Contraction Methods for Convex Optimization and monotone variational inequalities – No.13

Alternating direction method of multipliers in sense of customized PPA

Bingsheng He
Department of Mathematics
Nanjing University
hebma@nju.edu.cn

The context of this lecture is based on the manuscript [2]
1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

\[
\min \{ \theta_1(x) + \theta_2(y) \mid Ax + By = b, \ x \in \mathcal{X}, \ y \in \mathcal{Y} \} \quad (1.1)
\]

where \( \theta_1(x) : \mathbb{R}^{n_1} \to \mathbb{R}, \ \theta_2(y) : \mathbb{R}^{n_2} \to \mathbb{R} \) are convex functions (but not necessarily smooth), \( A \in \mathbb{R}^{m \times n_1}, \ B \in \mathbb{R}^{m \times n_2} \) and \( b \in \mathbb{R}^m, \ \mathcal{X} \subset \mathbb{R}^{n_1}, \ \mathcal{Y} \subset \mathbb{R}^{n_2} \) are given closed convex sets.

The task of solving the problem (1.1) is to find an \((x^*, y^*, \lambda^*) \in \Omega\), such that

\[
\begin{align*}
\theta_1(x) - \theta_1(x^*) + (x - x^*)^T(-A^T \lambda^*) \geq 0, \\
\theta_2(y) - \theta_2(y^*) + (y - y^*)^T(-B^T \lambda^*) \geq 0, \\
(\lambda - \lambda^*)^T(Ax^* + By^* - b) \geq 0,
\end{align*}
\]

where \( \Omega = \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^m \).
By denoting

\[ u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T \lambda \\ -B^T \lambda \\ Ax + By - b \end{pmatrix} \]

and

\[ \theta(u) = \theta_1(x) + \theta_2(y), \]

the first order optimal condition (1.2) can be written in a compact form such as

\[ w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega. \tag{1.3} \]

Note that the mapping \( F \) is monotone. We use \( \Omega^* \) to denote the solution set of the variational inequality (1.3). For convenience we use the notations

\[ v = \begin{pmatrix} y \\ \lambda \end{pmatrix} \quad \text{and} \quad \mathcal{V}^* = \{ (y^*, \lambda^*) \mid (x^*, y^*, \lambda^*) \in \Omega^* \}. \]
Applied ADMM to the structure VI \((y^k, \lambda^k) \Rightarrow (y^{k+1}, \lambda^{k+1})\)

First, for given \((y^k, \lambda^k)\), \(\tilde{x}^k\) is the solution of the following problem

\[
\tilde{x}^k = \text{Argmin} \left\{ \begin{array}{c}
\theta_1(x) - (\lambda^k)^T (Ax + By^k - b) \\
+ \frac{\beta}{2} \|Ax + By^k - b\|^2
\end{array} \right\} \quad x \in \mathcal{X}
\] (1.4a)

Use \(\lambda^k\) and the obtained \(\tilde{x}^k\), \(\tilde{y}^k\) is the solution of the following problem

\[
\tilde{y}^k = \text{Argmin} \left\{ \begin{array}{c}
\theta_2(y) - (\lambda^k)^T (A\tilde{x}^k + By - b) \\
+ \frac{\beta}{2} \|A\tilde{x}^k + By - b\|^2
\end{array} \right\} \quad y \in \mathcal{Y}
\] (1.4b)

\[
\tilde{\lambda}^k = \lambda^k - \beta (A\tilde{x}^k + B\tilde{y}^k - b).
\] (1.4c)

The sub-problems (1.4a) and (1.4b) are separately solved.
Classical Alternating Direction Method of Multipliers:

$v^{k+1} = \tilde{v}^k.$

Ye-Yuan’s Alternating Direction Method of Multipliers:

$v^{k+1} = v^k - \alpha_k (v^k - \tilde{v}^k), \quad \alpha_k = \gamma \alpha_k^*, \quad \gamma \in (0, 2) \quad (1.5a)$

where

$$\alpha_k^* = \frac{\|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T B (y^k - \tilde{y}^k)}{\|v^k - \tilde{v}^k\|_H^2} \quad (1.5b)$$

and

$$\|v^k - \tilde{v}^k\|_H^2 = \beta \|B (y^k - \tilde{y}^k)\|^2 + \frac{1}{\beta} \|\lambda^k - \tilde{\lambda}^k\|^2.$$

The convergence of the classical alternating direction method and Ye-Yuan’s ADMM are demonstrated in Lecture 11.
Ye-Yuan’s ADMM vs Classical ADMM:

The iteration number of Ye-Yuan’s ADMM is less than the one of the classical ADMM. However, in Ye-Yuan’s ADMM, we need to calculate the step size length $\alpha_k^*$ in each iteration.

