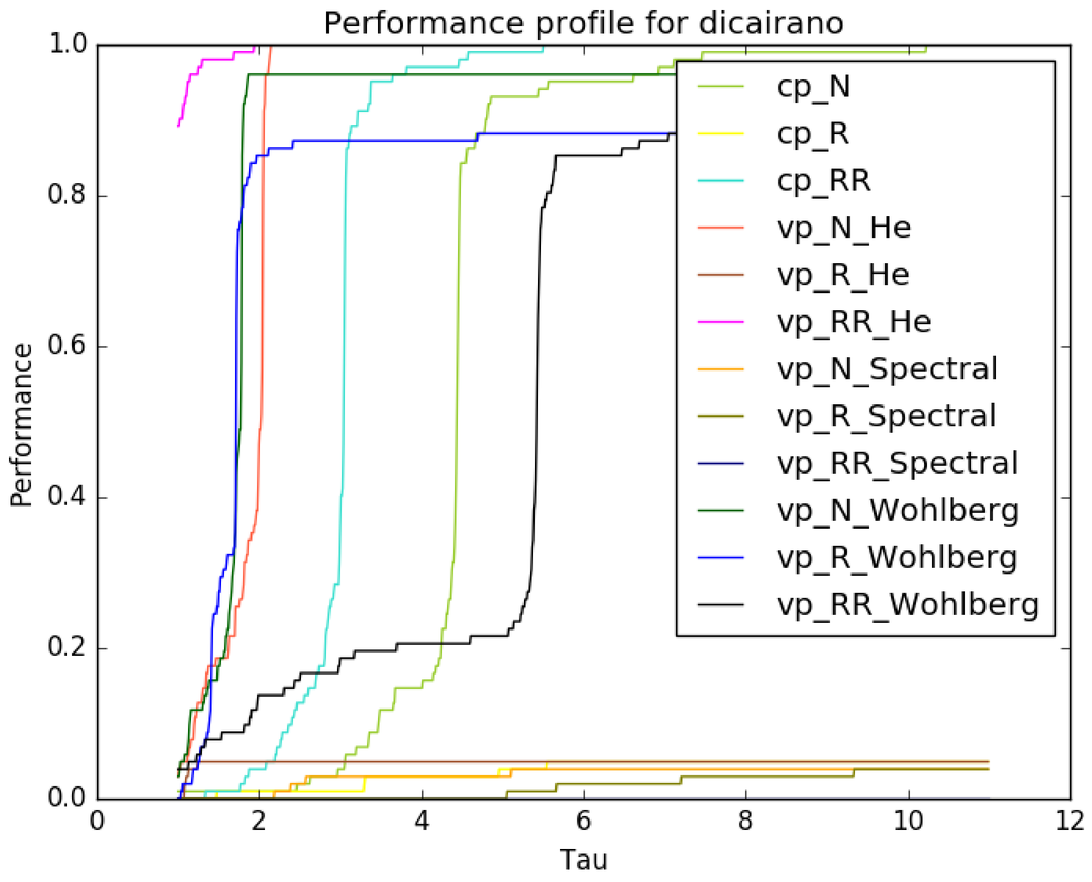

Figure 2: Comparison of algorithms for Di Cairano optimal penalty parameter.

7.4.3 Acary

Figure 2 shows that half of the algorithms suffer of robustness. Wohlberg and Spectral ρ-updates depict the worst behavior. Furthermore, the difference bewteen the constant and varying penalty parameter is only comparable in He ρ-update.

