

Alternating Direction Method of Multipliers (ADMM) for Frictional Contact

Vincent Acary, Yoshiro Kanno & Nicolas Molina

March 29, 2018



Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Finite-dimensional frictional contact problem

The problem in [1] reads:

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

The problem data are:

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

The unknowns are two vectors:

- ▶ $\mathbf{v} \in \mathbb{R}^n$, a velocity-like vector
- ▶ $\mathbf{r} \in \mathbb{R}^m$, 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 $K_{e,\mu}^*$ its dual cone, i.e.,

$$K_{e,\mu} = \prod_{\alpha=1 \dots n_c} K_{e,\mu}^\alpha = \prod_{\alpha=1 \dots n_c} \{(x_1, \mathbf{x}_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|\mathbf{x}_2\| \leq \mu x_1\} \quad (2)$$

where $x_1 = \mathbf{x}^\top \mathbf{e}$, $\mathbf{x}_2 = \mathbf{x} - x_1 \mathbf{e}$ and $K_{e,\mu}^* = K_{e, \frac{1}{\mu}}$.

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

$$\Phi(\mathbf{u}) = [\mu^\alpha \|E_t \mathbf{u}^\alpha\| \hat{\mathbf{e}}_n, \alpha = 1 \dots n_c]^\top \quad (3)$$

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

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Fundamentals

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

$$\text{Minimize} \quad f(\mathbf{x}) + g(\mathbf{y}) \quad (4a)$$

$$\text{subject to} \quad A\mathbf{x} + B\mathbf{y} = \mathbf{c} \quad (4b)$$

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

Fundamentals

The augmented Lagrangian of problem (4) is defined as

$$L_\rho(\mathbf{x}; \mathbf{y}; \mathbf{w}) = f(\mathbf{x}) + g(\mathbf{y}) + \mathbf{w}^\top (A\mathbf{x} + B\mathbf{y} - \mathbf{c}) + \frac{\rho}{2} \|A\mathbf{x} + B\mathbf{y} - \mathbf{c}\|^2 \quad (5)$$

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

$$\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_\rho(\mathbf{x}; \mathbf{y}^k; \mathbf{w}^k) \quad (6)$$

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

$$\mathbf{w}^{k+1} := \mathbf{w}^k + \rho(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}) \quad (8)$$

where the dual variable is updated via dual ascent method.

Fundamentals

ADMM is often described in a slightly different form. Define \mathbf{z} by $\mathbf{z} = \mathbf{w}/\rho$. Then the augmented Lagrangian in (5) is reduced to

$$L_\rho(\mathbf{x}; \mathbf{y}; \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{y}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c} + \mathbf{z}\|^2 - \frac{\rho}{2} \|\mathbf{z}\|^2. \quad (9)$$

Using L_ρ in (9), we can express ADMM as follows:

$$\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_\rho(\mathbf{x}; \mathbf{y}^k; \mathbf{z}^k), \quad (10)$$

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

$$\mathbf{z}^{k+1} := \mathbf{z}^k + \mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c} \quad (12)$$

The expression in (10), (11), and (12) is called the ADMM in the scaled form, and \mathbf{z} is called the scaled dual variable.

Stop criterion

The residuals of the optimality conditions can be related to an approximated bound:

$$\begin{aligned}\|\mathbf{r}^k\| &\leq \epsilon^{pri} \\ \|\mathbf{s}^k\| &\leq \epsilon^{dual}\end{aligned}\tag{13}$$

where $\mathbf{r}^{k+1} = A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}$ and $\mathbf{s}^{k+1} = \rho A^T B (\mathbf{y}^{k+1} - \mathbf{y}^k)$ 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 chosen using an absolute and relative criterion, such as

$$\begin{aligned}\epsilon^{pri} &= \sqrt{l}\epsilon^{abs} + \epsilon^{rel} \max\{\|A\mathbf{x}^{k+1}\|, \|B\mathbf{y}^{k+1}\|, \|\mathbf{c}\|\} \\ \epsilon^{dual} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|A^T \rho \mathbf{z}^{k+1}\|\end{aligned}\tag{14}$$

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

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Coulomb friction problem via convex optimization