2 ADMM based customized PPA

The $k$-th iteration of the proposed Alternating Direction Method of Multipliers in this section is also from a pair of $(y^k, \lambda^k)$ to a new pair of $(y^{k+1}, \lambda^{k+1})$. In the prediction step, we generate a $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$ which satisfies

$$\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + Q(\tilde{v}^k - v^k)\} \geq 0, \quad \forall w \in \Omega, \quad (2.1)$$

where $Q$ is a $3 \times 2$ block matrix whose first row is zero, and the rest sub-matrix is symmetric and positive semi-definite. In details, the matrices $Q$ and $M$ have
the following forms

\[
Q = \begin{pmatrix}
0 & 0 \\
\beta B^T B & -B^T \\
-B & \frac{1}{\beta} I_m
\end{pmatrix}
\quad \text{and} \quad
H = \begin{pmatrix}
\beta B^T B & -B^T \\
-B & \frac{1}{\beta} I_m
\end{pmatrix}.
\] (2.2)

Note that the matrix \( H \) is symmetric and positive semidefinite. If we replace \( Q(\tilde{v}^k - v^k) \) by \( G(\tilde{w}^k - w^k) \) with a symmetric positive definite matrix \( G \), then (2.1) becomes a sub-problem of the proximal point algorithm. Thus, the method in this lecture is called the ADMM-based customized PPA or Alternating direction method in the sense of customized PPA.

2.1 Motivation

In the classical ADMM, the variable \( x \) is not a part of the state. \( \tilde{x}^k \) is only an intermediate result computed from the previous state \((y^k, \lambda^k)\). Note that \( \tilde{x}^k \) is
the minimizer of the augmented Lagrangian function with \( y = y^k \), i.e.,

\[
\tilde{x}^k = \text{Argmin}\{\theta_1(x) - (\lambda^k)^T(Ax + By^k - b) + \frac{\beta}{2} \|Ax + By^k - b\|^2 \mid x \in \mathcal{X}\}.
\]

(2.3)

Thus, we have \( \tilde{x}^k \in \mathcal{X} \) and

\[
\theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\lambda^k + \beta A^T(A\tilde{x}^k + By^k - b)\} \geq 0, \ \forall \ x \in \mathcal{X}.
\]

(2.4)

If we write the above variational inequality as

\[
\theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\lambda^k\} \geq 0, \ \forall \ x \in \mathcal{X},
\]

it implies that

\[
\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b).
\]

(2.5)

According to the above definition, for any \( \tilde{y}^k \in \mathcal{Y} \), we have

\[
(A\tilde{x}^k + B\tilde{y}^k - b) - B(\tilde{y}^k - y^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0.
\]

(2.6)
Combining (2.4) and (2.6) together, we get \((\tilde{x}^k, \tilde{\lambda}^k) \in \mathcal{X} \times \mathbb{R}^m\),

\[
\theta_1(x) - \theta_1(\tilde{x}^k) \\
+ \left( x - \tilde{x}^k \right)^T \left\{ \left( \begin{array}{c} -A^T \tilde{\lambda}^k \\
A \tilde{x}^k + B \tilde{y}^k - b \end{array} \right) + \left( \begin{array}{cc} 0 & 0 \\
-B & \frac{1}{\beta} \end{array} \right) \left( \begin{array}{c} \tilde{y}^k - y^k \\
\tilde{\lambda}^k - \lambda^k \end{array} \right) \right\} \geq 0, \quad (2.7)
\]

for all \((x, \lambda) \in \mathcal{X} \times \mathbb{R}^m\). In order to get \(\tilde{w}^k \in \Omega\), such that

