Alternating Direction Method of Multipliers (ADMM) for Frictional Contact

Vincent Acary, Yoshiro Kanno & Nicolas Molina

March 28, 2018

Outline

Outline

Finite-dimensional frictional contact problem

The problem in [1] reads:

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

The problem data are:

- $M \in \mathbb{R}^{n\times n}$, a positive definite matrix
- $f \in \mathbb{R}^n$
- $H \in \mathbb{R}^{n\times m}$, with $m = 3n_c$,
- $w \in \mathbb{R}^m$
- $\mu \in \mathbb{R}^{n_c}$, a vector of coefficients of friction

The unknowns are two vectors:

- $v \in \mathbb{R}^n$, a velocity-like vector
- $r \in \mathbb{R}^m$, a contact reaction or impulse

Finite-dimensional frictional contact problem

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

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

where $\hat{\boldsymbol{e}}_t = [1, 0, 0]^{\top}$ is the normal 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 $\tilde{\boldsymbol{u}}$.

Outline

Fundamentals

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 $\boldsymbol{y} \in \mathbb{R}^m$:

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

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.

Fundamentals

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

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

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

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

where the dual variable is updated via dual ascent method.

Fundamentals

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

$$L_\rho(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \frac{\rho}{2}\|A\boldsymbol{x} + B\boldsymbol{y} - \boldsymbol{c} + \boldsymbol{z}\|^2 - \frac{\rho}{2}\|\boldsymbol{z}\|^2. \tag{10}$$

Using L_ρ in (10), we can express ADMM as follows:

$$\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k), \tag{11}$$

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

$$\boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c} \tag{13}$$

The expression in (11), (12), and (13) is called the ADMM in the scaled form, and $\boldsymbol{z}$ is called the scaled dual variable.

Stop criterion

The residuals of the optimality conditions can be related to an approximated bound on the KKT conditions (primal and dual feasibility) of the current point

$$\begin{aligned} \|\boldsymbol{r}^k\| &\leq \epsilon^{pri} \\ \|\boldsymbol{s}^k\| &\leq \epsilon^{dual} \end{aligned} \tag{14}$$

where $\boldsymbol{r}^{k+1} = A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$ and $\boldsymbol{s}^{k+1} = \rho A^T B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right)$ are the primal and dual residuals, respectively; and $\epsilon^{pri} > 0$ 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

$$\begin{aligned} \epsilon^{pri} &= \sqrt{l}\epsilon^{abs} + \epsilon^{rel} \max\{\|A\boldsymbol{x}^{k+1}\|, \|B\boldsymbol{y}^{k+1}\|, \|\boldsymbol{c}\|\} \\ \epsilon^{dual} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|A^T\rho\boldsymbol{z}^{k+1}\| \end{aligned} \tag{15}$$

where $\epsilon^{abs} > 0$ and $\epsilon^{dual} > 0$ is an absolute and relative tolerance, respectively. This values typically depend on the scale of the data size.

Outline

Coulomb friction problem via convex optimization

In [2], a parametric convex optimization formulation is presented for the dynamical Coulomb friction problem. The problem reads

$$\underset{\boldsymbol{v},\tilde{\boldsymbol{u}}}{\text{Minimize}} \quad \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} \tag{16a}$$

$$\text{subject to} \quad \tilde{\boldsymbol{u}} = H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}) \tag{16b}$$

$$\tilde{\boldsymbol{u}} \in K^*_{e,\mu} \tag{16c}$$

where $\boldsymbol{s} \in \mathbb{R}^m$ is a parameter vector and $\Phi(\boldsymbol{s}) = [\mu^\alpha \| E_t \boldsymbol{s}^\alpha \| \hat{e}_n, \alpha = 1...n_c]^\top$.

Coulomb friction problem via convex optimization

Let $\delta_{K^*_{e,\mu}} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ denote the indicator function of $K^*_{e,\mu}$, i.e.,

$$\delta_{K^*_{e,\mu}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in K^*_{e,\mu} \\ +\infty & \text{otherwise.} \end{cases} \tag{17}$$

Then problem (16) is equivalently rewritten as follows:

$$\underset{\boldsymbol{v},\tilde{\boldsymbol{u}}}{\text{Minimize}} \quad \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) \tag{18a}$$

$$\text{subject to} \quad \tilde{\boldsymbol{u}} = H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}) \tag{18b}$$

Iterations

Problem (18) is a minimization problem of a convex function under linear equality constraints. Its augmented Lagrangian (in the scaled form) is formulated as

$$L_\rho(\boldsymbol{v};\tilde{\boldsymbol{u}};\boldsymbol{\zeta}) = \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top\boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta} - \tilde{\boldsymbol{u}}\|^2 - \frac{\rho}{2}\|\boldsymbol{\zeta}\|^2 \tag{19}$$

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 (18) consists of iterating the updates (see (11), (12), and (13))

$$\boldsymbol{v}^{k+1} := \arg\min_{\boldsymbol{v}} L_\rho(\boldsymbol{v};\tilde{\boldsymbol{u}}^k;\boldsymbol{\zeta}^k) \tag{20}$$

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} L_\rho(\boldsymbol{v}^{k+1};\tilde{\boldsymbol{u}};\boldsymbol{\zeta}^k) \tag{21}$$

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{s}) - \tilde{\boldsymbol{u}}^{k+1} \tag{22}$$

Iterations

- v-update

The first step in (20) corresponds to the solution to the following linear system:

$$\left[M + \rho H^\top H\right] \boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho H^\top (\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{s}) - \boldsymbol{\zeta}^k) \tag{23}$$

- $\tilde{u}$-update

The second step in (21) can be computed independently for each $\alpha = 1...n_c$ as

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2} \| H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta}^k - \tilde{\boldsymbol{u}} \|^2 \tag{24}$$

which is identified as the proximity operator or the Moreau-Yosida regularization of $\delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}})$ and this can be written explicitly by using the projection onto the second-order cone as

$$\tilde{\boldsymbol{u}}^{k+1} := \Pi_{K^*_{e,\mu}}(H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{s}) + \boldsymbol{\zeta}^k) \tag{25}$$

Projection onto dual second-order cone

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

$$\Pi_{K^*_{e,\mu}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } -\boldsymbol{x} \in K_{e,\mu} \\ \boldsymbol{x} & \text{if } \boldsymbol{x} \in K^*_{e,\mu} \\ \dfrac{\mu^2\left(x_1 + \frac{1}{\mu}\|\boldsymbol{x}_2\|\right)}{1+\mu^2}\begin{bmatrix} 1 \\ \frac{1}{\mu}\boldsymbol{x}_2/\|\boldsymbol{x}_2\| \end{bmatrix} & \text{if } -\boldsymbol{x} \notin K_{e,\mu} \wedge \boldsymbol{x} \notin K^*_{e,\mu}, \end{cases} \tag{26}$$

- Finally, it is worthy to note that for (23) we carry out a Cholesky or LU factorization and (25) can be solved explicitly by using the formulan given above.