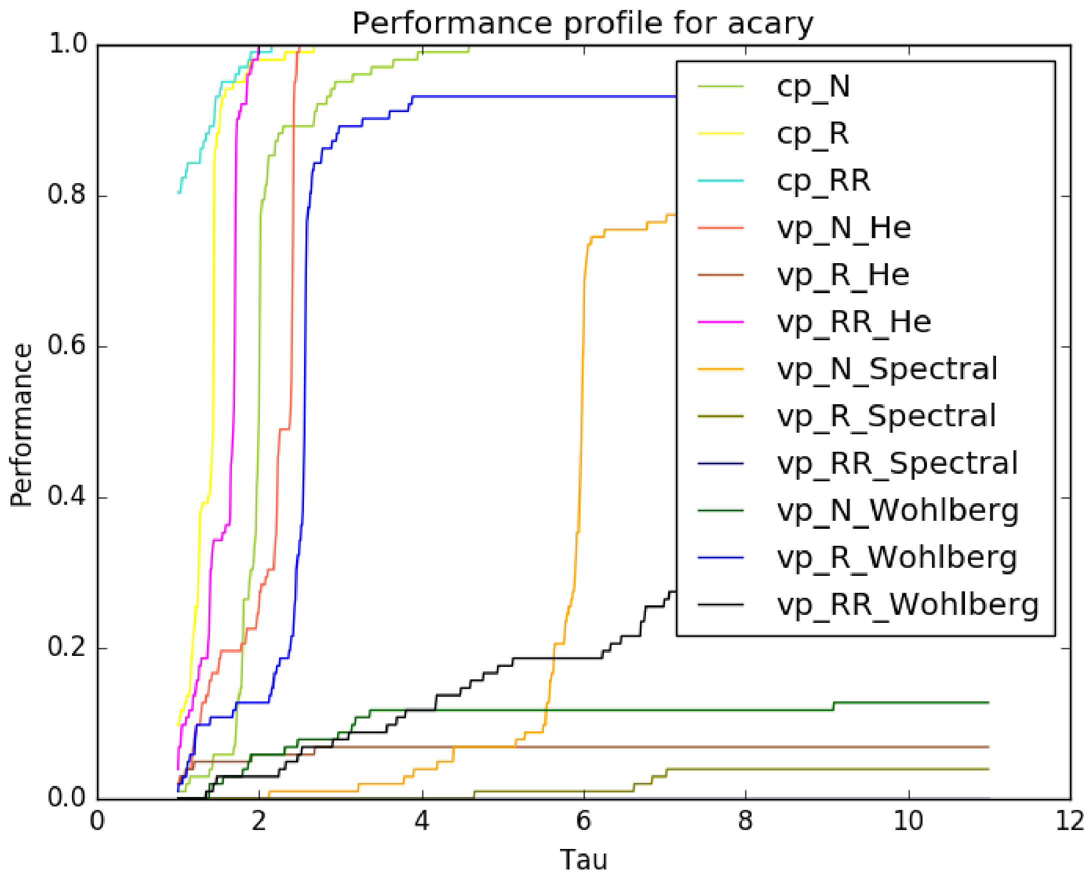

Figure 3: Comparison of algorithms for Acary optimal penalty parameter.

7.4.4 Normal

Figure 4 shows that most of the algorithms suffer of robustness. Wohlberg and Spectral ρ-updates depict the worst behavior. Furthermore, the difference bewteen the constant and varying penalty parameter is highly comparable in He ρ-update.