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

$$\underset{\mathbf{v}, \tilde{\mathbf{u}}}{\text{Minimize}} \quad \frac{1}{2} \mathbf{v}^\top M \mathbf{v} + \mathbf{f}^\top \mathbf{v} \quad (15a)$$

$$\text{subject to} \quad \tilde{\mathbf{u}} = H \mathbf{v} + \mathbf{w} + \Phi(\mathbf{s}) \quad (15b)$$

$$\tilde{\mathbf{u}} \in K_{e, \mu}^* \quad (15c)$$

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

Coulomb friction problem via convex optimization

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

$$\delta_{K_{e,\mu}^*}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in K_{e,\mu}^* \\ +\infty & \text{otherwise.} \end{cases} \quad (16)$$

Then problem (15) is equivalently rewritten as follows:

$$\underset{\mathbf{v}, \tilde{\mathbf{u}}}{\text{Minimize}} \quad \frac{1}{2} \mathbf{v}^\top M \mathbf{v} + \mathbf{f}^\top \mathbf{v} + \delta_{K_{e,\mu}^*}(\tilde{\mathbf{u}}) \quad (17a)$$

$$\text{subject to} \quad \tilde{\mathbf{u}} = H \mathbf{v} + \mathbf{w} + \Phi(\mathbf{s}) \quad (17b)$$

Iterations

Problem (17) is a minimization problem of a convex function under linear equality constraints. Its augmented Lagrangian (in the scaled form) is formulated as

$$L_{\rho}(\mathbf{v}; \tilde{\mathbf{u}}; \boldsymbol{\zeta}) = \frac{1}{2} \mathbf{v}^{\top} M \mathbf{v} + \mathbf{f}^{\top} \mathbf{v} + \delta_{K_{e, \mu}^*}(\tilde{\mathbf{u}}) + \frac{\rho}{2} \|H \mathbf{v} + \mathbf{w} + \Phi(\mathbf{s}) + \boldsymbol{\zeta} - \tilde{\mathbf{u}}\|^2 - \frac{\rho}{2} \|\boldsymbol{\zeta}\|^2 \quad (18)$$

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 (17) consists of iterating the updates (see (10), (11), and (12))

$$\mathbf{v}^{k+1} := \arg \min_{\mathbf{v}} L_{\rho}(\mathbf{v}; \tilde{\mathbf{u}}^k; \boldsymbol{\zeta}^k) \quad (19)$$

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

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H \mathbf{v}^{k+1} + \mathbf{w} + \Phi(\mathbf{s}) - \tilde{\mathbf{u}}^{k+1} \quad (21)$$

Iterations

► v-update

The first step in (19) corresponds to the solution to the following linear system:

$$\left[M + \rho H^\top H \right] \mathbf{v}^{k+1} = -\mathbf{f} + \rho H^\top (\tilde{\mathbf{u}}^k - \mathbf{w} - \Phi(\mathbf{s}) - \boldsymbol{\zeta}^k) \quad (22)$$

► $\tilde{\mathbf{u}}$ -update

The second step in (20) can be computed independently for each $\alpha = 1 \dots n_c$ as

$$\tilde{\mathbf{u}}^{k+1} := \arg \min_{\tilde{\mathbf{u}}} \delta_{K_{e,\mu}^*}(\tilde{\mathbf{u}}) + \frac{\rho}{2} \| H\mathbf{v}^{k+1} + \mathbf{w} + \Phi(\mathbf{s}) + \boldsymbol{\zeta}^k - \tilde{\mathbf{u}} \|^2 \quad (23)$$

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

$$\tilde{\mathbf{u}}^{k+1} := \Pi_{K_{e,\mu}^*}(H\mathbf{v}^{k+1} + \mathbf{w} + \Phi(\mathbf{s}) + \boldsymbol{\zeta}^k) \quad (24)$$

Projection onto dual second-order cone

The projection of \mathbf{x} onto $K_{e,\mu}^*$ could be written as follows

$$\Pi_{K_{e,\mu}^*}(\mathbf{x}) = \begin{cases} 0 & \text{if } -\mathbf{x} \in K_{e,\mu} \\ \mathbf{x} & \text{if } \mathbf{x} \in K_{e,\mu}^* \\ \frac{\mu^2 \left(x_1 + \frac{1}{\mu} \|\mathbf{x}_2\| \right)}{1 + \mu^2} \begin{bmatrix} 1 \\ \frac{1}{\mu} \mathbf{x}_2 / \|\mathbf{x}_2\| \end{bmatrix} & \text{if } -\mathbf{x} \notin K_{e,\mu} \wedge \mathbf{x} \notin K_{e,\mu}^*, \end{cases} \quad (25)$$