\[
\left( \begin{array}{c} \theta_1(x) - \theta_1(\tilde{x}^k) + \\
\theta_2(y) - \theta_2(\tilde{y}^k) \end{array} \right) + \left( \begin{array}{c} x - \tilde{x}^k \\
y - \tilde{y}^k \\
\lambda - \tilde{\lambda}^k \end{array} \right)^T \left\{ \left( \begin{array}{c} -A^T \tilde{\lambda}^k \\
-B^T \tilde{\lambda}^k \\
A \tilde{x}^k + B \tilde{y}^k - b \end{array} \right) \right\} \geq 0, \quad \forall w \in \Omega, \quad (2.8)
\]
we need only to find \( \tilde{y}^k \in Y \), such that
\[
\tilde{y}^k \in Y, \quad (\theta_2(y) - \theta_2(\tilde{y}^k)) + (y - \tilde{y}^k)^T \left\{-B^T \tilde{\lambda}^k + B^T (\beta B (\tilde{y}^k - y^k) - (\tilde{\lambda}^k - \lambda^k)) \right\} \geq 0, \quad \forall \ y \in Y. \quad (2.9)
\]
By using (2.5), we have
\[
\beta B (\tilde{y}^k - y^k) - (\tilde{\lambda}^k - \lambda^k) = \beta \left( A \tilde{x}^k + B \tilde{y}^k - b \right).
\]
Thus, the variational inequality (2.9) is
\[
(\theta_2(y) - \theta_2(\tilde{y}^k)) + (y - \tilde{y}^k)^T \left\{-B^T \tilde{\lambda}^k + \beta B^T \left( A \tilde{x}^k + B \tilde{y}^k - b \right) \right\} \geq 0, \quad \forall \ y \in Y.
\]
For given \( \tilde{x}^k \) and the defined \( \tilde{\lambda}^k \) in (2.5), such a \( \tilde{y}^k \) can be obtained via solving the following convex optimization problem:
\[
\tilde{y}^k = \text{Argmin} \{ \theta_2(y) + \frac{\beta}{2} \| A \tilde{x}^k + B y - b - \frac{1}{\beta} \tilde{\lambda}^k \|^2 | y \in Y \}. \quad (2.10)
\]
The above analysis guides us to construct the ADMM based customized PPA.
2.2 The proposed ADMM based customized PPA

From given $v^k = (y^k, \lambda^k)$, the prediction step produces $w^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$.

The prediction step:

1. First, for given $(y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) + \frac{\beta}{2} \|Ax + By^k - b - \frac{1}{\beta} \lambda^k\|^2 \mid x \in \mathcal{X}\} \quad (2.11a)$$

2. Set the multipliers by

$$\tilde{\lambda}^k = \lambda^k - \beta (A\tilde{x}^k + By^k - b). \quad (2.11b)$$

3. Finally, use the obtained $\tilde{x}^k$ and $\tilde{\lambda}^k$, find $\tilde{y}^k$ by

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \frac{\beta}{2} \|A\tilde{x}^k + By - b - \frac{1}{\beta} \tilde{\lambda}^k\|^2 \mid y \in \mathcal{Y}\} \quad (2.11c)$$

In the ADMM view of point, we generate the predictor in the order

$$\tilde{x}^k, \quad \tilde{\lambda}^k \quad \text{and} \quad \tilde{y}^k.$$
As illustrated in the motivation, we get (2.8). This variational inequality can be written in the form of

\[ \tilde{w}^k \in \Omega, \ (w - \tilde{w}^k)^T \{ F(\tilde{w}^k) + Q(\tilde{v}^k - v^k) \} \geq 0, \ \forall w \in \Omega, \quad (2.12) \]

where \( Q \) is just the same matrix defined in (2.2). The above variational inequality is essential in the unified framework of the contraction methods.

The correction step: Update the new iterate \( v^{k+1} \) by

\[ v^{k+1} = v^k - \gamma(v^k - \tilde{v}^k), \quad \gamma \in (0, 2). \quad (2.13) \]

To get the new iterate \( v^{k+1} \), this method does not need to calculate the step size.

### 2.3 Convergence of the ADMM in sense of customized PPA

Based on the analysis in the last subsection, we have the following lemma.

**Lemma 2.1** Let \( \tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega \) be generated by (2.11) from the given
\(v^k = (y^k, \lambda^k)\). Then, we have

\[(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq 0, \ \forall w^* \in \Omega^*,\] (2.14)

where the matrix \(Q\) is defined in (2.2).

**Proof.** Setting \((x, y, \lambda) = (x^*, y^*, \lambda^*)\) in (2.8), we obtain

\[(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k).\] (2.15)

Since \(F\) is monotone and \(\tilde{w}^k \in \Omega\), it follows that

\[
\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \\
\geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0.
\]

The last inequality is due to \(\tilde{w}^k \in \Omega\) and \(w^* \in \Omega^*\) (see (1.3)). Therefore, the right hand side of (2.15) is non-negative and the lemma is proved. \(\square\)

**Lemma 2.2** Let \(\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega\) be generated by (2.11) from the given
\( \mathbf{v}^k = (\mathbf{y}^k, \lambda^k) \). Then, we have

\[
(\mathbf{v}^k - \mathbf{v}^*)^T \mathbf{H}(\mathbf{v}^k - \tilde{\mathbf{v}}^k) \geq \|\mathbf{v}^k - \tilde{\mathbf{v}}^k\|_H^2, \quad \forall \mathbf{v}^* \in \mathcal{V}^*,
\]

where \( \mathcal{M} \) is defined in (2.2).

