Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2014)

License Information for Mathematics of Data Slides

Outline

- Today
  1. Convex constrained optimization and motivating examples
  2. Optimality condition
  3. Conjugate functions
  4. Monotone inclusion and monotone mixed variational inequality formulations
  5. Chambolle-Pock's primal-dual method
  6. Primal-dual hybrid gradient method
  7. Splitting methods
  8. Model-based excessive gap primal-dual method
- Next week
  1. Disciplined convex programming

Motivation

Motivation

- **Primal-dual convex optimization methods** are powerful for solving saddle point problems as well as constrained convex optimization problems.
- This lecture aims at presenting some emerging primal-dual methods which have been recently used to solve many practical problems in signal/image processing, machine learning and statistics.
- This lecture is a continuation of Lecture 7.

Motivating example: image denoising via anisotropicTV-norm

We consider an image denoising problem with anisotropic total variation norm:

- Given a noisy image $\mathbf{b} \in \mathbb{R}^{m \times n}$. The goal is to recover a clean image from $\mathbf{b}$ using anisotropic total variation norm.

Original image

Noisy image

Motivating example: image denoising via anisotropicTV-norm

We consider an image denoising problem with anisotropic total variation norm:

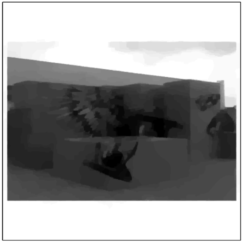
- Given a noisy image $\mathbf{b} \in \mathbb{R}^{m \times n}$. The goal is to recover a clean image from $\mathbf{b}$ using anisotropic total variation norm.

Original image

Noisy image

Anisotropic TV denoising ($\rho = 20$)

Anisotropic TV denoising ($\rho = 40$)



Motivating example: image denoising via anisotropicTV-norm

We consider an image denoising problem with anisotropic total variation norm:

- Given a noisy image $\mathbf{b} \in \mathbb{R}^{m \times n}$. The goal is to recover a clean image from $\mathbf{b}$ using anisotropic total variation norm.
- **This problem** can be formulated as a convex optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^{m \times n}} \frac{1}{2}\|\mathbf{x} - \mathbf{b}\|_F^2 + \rho\|\mathbf{D}\mathbf{x}\|_1, \tag{1}$$

  where $\rho > 0$ is a regularization parameter and $\mathbf{D}$ is a given matrix representing the total variation of $\mathbf{x}$.

There are different ways to reformulate problem (1), for example:

- Since $\|\mathbf{z}\|_1 = \max_{\|\mathbf{u}\|_\infty \leq 1} \mathbf{u}^T\mathbf{z}$, we can reformulate (1) as a saddle point problem:

$$\min_{\mathbf{x} \in \mathbb{R}^{m \times n}} \max_{\|\mathbf{u}\|_\infty \leq 1} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{b}\|_F^2 + \rho\mathbf{u}^T\mathbf{D}\mathbf{x} \right\},$$

- We can also reformulate (1) as a constrained convex minimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x},\mathbf{r}} & \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{b}\|_F^2 + \rho\|\mathbf{r}\|_1 \right\}, \\ \text{s.t.} & \mathbf{D}\mathbf{x} - \mathbf{r} = 0. \end{array}$$

In this lecture, we present several emerging methods to solve both the saddle point formulation and the constrained formulation of (1).

Mathematical form of constrained convex optimization

Constrained convex optimization setting

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \tag{2}$$

- $f \in \mathcal{F}(\mathbb{R}^p)$ is a proper, closed and convex function (see Lecture 2).
- $\mathbf{A} \in \mathbb{R}^{n\times p}$ is full-row rank $(n < p)$, $\mathbf{b} \in \mathbb{R}^n$.

We can incorporating constraints $\mathbf{x} \in \mathcal{X}$ for a given closed and convex set $\mathcal{X}$ via its indicator function $\iota_{\mathcal{X}}$, i.e.:

$$\boxed{f(\mathbf{x}) \leftarrow f(\mathbf{x}) + \iota_{\mathcal{X}}(\mathbf{x})} \quad \text{where} \quad \iota_{\mathcal{X}}(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{X} \\ +\infty & \text{otherwise.} \end{cases}$$

Mathematical form of constrained convex optimization

Constrained convex optimization setting

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \tag{2}$$

- $f \in \mathcal{F}(\mathbb{R}^p)$ is a proper, closed and convex function (see Lecture 2).
- $\mathbf{A} \in \mathbb{R}^{n\times p}$ is full-row rank $(n < p)$, $\mathbf{b} \in \mathbb{R}^n$.

We can incorporating constraints $\mathbf{x} \in \mathcal{X}$ for a given closed and convex set $\mathcal{X}$ via its indicator function $\iota_{\mathcal{X}}$, i.e.:

$$\boxed{f(\mathbf{x}) \leftarrow f(\mathbf{x}) + \iota_{\mathcal{X}}(\mathbf{x})} \quad \text{where} \quad \iota_{\mathcal{X}}(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{X} \\ +\infty & \text{otherwise.} \end{cases}$$

Common structures

As in Lecture 7, methods presented in this lecture also rely on the two common structures:

- **Decomposability** of f.
- **Tractable proximity** of f.

Structures of constrained convex optimization problems

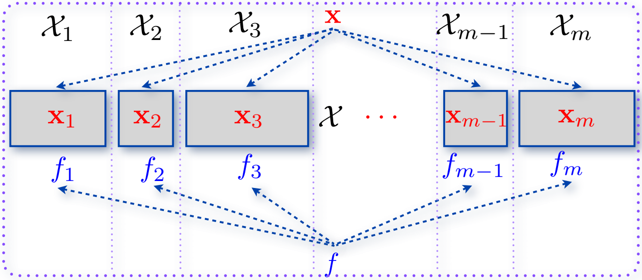
Decomposable structure

The function f can be decomposed as

$$f(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}_i) \tag{3}$$

where $m \geq 1$ is the number of components, $\mathbf{x}_i$ is a sub-vector (component) of $\mathbf{x}$, $f_i : \mathbb{R}^{p_i} \to \mathbb{R} \cup \{+\infty\}$ is convex and $\sum_{i=1}^{m} p_i = p$.

Special case: $m = 2$, which already covers many important instances (see Lecture 7).



Structures of constrained convex optimization problems

Tractable proximity

Each **component** f_i has a **"tractablly proximal operator"** $(i = 1, \ldots, m)$, i.e.:

$$\boxed{\mathrm{prox}_{f_i}(\mathbf{x}_i) := \underset{\mathbf{z}_i \in \mathbb{R}^{p_i}}{\mathrm{argmin}} \left\{ f_i(\mathbf{z}_i) + (1/2)\|\mathbf{z}_i - \mathbf{x}_i\|_2^2 \right\}} \quad (4)$$

can be solved "efficiently":

- (4) has a closed form solution (with low computational cost)
- (4) can be solved in polynomial time.

Example (Tractable proximity functions)

- One-variable functions
  - Smooth functions, e.g., $f(x) := x - 2\log(1 + x)$
  - Nonsmooth functions, e.g., $f(x) := |x|$
- Separable functions, e.g., $f(\mathbf{x}) := \sum_{i=1}^{p} \|\mathbf{x}_i\|_2$, where $\mathbf{x} := (\mathbf{x}_1^T, \cdots, \mathbf{x}_p^T)^T$.
- The indicator function ι of boxes, cones ($\mathbb{R}_+^p$, $\mathbb{S}_+^p$ and Lorentz cone) and simplex.
- More examples can be found in Lectures 4 and 5.

Outline

- Today
  1. Convex constrained optimization and motivating examples
  2. Optimality condition
  3. Conjugate functions
  4. Monotone inclusion and monotone mixed variational inequality formulations
  5. Chambolle-Pock's primal-dual method
  6. Primal-dual hybrid gradient method
  7. Splitting methods
  8. Model-based excessive gap primal-dual method
- Next week
  1. Disciplined convex programming

Solutions and solution set

Optimal solutions and optimal solution set

We define the feasible set of (2): (cf. $f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}\}$) as:

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^p \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\}.$$

A feasible point $\mathbf{x}^\star \in \mathcal{D}$ is called a globally optimal solution (or solution) of (2) if

$$f(\mathbf{x}^\star) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}.$$

All solutions of (2) forms the solution set $\mathcal{X}^\star$ of (2).

- The solution set $\mathcal{X}^\star$ is closed and convex.
- Numerical solution methods often try to find an approximation $\mathbf{x}^\star_\epsilon$ of one solution $\mathbf{x}^\star \in \mathcal{X}^\star$ in the following sense:

Solutions and solution set

Optimal solutions and optimal solution set

We define the feasible set of (2): (cf. $f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}\}$) as:

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^p \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\}.$$

A feasible point $\mathbf{x}^\star \in \mathcal{D}$ is called a globally optimal solution (or solution) of (2) if

$$f(\mathbf{x}^\star) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}.$$

All solutions of (2) forms the solution set $\mathcal{X}^\star$ of (2).

- The solution set $\mathcal{X}^\star$ is closed and convex.
- Numerical solution methods often try to find an approximation $\mathbf{x}^\star_\epsilon$ of one solution $\mathbf{x}^\star \in \mathcal{X}^\star$ in the following sense:

Approximate solution

Given a tolerance $\epsilon \geq 0$, a point $\mathbf{x}^\star_\epsilon \in \mathbb{R}^p$ is called an ϵ-solution of (2) if

$$\begin{cases} |f(\mathbf{x}^\star_\epsilon) - f^\star| \leq \epsilon & \text{(objective residual)}, \\ \|\mathbf{A}\mathbf{x}^\star_\epsilon - \mathbf{b}\| \leq \epsilon & \text{(feasibility gap)}. \end{cases}$$

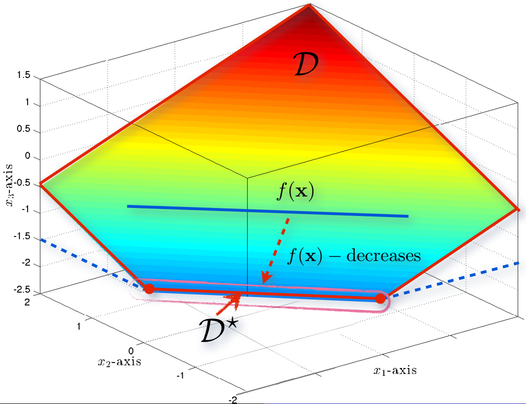
Remark: we can use different tolerances for the objective residual and feasibility gap.

Example: Feasible set and solution set

Consider a constrained convex problem:

$$\min_{\mathbf{x}\in\mathbb{R}^3} \quad \{f(\mathbf{x}) := x_3\},$$
$$\text{s.t.} \quad 2x_3 - x_1 - x_2 = -1,$$
$$\mathbf{x} \in \mathcal{X} := [-2,2]\times[-2,2]\times[-1.5,1.5].$$

The **feasible set** $\mathcal{D} := \{\mathbf{x}\in\mathbb{R}^3 : 2x_3 - x_1 - x_2 = -1, \mathbf{x}\in\mathcal{X}\}$ and the **solution set** $\mathcal{D}^\star$ of this problem are plotted in the figure below. $\mathcal{D}^\star$ is in fact a segment (many solutions), which is the lowest edge of the polytope $\mathcal{D}$.