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Optimal penalty parameter selection

- ▶ Ghadimi
- ▶ Di Cairano
- ▶ Acary
- ▶ Normal

Optimal penalty parameter selection

► Ghadimi

The ρ^* that minimizes the rate of convergence is

$$\rho^* := \left(\sqrt{\lambda_{\min}(HM^{-1}H^\top)\lambda_{\max}(HM^{-1}H^\top)} \right)^{-1} \quad (26)$$

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 ρ^* that minimizes the rate of convergence is

$$\rho^* := \sqrt{\lambda_{\min}(M)\lambda_{\max}(M)} \quad (27)$$

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 (22)

$$\rho^* := \frac{\|M\|_1}{\|H\|_1} \quad (28)$$

► Normal

Finally, a less expensive and standard method is to set

$$\rho^* := 1 \quad (29)$$

- └ Penalty parameter
 - └ Varying penalty parameter

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.

$$\rho^{k+1} := \begin{cases} \tau \rho^k & \text{if } \|\mathbf{r}^k\| > \varphi \|\mathbf{s}^k\| \\ \tau^{-1} \rho^k & \text{if } \|\mathbf{s}^k\| > \varphi \|\mathbf{r}^k\| \\ \rho^k & \text{otherwise,} \end{cases} \quad (30)$$

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

Varying penalty parameter

► Wohlberg

The behavior of ADMM under scaling is addressed by Wohlberg [6]. The heuristic implemented by He et al. is appropriately scaled to maintain invariance of the algorithm as:

$$\rho^{k+1} := \begin{cases} \tau \rho^k & \text{if } \|\mathbf{r}_{rel}^k\| > \xi \varphi \|\mathbf{s}_{rel}^k\| \\ \tau^{-1} \rho^k & \text{if } \|\mathbf{s}_{rel}^k\| > \xi \varphi \|\mathbf{r}_{rel}^k\| \\ \rho^k & \text{otherwise,} \end{cases} \quad (31)$$

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

$$\begin{aligned} \mathbf{r}_{rel}^{k+1} &:= \frac{\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c}}{\max\{\|\mathbf{A}\mathbf{x}^{k+1}\|, \|\mathbf{B}\mathbf{y}^{k+1}\|, \|\mathbf{c}\|\}} \\ \mathbf{s}_{rel}^{k+1} &:= \frac{\rho \mathbf{A}^T \mathbf{B} (\mathbf{y}^{k+1} - \mathbf{y}^k)}{\|\mathbf{A}^T \mathbf{w}^{k+1}\|} = \frac{\mathbf{A}^T \mathbf{B} (\mathbf{y}^{k+1} - \mathbf{y}^k)}{\|\mathbf{A}^T \boldsymbol{\zeta}^{k+1}\|} \end{aligned} \quad (32)$$

Varying penalty parameter

The fixed multiplier τ is a potential weakness of the penalty update policies. A straightforward solution is to adapt τ at each iteration

$$\tau^{k+1} := \begin{cases} \sqrt{\xi^{-1} \frac{\|r_{rel}^k\|}{\|s_{rel}^k\|}} & \text{if } 1 \leq \sqrt{\xi^{-1} \frac{\|r_{rel}^k\|}{\|s_{rel}^k\|}} < \tau_{max} \\ \sqrt{\xi \frac{\|s_{rel}^k\|}{\|r_{rel}^k\|}} & \text{if } \tau_{max}^{-1} < \sqrt{\xi^{-1} \frac{\|r_{rel}^k\|}{\|s_{rel}^k\|}} < 1 \\ \tau_{max} & \text{otherwise} \end{cases} \quad (33)$$

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 (4).

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 $k_0 < k$

$$\mathbf{z}_S^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left(\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^k - \mathbf{c} \right) \quad (34)$$

$$\Delta \hat{\mathbf{w}}_k := \rho^k \mathbf{z}_S^{k+1} - \rho^{k_0} \mathbf{z}_S^{k_0} \quad (35)$$

$$\Delta \hat{F}_k := A \left(\mathbf{x}^{k+1} - \mathbf{x}^{k_0} \right) \quad (36)$$

$$\Delta \hat{G}_k := B \left(\mathbf{y}^{k+1} - \mathbf{y}^{k_0} \right) \quad (37)$$