**Proof.** Recall the matrices \( \mathbf{Q} \) and \( \mathbf{H} \) in (2.2). It follows from (2.14) that

\[
(\tilde{\mathbf{v}}^k - \mathbf{v}^*)^T \mathbf{H}(\mathbf{v}^k - \tilde{\mathbf{v}}^k) \geq 0, \quad \forall \mathbf{v}^* \in \mathcal{V}^*.
\]

Assertion (2.16) follows from the last inequality directly. \( \square \)

The matrix \( \mathbf{H} \) is symmetric and positive semi-definite. We still use \( \|\mathbf{v} - \tilde{\mathbf{v}}\|_H \) to denote that

\[
\|\mathbf{v} - \tilde{\mathbf{v}}\|_H = \sqrt{(\mathbf{v} - \tilde{\mathbf{v}})^T \mathbf{H}(\mathbf{v} - \tilde{\mathbf{v}})}.
\]

If \( \|\mathbf{v}^k - \tilde{\mathbf{v}}^k\|_H^2 = 0 \), because \( \mathbf{H} \) is symmetric and positive semi-definite, we have \( \mathbf{H}(\mathbf{v}^k - \tilde{\mathbf{v}}^k) = 0 \). In this case, \( \tilde{\mathbf{w}}^k \) is a solution of the variational inequality (see (1.2) and (2.8)). Thus, we can take \( \|\mathbf{v}^k - \tilde{\mathbf{v}}^k\|_H^2 \leq \epsilon \) as the stopping criterium in the iteration process.
Theorem 2.1 Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ be generated by (2.11) from the given $v^k = (y^k, \lambda^k)$ and the new iterate $v^{k+1}$ be given by (2.13). Then we have
\[ \|v^{k+1} - v^*\|^2_H \leq \|v^k - v^*\|^2_H - \gamma (2 - \gamma) \|v^k - \tilde{v}^k\|^2_H, \quad \forall \, v^* \in V^*. \quad (2.17) \]

Proof. By a simple manipulation, we obtain

\[
\begin{align*}
\|v^{k+1} - v^*\|^2_H & \overset{(2.16)}{=} \|(v^k - v^*) - \gamma (v^k - \tilde{v}^k)\|^2_H \\
& = \|v^k - v^*\|^2_H - 2\gamma (v^k - v^*)^T H (v^k - \tilde{v}^k) + \gamma^2 \|v^k - \tilde{v}^k\|^2_H \\
& \overset{(2.13)}{\leq} \|v^k - v^*\|^2_H - 2\gamma \|v^k - \tilde{v}^k\|^2_H + \gamma^2 \|v^k - \tilde{v}^k\|^2_H \\
& = \|v^k - v^*\|^2_H - \gamma (2 - \gamma) \|v^k - \tilde{v}^k\|^2_H.
\end{align*}
\]

This is true for any $v^* \in V^*$ and the theorem is proved. \qed

The inequality (2.17) is essential for the convergence of the proposed alternating direction method. The detailed convergence proof can be found in [2]. For the convergence rate of the customized PPA, the reader are refereed to [9].
2.4 Ensure the matrix $H$ to be positive definite

In the ADMM based customized PPA (2.11), the subproblem (2.11c) can be written as

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \frac{\beta}{2} \| By - p^k \|^2 \mid y \in \mathcal{Y}\}, \quad (2.18)$$

where

$$p^k = b + \frac{1}{\beta} \tilde{\lambda}^k - A \tilde{x}^k.$$

If we add an additional term $\frac{\delta \beta}{2} \| B(y - y^k) \|^2$ (with any small $\delta > 0$) to the objective function of the subproblem (2.11c), we will get $\tilde{y}^k$ via

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \frac{\beta}{2} \| By - p^k \|^2 + \frac{\delta \beta}{2} \| B(y - y^k) \|^2 \mid y \in \mathcal{Y}\}.$$