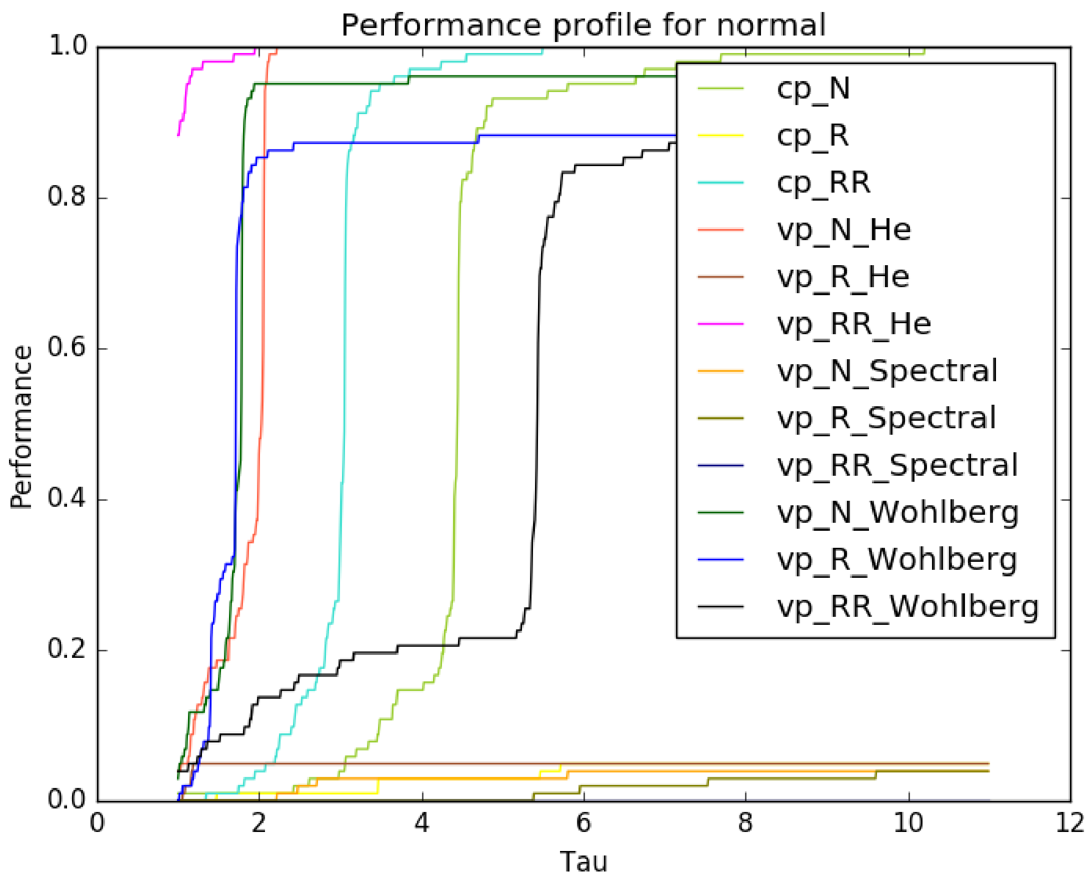

Figure 4: Comparison of algorithms for Normal optimal penalty parameter.

7.5 Discussion

In Table 2 are depicted the best algorithms with their respective optimal penalty parameter. Although their behavior are very close, Di Cairano takes into account the data size of the problem, which makes it a more general solver.

	vp-RR-He (6) (Di Cairano)	vp-RR-He (6) (Normal)
$\rho_s(0)$	0.892	0.882
$\tau_{\rho_s=1}$	1.94	1.95
Total CPU time (s)	45.43	45.32

Table 2: Comparison of best algorithms.

It is worth noting that for the penalty parameter proposed in this paper (Acary), its best performance is with cp-RR, i.e., this value is the closest one to the optimal and there is no need to update it under any scheme.

8 S-update

To solve the friction problem, the parameter s in problem (18) should be updated. One obvious way is that, once we solve problem (18) with fixed s by ADMM, then update s by using the obtained solution, and repeat this procedure. This is what we called *external update.* Another possibility is to update s at each iteration of the ADMM. This is what we called *internal update.* Intuitively, the latter saves the total computational cost, but stability of the algorithm in this is not clear, there is no proof of existence or uniqueness in comparison to the first one [4].

8.1 Initial guess

Another issue is the initial guess of s to improve the rate of convergence. One simple method that performs well in our tests (results not shown) is $\boldsymbol{s}^0 = 0 \in \mathbb{R}^m$.