Outline

Optimal penalty parameter selection

- Ghadimi
- Di Cairano
- Acary
- Normal

Optimal penalty parameter selection

- Ghadimi

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

$$\rho^\star := \left(\sqrt{\lambda_{min}(HM^{-1}H^\top)\lambda_{max}(HM^{-1}H^\top)} \right)^{-1} \tag{27}$$

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 [3]. It is worth noting that the matrix $HM^{-1}H^\top$ is called *Delassus Matrix* and is denoted by W.

- Di Cairano

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

$$\rho^\star := \sqrt{\lambda_{min}(M)\lambda_{max}(M)} \tag{28}$$

In this case, the nullity of H equal to zero is avoided [4].

Optimal penalty parameter selection

- 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 (23)

$$\rho^\star := \frac{\|M\|_1}{\|H\|_1} \tag{29}$$

- Normal

Finally, a less expensive and standard method is to set

$$\rho^\star := 1 \tag{30}$$

Varying penalty parameter

- He
- Wohlberg
- Spectral

Varying penalty parameter

- He

He et al. [5] 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

$$\rho^{k+1} := \begin{cases} \tau\rho^k & \text{if } \|\boldsymbol{r}^k\| > \varphi\|\boldsymbol{s}^k\| \\ \tau^{-1}\rho^k & \text{if } \|\boldsymbol{s}^k\| > \varphi\|\boldsymbol{r}^k\| \\ \rho^k & \text{otherwise,} \end{cases} \tag{31}$$

where $\tau > 1$ and $\varphi > 1$. This method appears to becoming quite popular in the literature (see [6] for references).

Varying penalty parameter

- Wohlberg

The behavior of ADMM under scaling is addresed by Wohlberg [6]. 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:

$$\rho^{k+1} := \begin{cases} \tau\rho^k & \text{if } \|\boldsymbol{r}_{rel}^k\| > \xi\varphi\|\boldsymbol{s}_{rel}^k\| \\ \tau^{-1}\rho^k & \text{if } \|\boldsymbol{s}_{rel}^k\| > \xi\varphi\|\boldsymbol{r}_{rel}^k\| \\ \rho^k & \text{otherwise,} \end{cases} \tag{32}$$

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

$$\begin{aligned} \boldsymbol{r}_{rel}^{k+1} &:= \frac{A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}}{\max\{\|A\boldsymbol{x}^{k+1}\|, \|B\boldsymbol{y}^{k+1}\|, \|\boldsymbol{c}\|\}} \\ \boldsymbol{s}_{rel}^{k+1} &:= \frac{\rho A^T B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right)}{\|A^T\boldsymbol{w}^{k+1}\|} = \frac{A^T B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right)}{\|A^T\boldsymbol{\zeta}^{k+1}\|} \end{aligned} \tag{33}$$

Varying penalty parameter

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

$$\tau^{k+1} := \begin{cases} \sqrt{\xi^{-1} \frac{\|\boldsymbol{r}_{rel}^k\|}{\|\boldsymbol{s}_{rel}^k\|}} & \text{if } 1 \leq \sqrt{\xi^{-1} \frac{\|\boldsymbol{r}_{rel}^k\|}{\|\boldsymbol{s}_{rel}^k\|}} < \tau_{max} \\ \sqrt{\xi \frac{\|\boldsymbol{s}_{rel}^k\|}{\|\boldsymbol{r}_{rel}^k\|}} & \text{if } \tau_{max}^{-1} < \sqrt{\xi^{-1} \frac{\|\boldsymbol{r}_{rel}^k\|}{\|\boldsymbol{s}_{rel}^k\|}} < 1 \\ \tau_{max} & \text{otherwise} \end{cases} \tag{34}$$

where τ_{max} provides a bound on τ, i.e., the convergence results in [5] still hold for this extension.

Varying penalty parameter

- Spectral

Xu et al. [7] propose to use a 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 $k_0 < k$

$$\boldsymbol{z}_S^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^k - \boldsymbol{c}\right) \tag{35}$$

$$\Delta\hat{w}_k := \rho^k \boldsymbol{z}_S^{k+1} - \rho^{k_0}\boldsymbol{z}_S^{k_0} \tag{36}$$

$$\Delta\hat{F}_k := A\left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^{k_0}\right) \tag{37}$$

$$\Delta\hat{G}_k := B\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^{k_0}\right) \tag{38}$$

Varying penalty parameter

The curvatures are estimated via least squares yielding

$$\begin{aligned} \hat{\alpha}_k^{SD} &:= \frac{\langle \Delta \hat{w}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle} \qquad \hat{\beta}_k^{SD} := \frac{\langle \Delta \hat{w}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle} \\ \hat{\alpha}_k^{MG} &:= \frac{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{F}_k, \Delta \hat{F}_k \rangle} \qquad \hat{\beta}_k^{MG} := \frac{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{G}_k, \Delta \hat{G}_k \rangle} \end{aligned} \tag{39}$$

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

$$\begin{aligned} \hat{\alpha}_k &:= \begin{cases} \hat{\alpha}_k^{SD} & \text{if } 2\hat{\alpha}_k^{MG} > \hat{\alpha}_k^{SD} \\ \hat{\alpha}_k^{MG} & \text{otherwise} \end{cases} \\ \hat{\beta}_k &:= \begin{cases} \hat{\beta}_k^{SD} & \text{if } 2\hat{\beta}_k^{MG} > \hat{\beta}_k^{SD} \\ \hat{\beta}_k^{MG} & \text{otherwise} \end{cases} \end{aligned} \tag{40}$$

Varying penalty parameter

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:

$$\alpha_k^{cor} := \frac{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle}{\|\Delta \hat{F}_k\| \|\Delta \hat{w}_k\|} \wedge \beta_k^{cor} := \frac{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle}{\|\Delta \hat{G}_k\| \|\Delta \hat{w}_k\|} \tag{41}$$

Finally, the safeguard spectral adaptative penalty rule is

$$\rho_{k+1} := \begin{cases} \sqrt{\hat{\alpha}_k \hat{\beta}_k} & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \hat{\alpha}_k & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} \leq \epsilon^{cor} \\ \hat{\beta}_k & \text{if } \alpha_k^{cor} \leq \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \rho_k & \text{otherwise} \end{cases} \tag{42}$$