Optimality condition

Lagrange function

$$\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

Here, $\lambda \in \mathbb{R}^n$ is the vector of Lagrange multipliers (or dual variables).

Optimality condition

The **optimality condition** of (2) can be written as

$$\begin{cases} 0 & \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star), \\ 0 & = \mathbf{A}\mathbf{x}^\star - \mathbf{b}. \end{cases} \tag{5}$$

Here the subdifferential of f at $\mathbf{x}^\star$ is defined as (see Lecture 2):

$$\partial f(\mathbf{x}^\star) := \{\mathbf{z} \in \mathbb{R}^p \ : \ f(\mathbf{y}) \geq f(\mathbf{x}^\star) + \mathbf{z}^T(\mathbf{y} - \mathbf{x}^\star), \ \forall \mathbf{y} \in \mathbb{R}^p\}.$$

- The condition (5) is the KKT (Karush-Kuhn-Tucker) condition.
- Any point $(\mathbf{x}^\star, \lambda^\star)$ satisfying (5) is called a KKT point.
- $\mathbf{x}^\star$ is called a stationary point and $\lambda^\star$ is the corresponding multipliers.

Conjugation of functions

- **Duality** is a central concept in optimization, especially in convex optimization.
- We review the notion of Fenchel's conjugate function and its basic properties which will be used to define the dual problem.
- We limit our definition to the class of convex functions $f \in \mathcal{F}(\mathbb{R}^p)$.

Definition

Let $\mathcal{Q}$ be a predefined Euclidean space and Q^* be its dual space. Given a proper, closed and convex function $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$, the function $f^* : Q^* \to \mathbb{R} \cup \{+\infty\}$ such that

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \text{dom}(f)} \left\{ \mathbf{y}^T \mathbf{x} - f(\mathbf{x}) \right\}$$

is called the Fenchel conjugate (or conjugate) of f.

Conjugation of functions

- **Duality** is a central concept in optimization, especially in convex optimization.
- We review the notion of Fenchel's conjugate function and its basic properties which will be used to define the dual problem.
- We limit our definition to the class of convex functions $f \in \mathcal{F}(\mathbb{R}^p)$.

Definition

Let $\mathcal{Q}$ be a predefined Euclidean space and Q^* be its dual space. Given a proper, closed and convex function $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$, the function $f^* : Q^* \to \mathbb{R} \cup \{+\infty\}$ such that

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathrm{dom}(f)} \left\{ \mathbf{y}^T \mathbf{x} - f(\mathbf{x}) \right\}$$

is called the Fenchel conjugate (or conjugate) of f.

- f^* is a convex and lower, semicontinuous function by construction (as the supremum of affine functions of $\mathbf{y}$).
- The conjugate of the conjugate of a convex function f is ... the same function f; i.e., $f^{**} = f$ for $f \in \mathcal{F}(\mathcal{Q})$.

Two basic properties of the function and its conjugation

Lemma (Fenchel-Young inequality (Property 1))

Let $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$ *and* $f^* : Q^* \to \mathbb{R} \cup \{+\infty\}$ *be a function and its conjugation; here* Q^* *be the dual space of* $\mathcal{Q}$*. Then, the following inequality holds true:*

$$f(\mathbf{x}) + f^*(\mathbf{y}) \geq \mathbf{x}^T\mathbf{y}, \quad \forall \mathbf{x} \in Q, \mathbf{y} \in Q^*.$$

Two basic properties of the function and its conjugation

Lemma (Fenchel-Young inequality (Property 1))

Let $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$ and $f^* : Q^* \to \mathbb{R} \cup \{+\infty\}$ be a function and its conjugation; here Q^* be the dual space of $\mathcal{Q}$. Then, the following inequality holds true:

$$f(\mathbf{x}) + f^*(\mathbf{y}) \geq \mathbf{x}^T\mathbf{y}, \quad \forall \mathbf{x} \in Q, \mathbf{y} \in Q^*.$$

- Since f^* is proper, closed and convex, its subdifferential ∂f^* exists for any $\mathbf{y}$ in the relative interior of its domain.
- For $f \in \mathcal{F}(\mathcal{Q})$, if the subdifferential of f and f^* exists, then we have the following relation:

Lemma (Subgradient property (Property 2))

Let $\mathbf{y} \in \partial f(\mathbf{x})$ for some $\mathbf{x} \in dom(f)$. Then $\mathbf{y} \in dom(f^*)$ and vise versa. Moreover, we have

$$\mathbf{u} \in \partial f(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \partial f^*(\mathbf{u}).$$

Conjugation of functions

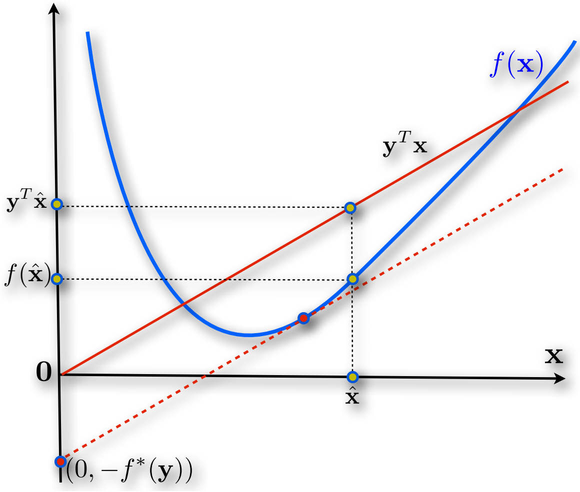


Figure: The conjugate function $f^*(\mathbf{y})$ is the maximum gap between the linear function $\mathbf{x}^T\mathbf{y}$ (red line) and $f(\mathbf{x})$, as shown in dashed line.

Example 1: Convex quadratic function

Example (Strictly convex quadratic function)

- Given a symmetric positive definite matrix $\mathbf{\Phi}$, $(\mathbf{\Phi} \succ 0)$ and a vector $\mathbf{b}$.
- Let $f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{\Phi}\mathbf{x} + \mathbf{b}^T\mathbf{x}$ be a quadratic function for $\mathbf{x} \in \mathbb{R}^p$.
- It is clear that $\text{dom}(f) = \mathbf{R}^p$.
- By definition,

$$f^*(\mathbf{y}) := \sup_{\mathbf{x}\in\mathbb{R}^p} \left\{ \mathbf{y}^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\mathbf{\Phi}\mathbf{x} - \mathbf{b}^T\mathbf{x} \right\}.$$

- Since this is an unconstrained convex problem, the maximum is attained when $\mathbf{\Phi}\mathbf{x}^* + \mathbf{b} = \mathbf{y}$, which leads to $\mathbf{x}^\star = \mathbf{\Phi}^{-1}(\mathbf{y} - \mathbf{b})$.
- Hence, we have

$$f^*(\mathbf{y}) = \frac{1}{2}(\mathbf{y} - \mathbf{b})^T\mathbf{\Phi}^{-1}(\mathbf{y} - \mathbf{b}) = \frac{1}{2}\mathbf{y}^T\mathbf{\Phi}^{-1}\mathbf{y} - (\mathbf{\Phi}^{-1}\mathbf{b})^T\mathbf{y} + \frac{1}{2}\mathbf{b}^T\mathbf{\Phi}^{-1}\mathbf{b},$$

  which is again a convex quadratic function with $\text{dom}(f^*) = \mathbb{R}^p$.

- Since $\nabla f(\mathbf{x}) = \mathbf{\Phi}\mathbf{x} + \mathbf{b} := \mathbf{u}$ and $\nabla f^*(\mathbf{y}) = \mathbf{\Phi}^{-1}(\mathbf{y} - \mathbf{b})$, we can see that $\mathbf{x} = \mathbf{\Phi}^{-1}(\mathbf{u} - \mathbf{b}) = \nabla f^*(\mathbf{u})$.

Example 2: Log-determinant

Example (Log-determinant function)

- Let $f(\mathbf{X}) := -\log\det(\mathbf{X})$, where $\mathsf{dom}(f) \equiv \mathbb{S}^p_{++}$.
- By definition, we have

$$f^*(\mathbf{Y}) = \sup_{\mathbf{X}\in\mathsf{dom}(f)} \left\{\mathsf{tr}(\mathbf{YX}) + \log\det(\mathbf{X})\right\},$$

- One can show that the above is unbounded above unless $\mathbf{Y} \prec 0$.
- To find the maximum of the above problem, we have:

$$\nabla\left(\mathsf{tr}(\mathbf{YX}) + \log\det(\mathbf{X})\right) = 0 \Rightarrow \mathbf{X}^\star = -\mathbf{Y}^{-1},$$

and thus,

$$f^*(\mathbf{Y}) = -\log\det\left(-\mathbf{Y}\right) - p, \quad \text{with} \quad \mathsf{dom}(f^*) = -\mathbb{S}^p_{++}.$$

- Since $\nabla f(\mathbf{X}) = -\mathbf{X}^{-1} := \mathbf{U}$ and $\nabla f^*(\mathbf{Y}) = -\mathbf{Y}^{-1}$, we have $\mathbf{X} = -\mathbf{U}^{-1} = \nabla f^*(\mathbf{U})$.

Conjugation of functions

Example

$f(\mathbf{x})$	dom(f)/dom(f^*)	$f^*(\mathbf{y})$
$f(\alpha\mathbf{x})$ (where $a \neq 0$)	$\mathcal{Q}/\mathcal{Q}^*$	$f^*\left(\frac{\mathbf{y}}{\alpha}\right)$
$f(\mathbf{x}+\beta)$	$\mathcal{Q}/\mathcal{Q}^*$	$f^*(\mathbf{y}) - \langle\beta, \mathbf{y}\rangle$
$\alpha f(\mathbf{x})$ (where $\alpha > 0$)	$\mathcal{Q}/\mathcal{Q}^*$	$\alpha f^*\left(\frac{\mathbf{y}}{\alpha}\right)$
$\frac{\\|\mathbf{x}\\|^r}{r}$ (where $r > 1$)	$\mathbb{R}^p/\mathbb{R}^p$	$\frac{\\|\mathbf{y}\\|^q}{q}$ (where $\frac{1}{r}+\frac{1}{q}=1$)
$-\log(x)$	$\mathbb{R}_{++}/\mathbb{R}_{--}$	$-(1+\log(y))$
e^x	$\mathbb{R}/\mathbb{R}_+$	$\begin{cases} y\log(y) - y & \text{if } y > 0 \\ 0 & \text{if } y = 0 \end{cases}$

Table: Legendre transforms (conjugations) for many common functions as well as a few useful properties.

Dual problem

- In Lecture 7 we have used the Lagrange duality theory to present **methods of multipliers**.
- In this lecture, we use the Frenchel duality theory to define the **dual problem** formulation and develop **primal-dual methods** for solving (2).