By a manipulation, the solution point of the above subproblem is obtained via

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \frac{(1+\delta)\beta}{2} \| By - q^k \|^2 \mid y \in \mathcal{Y}\}, \quad (2.19)$$

where

$$q^k = \frac{1}{1+\delta}(p^k + \delta y^k).$$
In this way, the matrix $Q$ in (2.12) will be modified to

$$Q = \begin{pmatrix}
0 & 0 \\
(1 + \delta)\beta B^T B & -B^T \\
-B & \frac{1}{\beta} I_m
\end{pmatrix},$$

and the related matrix $H$ in (2.2) becomes

$$H = \begin{pmatrix}
(1 + \delta)\beta B^T B & -B^T \\
-B & \frac{1}{\beta} I_m
\end{pmatrix} = \begin{pmatrix}
\sqrt{\beta} B^T & 0 \\
0 & \sqrt{\frac{1}{\beta} I_m}
\end{pmatrix} \begin{pmatrix}
(1 + \delta)I & -I \\
-I & I_m
\end{pmatrix} \begin{pmatrix}
\sqrt{\beta} B & 0 \\
0 & \sqrt{\frac{1}{\beta} I_m}
\end{pmatrix}. \quad (2.20)$$

Thus, for any $\delta > 0$, $H$ is positive definite when $B$ is a full rank matrix. In other words, instead of (2.18), using (2.19) to get $\tilde{y}^k$, it will ensure the positivity of $H$ theoretically. However, in practical computation, it works still well by using $\delta = 0$. 
3 Application and Numerical Experiments

3.1 Applications to least-squares problems

We consider the following problem:

$$\min \left\{ \frac{1}{2} \| X - C \|_F^2 \mid X \in S^n_+ \cap S_B \right\}, \quad (3.1)$$

where

$$S^n_+ = \{ H \in \mathbb{R}^{n \times n} \mid H^T = H, \ H \succeq 0 \}. \quad (3.2)$$

and

$$S_B = \{ H \in \mathbb{R}^{n \times n} \mid H^T = H, \ H_L \leq H \leq H_U \}. \quad (3.3)$$

Use the following MATLAB Code to produce the matrices $C$, $H_L$ and $H_U$

```matlab
rand(‘state’,0); C=rand(n,n); C=(C’+C)-ones(n,n)+eye(n);
% C is symmetric and C_{ij} is in (-1,1), C_{jj} is in (0,2)
HU=ones(n)*0.1; HL=-HU; for i=1:n HU(i,i)=1; HL(i,i)=1; end;
```
The problem is converted to the following equivalent one:

\[
\min \quad \frac{1}{2} \|X - C\|_2^2 + \frac{1}{2} \|Y - C\|_2^2 \\
\text{s.t} \quad X - Y = 0, \quad X \in S^n_+, \; Y \in S_B.
\]

The basic sub-problems in the ADMM based customized PPA

- For fixed \( Y^k \) and \( Z^k \),
  \[
  \tilde{X}^k = \text{Argmin}\left\{ \frac{1}{2} \|X - C\|_F^2 - \text{Tr}(Z^k X) + \frac{\beta}{2} \|X - Y^k\|_F^2 \mid X \in S^n_+ \right\}
  \]

- Set \( \tilde{Z}^k \) by
  \[
  \tilde{Z}^k = Z^k - \beta(\tilde{X}^k - Y^k).
  \]

- With fixed \( \tilde{X}^k \) and \( \tilde{Z}^k \),
  \[
  \tilde{Y}^k = \text{Argmin}\left\{ \frac{1}{2} \|Y - C\|_F^2 + \text{Tr}(\tilde{Z}^k Y) + \frac{\beta}{2} \|\tilde{X}^k - Y\|_F^2 \mid Y \in S_B \right\}
  \]
\(\tilde{X}^k\) can be directly obtained via

\[
\tilde{X}^k = P_{S^n_+} \left\{ \frac{1}{1 + \beta} (\beta \tilde{Y}^k + Z^k + C) \right\}.
\]  
(3.3)

\[ P_{S^n_+}(A) = U \Lambda^+ U^T, \quad [U, \Lambda] = \text{eig}(A), \quad \Lambda^+ = \max(\Lambda, 0). \]

Similarly, \(\tilde{Y}^k\) is given by

\[
\tilde{Y}^k = P_{S_B} \left\{ \frac{1}{1 + \beta} (\beta \tilde{X}^k - \tilde{Z}^k + C) \right\}. \quad \]  
(3.4)

\[ S_B = \{ H \mid H_L \leq H \leq H_U \}, \quad P_{S_B}(A) = \min(\max(H_L, A), H_U) \]