8.2 Internal update

For the next development, we consider

$$\Phi(\boldsymbol{u}) = [\mu^\alpha \|E_t \boldsymbol{u}^\alpha\| \hat{\boldsymbol{e}}_t, \alpha = 1...n_c]^T = [\mu^\alpha \|E_t \left(H\boldsymbol{v} + \boldsymbol{w}\right)^\alpha \| \hat{\boldsymbol{e}}_t, \alpha = 1...n_c]^T \tag{56}$$

therefore $\Phi(\boldsymbol{u})$ could be seen as $\Phi(\boldsymbol{v})$. Then, the problem (21) is a minimization problem of a convex function under non-linear equality constraints. Its augmented Lagrangian (in the scaled form) is formulated as

$$L_\rho(\boldsymbol{v}; \tilde{\boldsymbol{u}}; \boldsymbol{\zeta}) = \frac{1}{2}\boldsymbol{v}^\top M \boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{v}) + \boldsymbol{\zeta} - \tilde{\boldsymbol{u}}\|^2 - \frac{\rho}{2}\|\boldsymbol{\zeta}\|^2, \tag{57}$$

where ρ is the penalty parameter, and $\boldsymbol{\zeta} \in \mathbb{R}^m$ is the scaled dual variable (the scaled Lagrange multiplier). The ADMM solving problem (20) consists of iterating the updates

$$\boldsymbol{v}^{k+1} := \arg\min_{\boldsymbol{v}} L_\rho(\boldsymbol{v}; \tilde{\boldsymbol{u}}^k; \boldsymbol{\zeta}^k), \tag{58}$$

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} L_\rho(\boldsymbol{v}^{k+1}; \tilde{\boldsymbol{u}}; \boldsymbol{\zeta}^k), \tag{59}$$

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H\boldsymbol{v}^{k+1} - \tilde{\boldsymbol{u}}^{k+1} + \Phi(\boldsymbol{v}^{k+1}). \tag{60}$$

The first step in (58) corresponds to unconstrained minimization of a convex function. It follows from the stationarity condition of this convex function that $\boldsymbol{v}^{k+1}$ is the solution to the following system of linear equations:

$$\left[M + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k+1})\right)H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k+1})\right)(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^{k+1}) - \boldsymbol{\zeta}^k) \tag{61}$$

this non-linearity is difficult to handle, however, if the previous iteration is considered in the $\Phi(\boldsymbol{v})$ term, we get the following equation

$$\left[M + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k})\right)H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k})\right)(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^{k}) - \boldsymbol{\zeta}^k) \tag{62}$$

which follows the same structure of (25) and allows us to perform a LU or Cholesky factorization. If $\|E_t \boldsymbol{u}^\alpha\| \neq 0$

$$\partial_{\boldsymbol{v}} \Phi(\boldsymbol{v}) = \left[\mu^\alpha \hat{\boldsymbol{e}}_t^\top \frac{E_t \left(H\boldsymbol{v} + \boldsymbol{w}\right)^\alpha}{\|E_t \left(H\boldsymbol{v} + \boldsymbol{w}\right)^\alpha \|} E_t^\top H^{\alpha^\top}, \alpha = 1...n_c\right]^T = \Psi \tag{63}$$

Therefore

$$\left[M + \rho \left(H^\top + \Psi\right) H\right] \boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho \left(H^\top + \Psi\right) (\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^k) - \boldsymbol{\zeta}^k). \tag{64}$$

The second step in (59) can be computed independently for each $\alpha = 1...n_c$ as

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{v}^k) + \boldsymbol{\zeta}^k - \tilde{\boldsymbol{u}}\|^2. \tag{65}$$

(65) can be rewritten by using the projection onto the second-order cone as

$$\tilde{\boldsymbol{u}}^{k+1} := \Pi_{K^*_{e,\mu}}(H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \boldsymbol{\zeta}^k + \Phi(\boldsymbol{v}^k)) \tag{66}$$

8.3 External update

In this case, the external error measurement is given for each $\alpha = 1...n_c$ by

$$\left[\frac{\Phi(\boldsymbol{v}^{k+1}) - \Phi(\boldsymbol{v}^k)}{\Phi(\boldsymbol{v}^k)}\right]^\alpha \leq \epsilon^{ext} \tag{67}$$

where ϵ^{ext} is the external tolerance, with a value 10^{-3}.