Dual problem

- In Lecture 7 we have used the Lagrange duality theory to present **methods of multipliers**.
- In this lecture, we use the Frenchel duality theory to define the **dual problem** formulation and develop **primal-dual methods** for solving (2).

Dual formulation

- From the optimality condition (5): $\begin{cases} 0 & \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star) \\ 0 & = \mathbf{A}\mathbf{x}^\star - \mathbf{b} \end{cases}$ we have

$$-\mathbf{A}^T\lambda^\star \in \partial f(\mathbf{x}^\star).$$

- The last expression is equivalent to $\mathbf{x}^\star \in \partial f^*(-\mathbf{A}^T\lambda^\star)$ (see Property 2).
- Since $\mathbf{A}\mathbf{x}^\star - \mathbf{b} = 0$, using $\mathbf{x}^\star \in \partial f^*(-\mathbf{A}^T\lambda^\star)$, we have

$$\boxed{0 \in \mathbf{A}\partial f^*(-\mathbf{A}^T\lambda^\star) - \mathbf{b}.} \tag{6}$$

(6) is exactly the optimality condition of

$$\boxed{\max_{\lambda\in\mathbb{R}^n} \left\{-f^*(-\mathbf{A}^T\lambda) - \mathbf{b}^T\lambda\right\}.} \tag{7}$$

(7) is still a convex problem and equivalent to $\min_\lambda\{f^*(-\mathbf{A}^T\lambda) + \mathbf{b}^T\lambda\}$.

Decomposable structure

Dual problem of the decomposable objective function

If $f(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}_i)$ then

$$\max_{\lambda \in \mathbb{R}^n} \Big\{ -\sum_{i=1}^{m} f_i^*(\mathbf{A}_i^T \lambda) - \mathbf{b}^T \lambda \Big\}.$$

where $\mathbf{A} \equiv [\mathbf{A}_1, \cdots, \mathbf{A}_m]$.

Note: The evaluation of the **dual objective function** and its **gradient** can be computed in **parallel**.

Decomposable structure

Dual problem of the decomposable objective function

If $f(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}_i)$ then

$$\max_{\lambda \in \mathbb{R}^n} \Big\{ -\sum_{i=1}^{m} f_i^*(\mathbf{A}_i^T \lambda) - \mathbf{b}^T \lambda \Big\}.$$

where $\mathbf{A} \equiv [\mathbf{A}_1, \cdots, \mathbf{A}_m]$.

Note: The evaluation of the **dual objective function** and its **gradient** can be computed in **parallel**.

Dual formulation of empirical risk minimization

An empirical risk minimization problem can be expressed as

$$\min_{\mathbf{z} \in \mathbb{R}^n} \sum_{i=1}^{m} f_i(\mathbf{A}_i^T \mathbf{z} + \mathbf{b}_i).$$

Its Fenchel dual problem therefore can be written as:

$$\min_{\mathbf{x} \in \mathbb{R}^p} \sum_{i=1}^{m} \Big\{ f_i^*(\mathbf{x}_i) - \mathbf{b}_i^T \mathbf{x}_i \Big\} \quad \text{s.t.} \quad \sum_{i=1}^{m} \mathbf{A}_i \mathbf{x}_i = 0.$$

Example: Dual problem of the basis pursuit

Basis pursuit

Consider the following basis pursuit problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & \left\{ f(\mathbf{x}) := \|\mathbf{x}\|_1 = \sum_{i=1}^{p} |x_i| \right\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

Example: Dual problem of the basis pursuit

Basis pursuit

Consider the following basis pursuit problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & \left\{ f(\mathbf{x}) := \|\mathbf{x}\|_1 = \textstyle\sum_{i=1}^p |x_i| \right\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

Dual problem of basis pursuit

- The Fenchel dual function $f^*(\mathbf{u}) := \sup_{\mathbf{x}\in\mathbb{R}^p} \{\mathbf{u}^T\mathbf{x} - \|\mathbf{x}\|_1\}$ becomes

$$f^*(\mathbf{u}) = \iota_{\{\|\mathbf{u}\|_\infty \le 1\}}(\mathbf{u}) = \begin{cases} 0 & \text{if } \|\mathbf{u}\|_\infty \le 1, \\ +\infty & \text{otherwise.} \end{cases}$$

- The dual problem

$$\max_{\lambda} \left\{ -\iota_{\{\|\mathbf{A}^T\lambda\|_\infty \le 1\}}(-\mathbf{A}^T\lambda) - \mathbf{b}^T\lambda \right\}$$

- Equivalent expression:

$$\boxed{\min_{\lambda\in\mathbb{R}^n} \mathbf{b}^T\lambda \;\; \text{s.t.} \;\; \|\mathbf{A}^T\lambda\|_\infty \le 1.}$$

Set-valued mappings

In the previous slide, we have seen the subdifferential $\partial f(\mathbf{x})$ of a convex function f at a given point $\mathbf{x}$:

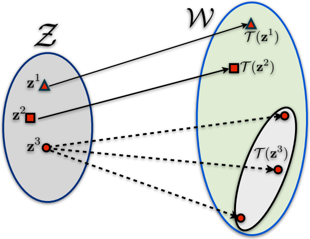
$$\partial f(\mathbf{x}) := \text{set of all subgradients of } f \text{ at } \mathbf{x}$$

Example: Subdifferential of $|\mathbf{x}|$ in $\mathbb{R}$:

$$\mathcal{T}(\mathbf{x}) = \partial|\mathbf{x}| = \begin{cases} +1 & \text{if } x > 0 \quad \text{(single value)}, \\ -1 & \text{if } x < 0 \quad \text{(single value)}, \\ [-1, 1] & \text{if } x = 0 \quad \text{(multiple values)}. \end{cases}$$

Set-valued mappings:

- For any convex set $\mathcal{W}$, we denote by $2^{\mathcal{W}}$ the set of all subsets of $\mathcal{W}$.
- $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{W}}$ is a set-valued mapping if for $\mathbf{z} \in \mathcal{Z}$, $\mathcal{T}(\mathbf{z})$ is a subset in $\mathcal{W}$.



Roughly speaking, a mapping that produces more than one output values for at least one input is called a set-valued mapping.

Monotone operators

For a **set-valued mapping** $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$, we define

- The domain of $\mathcal{T}$ as $\mathrm{dom}(\mathcal{T}) := \{\mathbf{z} \in \mathcal{Z} \ : \ \mathcal{T}(\mathbf{z}) \neq \emptyset\}$.
- The graph of $\mathcal{T}$ is $\mathrm{graph}(\mathcal{T}) := \{(\mathbf{z}, \mathbf{v}) \ : \ \mathbf{v} \in \mathcal{T}(\mathbf{z}), \ \mathbf{z} \in \mathsf{dom}(\mathcal{T})\}$.

Monotone operators

For a **set-valued mapping** $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$, we define

- The domain of $\mathcal{T}$ as $\mathrm{dom}(\mathcal{T}) := \{\mathbf{z} \in \mathcal{Z} \ : \ \mathcal{T}(\mathbf{z}) \neq \emptyset\}$.
- The graph of $\mathcal{T}$ is $\mathrm{graph}(\mathcal{T}) := \{(\mathbf{z}, \mathbf{v}) \ : \ \mathbf{v} \in \mathcal{T}(\mathbf{z}), \ \mathbf{z} \in \mathrm{dom}(\mathcal{T})\}$.

Monotonicity can be considered as an equivalent property of convexity acting on the differential ∇f or subdifferential ∂f of the function instead of the function value.

For smooth and convex function, monotonicity of ∇f means that

$$(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) \geq 0, \quad \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathrm{dom}(f).$$

This inequality is the sum of $f(\mathbf{x}) \geq f(\hat{\mathbf{x}}) + \nabla f(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})$ and $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})$.

Monotone operators

For a **set-valued mapping** $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$, we define

- The domain of $\mathcal{T}$ as $\mathrm{dom}(\mathcal{T}) := \{\mathbf{z} \in \mathcal{Z} \ : \ \mathcal{T}(\mathbf{z}) \neq \emptyset\}$.
- The graph of $\mathcal{T}$ is $\mathrm{graph}(\mathcal{T}) := \{(\mathbf{z}, \mathbf{v}) \ : \ \mathbf{v} \in \mathcal{T}(\mathbf{z}), \ \mathbf{z} \in \mathrm{dom}(\mathcal{T})\}$.

Monotonicity can be considered as an equivalent property of convexity acting on the differential ∇f or subdifferential ∂f of the function instead of the function value.

For smooth and convex function, monotonicity of ∇f means that

$$(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) \geq 0, \quad \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathrm{dom}(f).$$

This inequality is the sum of $f(\mathbf{x}) \geq f(\hat{\mathbf{x}}) + \nabla f(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})$ and $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})$.

Monotonicity

Given a convex set $\mathcal{Z}$ and a set-valued mapping $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$.

- $\mathcal{T}$ is called μ-strongly monotone on $\mathcal{Z}$ if for any $\mathbf{z}$ and $\hat{\mathbf{z}}$ in $\mathcal{Z}$:

$$\boxed{(\mathbf{u} - \hat{\mathbf{u}})^T(\mathbf{z} - \hat{\mathbf{z}}) \geq \mu\|\mathbf{z} - \hat{\mathbf{z}}\|^2, \ \forall \mathbf{u} \in \mathcal{T}(\mathbf{z}), \hat{\mathbf{u}} \in \mathcal{T}(\hat{\mathbf{z}}).}$$

- If $\mu = 0$, then we say that $\mathcal{T}$ is monotone.
- If $\mu > 0$, then we say that $\mathcal{T}$ is strongly monotone with the parameter μ.

Monotone operators

For a **set-valued mapping** $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$, we define

- The domain of $\mathcal{T}$ as $\mathrm{dom}(\mathcal{T}) := \{\mathbf{z} \in \mathcal{Z} \ : \ \mathcal{T}(\mathbf{z}) \neq \emptyset\}$.
- The graph of $\mathcal{T}$ is $\mathrm{graph}(\mathcal{T}) := \{(\mathbf{z}, \mathbf{v}) \ : \ \mathbf{v} \in \mathcal{T}(\mathbf{z}), \ \mathbf{z} \in \mathrm{dom}(\mathcal{T})\}$.

Monotonicity can be considered as an equivalent property of convexity acting on the differential ∇f or subdifferential ∂f of the function instead of the function value.

For smooth and convex function, monotonicity of ∇f means that

$$(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) \geq 0, \quad \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathrm{dom}(f).$$

This inequality is the sum of $f(\mathbf{x}) \geq f(\hat{\mathbf{x}}) + \nabla f(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})$ and $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})$.

Monotonicity

Given a convex set $\mathcal{Z}$ and a set-valued mapping $\mathcal{T} : \mathcal{Z} \rightrightarrows 2^{\mathcal{Z}}$.