Varying penalty parameter

The curvatures are estimated via least squares yielding

$$\begin{aligned}\hat{\alpha}_k^{SD} &:= \frac{\langle \Delta \hat{w}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle} & \hat{\beta}_k^{SD} &:= \frac{\langle \Delta \hat{w}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle} \\ \hat{\alpha}_k^{MG} &:= \frac{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{F}_k, \Delta \hat{F}_k \rangle} & \hat{\beta}_k^{MG} &:= \frac{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle}{\langle \Delta \hat{G}_k, \Delta \hat{G}_k \rangle}\end{aligned}\quad (38)$$

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

$$\begin{aligned}\hat{\alpha}_k &:= \begin{cases} \hat{\alpha}_k^{SD} & \text{if } 2\hat{\alpha}_k^{MG} > \hat{\alpha}_k^{SD} \\ \hat{\alpha}_k^{MG} & \text{otherwise} \end{cases} \\ \hat{\beta}_k &:= \begin{cases} \hat{\beta}_k^{SD} & \text{if } 2\hat{\beta}_k^{MG} > \hat{\beta}_k^{SD} \\ \hat{\beta}_k^{MG} & \text{otherwise} \end{cases}\end{aligned}\quad (39)$$

Varying penalty parameter

Xu et al. propose to safeguard the method by assessing the quality of the curvature estimates, and only updating the stepsize if the curvature estimates satisfy a reliability criterion:

$$\alpha_k^{cor} := \frac{\langle \Delta \hat{F}_k, \Delta \hat{w}_k \rangle}{\|\Delta \hat{F}_k\| \|\Delta \hat{w}_k\|} \wedge \beta_k^{cor} := \frac{\langle \Delta \hat{G}_k, \Delta \hat{w}_k \rangle}{\|\Delta \hat{G}_k\| \|\Delta \hat{w}_k\|} \quad (40)$$

Finally, the safeguard spectral adaptative penalty rule is

$$\rho_{k+1} := \begin{cases} \sqrt{\hat{\alpha}_k \hat{\beta}_k} & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \hat{\alpha}_k & \text{if } \alpha_k^{cor} > \epsilon^{cor} \wedge \beta_k^{cor} \leq \epsilon^{cor} \\ \hat{\beta}_k & \text{if } \alpha_k^{cor} \leq \epsilon^{cor} \wedge \beta_k^{cor} > \epsilon^{cor} \\ \rho_k & \text{otherwise} \end{cases} \quad (41)$$

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

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Relaxed ADMM

A relaxed version of ADMM was proposed in [8]. This method applies Nesterov's type overrelaxation scheme to the updates of \mathbf{y}^k and \mathbf{z}^k which takes into account past iterations. It is shown that the method converges in the primal and dual residuals with rate $O(1/k^2)$.

$$\alpha_{k+1} := \frac{1}{2} \left(1 + \sqrt{1 + 4\alpha_k^2} \right) \quad (42a)$$

$$\hat{\mathbf{y}}^{k+1} := \mathbf{y}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}} (\mathbf{y}^{k+1} - \mathbf{y}^k) \quad (42b)$$

$$\hat{\mathbf{z}}^{k+1} := \mathbf{z}^{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}} \left(\mathbf{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k} \mathbf{z}^k \right). \quad (42c)$$

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 residual, which measures both the primal and dual error simultaneously:

$$e_k := \rho^k \|\mathbf{z}^{k+1} - \frac{\rho^{k-1}}{\rho^k} \hat{\mathbf{z}}^k\|^2 + \rho^k \|B(\mathbf{y}^{k+1} - \hat{\mathbf{y}}^k)\|^2 \quad (43)$$

if $e_k < \eta e_{k-1}$ the relaxed ADMM (42) is performed, otherwise, the restart rule states