8.4 Comparison

In most of the test examples the behavior dispicted in Figure 5-7 is observed. This is analyzed with the best solver of Section 7 and the test example *Box Stacks-i1000-352-6*.

In Figure 5 is observed a convergence after 7 external iterations. Note that the first value is not shown because is zero.

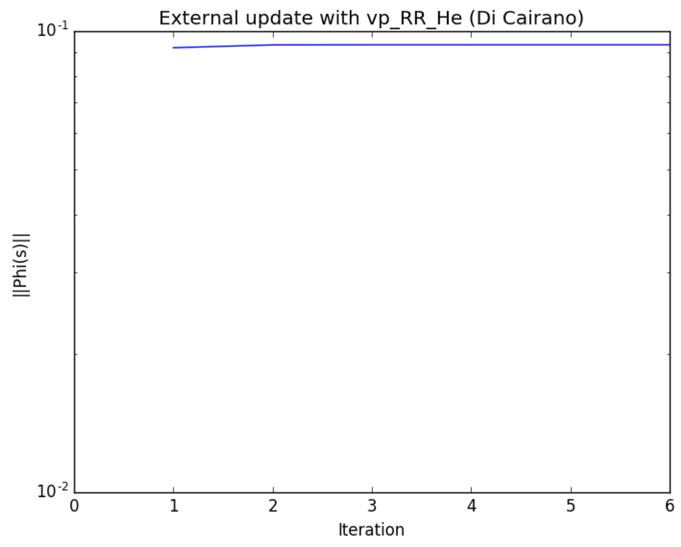

Figure 5: External update with vp-RR-He (Di Cairano).

In Figure 6 is observed a convergence after 43 internal iterations. From iteration 25 the same value holds, but the stop criterion (16) is not satisfied.

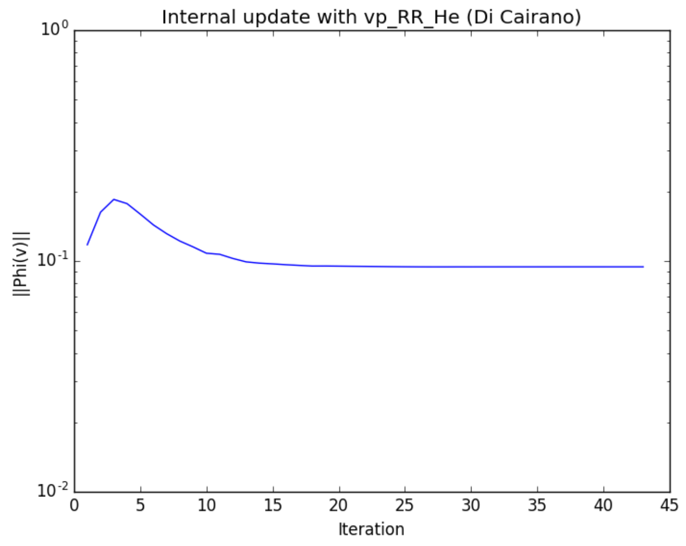

Figure 6: Internal update with vp-RR-He (Di Cairano).

In Figure 7 is despicted the same internal update, but the stop criterion (16) is changed for the external one (67). It is observed that $\|\Phi(\boldsymbol{v})\|$ holds from iteration 25, but $\Phi(\boldsymbol{v})^{\alpha}$ is not the same from iteration to iteration, the test was stopped at 1000 iterations.