- $\mathcal{T}$ is called μ-strongly monotone on $\mathcal{Z}$ if for any $\mathbf{z}$ and $\hat{\mathbf{z}}$ in $\mathcal{Z}$:

$$\boxed{(\mathbf{u} - \hat{\mathbf{u}})^T(\mathbf{z} - \hat{\mathbf{z}}) \geq \mu\|\mathbf{z} - \hat{\mathbf{z}}\|^2, \ \forall \mathbf{u} \in \mathcal{T}(\mathbf{z}), \hat{\mathbf{u}} \in \mathcal{T}(\hat{\mathbf{z}}).}$$

- If $\mu = 0$, then we say that $\mathcal{T}$ is monotone.
- If $\mu > 0$, then we say that $\mathcal{T}$ is strongly monotone with the parameter μ.

If $\mathcal{T}$ is single-valued, then the condition reduces to

$$(\mathbf{z} - \hat{\mathbf{z}})^T(\mathcal{T}(\mathbf{z}) - \mathcal{T}(\hat{\mathbf{z}})) \geq \mu\|\mathbf{z} - \hat{\mathbf{z}}\|^2, \quad \forall \ \mathbf{z}, \hat{\mathbf{z}} \in \mathcal{Z}.$$

Monotone function vs nonmonotone functions

$(T(\hat{\mathbf{x}}) - T(\mathbf{x}))^T(\hat{\mathbf{x}} - \mathbf{x}) \geq 0$ $\quad$ $(T(\hat{\mathbf{x}}) - T(\mathbf{x}))^T(\hat{\mathbf{x}} - \mathbf{x}) \leq 0$ $\quad$ $(T(\hat{\mathbf{x}}) - T(\mathbf{x}))^T(\hat{\mathbf{x}} - \mathbf{x}) \leq 0$

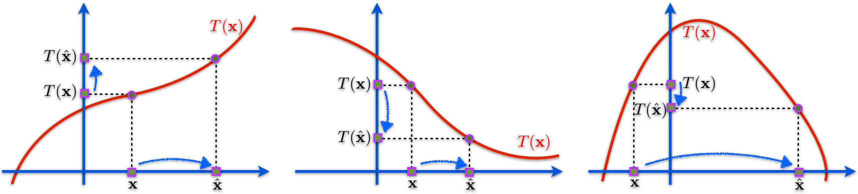


Figure: Monotone function - Nonmonotone function - Nonmonotone function

increasing function - decreasing function -

Maximal monotone operators

Definition (Maximal monotonicity)

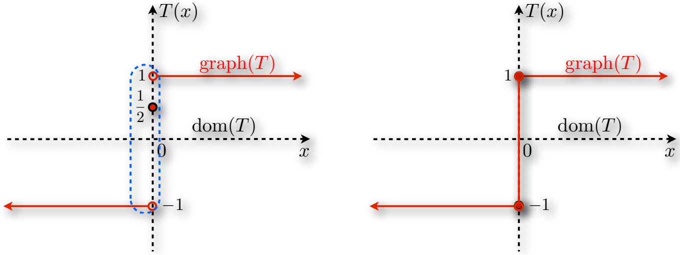
A set-valued mapping $\mathcal{T}$ is called maximal monotone if:

- $\mathcal{T}$ is monotone
- There exists no other monotone mapping $\tilde{\mathcal{T}}$ such that $\mathrm{graph}(\mathcal{T}) \subset \mathrm{graph}(\tilde{\mathcal{T}})$.

Example (Visualization of a maximal monotone operator)

Consider the mapping $\mathcal{T}$ from $\mathbb{R}$ to $2^{\mathbb{R}}$ as follows:

- Nonmaximal monotone (left figure): $T(x) = 1$ if $x > 0$, $T(x) = -1$ if $x < 0$ and $T(x) = 1/2$ if $x = 0 \Rightarrow T$ is not maximal monotone.
- Maximal monotone (right figure): $T(x) = 1$ if $x > 0$, $T(x) = -1$ if $x < 0$ and $T(x) = [-1, 1]$ if $x = 0 \Rightarrow T$ is maximal monotone.



Example of maximal monotone operators

- Affine mapping $\mathcal{T}(\mathbf{z}) := \mathbf{H}\mathbf{z} + \mathbf{h}$ is maximal monotone if $\mathbf{H}$ is positive semidefinite.
  - We have
$$(\mathcal{T}(\mathbf{z}) - \mathcal{T}(\hat{\mathbf{z}}))^T(\mathbf{z} - \hat{\mathbf{z}}) = (\mathbf{z} - \hat{\mathbf{z}})^T\mathbf{H}(\mathbf{z} - \hat{\mathbf{z}}) \geq \sigma_{\min}(\mathbf{H})\|\mathbf{z} - \hat{\mathbf{z}}\|^2,$$
    where $\mu = \sigma_{\min}(\mathbf{H})$ the smallest singular value of $\mathbf{H}$
  - $\mathcal{T}$ is strongly monotone if $\mathbf{H}$ is positive definite.
- Any nondecreasing function $f : \mathbb{R} \to \mathbb{R}$ is monotone.

Example of maximal monotone operators

- Affine mapping $\mathcal{T}(\mathbf{z}) := \mathbf{H}\mathbf{z} + \mathbf{h}$ is maximal monotone if $\mathbf{H}$ is positive semidefinite.
  - We have

$$(\mathcal{T}(\mathbf{z}) - \mathcal{T}(\hat{\mathbf{z}}))^T(\mathbf{z} - \hat{\mathbf{z}}) = (\mathbf{z} - \hat{\mathbf{z}})^T\mathbf{H}(\mathbf{z} - \hat{\mathbf{z}}) \geq \sigma_{\min}(\mathbf{H})\|\mathbf{z} - \hat{\mathbf{z}}\|^2,$$

    where $\mu = \sigma_{\min}(\mathbf{H})$ the smallest singular value of $\mathbf{H}$
  - $\mathcal{T}$ is strongly monotone if $\mathbf{H}$ is positive definite.
- Any nondecreasing function $f : \mathbb{R} \to \mathbb{R}$ is monotone.
- The subdifferential ∂f of a convex function f is maximal monotone.
  - By definition, any $\mathbf{u} \in \partial f(\mathbf{x})$, one has $f(\hat{\mathbf{x}}) - f(\mathbf{x}) \geq \mathbf{u}^T(\hat{\mathbf{x}} - \mathbf{x})$ for any $\hat{\mathbf{x}}$.
  - Similarly, $\hat{\mathbf{u}} \in \partial f(\hat{\mathbf{x}})$, then $f(\mathbf{x}) - f(\hat{\mathbf{x}}) \geq \hat{\mathbf{u}}^T(\mathbf{x} - \hat{\mathbf{x}})$.
  - Summing up these inequalities, we obtain $(\mathbf{u} - \hat{\mathbf{u}})^T(\mathbf{x} - \hat{\mathbf{x}}) \geq 0$.
- If f is strongly convex with the convexity parameter μ, then ∂f is strongly monotone with monotonicity parameter μ.

Example of maximal monotone operators

- Affine mapping $\mathcal{T}(\mathbf{z}) := \mathbf{H}\mathbf{z} + \mathbf{h}$ is maximal monotone if $\mathbf{H}$ is positive semidefinite.
  - We have

$$(\mathcal{T}(\mathbf{z}) - \mathcal{T}(\hat{\mathbf{z}}))^T(\mathbf{z} - \hat{\mathbf{z}}) = (\mathbf{z} - \hat{\mathbf{z}})^T\mathbf{H}(\mathbf{z} - \hat{\mathbf{z}}) \geq \sigma_{\min}(\mathbf{H})\|\mathbf{z} - \hat{\mathbf{z}}\|^2,$$

    where $\mu = \sigma_{\min}(\mathbf{H})$ the smallest singular value of $\mathbf{H}$
  - $\mathcal{T}$ is strongly monotone if $\mathbf{H}$ is positive definite.
- Any nondecreasing function $f : \mathbb{R} \to \mathbb{R}$ is monotone.
- The subdifferential ∂f of a convex function f is maximal monotone.
  - By definition, any $\mathbf{u} \in \partial f(\mathbf{x})$, one has $f(\hat{\mathbf{x}}) - f(\mathbf{x}) \geq \mathbf{u}^T(\hat{\mathbf{x}} - \mathbf{x})$ for any $\hat{\mathbf{x}}$.
  - Similarly, $\hat{\mathbf{u}} \in \partial f(\hat{\mathbf{x}})$, then $f(\mathbf{x}) - f(\hat{\mathbf{x}}) \geq \hat{\mathbf{u}}^T(\mathbf{x} - \hat{\mathbf{x}})$.
  - Summing up these inequalities, we obtain $(\mathbf{u} - \hat{\mathbf{u}})^T(\mathbf{x} - \hat{\mathbf{x}}) \geq 0$.
- If f is strongly convex with the convexity parameter μ, then ∂f is strongly monotone with monotonicity parameter μ.
- The normal cone $\mathcal{N}_{\mathcal{X}}$ of a nonempty, closed and convex set $\mathcal{X}$ is also a monotone mapping
  - Since it is the subdifferential of the indicator function $\iota_{\mathcal{X}}$, which is proper, closed and convex.

Monotonicity of the normal cone

Given a nonempty, closed and convex set $\mathcal{X}$. The **normal cone** of $\mathcal{X}$ at $\mathbf{x}$ is defined as

$$\mathcal{N}_{\mathcal{X}}(\mathbf{x}) := \begin{cases} \{\mathbf{u} \ : \ \mathbf{u}^T(\mathbf{x}-\mathbf{y}) \geq 0, \ \forall \mathbf{y} \in \mathcal{X}\} & \text{if } \mathbf{x} \in \mathcal{X}, \\ \emptyset & \text{otherwise.} \end{cases}$$

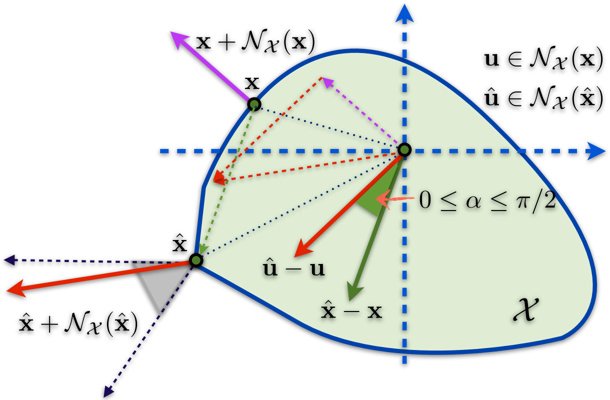
Then $\mathcal{N}_{\mathcal{X}}(\cdot)$ is a set-valued mapping and is monotone.

Monotonicity of the normal cone

Given a nonempty, closed and convex set $\mathcal{X}$. The **normal cone** of $\mathcal{X}$ at $\mathbf{x}$ is defined as

$$\mathcal{N}_{\mathcal{X}}(\mathbf{x}) := \begin{cases} \{\mathbf{u} \;:\; \mathbf{u}^T(\mathbf{x}-\mathbf{y}) \geq 0,\ \forall \mathbf{y} \in \mathcal{X}\} & \text{if } \mathbf{x} \in \mathcal{X}, \\ \emptyset & \text{otherwise.} \end{cases}$$

Then $\mathcal{N}_{\mathcal{X}}(\cdot)$ is a set-valued mapping and is monotone.