$$\alpha_{k+1} := 1 \quad (44a)$$

$$\hat{\mathbf{y}}^{k+1} := \mathbf{y}^{k+1} \quad (44b)$$

$$\hat{\mathbf{z}}^{k+1} := \mathbf{z}^{k+1} \quad (44c)$$

$$e_k \leftarrow \frac{e_{k-1}}{\eta} \quad (44d)$$

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

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

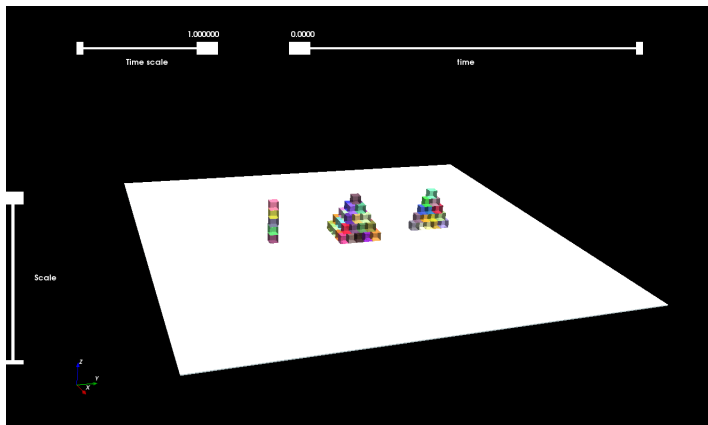


Figure: Box Stacks, set of examples in the comparison.

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

$$t_{p,s} = \text{computing time required for } s \text{ to solve } p \text{ at precision } tol \quad (45)$$

A performance ratio over all the solvers is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s}, s \in S\}} \geq 1 \quad (46)$$

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

$$\rho_s(\tau) = \frac{1}{n_p} \text{card}\{p \in P, r_{p,s} \leq \tau\} \leq 1 \quad (47)$$

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Family of optimal penalty parameters

- ▶ Ghadimi
- ▶ Di Cairano
- ▶ Acary
- ▶ Normal

Ghadimi

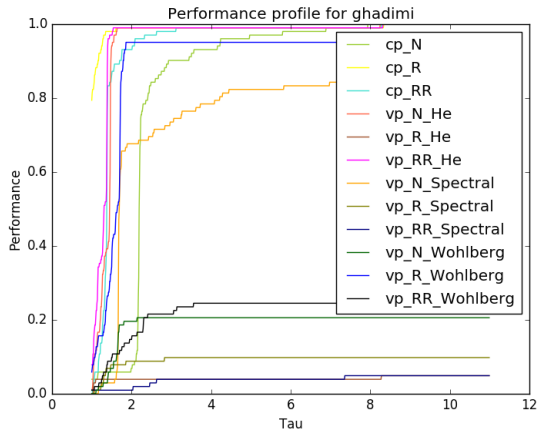


Figure: Comparison of algorithms for Ghadimi optimal penalty parameter.

Di Cairano

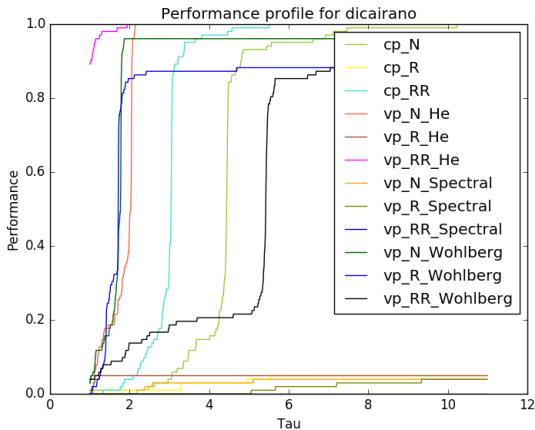


Figure: Comparison of algorithms for Di Cairano optimal penalty parameter.

Acary

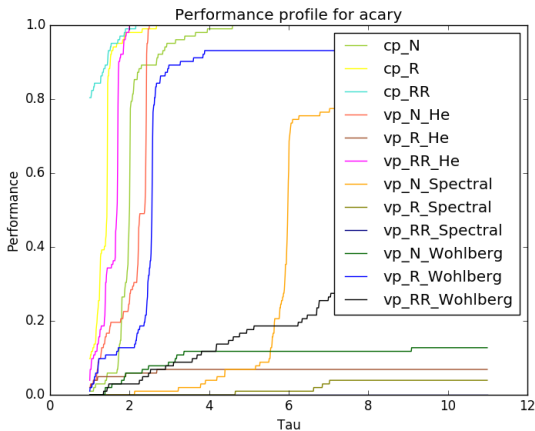


Figure: Comparison of algorithms for Acary optimal penalty parameter.