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

Outline

Relaxed ADMM

A relaxed version of ADMM was proposed in [8]. 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 $O(1/k^2)$.

$$\alpha_{k+1} := \frac{1}{2}\left(1 + \sqrt{1 + 4\alpha_k^2}\right) \tag{43a}$$

$$\hat{\boldsymbol{y}}^{k+1} := \boldsymbol{y}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k) \tag{43b}$$

$$\hat{\boldsymbol{z}}^{k+1} := \boldsymbol{z}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\boldsymbol{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k}\boldsymbol{z}^k). \tag{43c}$$

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 [8]. The restart rule relies on a combined resiual, which measures both the primal and dual error simultaneously:

$$e_k := \rho^k \|\boldsymbol{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k}\hat{\boldsymbol{z}}^k\|^2 + \rho^k \|B(\boldsymbol{y}^{k+1} - \hat{\boldsymbol{y}}^k)\|^2 \tag{44}$$

if $e_k < \eta e_{k-1}$ the relaxed ADMM (43) is performed, otherwise, the restart rule states

$$\alpha_{k+1} := 1 \tag{45a}$$

$$\hat{\boldsymbol{y}}^{k+1} := \boldsymbol{y}^{k+1} \tag{45b}$$

$$\hat{\boldsymbol{z}}^{k+1} := \boldsymbol{z}^{k+1} \tag{45c}$$

$$e_k \leftarrow \frac{e_{k-1}}{\eta} \tag{45d}$$

Outline

Performance profiles

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

$$t_{p,s} = \text{computing time required for } s \text{ to solve } p \text{ at precision } tol \tag{46}$$

A performance ratio over all the solvers is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}, s \in S\}} \geq 1 \tag{47}$$

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

$$\rho_s(\tau) = \frac{1}{n_p} card\{p \in P, r_{p,s} \leq \tau\} \leq 1 \tag{48}$$

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: Nomenclature for the algorithms.

Set of examples

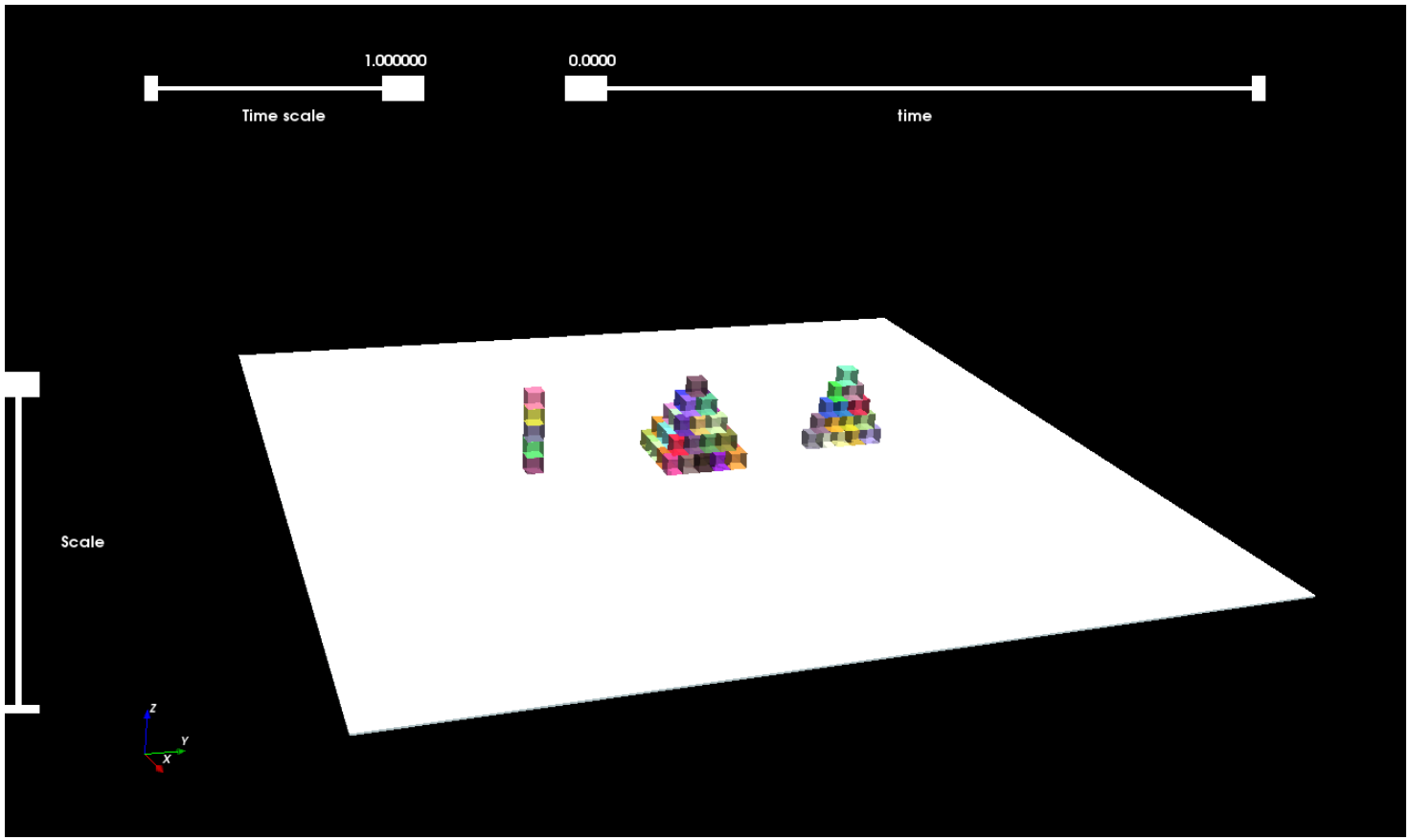

Figure: Box Stacks, set of examples in the comparison.