Resolvent and relation to prox-operator

Resolvent of a maximal monotone mapping

Given a maximal monotone mapping $\mathcal{T}$.

- The **resolvent** $\mathcal{J}_{\mathcal{T}}$ of $\mathcal{T}$ at $\mathbf{w}$ is defined as a solution of the *inclusion* w.r.t. $\mathbf{z}$:

$$\mathbf{w} \in \mathbf{z} + \mathcal{T}(\mathbf{z}).$$

- Conventionally, we can write $\boxed{\mathcal{J}_{\mathcal{T}}(\mathbf{w}) := (\mathbb{I} + \mathcal{T})^{-1}(\mathbf{w})}$.

Resolvent and relation to prox-operator

Resolvent of a maximal monotone mapping

Given a maximal monotone mapping $\mathcal{T}$.

- The **resolvent** $\mathcal{J}_{\mathcal{T}}$ of $\mathcal{T}$ at $\mathbf{w}$ is defined as a solution of the *inclusion* w.r.t. $\mathbf{z}$:

$$\mathbf{w} \in \mathbf{z} + \mathcal{T}(\mathbf{z}).$$

- Conventionally, we can write $\boxed{\mathcal{J}_{\mathcal{T}}(\mathbf{w}) := (\mathbb{I} + \mathcal{T})^{-1}(\mathbf{w})}$.

Lemma (Well-definedness [12])

If $\mathcal{T}$ is maximal monotone then $\mathcal{J}_{\mathcal{T}}(\mathbf{w})$ is well-defined and single-valued.

Remark: If $\mathcal{T}$ is not maximal monotone, then $\mathcal{J}_{\mathcal{T}}(\mathbf{w})$ may not be well-defined.

Resolvent and relation to prox-operator

Resolvent of a maximal monotone mapping

Given a maximal monotone mapping $\mathcal{T}$.

- The **resolvent** $\mathcal{J}_{\mathcal{T}}$ of $\mathcal{T}$ at $\mathbf{w}$ is defined as a solution of the *inclusion* w.r.t. $\mathbf{z}$:

$$\mathbf{w} \in \mathbf{z} + \mathcal{T}(\mathbf{z}).$$

- Conventionally, we can write $\boxed{\mathcal{J}_{\mathcal{T}}(\mathbf{w}) := (\mathbb{I} + \mathcal{T})^{-1}(\mathbf{w})}$.

Lemma (Well-definedness [12])

If $\mathcal{T}$ is maximal monotone then $\mathcal{J}_{\mathcal{T}}(\mathbf{w})$ is well-defined and single-valued.

Remark: If $\mathcal{T}$ is not maximal monotone, then $\mathcal{J}_{\mathcal{T}}(\mathbf{w})$ may not be well-defined.

Relation to prox operator

Let $\mathcal{T} := \partial f$ the subdifferential of a proper, closed and convex function $f \in \mathcal{F}(\mathbb{R}^p)$. Then $\mathcal{T}$ is maximal monotone and

$$\boxed{\mathcal{J}_{\partial f}(\cdot) \equiv \mathrm{prox}_f(\cdot).}$$

Example: Resolvent of the normal cone

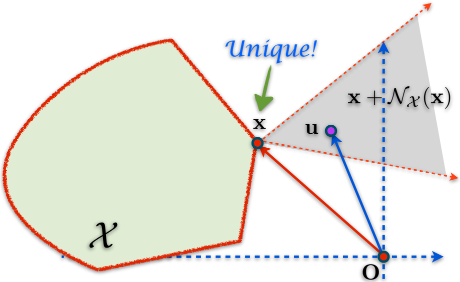
Consider the inclusion $\mathbf{u} \in \mathbf{x} + \mathcal{N}_{\mathcal{X}}(\mathbf{x})$. We can write $\mathbf{u} - \mathbf{x} \in \mathcal{N}_{\mathcal{X}}(\mathbf{x})$. By definition of $\mathcal{N}_{\mathcal{X}}(\mathbf{x})$ we have $(\mathbf{u}-\mathbf{x})^T\mathbf{x} \geq (\mathbf{u}-\mathbf{x})^T\mathbf{y}$ for all $\mathbf{y} \in \mathcal{X}$. Hence, we can write

$$(\mathbf{x}-\mathbf{u})^T(\mathbf{y}-\mathbf{u}) \geq \|\mathbf{x}-\mathbf{u}\|_2^2, \quad \forall \mathbf{y} \in \mathcal{X}.$$

This inequality shows that $\mathbf{x}$ is the solution of

$$\mathbf{x} = \arg\min_{\mathbf{y}\in\mathcal{X}} \|\mathbf{y}-\mathbf{u}\|_2^2$$

which is indeed the projection of $\mathbf{u}$ onto $\mathcal{X}$, i.e. $\mathcal{S}_{\mathcal{N}_{\mathcal{X}}(\cdot)}(\mathbf{u}) = \mathbf{x} = \pi_{\mathcal{X}}(\mathbf{u})$.



From equations to inclusions

Before presenting methods for solving (2), we review a notion in convex analysis called inclusion. Let us motivate this concept by starting from a **system of equations**.

For a single-valued mapping $J : \mathbb{R}^p \to \mathbb{R}^p$, we consider the **system of equations**:

$$\mathbf{x} = J(\mathbf{x}). \tag{8}$$

From equations to inclusions

Before presenting methods for solving (2), we review a notion in convex analysis called inclusion. Let us motivate this concept by starting from a **system of equations**.

For a single-valued mapping $J : \mathbb{R}^p \to \mathbb{R}^p$, we consider the **system of equations**:

$$\mathbf{x} = J(\mathbf{x}). \tag{8}$$

Fixed point iteration scheme

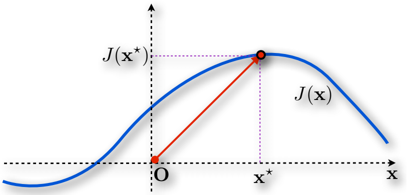
If the mapping J is contractive, i.e., $\exists \kappa \in [0, 1)$ such that

$$\|J(\mathbf{x}) - J(\hat{\mathbf{x}})\| \leq \kappa \|\mathbf{x} - \hat{\mathbf{x}}\|, \quad \forall \mathbf{x}, \hat{\mathbf{x}},$$

then, by the Banach contraction mapping principle, the sequence $\{\mathbf{x}^k\}$ generated by

$$\mathbf{x}^{k+1} := J(\mathbf{x}^k), \quad k \geq 0,$$

starting from $\mathbf{x}^0$ converges to the unique fixed point $\mathbf{x}^\star$ of J, i.e., $\mathbf{x}^\star = J(\mathbf{x}^\star)$.



From equations ...

Now, given a single-valued mapping $T : \mathbb{R}^p \to \mathbb{R}^p$, let us consider a **general system of equations**:

$$T(\mathbf{x}) = 0, \tag{9}$$

A simple way to transform (9) into (8), i.e., $\mathbf{x} = J(\mathbf{x})$, is:

$$\mathbf{x} = \mathbf{x} - \gamma T(\mathbf{x}) := J_T^{\gamma}(\mathbf{x}), \quad \gamma \neq 0.$$

Then, we can generate a fixed-point scheme for solving (9) as

$$\mathbf{x}^{k+1} := J_T^{\gamma}(\mathbf{x}^k). \tag{10}$$

If J_T^{γ} is contractive for given $\gamma \neq 0$, then $\{\mathbf{x}^k\}$ generated by (10) converges to a solution $\mathbf{x}^\star$ of (8).

From equations ...

Now, given a single-valued mapping $T : \mathbb{R}^p \to \mathbb{R}^p$, let us consider a **general system of equations**:

$$T(\mathbf{x}) = 0, \tag{9}$$

A simple way to transform (9) into (8), i.e., $\mathbf{x} = J(\mathbf{x})$, is:

$$\mathbf{x} = \mathbf{x} - \gamma T(\mathbf{x}) := J_T^{\gamma}(\mathbf{x}), \quad \gamma \neq 0.$$

Then, we can generate a fixed-point scheme for solving (9) as

$$\mathbf{x}^{k+1} := J_T^{\gamma}(\mathbf{x}^k). \tag{10}$$

If J_T^{γ} is contractive for given $\gamma \neq 0$, then $\{\mathbf{x}^k\}$ generated by (10) converges to a solution $\mathbf{x}^\star$ of (8).

Example (Gradient method)

Let us consider the **optimality condition** of the unconstrained smooth convex problem: $f^\star = \min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$ as

$$\nabla f(\mathbf{x}) = 0.$$

Since ∇f is single-valued, using the same trick as in the previous slide, we can write the fixed-point iterative scheme as

$$\mathbf{x}^{k+1} := \mathbf{x}^k - \gamma_k \nabla f(\mathbf{x}^k) = J_T^{\gamma_k}(\mathbf{x}).$$

If f is μ-strongly convex with $\mu > 0$ and ∇f is L_f-Lipschitz continuous, then J_T^{γ} is contractive for any $\gamma \in (0, 2\mu/L^2)$.

Proof of contractivity of J_T^γ

Proof.

1. If f is smooth and strongly convex with the strong convexity parameter μ. Then
$$(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}))^T(\mathbf{x} - \hat{\mathbf{x}}) \geq \mu\|\mathbf{x} - \hat{\mathbf{x}}\|^2, \ \forall \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p.$$
2. Since $J_T^\gamma(\mathbf{x}) := \mathbf{x} - \gamma\nabla f(\mathbf{x})$, for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$, we have
$$\begin{aligned}
\|J_T^\gamma(\mathbf{x}) - J_T^\gamma(\hat{\mathbf{x}})\|^2 &= \|\mathbf{x} - \hat{\mathbf{x}} - \gamma(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}))\|^2 \\
&= \|\mathbf{x} - \hat{\mathbf{x}}\|^2 - 2\gamma(\mathbf{x} - \hat{\mathbf{x}})^T(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}})) \\
&+ \gamma^2\|\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}})\|^2 \\
&\leq \|\mathbf{x} - \hat{\mathbf{x}}\|^2 - 2\gamma \underbrace{\mu\|\mathbf{x} - \hat{\mathbf{x}}\|^2}_{\text{strong convexity}} + \gamma^2 \underbrace{L^2\|\mathbf{x} - \hat{\mathbf{x}}\|^2}_{\text{Lipschitz gradient}} \\
&\leq (1 - 2\gamma\mu + \gamma^2 L^2)\|\mathbf{x} - \hat{\mathbf{x}}\|^2.
\end{aligned}$$
3. Hence, J_T^γ is contractive if $1 - 2\gamma\mu + \gamma^2 L^2 \in [0, 1)$, which implies $\gamma \in (0, 2\mu/L^2)$.
4. We note that $1 - 2\gamma\mu + \gamma^2 L^2$ is minimized if $\gamma_\star = \mu/L^2$ and hence $1 - 2\gamma\mu + \gamma^2 L^2 = 1 - \mu^2/L^2$.