The most time consuming calculation is \([U, \Lambda] = \text{eig}(A), 9n^3\)
MATLAB Code – An iteration of the classical ADMM

\[ Y_0 = Y; \quad Z_0 = Z; \quad k = k+1; \]
\[ X = (Y_0*beta+Z_0+C)/(1+beta); \quad [V,D] = \text{eig}(X); \quad D = \text{max}(0,D); \]
\[ X = (V*D)*V'; \]
\[ Y = \text{min} \left( \text{max} \left( (X*beta-Z_0+C)/(1+beta), HL \right), HU \right); \]
\[ Z = Z_0-(X-Y)*beta; \]

MATLAB Code – An iteration of the new order ADMM

\[ Y_0 = Y; \quad Z_0 = Z; \quad k = k+1; \]
\[ X = (Y_0*beta+Z_0+C)/(1+beta); \quad [V,D] = \text{eig}(X); \quad D = \text{max}(0,D); \]
\[ X = (V*D)*V'; \quad Z = Z_0-(X-Y_0)*beta; \]
\[ Y = \text{min} \left( \text{max} \left( (X*beta-Z+C)/(1+beta), HL \right), HU \right); \]

MATLAB Code – An iteration of the extended ADMM

\[ Y_0 = Y; \quad Z_0 = Z; \quad k = k+1; \]
\[ X = (Y_0*beta+Z_0+C)/(1+beta); \quad [V,D] = \text{eig}(X); \quad D = \text{max}(0,D); \]
\[ X = (V*D)*V'; \quad Z = Z_0-(X-Y_0)*beta; \]
\[ Y = \text{min} \left( \text{max} \left( (X*beta-Z+C)/(1+beta), HL \right), HU \right); \]
\[ Y = Y_0-(Y_0-Y)*1.5; \]
\[ Z = Z_0-(Z_0-Z)*1.5; \]
**Numerical results for problem (3.1)**

\[ C = \text{rand}(n,n); \quad C = (C' + C) - \text{ones}(n,n) + \text{eye}(n) \]

\[ H_U = \text{ones}(n,n)/10; \quad H_L = -\text{ones}(n,n)/10; \quad H_U(jj) = H_L(jj) = 1. \]

Table 1. Numerical results

<table>
<thead>
<tr>
<th>( n \times n ) Matrix</th>
<th>Classical ADMM</th>
<th>Customized PPA</th>
<th>Extended C-PPA</th>
<th>( \beta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = )</td>
<td>No. It</td>
<td>CPU Sec.</td>
<td>No. It</td>
<td>CPU Sec.</td>
</tr>
<tr>
<td>100</td>
<td>46</td>
<td>1.39</td>
<td>44</td>
<td>1.37</td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>3.07</td>
<td>50</td>
<td>3.05</td>
</tr>
<tr>
<td>500</td>
<td>48</td>
<td>25.50</td>
<td>49</td>
<td>24.52</td>
</tr>
<tr>
<td>800</td>
<td>51</td>
<td>110.18</td>
<td>50</td>
<td>107.29</td>
</tr>
<tr>
<td>1000</td>
<td>51</td>
<td>208.93</td>
<td>52</td>
<td>212.74</td>
</tr>
<tr>
<td>2000</td>
<td>55</td>
<td>1578.96</td>
<td>55</td>
<td>1579.68</td>
</tr>
</tbody>
</table>
3.2 Applications to image restoration

The mathematical form of the image restoration problem is

$$\min \| \| \nabla x \|_1 + \frac{\mu}{2} \| Kx - f \|^2,$$  \hspace{1cm} (3.5)

where $\mu > 0$ is trade-off; $K$ is a blur operator and $f$ is observed image.

**The equivalent problem:**

$$\min \| \| y \|_1 + \frac{\mu}{2} \| Kx - f \|^2$$  \hspace{1cm} (3.6)

s. t. $\nabla x = y$,

This is a problem of form (1.1) where $\mathcal{X}$, $\mathcal{Y}$ are full spaces,

$$\theta_1(x) = \frac{\mu}{2} \| Kx - f \|^2,$$

$$\theta_2(y) = \| \| y \|_1,$$

$$A = \nabla, \quad B = -I \quad \text{and} \quad b = 0.$$
The augmented Lagrangian function

\[ L_A(x, y, \lambda) = \|y\|_1 + \frac{\mu}{2} \|Kx - f\|^2 - \lambda^T (\nabla x - y) + \frac{\beta}{2} \|\nabla x - y\|^2, \]

where \( \lambda \) is Lagrange multiplier and \( \beta \) is the penalty parameter.