Outline

Ghadimi

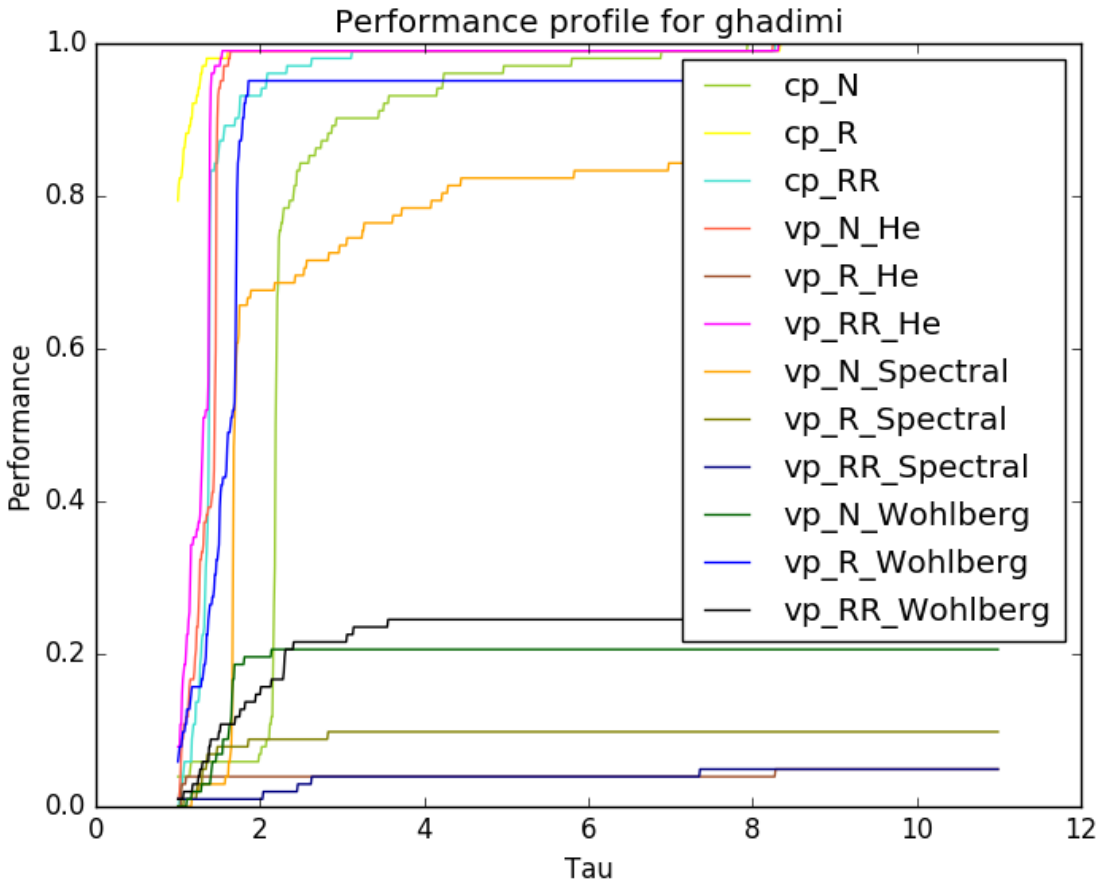

Figure: Comparison of algorithms for Ghadimi optimal penalty parameter.

Di Cairano

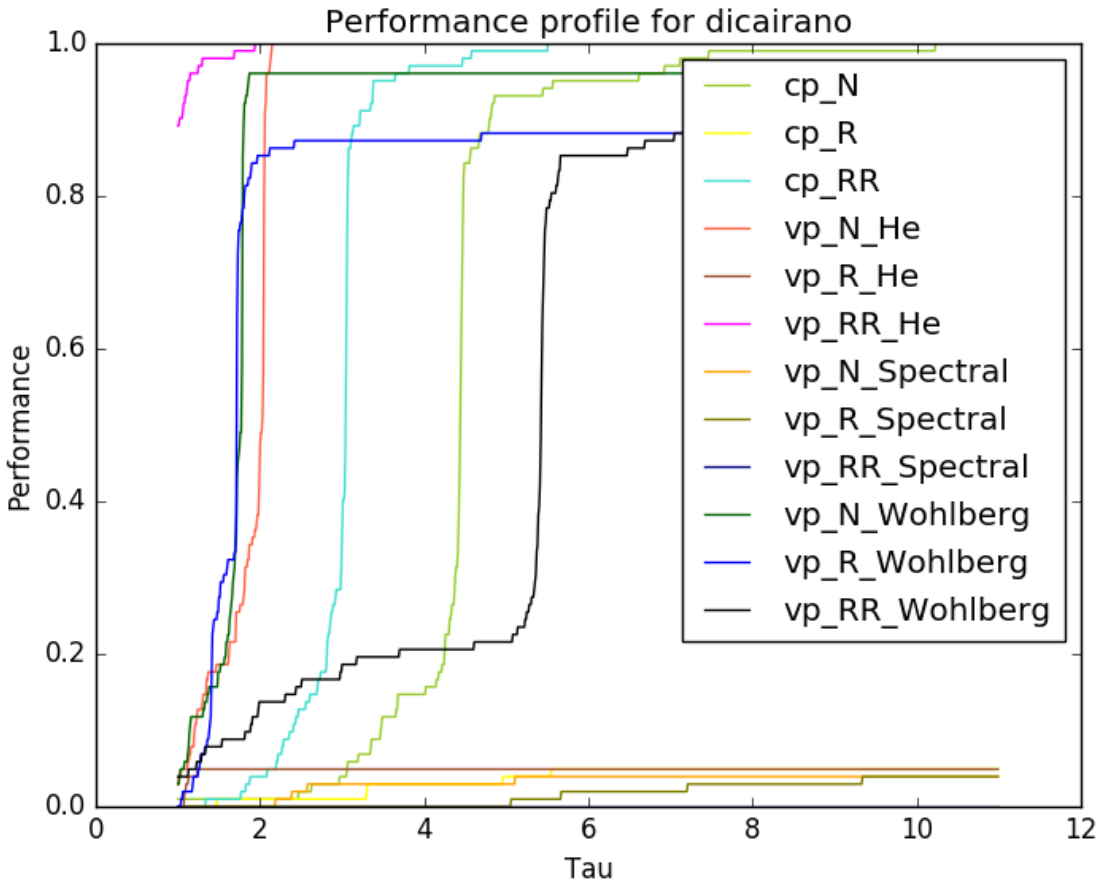

Figure: Comparison of algorithms for Di Cairano optimal penalty parameter.