Normal

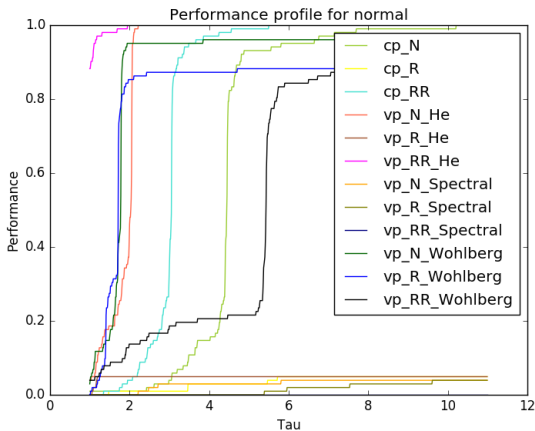


Figure: Comparison of algorithms for Normal optimal penalty parameter.

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)
$\rho_s(1)$	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 work (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

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

S-update

- ▶ External update
- ▶ Internal update
- ▶ Initial guess

Internal update

For the next development, we consider

$$\Phi(\mathbf{u}) = [\mu^\alpha \|E_t \mathbf{u}^\alpha\| \hat{\mathbf{e}}_n, \alpha = 1 \dots n_c]^T = [\mu^\alpha \|E_t (H\mathbf{v} + \mathbf{w})^\alpha\| \hat{\mathbf{e}}_n, \alpha = 1 \dots n_c]^T \quad (48)$$

therefore $\Phi(\mathbf{u})$ could be seen as $\Phi(\mathbf{v})$. Then, the problem (18) is a minimization problem of a convex function under non-linear equality constraints. Its augmented Lagrangian is formulated as

$$L_\rho(\mathbf{v}; \tilde{\mathbf{u}}; \boldsymbol{\zeta}) = \frac{1}{2} \mathbf{v}^\top M \mathbf{v} + \mathbf{f}^\top \mathbf{v} + \delta_{K_{\mathbf{e}, \mu}^*}(\tilde{\mathbf{u}}) + \frac{\rho}{2} \|H\mathbf{v} + \mathbf{w} + \Phi(\mathbf{v}) + \boldsymbol{\zeta} - \tilde{\mathbf{u}}\|^2 - \frac{\rho}{2} \|\boldsymbol{\zeta}\|^2 \quad (49)$$

where ρ is the penalty parameter, and $\boldsymbol{\zeta} \in \mathbb{R}^m$ is the scaled dual variable. The ADMM solving problem (17) consists of iterating the updates

$$\mathbf{v}^{k+1} := \arg \min_{\mathbf{v}} L_\rho(\mathbf{v}; \tilde{\mathbf{u}}^k; \boldsymbol{\zeta}^k) \quad (50)$$

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

$$\boldsymbol{\zeta}^{k+1} := \boldsymbol{\zeta}^k + H\mathbf{v}^{k+1} - \tilde{\mathbf{u}}^{k+1} + \Phi(\mathbf{v}^{k+1}) \quad (52)$$

Internal update

► v-update

The first step in (50) corresponds to unconstrained minimization of a convex function. It follows from the stationarity condition of this convex function that \mathbf{v}^{k+1} is the solution to the following system of non-linear equations:

$$\left[M + \rho \left(H^\top + \partial_{\mathbf{v}} \Phi(\mathbf{v}^{k+1}) \right) H \right] \mathbf{v}^{k+1} = -\mathbf{f} + \rho \left(H^\top + \partial_{\mathbf{v}} \Phi(\mathbf{v}^{k+1}) \right) (\tilde{\mathbf{u}}^k - \mathbf{w} - \Phi(\mathbf{v}^{k+1}) - \boldsymbol{\zeta}^k) \quad (53)$$

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

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

which follows the same structure of (22) and allows us to perform a LU or Cholesky factorization. If $\|E_t \mathbf{u}^\alpha\| \neq 0$

$$\partial_{\mathbf{v}} \Phi(\mathbf{v}) = \left[\mu^\alpha \hat{\mathbf{e}}_n^\top \frac{E_t (H\mathbf{v} + \mathbf{w})^\alpha}{\|E_t (H\mathbf{v} + \mathbf{w})^\alpha\|} E_t^\top H^{\alpha^\top}, \alpha = 1 \dots n_c \right]^\top = \Psi \quad (55)$$

Internal update

Therefore

$$\left[M + \rho \left(H^\top + \Psi \right) H \right] \mathbf{v}^{k+1} = -\mathbf{f} + \rho \left(H^\top + \Psi \right) (\tilde{\mathbf{u}}^k - \mathbf{w} - \Phi(\mathbf{v}^k) - \boldsymbol{\zeta}^k) \quad (56)$$

Note that if in (50) it is evaluated $\Phi(\mathbf{v})$ in the previous iteration before compute the stationary condition, it results in

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

which is a pseudo-update of (50) in comparison with (56), but it aims to be faster whitout the computation of Ψ , therefore, (57) will be used.