For given \((y^k, \lambda^k)\), get \((\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)\) as follows:

1. \(\tilde{x}^k\) is the solution of the following least square problem

\[ \tilde{x}^k = \arg\min_x \left\{ \frac{\mu}{2} \|Kx - f\|^2 - (\lambda^k)^T (\nabla x - y^k) + \frac{\beta}{2} \|\nabla x - y^k\|^2 \right\}. \]

2. Set \(\tilde{\lambda}^k\) by

\[ \tilde{\lambda}^k = \lambda^k - \beta (\nabla \tilde{x}^k - y^k). \]

3. Finally, with fixed \((\tilde{x}^k, \tilde{\lambda}^k), \tilde{y}^k\) are solutions of

\[ \tilde{y}^k = \arg\min_y \left\{ \|y\|_1 - (\tilde{\lambda}^k)^T (\nabla \tilde{x}^k - y) + \frac{\beta}{2} \|\nabla \tilde{x}^k - y\|^2 \right\}. \]
Solving the $x$ subproblem for getting $\tilde{x}^k$:

$$(\beta \nabla^T \nabla + \mu K^T K)\tilde{x}^k = \nabla^T (\beta y^k + \lambda^k) + \mu K^T f.$$  

- If $\nabla$ and $K$ satisfy some periodic boundary conditions, they can be factored by Fourier transform as $\nabla = \mathcal{F}^{-1} \Lambda_D \mathcal{F}$ and $K = \mathcal{F}^{-1} \Lambda_K \mathcal{F}$.

- If $\nabla$ and $K$ satisfy some reflective boundary conditions, they can be factored by discrete cosine transform as $\nabla = \mathcal{C}^{-1} \Lambda_D \mathcal{C}$ and $K = \mathcal{C}^{-1} \Lambda_K \mathcal{C}$.

Solving the $y$ subproblem for getting $\tilde{y}^k$:

$$\tilde{y}^k = \text{shrink}_{\frac{1}{\beta}} \left( \nabla \tilde{x}^k - \frac{\tilde{\lambda}^k}{\beta} \right),$$

where

$$\text{shrink}_c(v) = v - \min(c, \|v\|) \frac{v}{\|v\|}.$$
Note that

\[\text{shrink}_c(v) = v - P_{B^c_2}(v) \quad \text{where} \quad B^c_2 = \{v \in \mathbb{R}^n : \|v\|_2 \leq c\}.\]

**MATLAB Code – An iteration of the classical ADMM**

```matlab
%% step 1 \(x^{(k+1)}\)  
Temp = PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un = real(ifft2(fft2(Temp)./MDu));

%% step 2 \(y^{(k+1)}\)
dxun = Px(un);
dyun = Py(un);
sk1 = dxun - lbd11/beta;
sk2 = dyun - lbd12/beta;
nsk = sqrt(sk1.^2 + sk2.^2); nsk(nsk==0)=1;
nsk = max(1-1./(beta*nsk),0);
vn1 = sk1.*nsk;
vn2 = sk2.*nsk;

%% update \(\lambda\)
lbdn11 = lbd11 - beta*(dxun - vn1);
lbdn12 = lbd12 - beta*(dyun - vn2);

%% New iterative point
u = un; v1 = vn1; v2 = vn2; lbd11 = lbdn11; lbd12 = lbdn12;
```
MATLAB Code – An iteration of the new order ADMM

% step 1 x^{k+1}

Temp = PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un = real(ifft2(fft2(Temp)./MDu));
dxun = Px(un);
dyun = Py(un);

% update \lambda

lbdn11 = lbd11 - beta*(dxun - v1);
lbdn12 = lbd12 - beta*(dyun - v2);

% step 2 y^{k+1}

sk1 = dxun - lbdn11/beta;
sk2 = dyun - lbdn12/beta;
nsk = sqrt(sk1.^2 + sk2.^2);
nsk(nsk==0)=1;
nsk = max(1-1./(beta*nsk),0);
vn1 = sk1.*nsk;
vn2 = sk2.*nsk;

% New iterative point

u = un; v1 = vn1; v2 = vn2; lbd11 = lbdn11; lbd12 = lbdn12;
MATLAB Code – An iteration of the extended C-PPA

%%% step 1 \( x^{(k+1)} \) %%%
Temp = PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un = real(ifft2(fft2(Temp)./MDu));
dxun = Px(un);
dyun = Py(un);