Acary

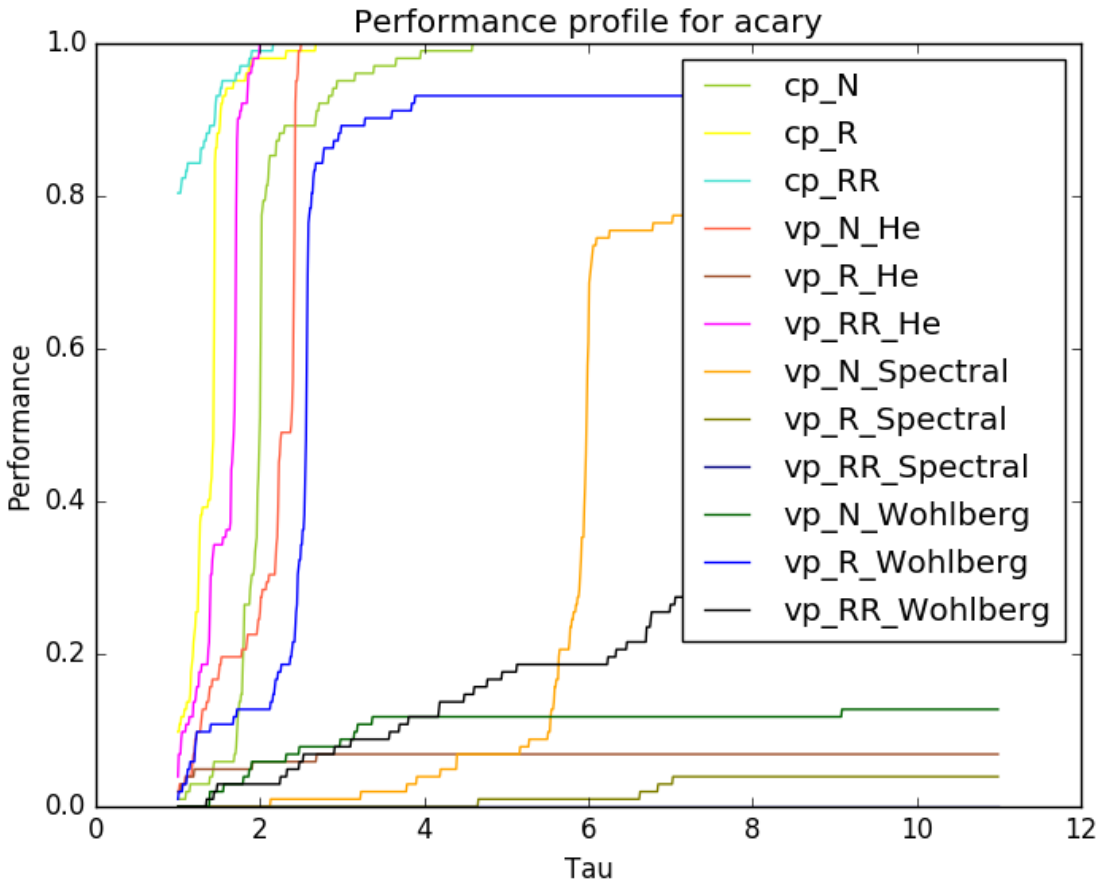

Figure: Comparison of algorithms for Acary optimal penalty parameter.

Normal

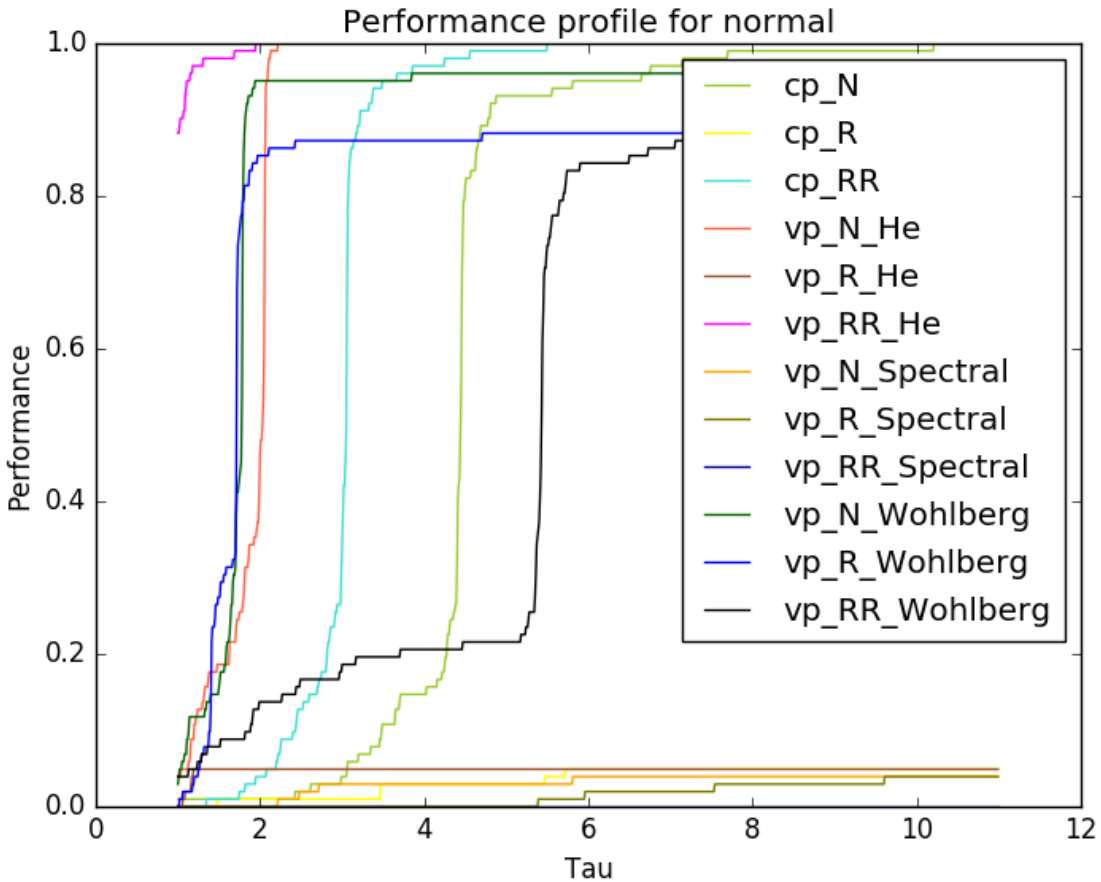

Figure: Comparison of algorithms for Normal optimal penalty parameter.

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: 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.

Outline

S-update

- To solve the friction problem, the parameter s in problem (16) should be updated. One obvious way is that, once we solve problem (16) 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 [2].
- 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 $\boldsymbol{s}^0 = 0 \in \mathbb{R}^m$.