► $\tilde{\mathbf{u}}$ -update

The second step in (51) can be computed independently for each $\alpha = 1 \dots n_c$ as

$$\tilde{\mathbf{u}}^{k+1} := \arg \min_{\tilde{\mathbf{u}}} \delta_{K_{e,\mu}^*}(\tilde{\mathbf{u}}) + \frac{\rho}{2} \| H\mathbf{v}^{k+1} + \mathbf{w} + \Phi(\mathbf{v}^k) + \boldsymbol{\zeta}^k - \tilde{\mathbf{u}} \|^2 \quad (58)$$

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

$$\tilde{\mathbf{u}}^{k+1} := \Pi_{K_{e,\mu}^*}(H\mathbf{v}^{k+1} + \mathbf{w} + \boldsymbol{\zeta}^k + \Phi(\mathbf{v}^{k+1})) \quad (59)$$

S-update

► External update

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

$$\left[\frac{\Phi(\mathbf{v}^{k+1}) - \Phi(\mathbf{v}^k)}{\Phi(\mathbf{v}^k)} \right]^\alpha \leq \epsilon^{ext} \quad (60)$$

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

$$\epsilon^{SIC} = \sqrt{\|M\mathbf{v} + \mathbf{f} - H^\top \mathbf{r}\|^2 + \|\tilde{\mathbf{u}} - \Pi_{K_{e,\mu}^*}(\tilde{\mathbf{u}} - \rho \mathbf{r})\|^2} \quad (61)$$

Outline

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

External S-update

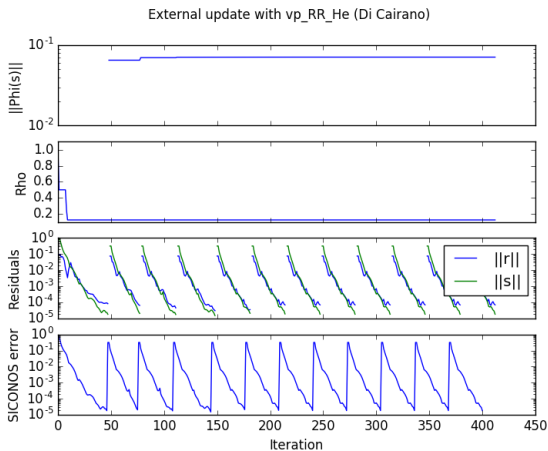


Figure: External update with vp-RR-He (Di Cairano).

Internal S-update

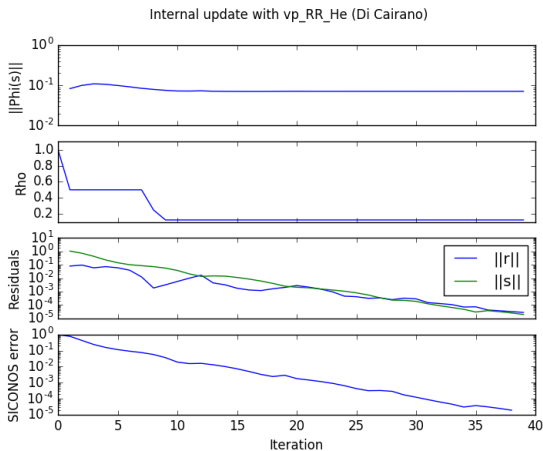


Figure: Internal update with vp-RR-He (Di Cairano).

Internal S-update

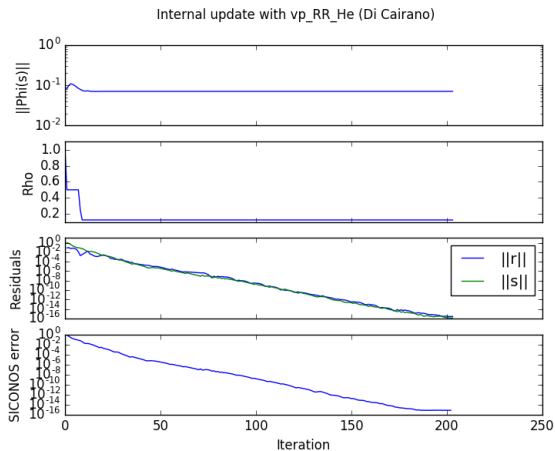


Figure: Internal update with vp-RR-He (Di Cairano) and stop criterion in (60).