□

... To inclusions

In the example presented previously, if we no longer assume f to be smooth, then the **optimality condition** turns out to be

$$0 \in \partial f(\mathbf{x}).$$

Since ∂f is a set-valued mapping, this condition is called an **inclusion**.

We can generalize this inclusion to any set-valued mapping $\mathcal{T}$ from $\mathbb{R}^p$ to $2^{\mathbb{R}^p}$ as

$$0 \in \mathcal{T}(\mathbf{x}). \tag{11}$$

In general, solving the inclusion (11) is much more difficulty than solving the equation system $T(\mathbf{x}) = 0$. Methods for solving (11) on the one hand can inherit from methods of solving equations, but on the other hand, require new mathematical tools.

... To inclusions

In the example presented previously, if we no longer assume f to be smooth, then the **optimality condition** turns out to be

$$0 \in \partial f(\mathbf{x}).$$

Since ∂f is a set-valued mapping, this condition is called an **inclusion**.

We can generalize this inclusion to any set-valued mapping $\mathcal{T}$ from $\mathbb{R}^p$ to $2^{\mathbb{R}^p}$ as

$$0 \in \mathcal{T}(\mathbf{x}). \tag{11}$$

In general, solving the inclusion (11) is much more difficulty than solving the equation system $T(\mathbf{x}) = 0$. Methods for solving (11) on the one hand can inherit from methods of solving equations, but on the other hand, require new mathematical tools.

Additional mathematical tools

Since we are working with the possibly nonsmooth and constrained convex problem (2), where $\mathcal{X}$ is not specified, its optimality condition will be reformulated as an inclusion. We will use additional mathematical tools from variational analysis such as:

- Monotone inclusions
- Monotone mixed variational inequalities
- Gap functions

Mixed variational inequality (MVI) formulation

Primal-dual mapping

We introduce a new primal-dual variable $\mathbf{z} := (\mathbf{x}^T, \lambda^T)^T \in \mathbb{R}^{p+n}$ and two mappings:

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix} \text{ and } \mathcal{T}(\mathbf{z}) := \left\{ \begin{pmatrix} \xi \\ 0^n \end{pmatrix} \in \mathbb{R}^{p+n} \;:\; \xi \in \partial f(\mathbf{x}) \right\}. \quad (12)$$

- Then $M : \mathbb{R}^{p+n} \to \mathbb{R}^{p+n}$ is a single-valued mapping (linear mapping).
- If f is not differentiable, then $\mathcal{T} : \mathbb{R}^{p+n} \rightrightarrows 2^{\mathbb{R}^{p+n}}$ is a set-valued mapping.

Mixed variational inequality (MVI) formulation

Primal-dual mapping

We introduce a new primal-dual variable $\mathbf{z} := (\mathbf{x}^T, \lambda^T)^T \in \mathbb{R}^{p+n}$ and two mappings:

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix} \text{ and } \mathcal{T}(\mathbf{z}) := \left\{ \begin{pmatrix} \xi \\ 0^n \end{pmatrix} \in \mathbb{R}^{p+n} \ : \ \xi \in \partial f(\mathbf{x}) \right\}. \tag{12}$$

- Then $M : \mathbb{R}^{p+n} \to \mathbb{R}^{p+n}$ is a single-valued mapping (linear mapping).
- If f is not differentiable, then $\mathcal{T} : \mathbb{R}^{p+n} \rightrightarrows 2^{\mathbb{R}^{p+n}}$ is a set-valued mapping.

Inclusion and MVI formulation

- The **optimality condition** (5) can be written as an **inclusion**:

$$\boxed{0 \in \mathcal{P}(\mathbf{z}) := M(\mathbf{z}) + \mathcal{T}(\mathbf{z}).}$$

- (5) can also be expressed as a **mixed variational inequality** (MVI):

$$\boxed{f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathbb{R}^{p+n}.} \tag{13}$$

Optimality condition as a monotone inclusion/VI problem

Lemma (Monotonicity of primal-dual mapping)

The mapping M and $\mathcal{T}$ defined in (12)*:*

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix} \quad \textit{and} \quad \mathcal{T}(\mathbf{z}) := \left\{ \begin{pmatrix} \xi \\ 0^n \end{pmatrix} \in \mathbb{R}^{p+n} \ : \ \xi \in \partial f(\mathbf{x}) \right\}.$$

are **maximal monotone**. *Consequently, $\mathcal{P} := M + \mathcal{T}$ is also* ***maximal monotone***.

To show the monotonicity of M, we can write $M(\mathbf{z})$ as

$$M(\mathbf{z}) := \mathbf{H}\mathbf{z} + \mathbf{h} \equiv \begin{bmatrix} 0 & \mathbf{A}^T \\ -\mathbf{A} & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}$$

It is clear that

$$\mathbf{z}^T\mathbf{H}\mathbf{z} = \mathbf{x}^T(\mathbf{A}^T\lambda) - \lambda^T\mathbf{A}\mathbf{x} = 0,$$

which shows that M is monotone.

Optimality condition as a monotone inclusion/VI problem

Lemma (Monotonicity of primal-dual mapping)

The mapping M and $\mathcal{T}$ defined in (12)*:*

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix} \quad \textit{and} \quad \mathcal{T}(\mathbf{z}) := \left\{ \begin{pmatrix} \xi \\ 0^n \end{pmatrix} \in \mathbb{R}^{p+n} \ : \ \xi \in \partial f(\mathbf{x}) \right\}.$$

are **maximal monotone***. Consequently, $\mathcal{P} := M + \mathcal{T}$ is also* ***maximal monotone****.*

To show the monotonicity of M, we can write $M(\mathbf{z})$ as

$$M(\mathbf{z}) := \mathbf{H}\mathbf{z} + \mathbf{h} \equiv \begin{bmatrix} 0 & \mathbf{A}^T \\ -\mathbf{A} & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}$$

It is clear that

$$\mathbf{z}^T\mathbf{H}\mathbf{z} = \mathbf{x}^T(\mathbf{A}^T\lambda) - \lambda^T\mathbf{A}\mathbf{x} = 0,$$

which shows that M is monotone.

Monotone inclusion and monotone variational inequality

- The **inclusion** $0 \in \mathcal{P}(\mathbf{z}^\star)$ is called a monotone inclusion if $\mathcal{P}$ is maximal monotone.
- The **mixed variational inequality**

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathbb{R}^{p+n}$$

  is called monotone if f is proper, closed and convex and M is maximal monotone.

Gap function for the MVI problem

Gap function

Let us consider a monotone MVIP problem of the form:

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathbb{R}^{p+n}$$

The **gap function** associated with this problem is defined as follows

$$\boxed{G(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathbb{R}^{p+n}} \Big\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \Big\}.} \tag{14}$$

Gap function for the MVI problem

Gap function

Let us consider a monotone MVIP problem of the form:

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathbb{R}^{p+n}$$

The **gap function** associated with this problem is defined as follows

$$\boxed{G(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathbb{R}^{p+n}} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\}.} \tag{14}$$

Properties

- Computing G and its **gradient** require to solve the convex problem in (14).
- G is nonnegative, i.e.: $G(\mathbf{z}) \geq 0$ for all $\mathbf{z} \in \mathbb{R}^{p+n}$.
  - Indeed, we have $G(\mathbf{z}) = \max_{\hat{\mathbf{z}} \in \mathbb{R}^{p+n}} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\} \geq f(\mathbf{x}) - f(\mathbf{x}) + M(\mathbf{z})^T(\mathbf{z} - \mathbf{z}) = 0,$ with $\hat{\mathbf{z}} = \mathbf{z}$.
- $\boxed{G(\mathbf{z}^\star) = 0}$ iff $\mathbf{z}^\star$ is a KKT point of (5).

  Exercise: By writing the optimality condition of the maximization problem (14) and rearranging it, we obtain exactly the KKT condition (5).

Example: Gap function

Consider the following constrained convex problem:

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{R}^2} \quad & \{f(\mathbf{x}) := |x_1| + x_2^2\}, \\ \text{s.t.} \quad & x_1 + x_2 = 1.\end{aligned}$$

We have $\mathbf{z} = (x_1, x_2, \lambda)^T \in \mathbb{R}^3$ and $M(\mathbf{z}) := (\lambda, \lambda, 1 - x_1 - x_2)^T$.

The **gap function** associated with the optimality condition of this problem becomes:

$$\begin{aligned}G(\mathbf{z}) &:= \max_{(y_1,y_2,\eta)^T\in\mathbb{R}^3} \Big\{|x_1| + x_2^2 - |y_1| - y_2^2 + \eta(x_1 + x_2 - 1) - \lambda(y_1 + y_2 - 1)\Big\} \\ &= \max_{(y_1,y_2)^T\in\mathbb{R}^2} \Big\{ - |y_1| - y_2^2 - \lambda(y_1 + y_2 - 1)\Big\} + \max_{\eta\in\mathbb{R}} \Big\{|x_1| + x_2^2 + \eta(x_1 + x_2 - 1)\Big\} \\ &= \begin{cases} |x_1| + x_2^2 - d(\lambda) & \text{if } x_1 + x_2 - 1, \\ +\infty & \text{otherwise,}\end{cases}\end{aligned}$$

where $d(\lambda) := \min_{(y_1,y_2)^T\in\mathbb{R}^2} \Big\{|y_1| + y_2^2 + \lambda(y_1 + y_2 - 1)\Big\}$ is the dual function.

Outline

- Today
  1. Convex constrained optimization and motivating examples
  2. Optimality condition
  3. Conjugate functions
  4. Monotone inclusion and monotone mixed variational inequality formulations
  5. Chambolle-Pock's primal-dual method
  6. Primal-dual hybrid gradient method
  7. Splitting methods
  8. Model-based excessive gap primal-dual method
- Next week
  1. Disciplined convex programming

A special class of constrained convex problems (2)

We first consider a constrained reformulation of composite convex minimization problems considered in Lecture 5.

Constrained convex reformulation

We consider the following special case of (2):

$$\begin{cases} \min\limits_{\mathbf{x}:=(\mathbf{u},\mathbf{v})\in\mathbb{R}^{p_1+p_2}} & \left\{F(\mathbf{x}) := f(\mathbf{u}) + g(\mathbf{v})\right\} \\ \text{s.t.} & \mathbf{K}\mathbf{u} - \mathbf{v} = 0. \end{cases} \tag{15}$$

where $\mathbf{K}$ is a **linear operator**, $f \in \mathcal{F}(\mathbb{R}^{p_1})$ and $g \in \mathcal{F}(\mathbb{R}^{p_2})$ are two convex functions.

A special class of constrained convex problems (2)

We first consider a constrained reformulation of composite convex minimization problems considered in Lecture 5.