%%% update \( \lambda \) %%%
lbdn11 = lbd11 - beta * (dxun - v1);
lbdn12 = lbd12 - beta * (dyun - v2);

%%% step 2 \( y^{(k+1)} \) %%%
sk1 = dxun - lbd11/beta;
sk2 = dyun - lbd12/beta;
nsk = sqrt(sk1.^2 + sk2.^2); nsk(nsk==0)=1;
nsk = max(1-1./(beta * nsk),0);
v1n = sk1.*nsk;
v2n = sk2.*nsk;

%%% New iterative point %%%
u = un;
v1 = v1 - gamma*(v1-vn1);
v2 = v2 - gamma*(v2-vn2);
lbd11 = lbd11 - gamma*(lbd11-lbdn11);
lbd12 = lbd12 - gamma*(lbd12-lbdn12);
Numerical results for image restoration

\[
I = \text{double}(\text{imread('chart.tiff'))}/255; \quad I = \text{double}(\text{imread('house.png'))}/255; \\
h = \text{fspecial('disk',7);} \quad x0 = \text{imfilter}(I,h,'circular')+0.02*\text{randn(size(I))};
\]

**Figure 1:** Original and degraded images. Left: Chart. Right: House
Figure 2: Performances of ADMM and two variants methods on TV-l2. Left: Chart. Right: House.
Figure 3: Performances of Algorithm 2 with different values of $\gamma$ for Chart. Top: fixed $\gamma$. Bottom: random generated $\gamma$. 
Figure 4: Restorations. From left column to right column: ADMM, new order ADMM, and the extended new order ADMM.
Table 1: Numerical comparisons of the classical ADMM (ADMM), the customized PPA and the extended customized PPA for TV-$l^2$ image restoration.

<table>
<thead>
<tr>
<th></th>
<th>Chart</th>
<th>House</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADMM</td>
<td>C-PPA</td>
</tr>
<tr>
<td>$\gamma = 1$</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>$\gamma = 1.8$</td>
<td>2.32</td>
<td>2.29</td>
</tr>
<tr>
<td>SNR</td>
<td>19.01</td>
<td>19.01</td>
</tr>
</tbody>
</table>

SNR = $20 \log_{10} \frac{\|x\|}{\|x-I\|}$, where $x$ is restoration and $I$ is original image.

It seems that the new order ADMM is so good as the classical one. However, the extended-ADMM converges much faster than the other both ADMMs.

**Remark**

For solving the structured convex optimization problem (1.1), the
classical alternating direction method is described in (1.4) and then the new iterate is updated by \((y^{k+1}, \lambda^{k+1}) = (\tilde{y}^k, \tilde{\lambda}^k)\).

In [7], it was shown that the ADMM is the application of the Douglas-Rachford splitting method [13] to the dual of (1.1). Then, in [4], Eckstein and Bertsekas demonstrated that the Douglas-Rachford splitting method is a special form of the proximal point algorithm (PPA) in [14], and inspired by the relaxed PPA in [8], they proposed the generalized alternating direction method of multipliers

\[
\begin{align*}
  x^{k+1} &= \arg\min \left\{ \theta_1(x) - x^T A^T \lambda^k + \frac{\beta}{2} \| A x + B y^k - b \|^2 \mid x \in X \right\}, \\
  y^{k+1} &= \arg\min \left\{ \theta_2(y) - y^T B^T \lambda^k + \frac{\beta}{2} \| [\alpha A x^{k+1} - (1 - \alpha)(B y^k - b)] + B y - b \|^2 \mid y \in Y \right\}, \\
  \lambda^{k+1} &= \lambda^k - \beta \left\{ [\alpha A x^{k+1} - (1 - \alpha)(B y^k - b)] + B y^{k+1} - b \right\},
\end{align*}
\]

(3.7)

where the parameter \(\alpha \in (0, 2)\) is a relaxation factor. The numerical efficiency of the recursion (3.7) with an over-relaxed choice of \(\alpha\), especially \(\alpha \in [1.5, 1.8]\) empirically, has been shown in [5, 6]. Some of young researcher told me that the
numerical behaviors of the customized PPA based ADMM (2.11)-(2.13) are almost the same as the relaxed ADMM (3.7). It seems possible to prove the equivalence of the two methods [3]. We emphasize here that the explanation of this lecture is in the frame of our lecture series. Use such explanation, it is easy to prove the contraction and the $O(1/t)$ convergence rate of the customized PPA based ADMM and its linearized variant, for details, see the next lecture.

References