Internal update

For the next development, we consider

$$\Phi(\boldsymbol{u}) = [\mu^{\alpha} \| E_t \boldsymbol{u}^{\alpha} \| \hat{\boldsymbol{e}}_n, \alpha = 1...n_c]^T = [\mu^{\alpha} \| E_t \left(H\boldsymbol{v} + \boldsymbol{w}\right)^{\alpha} \| \hat{\boldsymbol{e}}_t, \alpha = 1...n_c]^T \tag{49}$$

therefore $\Phi(\boldsymbol{u})$ could be seen as $\Phi(\boldsymbol{v})$. Then, the problem (19) is a minimization problem of a convex function under non-linear equality constraints. Its augmented Lagrangian is formulated as

$$L_\rho(\boldsymbol{v}; \tilde{\boldsymbol{u}}; \boldsymbol{\zeta}) = \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} + \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v} + \boldsymbol{w} + \Phi(\boldsymbol{v}) + \boldsymbol{\zeta} - \tilde{\boldsymbol{u}}\|^2 - \frac{\rho}{2}\|\boldsymbol{\zeta}\|^2 \tag{50}$$

where ρ is the penalty parameter, and $\boldsymbol{\zeta} \in \mathbb{R}^m$ is the scaled dual variable. The ADMM solving problem (18) consists of iterating the updates

$$\boldsymbol{v}^{k+1} := \arg\min_{\boldsymbol{v}} L_\rho(\boldsymbol{v}; \tilde{\boldsymbol{u}}^k; \boldsymbol{\zeta}^k) \tag{51}$$

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} L_\rho(\boldsymbol{v}^{k+1}; \tilde{\boldsymbol{u}}; \boldsymbol{\zeta}^k) \tag{52}$$

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H\boldsymbol{v}^{k+1} - \tilde{\boldsymbol{u}}^{k+1} + \Phi(\boldsymbol{v}^{k+1}) \tag{53}$$

Internal update

- v-update

The first step in (51) 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:

$$\left[M + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k+1})\right)H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k+1})\right)(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^{k+1}) - \boldsymbol{\zeta}^k) \tag{54}$$

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

$$\left[M + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k})\right)H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho\left(H^\top + \partial_{\boldsymbol{v}}\Phi(\boldsymbol{v}^{k})\right)(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^{k}) - \boldsymbol{\zeta}^k) \tag{55}$$

which follows the same structure of (23) and allows us to perform a LU or Cholesky factorization. If $\|E_t\boldsymbol{u}^\alpha\| \neq 0$

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

Internal update

Therefore

$$\left[M + \rho\left(H^\top + \Psi\right)H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho\left(H^\top + \Psi\right)(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^k) - \boldsymbol{\zeta}^k) \tag{57}$$

Note that if in (51) it is evaluated $\Phi(\boldsymbol{v})$ in the previous iteration before compute the stationary condition, it results in

$$\left[M + \rho H^\top H\right]\boldsymbol{v}^{k+1} = -\boldsymbol{f} + \rho H^\top(\tilde{\boldsymbol{u}}^k - \boldsymbol{w} - \Phi(\boldsymbol{v}^k) - \boldsymbol{\zeta}^k) \tag{58}$$

which is a pseudo-update of (51) in comparison with (57), but it aims to be faster whitout the compuation of Ψ, therefore, (58) will be used.

- $\tilde{u}$-update

The second step in (52) can be computed independently for each $\alpha = 1...n_c$ as

$$\tilde{\boldsymbol{u}}^{k+1} := \arg\min_{\tilde{\boldsymbol{u}}} \delta_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}}) + \frac{\rho}{2}\|H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \Phi(\boldsymbol{v}^k) + \boldsymbol{\zeta}^k - \tilde{\boldsymbol{u}}\|^2 \tag{59}$$

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

$$\tilde{\boldsymbol{u}}^{k+1} := \Pi_{K^*_{e,\mu}}(H\boldsymbol{v}^{k+1} + \boldsymbol{w} + \boldsymbol{\zeta}^k + \Phi(\boldsymbol{v}^{k+1})) \tag{60}$$

S-update

- External update

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

$$\left[\frac{\Phi(\boldsymbol{v}^{k+1}) - \Phi(\boldsymbol{v}^{k})}{\Phi(\boldsymbol{v}^{k})}\right]^{\alpha} \leq \epsilon^{ext} \tag{61}$$

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

- SICONOS error

The primal residual measures the error in the constraint in (1), but also the error could be measured from the dynamic equation and the natural map (**??**) as

$$\epsilon^{SIC} = \sqrt{\|M\boldsymbol{v} + \boldsymbol{f} - H^{\top}\boldsymbol{r}\|^2 + \|\tilde{\boldsymbol{u}} - \Pi_{K^*_{e,\mu}}(\tilde{\boldsymbol{u}} - \rho\boldsymbol{r})\|^2} \tag{62}$$