Constrained convex reformulation

We consider the following special case of (2):

$$\begin{cases} \min_{\mathbf{x}:=(\mathbf{u},\mathbf{v})\in\mathbb{R}^{p_1+p_2}} & \left\{F(\mathbf{x}) := f(\mathbf{u}) + g(\mathbf{v})\right\} \\ \text{s.t.} & \mathbf{K}\mathbf{u} - \mathbf{v} = 0. \end{cases} \tag{15}$$

where $\mathbf{K}$ is a **linear operator**, $f \in \mathcal{F}(\mathbb{R}^{p_1})$ and $g \in \mathcal{F}(\mathbb{R}^{p_2})$ are two convex functions.

- By setting $\mathbf{A} := [\mathbf{K}, -\mathbb{I}]$ and $\mathbf{b} := 0^n$, we can formulate (15) into (2):

$$F^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & F(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases}$$

- Problem (15) can be written as a composite convex minimization:

$$F^\star := \min_{\mathbf{u}\in\mathbb{R}^{p_1}} \left\{F(\mathbf{u}) := f(\mathbf{u}) + g(\mathbf{K}\mathbf{u})\right\}. \tag{16}$$

Min-max formulation and dual problem

The min-max (saddle point) problem

By using the Fenchel conjugate g^* of g, we can write

$$g(\mathbf{K}\mathbf{u}) = \max_{\mathbf{v}\in\mathbb{R}^n}\{\langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle - g^*(\mathbf{v})\}.$$

Substituting this function into (16), we obtain:

$$\boxed{\min_{\mathbf{u}\in\mathbb{R}^{p}1}\ \max_{\mathbf{v}\in\mathbb{R}^n}\Big\{\ \langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v})\Big\}} \tag{17}$$

where g^* is the conjugate of g.

Min-max formulation and dual problem

The min-max (saddle point) problem

By using the Fenchel conjugate g^* of g, we can write

$$g(\mathbf{K}\mathbf{u}) = \max_{\mathbf{v}\in\mathbb{R}^n}\{\langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle - g^*(\mathbf{v})\}.$$

Substituting this function into (16), we obtain:

$$\boxed{\min_{\mathbf{u}\in\mathbb{R}^{p1}} \max_{\mathbf{v}\in\mathbb{R}^n} \Big\{ \langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v}) \Big\}} \tag{17}$$

where g^* is the conjugate of g.

Dual problem

By exchanging the min-max in (17) and note that

$$\min_{\mathbf{u}\in\mathbb{R}^{p1}} \{\langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle + f(\mathbf{u})\} = -\max_{\mathbf{u}\in\mathbb{R}^{p1}} \{\Big\langle -\mathbf{K}^T\mathbf{v}, \mathbf{u}\Big\rangle - f(\mathbf{u})\} = -f^*(-\mathbf{K}^T\mathbf{v})$$

we have

$$\boxed{\max_{\mathbf{v}\in\mathbb{R}^q} \Big\{ -f^*(-\mathbf{K}^T\mathbf{v}) - g^*(\mathbf{v}) \Big\}.} \tag{18}$$

Chambolle-Pock's algorithm: the main idea

Optimality condition

First, we write the optimality condition of (17) as follows:

$$\begin{cases}\mathbf{K}\mathbf{u}^\star & \in \partial g^*(\mathbf{v}^\star)\\ -\mathbf{K}^T\mathbf{v}^\star & \in \partial f(\mathbf{u}^\star).\end{cases} \tag{19}$$

Since problem is convex, condition (19) is necessary and sufficient for $(\mathbf{u}^\star, \mathbf{v}^\star)$ to be primal and dual optimal to (17):

$$\min_{\mathbf{u}\in\mathbb{R}^p}\max_{\mathbf{v}\in\mathbb{R}^q}\Big\{\langle\mathbf{K}\mathbf{u},\mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v})\Big\}.$$

Chambolle-Pock's algorithm: the main idea

Optimality condition

First, we write the optimality condition of (17) as follows:

$$\begin{cases} \mathbf{K}\mathbf{u}^\star & \in \partial g^*(\mathbf{v}^\star) \\ -\mathbf{K}^T\mathbf{v}^\star & \in \partial f(\mathbf{u}^\star). \end{cases} \tag{19}$$

Since problem is convex, condition (19) is necessary and sufficient for $(\mathbf{u}^\star, \mathbf{v}^\star)$ to be primal and dual optimal to (17):

Fixed-point expression

Second, from (19), for any $\sigma > 0$ and $\tau > 0$, we can write

$$\mathbf{v}^\star + \sigma\mathbf{K}\mathbf{u}^\star \in (\mathbb{I} + \sigma\partial g^*)(\mathbf{v}^\star) \text{ and } \mathbf{u}^\star - \tau\mathbf{K}^T\mathbf{v}^\star \in (\mathbb{I} + \tau\partial f)(\mathbf{u}^\star).$$

Using the proximal operator of τf and σg^*, we can write the last expression as

$$\begin{cases} \mathbf{v}^\star & = \mathrm{prox}_{\sigma g^*}\left(\mathbf{v}^\star + \sigma\mathbf{K}\mathbf{u}^\star\right) \\ \mathbf{u}^\star & = \mathrm{prox}_{\tau f}\left(\mathbf{u}^\star - \tau\mathbf{K}^T\mathbf{v}^\star\right). \end{cases} \tag{20}$$

This relation shows that $\mathbf{x}^\star := (\mathbf{u}^\star, \mathbf{v}^\star)$ is a fixed point of the **mapping** $\mathcal{T}$ with:

$$\mathcal{T}(\mathbf{x}) := (\mathrm{prox}_{\sigma g^*}(\mathbf{v} + \sigma\mathbf{K}\mathbf{u}), \mathrm{prox}_{\tau f}(\mathbf{u} - \tau\mathbf{K}^T\mathbf{v})).$$

The Chambolle-Pock algorithm

The **Chambolle-Pock algorithm** is rooted from the classical Arrow-Hurwicz method, which is based on the fixed-point expression (20).

Chambolle-Pock's algorithm (CPA) [2]

1. Choose $\tau > 0$, $\sigma > 0$, $\theta \in [0, 1]$, $\mathbf{u}^0 \in \mathbb{R}^p$ and $\mathbf{v}^0 \in \mathbb{R}^q$. Set $\hat{\mathbf{u}}^0 := \mathbf{u}^0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma g^*}\left(\mathbf{v}^k + \sigma \mathbf{K}\hat{\mathbf{u}}^k\right) \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau f}\left(\mathbf{u}^k - \tau \mathbf{K}^T \mathbf{v}^k\right) \\ \hat{\mathbf{u}}^{k+1} & := \mathbf{u}^{k+1} + \theta(\mathbf{u}^{k+1} - \mathbf{u}^k). \end{cases} \tag{21}$$

The Chambolle-Pock algorithm

The **Chambolle-Pock algorithm** is rooted from the classical Arrow-Hurwicz method, which is based on the fixed-point expression (20).

Chambolle-Pock's algorithm (CPA) [2]

1. Choose $\tau > 0$, $\sigma > 0$, $\theta \in [0,1]$, $\mathbf{u}^0 \in \mathbb{R}^p$ and $\mathbf{v}^0 \in \mathbb{R}^q$. Set $\hat{\mathbf{u}}^0 := \mathbf{u}^0$.
2. For $k = 0, 1, \cdots,$ perform:

$$\begin{cases} \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma g^*}\left(\mathbf{v}^k + \sigma \mathbf{K}\hat{\mathbf{u}}^k\right) \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau f}\left(\mathbf{u}^k - \tau \mathbf{K}^T\mathbf{v}^k\right) \\ \hat{\mathbf{u}}^{k+1} & := \mathbf{u}^{k+1} + \theta(\mathbf{u}^{k+1} - \mathbf{u}^k). \end{cases} \tag{21}$$

Remarks

- If $\theta = 0$, then $\hat{\mathbf{u}}^k = \mathbf{u}^k$ and (21) collapses to the Arrow-Hurwicz method:

$$\begin{cases} \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma g^*}\left(\mathbf{v}^k + \sigma \mathbf{K}\mathbf{u}^k\right) \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau f}\left(\mathbf{u}^k - \tau \mathbf{K}^T\mathbf{v}^k\right). \end{cases}$$

- The step sizes σ and τ and the parameter θ can keep constantly or adaptively updates.
- When $\mathbf{K} = \mathbb{I}$, the **Chambolle-Pock algorithm** is equivalent to **ADMM**.

Restricted gap function for optimality certification

Let us define $\mathbf{x} := (\mathbf{u}, \mathbf{v}) \equiv (\mathbf{u}^T, \mathbf{v}^T)^T$, $\phi(\mathbf{x}) := f(\mathbf{u}) + g^\star(\mathbf{v})$ and

$$M(\mathbf{x}) := \begin{bmatrix} -\mathbf{K}^T\mathbf{v} \\ \mathbf{K}\mathbf{u} \end{bmatrix}.$$

Then (19): $\begin{cases} \mathbf{K}\mathbf{u}^\star \in \partial g^*(\mathbf{v}^\star) \\ -\mathbf{K}^T\mathbf{v}^\star \in \partial f(\mathbf{u}^\star) \end{cases}$ can be written as a **monotone MVI problem**:

$$\boxed{\phi(\mathbf{x}) - \phi(\mathbf{x}^\star) + M(\mathbf{x}^\star)^T(\mathbf{x} - \mathbf{x}^\star) \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^{p_1+p_2}} \qquad (22)$$

where $\mathbf{x}^\star := (\mathbf{u}^\star, \mathbf{v}^\star)$.

Restricted gap function for optimality certification

Let us define $\mathbf{x} := (\mathbf{u}, \mathbf{v}) \equiv (\mathbf{u}^T, \mathbf{v}^T)^T$, $\phi(\mathbf{x}) := f(\mathbf{u}) + g^\star(\mathbf{v})$ and

$$M(\mathbf{x}) := \begin{bmatrix} -\mathbf{K}^T\mathbf{v} \\ \mathbf{K}\mathbf{u} \end{bmatrix}.$$

Then (19): $\begin{cases} \mathbf{K}\mathbf{u}^\star \in \partial g^*(\mathbf{v}^\star) \\ -\mathbf{K}^T\mathbf{v}^\star \in \partial f(\mathbf{u}^\star) \end{cases}$ can be written as a **monotone MVI problem**:

$$\boxed{\phi(\mathbf{x}) - \phi(\mathbf{x}^\star) + M(\mathbf{x}^\star)^T(\mathbf{x} - \mathbf{x}^\star) \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^{p_1+p_2}} \tag{22}$$

where $\mathbf{x}^\star := (\mathbf{u}^\star, \mathbf{v}^\star)$.