Discussion

In Table 3 are shown the values of the final $\|\Phi(s)\|$ with its respective 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

Introduction

ADMM method

ADMM formulation of problem

Penalty parameter

Iteration improvements

Numerical experience

Results

S-update

Numerical experience: S-update

Final remarks

Final remarks

- ▶ The best improvement is given by vp-RR-He with Di Cairano and Normal optimal penalty parameter.
- ▶ Open question in the same numerical value of convergence in $\Phi(s)$ with external and internal update.
- ▶ Comparison with other numerical methods (Fixed point and projection methods for VI formulation, Newton based methods, Splitting techniques and proximal point algorithm, etc.).

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 desirable 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: \mathbf{y}^0 , \mathbf{z}^0 , and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \mathbf{y}^k; \mathbf{z}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \mathbf{z}^k)$ 
4:    $\mathbf{z}^{k+1} := \mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}$ 
5:   Stop criterion in (13)
6: end for

```

Constant penalty parameter

Algorithm 2 Relaxed ADMM.

Require: $\mathbf{y}^0 = \hat{\mathbf{y}}^0$, $\mathbf{z}^0 = \hat{\mathbf{z}}^0$, $\alpha_0 = 1$, and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}$ 
5:   Stop criterion in (13)
6:   Relaxation in (42)
7: end for
```

Algorithm 3 Relaxed + Restart ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}$ 
5:   Stop criterion in (13)
6:   Relaxation + Restart in (42) (44)
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: $\mathbf{y}^0, \mathbf{z}^0, \varphi = 10, \tau = 2$, and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \mathbf{y}^k; \mathbf{z}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \mathbf{z}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Update of  $\rho$  in (He)
7: end for
```

Varying penalty parameter - He

Algorithm 5 Relaxed ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left( \mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c} \right)$ 
5:   Stop criterion in (13)
6:   Relaxation in (42)
7:   Update of  $\rho$  in (He)
8: end for
    
```

Algorithm 6 Relaxed + Restart ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} \left( \mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c} \right)$ 
5:   Stop criterion in (13)
6:   Relaxation + Restart in (42) (44)
7:   Update of  $\rho$  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: $\mathbf{y}^0, \mathbf{z}^0, \xi = 1, \varphi = 10, \tau_{max} = 100$, and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \mathbf{y}^k; \mathbf{z}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \mathbf{z}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Update of  $\rho$  in (Wohlberg)
7: end for
```

Varying penalty parameter - Wohlberg

Algorithm 8 Relaxed ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Relaxation in (42)
7:   Update of  $\rho$  in (Wohlberg)
8: end for

```

Algorithm 9 Relaxed + Restart ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Relaxation + Restart in (42) (44)
7:   Update of  $\rho$  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: $\mathbf{y}^0, \mathbf{z}^0, T_f = 2, \epsilon^{cor} = 0.2$, and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \mathbf{y}^k; \mathbf{z}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \mathbf{z}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   if  $\text{mod}(k, T_f) = 1$  then
7:     Update of  $\rho$  in (Spectral)
8:   else
9:      $\rho^{k+1} \leftarrow \rho^k$ 
10:  end if
11: end for
```

Varying penalty parameter - Spectral

Algorithm 11 Relaxed ADMM.

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

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Relaxation in (42)
7:   if  $\text{mod}(k, T_f) = 1$  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: $\mathbf{y}^0 = \hat{\mathbf{y}}^0$, $\mathbf{z}^0 = \hat{\mathbf{z}}^0$, $\alpha_0 = 1$, $\eta \approx 1$, $T_f = 2$, $\epsilon^{cor} = 0.2$, and $\rho > 0$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $\mathbf{x}^{k+1} := \arg \min_{\mathbf{x}} L_{\rho}(\mathbf{x}; \hat{\mathbf{y}}^k; \hat{\mathbf{z}}^k)$ 
3:    $\mathbf{y}^{k+1} := \arg \min_{\mathbf{y}} L_{\rho}(\mathbf{x}^{k+1}; \mathbf{y}; \hat{\mathbf{z}}^k)$ 
4:    $\mathbf{z}^{k+1} := \frac{\rho^{k-1}}{\rho^k} (\mathbf{z}^k + A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c})$ 
5:   Stop criterion in (13)
6:   Relaxation + Restart in (42) (44)
```