Outline

External S-update

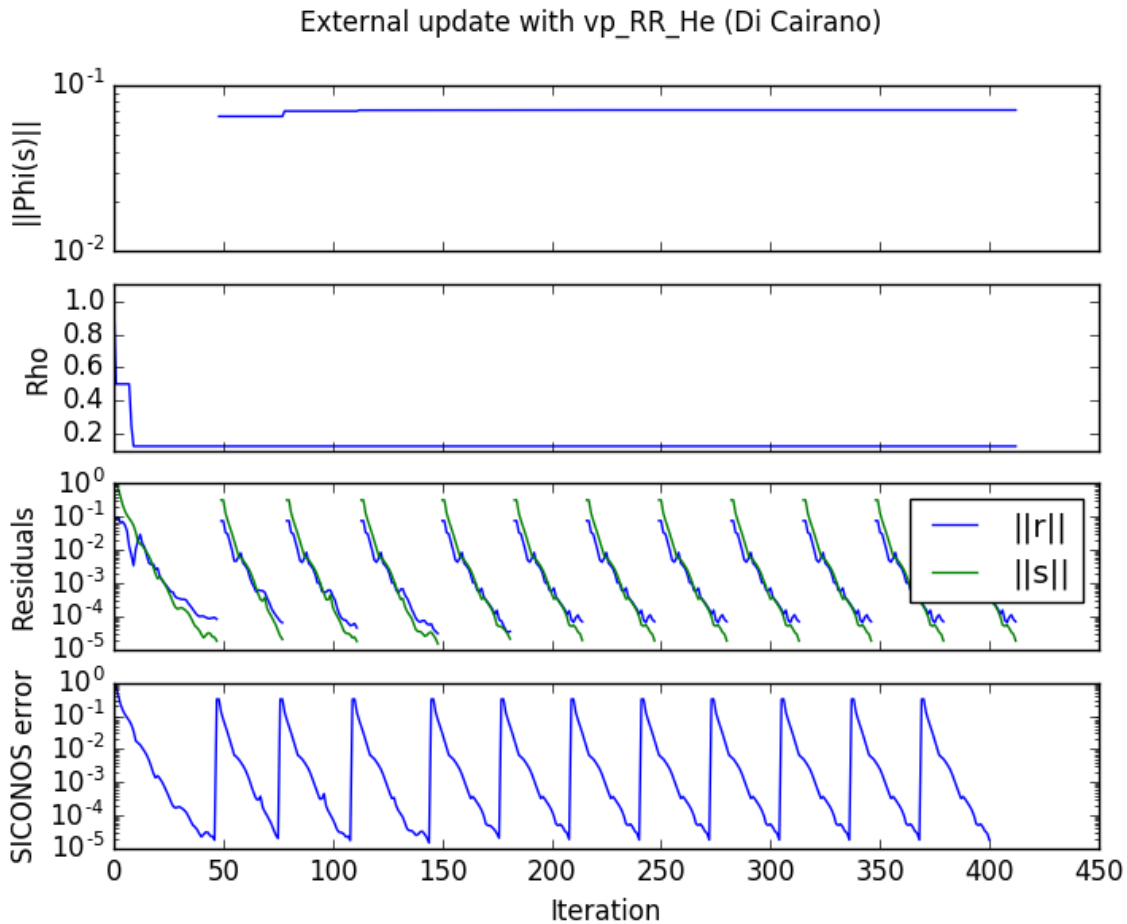

Figure: External update with vp-RR-He (Di Cairano).

Internal S-update

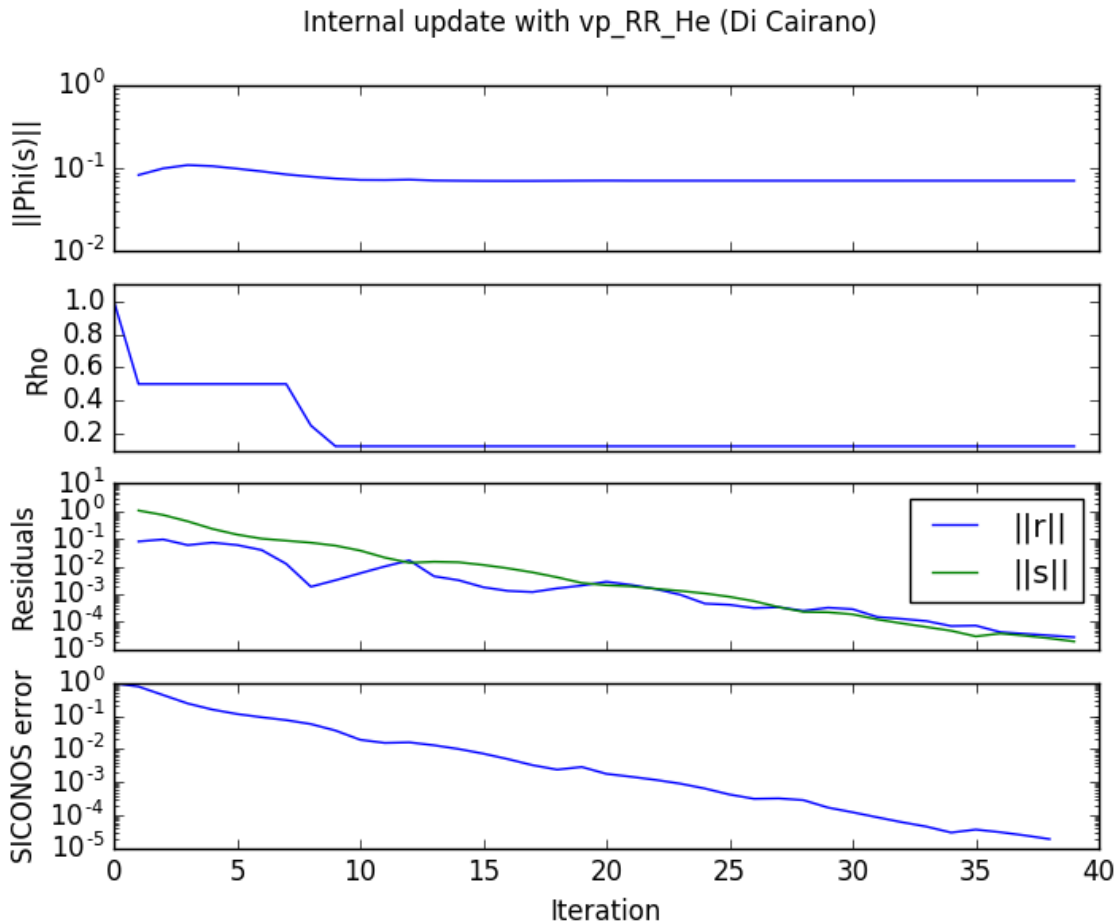

Figure: Internal update with vp-RR-He (Di Cairano).