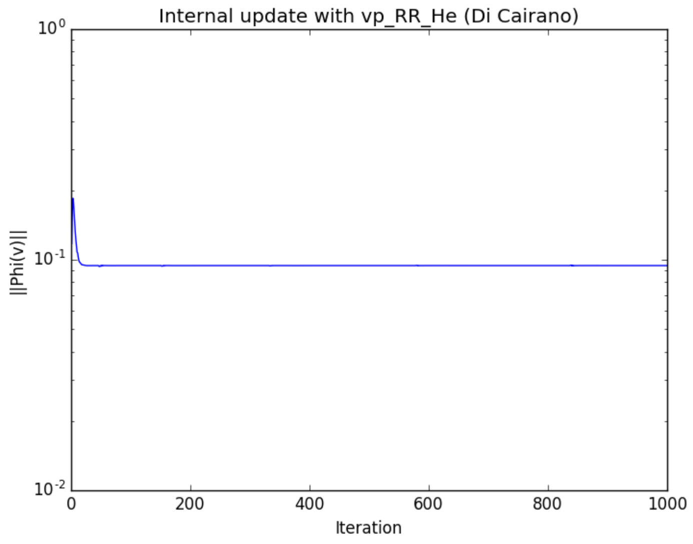

Figure 7: Internal update with vp-RR-He (Di Cairano) and stop criterion in (67).

In Table 3 are shown the values of the final $\|\Phi(s)\|$ with its respectives first and last contact vector $\Phi(s)^{\alpha}$:

	External	Internal	Internal-s
$\|\Phi(s)\|$	9.34e-02	9.43e-02	9.43e-02
$\Phi(s)^{\alpha=1}$	[0.00e+00 5.19e-06 5.19e-06]	[0.00e+00 5.67e-06 5.67e-06]	[0.00e+00 3.89e-06 3.89e-06]
$\Phi(s)^{\alpha=n_c}$	[0.00e+00 5.02e-04 5.02e-04]	[0.00e+00 4.78e-04 4.78e-04]	[0.00e+00 4.73e-04 4.73e-04]

Table 3: Comparison of s-update

Both external and internal update converges to almost the same value of $\|\Phi(s)\|$, but this is an apparent result of uniqueness when $\Phi(s)^{\alpha}$ is compared in each case. Therefore, the internal update does not guarantee the same result for $\Phi(s)$.

References

[1] Vincent Acary, Maurice Brémond, and Olivier Huber. On solving contact problems with Coulomb friction: formulations and numerical comparisons. Research Report RR-9118, INRIA, November 2017.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[3] Vincent Acary and Florent Cadoux. *Applications of an Existence Result for the Coulomb Friction Problem*, pages 45–66. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[4] Vincent Acary, Florent Cadoux, Claude Lemaréchal, and Jérôme Malick. A formulation of the linear discrete coulomb friction problem via convex optimization. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 91(2):155–175, 2011.

[5] Shunsuke Hayashi, Nobuo Yamashita, and Masao Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization*, 15(2):593–615, 2005.

[6] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on optimization*, 12(2):436–460, 2002.

[7] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

[8] Jonathan Eckstein and W Yao. Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32:3, 2012.

[9] Euhanna Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.

[10] Arvind U Raghunathan and Stefano Di Cairano. Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection. In *American Control Conference (ACC), 2014*, pages 4324–4329. IEEE, 2014.

[11] B. S. He, H. Yang, and S. L. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, Aug 2000.

[12] Brendt Wohlberg. Admm penalty parameter selection by residual balancing. *arXiv preprint arXiv:1704.06209*, 2017.

[13] Zheng Xu, Mário AT Figueiredo, and Tom Goldstein. Adaptive admm with spectral penalty parameter selection. *arXiv preprint arXiv:1605.07246*, 2016.

[14] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[15] S. L. Wang and L. Z. Liao. Decomposition method with a variable parameter for a class of monotone variational inequality problems. *Journal of Optimization Theory and Applications*, 109(2):415–429, May 2001.

[16] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, Jan 2002.