Internal S-update

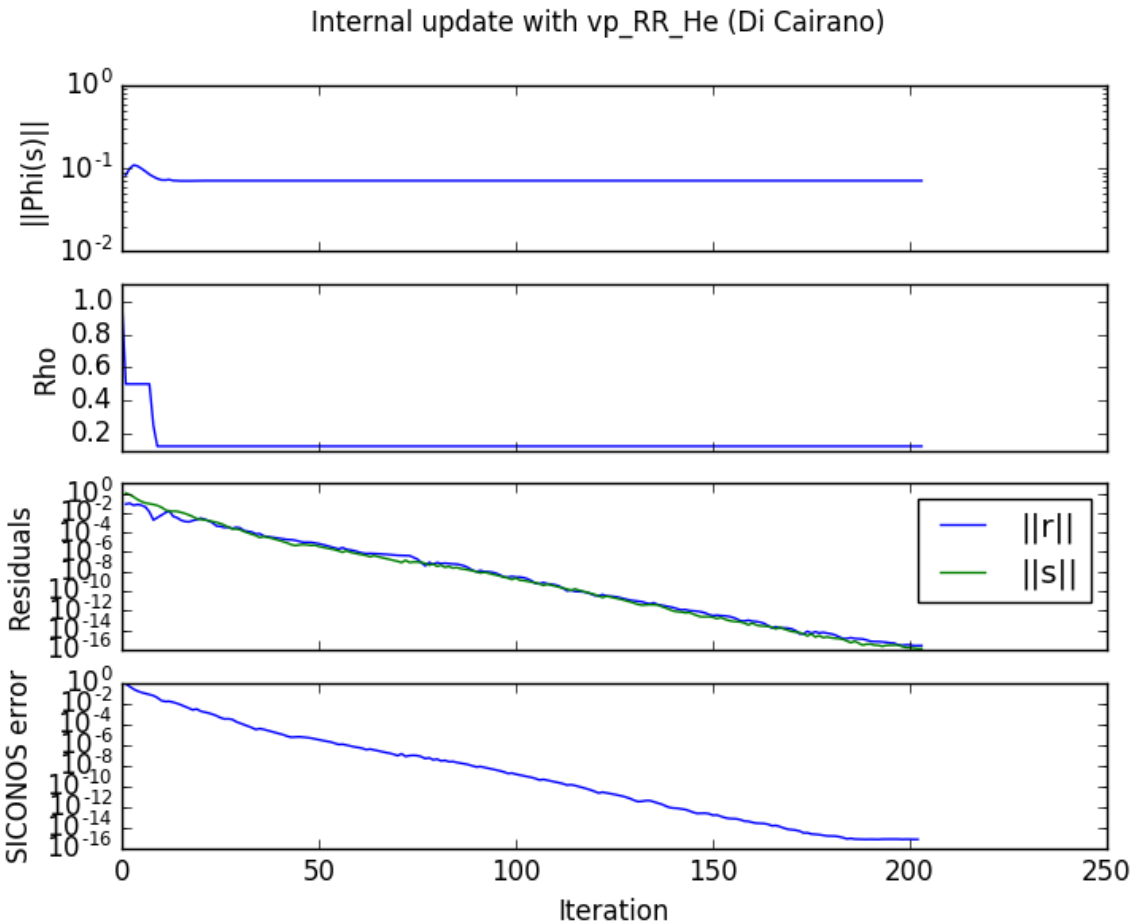

Figure: Internal update with vp-RR-He (Di Cairano) and stop criterion in (61).

Discussion

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
$\|\Phi(s)\|$	7.10e-02	7.10e-02	7.10e-02
$\Phi(s)^{\alpha=1}$	[8.31e-06 0.00e+00 0.00e+00]	[6.49e-06 0.00e+00 0.00e+00]	[7.45e-06 0.00e+00 0.00e+00]
$\Phi(s)^{\alpha=n_c}$	[1.02e-05 0.00e+00 0.00e+00]	[1.19e-05 0.00e+00 0.00e+00]	[7.85e-06 0.00e+00 0.00e+00]

Table: Comparison of s-update

Outline

Final remarks

- Comparison with other numerical methods (Fixed point and projection methods for VI formulation, Newton based methods, Splitting techniques and proximal point algorithm, etc.).
- Open question in the same numerical value of convergence in $\Phi(s)$ with internal update.

References I

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

[2] 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.

[3] 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.

[4] 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.

References II

[5] 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.

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

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

[8] 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.

[9] 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.

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 [8]. In our case $\eta = 0.999$ was used.

Algorithm 1 ADMM.

Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3: $\quad \boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4: $\quad \boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5: $\quad$ Stop criterion in (14)
6: **end for**

Constant penalty parameter

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** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3: $\quad \boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4: $\quad \boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Relaxation in (43)
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** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \hat{\boldsymbol{y}}^k; \hat{\boldsymbol{z}}^k)$
3: $\quad \boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \hat{\boldsymbol{z}}^k)$
4: $\quad \boldsymbol{z}^{k+1} := \boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Relaxation + Restart in (43) (45)
7: **end for**

Varying penalty parameter - He

In Wang et al. the value $\tau = 2$ generally performs well in the ρ update [9].

Algorithm 4 ADMM.

Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3: $\quad \boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4: $\quad \boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c} \right)$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Update of ρ in (He)
7: **end for**

Varying penalty parameter - He

Algorithm 5 Relaxed ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
3: $y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
4: $z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
5: Stop criterion in (14)
6: Relaxation in (43)
7: Update of ρ in (He)
8: **end for**

Algorithm 6 Relaxed + Restart ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $\varphi = 10$, $\tau = 2$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
3: $y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
4: $z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
5: Stop criterion in (14)
6: Relaxation + Restart in (43) (45)
7: Update of ρ in (He)
8: **end for**

Varying penalty parameter - Wohlberg

In He et al. the value $\tau_{max} = 100$ generally performs well in the ρ update [6].

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: $\quad \boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
3: $\quad \boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
4: $\quad \boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c} \right)$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Update of ρ in (Wohlberg)
7: **end for**

Varying penalty parameter - Wohlberg

Algorithm 8 Relaxed ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $\xi = 1$, $\varphi = 10$, $\tau_{max} = 100$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
3: $\quad y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
4: $\quad z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Relaxation in (43)
7: $\quad$ Update of ρ in (Wohlberg)
8: **end for**

Algorithm 9 Relaxed + Restart ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $\xi = 1$, $\varphi = 10$, $\tau_{max} = 100$, and $\rho > 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
3: $\quad y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
4: $\quad z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
5: $\quad$ Stop criterion in (14)
6: $\quad$ Relaxation + Restart in (43) (45)
7: $\quad$ Update of ρ in (Wohlberg)
8: **end for**

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 [7].

Algorithm 10 ADMM.

Require: $\boldsymbol{y}^0$, $\boldsymbol{z}^0$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$

for $k = 0, 1, 2, \ldots$ **do**
  $\boldsymbol{x}^{k+1} := \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}; \boldsymbol{y}^k; \boldsymbol{z}^k)$
  $\boldsymbol{y}^{k+1} := \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{k+1}; \boldsymbol{y}; \boldsymbol{z}^k)$
  $\boldsymbol{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(\boldsymbol{z}^k + A\boldsymbol{x}^{k+1} + B\boldsymbol{y}^{k+1} - \boldsymbol{c}\right)$
  Stop criterion in (14)
  if $mod(k, T_f) = 1$ **then**
    Update of ρ in (Spectral)
  else
    $\rho^{k+1} \leftarrow \rho^k$
  end if
end for

Varying penalty parameter - Spectral

Algorithm 11 Relaxed ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$

for $k = 0, 1, 2, \ldots$ **do**
  $x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
  $y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
  $z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
  Stop criterion in (14)
  Relaxation in (43)
  if $mod(k, T_f) = 1$ **then**
    Update of ρ in (Spectral)
  else
    $\rho^{k+1} \leftarrow \rho^k$
  end if
end for

Algorithm 12 Relaxed + Restart ADMM.

Require: $y^0 = \hat{y}^0$, $z^0 = \hat{z}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$

for $k = 0, 1, 2, \ldots$ **do**
  $x^{k+1} := \arg\min_x L_\rho(x; \hat{y}^k; \hat{z}^k)$
  $y^{k+1} := \arg\min_y L_\rho(x^{k+1}; y; \hat{z}^k)$
  $z^{k+1} := \frac{\rho^{k-1}}{\rho^k}\left(z^k + Ax^{k+1} + By^{k+1} - c\right)$
  Stop criterion in (14)
  Relaxation + Restart in (43) (45)