Definition (Restricted gap function)

Let $\mathcal{X} \subseteq \mathbb{R}^{p_1+p_2}$ be a nonempty, closed, convex and bounded set. We define a **restricted gap function** of (22) restricted on $\mathcal{X}$ as

$$\boxed{G_{\mathcal{X}}(\mathbf{x}) := \max_{\tilde{\mathbf{x}} \in \mathcal{X} \subseteq \mathbb{R}^{p_1+p_2}} \left\{ \phi(\mathbf{x}) - \phi(\tilde{\mathbf{x}}) + M(\mathbf{x})^T(\mathbf{x} - \tilde{\mathbf{x}}) \right\}} \tag{23}$$

Convergence theorem

Theorem (Convergence)

Assumptions:

- (17): $\min_{\mathbf{u}\in\mathbb{R}^{p1}} \max_{\mathbf{v}\in\mathbb{R}^{p2}} \Big\{ \langle \mathbf{K}\mathbf{u}, \mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v}) \Big\}$ *has a saddle point* $\mathbf{z}^\star := (\mathbf{u}^\star, \mathbf{v}^\star)$.
- $\{(\mathbf{u}^k, \mathbf{v}^k)\}$ *be the sequence generated by the* ***Chambolle-Pock algorithm****.*
- *If we choose* $\theta := 1$, $\sigma > 0$ *and* $\tau > 0$ *such that* $\tau\sigma\|\mathbf{K}\|^2 < 1$.

Conclusion:

- *The sequence* $\{\bar{\mathbf{x}}^k\}_{k\geq 0}$ *defined by*

$$\bar{\mathbf{x}}^k = (\bar{\mathbf{u}}^k, \bar{\mathbf{v}}^k) := \frac{1}{(k+1)} \sum_{j=0}^{k} (\mathbf{u}^j, \mathbf{v}^j)$$

satisfies

$$G_{\mathcal{X}}(\bar{\mathbf{x}}^k) \leq \frac{1}{k+1}\Big[\max_{\mathbf{x}:=(\mathbf{u},\mathbf{v})\in\mathcal{X}} \Big\{ (1/(2\tau))\|\mathbf{u}-\mathbf{u}^0\|^2 + (1/(2\sigma))\|\mathbf{v}-\mathbf{v}^0\|^2 \Big\} \Big]. \quad (24)$$

- $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ *converges to a saddle point* $\mathbf{z}^\star$ *of* (17) *at the* $\mathcal{O}(1/k)$ *rate w.r.t. the restricted gap function* $G_{\mathcal{X}}$ *(in the ergodic sense).*

Convergence theorem: Remarks

$$G_{\mathcal{X}}(\bar{\mathbf{x}}^k) \leq \frac{1}{k+1}\Big[\max_{\mathbf{x}:=(\mathbf{u},\mathbf{v})\in\mathcal{X}}\Big\{(1/(2\tau))\|\mathbf{u}-\mathbf{u}^0\|^2 + (1/(2\sigma))\|\mathbf{v}-\mathbf{v}^0\|^2\Big\}\Big]. \quad (24)$$

- The right-hand side of the estimate (24) depends on the choice of $\mathcal{X}$. Theoretically, we need to choose $\mathcal{X}$ such that $\mathcal{X}^\star \subseteq \mathcal{X}$, where $\mathcal{X}^\star$ is the solution set of (17), which is unknown.
- The estimate (24) does not imply the convergence rate of $\{\bar{\mathbf{u}}^k\}$ and $\{\bar{\mathbf{v}}^k\}$ separately.

Acceleration - Case 1: f or $g^\star$ is strongly convex

If either f or $g^\star$ is strongly convex, the **Chambolle-Pock algorithm** can be accelerated to get better convergence rate.

Assumption A.1.

The function f is strongly convex with a strong convexity parameter $\sigma_f > 0$.

- Under Assumption A.1., the conjugate f^* is smooth and has the Lipschitz gradient.
- The **Chambolle-Pock algorithm** can be accelerated to get $\mathcal{O}(1/k^2)$ convergence rate.

Acceleration - Case 1: f or $g^\star$ is strongly convex

If either f or $g^\star$ is strongly convex, the **Chambolle-Pock algorithm** can be accelerated to get better convergence rate.

Assumption A.1.

The function f is strongly convex with a strong convexity parameter $\sigma_f > 0$.

- Under Assumption A.1., the conjugate f^* is smooth and has the Lipschitz gradient.
- The **Chambolle-Pock algorithm** can be accelerated to get $\mathcal{O}(1/k^2)$ convergence rate.

Chambolle-Pock's algorithm for strongly convex (CPA$_1$)

1. Choose $\tau > 0$, $\sigma > 0$ such that $\sigma_0\tau_0 L^2 \leq 1$, $\mathbf{u}^0 \in \mathbb{R}^p$ and $\mathbf{v}^0 \in \mathbb{R}^q$. Set $\hat{\mathbf{u}}^0 := \mathbf{u}^0$.

2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma_k g^*}\left(\mathbf{v}^k + \sigma_k \mathbf{K}\hat{\mathbf{u}}^k\right) \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau_k f}\left(\mathbf{u}^k - \tau_k \mathbf{K}^T\mathbf{v}^k\right) \\ \theta_k & := (1 + 2\sigma_f\tau_k)^{-1/2}, \\ \tau_{k+1} & := \theta_k\tau_k, \\ \sigma_{k+1} & := \theta_k^{-1}\sigma_k, \\ \hat{\mathbf{u}}^{k+1} & := \mathbf{u}^{k+1} + \theta_k(\mathbf{u}^{k+1} - \mathbf{u}^k). \end{cases} \tag{25}$$

Acceleration - Case 2: Both f and g^* are strongly convex

If both f and g^* are strongly convex, we can accelerate the **Chambolle-Pock algorithm** to obtain the linear convergence rate.

Assumption A.2.

The functions f and g^* are strongly convex with a strong convexity parameters $\sigma_f > 0$ and $\sigma_{g^*} > 0$, respectively.

Acceleration - Case 2: Both f and g^* are strongly convex

If both f and g^* are strongly convex, we can accelerate the **Chambolle-Pock algorithm** to obtain the linear convergence rate.

Assumption A.2.

The functions f and g^* are strongly convex with a strong convexity parameters $\sigma_f > 0$ and $\sigma_{g^*} > 0$, respectively.

Chambolle-Pock's algorithm for strongly convex f and g^* (CPA$_2$)

1. Choose $\omega \leq 2\sqrt{\sigma_f \sigma_{g^*}} L$.
2. Set $\tau := \omega/(2\sigma_f)$, $\sigma := \omega/(2\sigma_{g^*})$ and choose $\theta \in [(1+\omega)^{-1}, 1]$.
3. Find $\mathbf{u}^0 \in \mathbb{R}^p$ and $\mathbf{v}^0 \in \mathbb{R}^q$. Set $\hat{\mathbf{u}}^0 := \mathbf{u}^0$.
4. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma g^*}\left(\mathbf{v}^k + \sigma \mathbf{K} \hat{\mathbf{u}}^k\right) \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau f}\left(\mathbf{u}^k - \tau \mathbf{K}^T \mathbf{v}^k\right) \\ \hat{\mathbf{u}}^{k+1} & := \mathbf{u}^{k+1} + \theta(\mathbf{u}^{k+1} - \mathbf{u}^k). \end{cases} \quad (26)$$

Example: Strong convexity of f/g^*

Example (Strong convexity of f)

We consider the following image ℓ_1-TV denoising problem:

$$\min_{\mathbf{u}\in\mathbb{R}^{p_1}} (1/2)\|\mathbf{u}-\mathbf{b}\|_F^2 + \rho\|\mathbf{D}\mathbf{u}\|_1. \tag{27}$$

Here $\mathbf{b}$ is a noisy image, $\rho > 0$ is a regularization parameter, and $\mathbf{D}$ is a given matrix. We can write this problem into the following minmax form:

$$\min_{\mathbf{u}\in\mathbb{R}^{p_1}} \max_{\|\mathbf{v}\|_\infty\le 1} \left\{\rho\mathbf{v}^T\mathbf{D}\mathbf{u} + (1/2)\|\mathbf{u}-\mathbf{b}\|_F^2\right\}.$$

In this case, we have $f(\mathbf{u}) := (1/2)\|\mathbf{u}-\mathbf{b}\|_F^2$, which is strongly convex with the parameter $\mu_f = 1$, and $g^*(\mathbf{v}) = 0$.

Example (Strong convexity of both f and g^*)

If we apply Nesterov's smoothing technique to (27) with a simple prox-function $(1/2)\|\mathbf{v}\|_F^2$, we obtain the following problem:

$$\min_{\mathbf{u}\in\mathbb{R}^{p_1}} \max_{\|\mathbf{v}\|_\infty\le 1} \left\{\rho\mathbf{v}^T\mathbf{D}\mathbf{u} + (1/2)\|\mathbf{u}-\mathbf{b}\|_F^2 - (\gamma/2)\|\mathbf{v}\|_F^2\right\}.$$

where $\gamma > 0$ is a smoothness parameter. In this case, we can denote $g^*(\mathbf{v}) := (\gamma/2)\|\mathbf{v}\|_F^2$, which is strongly convex with the parameter $\gamma > 0$.

Convergence of CPA_1 and CPA_2

- **Assumptions**:
  - f is strongly convex with a strong convexity parameter $\sigma_f > 0$.
  - Let $\{(\mathbf{u}^k, \mathbf{v}^k)\}_{k\geq 0}$ be the sequence generated by CPA_1.
  - Let $\tau_0 > 0$ and $\sigma_0 := 1/(\tau_0 L^2)$.

Conclusion: Then for any $\epsilon > 0$, the exists K_0 (depending on ϵ and $\sigma_f \tau_0$) such that for any $k \geq K_0$,

$$\|\bar{\mathbf{u}}^k - \mathbf{u}^\star\|^2 \leq \frac{1+\epsilon}{(k+1)^2}\left(\frac{\|\mathbf{u}^0 - \mathbf{u}^\star\|^2}{\sigma_f^2 \tau_0^2} + \frac{L^2}{\sigma_f^2}\|\mathbf{v}^0 - \mathbf{v}^\star\|^2\right),$$

where $\bar{\mathbf{u}}^k := (k+1)^{-1}\sum_{j=0}^k \mathbf{u}^j$. The sequence $\{\bar{\mathbf{u}}^k\}_{k\geq 0}$ converges to $\mathbf{u}^\star$ at the $\mathcal{O}(1/k^2)$ rate.

Convergence of CPA_1 and CPA_2

- **Assumptions**:
  - f is strongly convex with a strong convexity parameter $\sigma_f > 0$.
  - Let $\{(\mathbf{u}^k, \mathbf{v}^k)\}_{k\geq 0}$ be the sequence generated by CPA_1.
  - Let $\tau_0 > 0$ and $\sigma_0 := 1/(\tau_0 L^2)$.

  Conclusion: Then for any $\epsilon > 0$, the exists K_0 (depending on ϵ and $\sigma_f \tau_0$) such that for any $k \geq K_0$,

$$\|\bar{\mathbf{u}}^k - \mathbf{u}^\star\|^2 \leq \frac{1+\epsilon}{(k+1)^2}\left(\frac{\|\mathbf{u}^0 - \mathbf{u}^\star\|^2}{\sigma_f^2 \tau_0^2} + \frac{L^2}{\sigma_f^2}\|\mathbf{v}^0 - \mathbf{v}^\star\|^2\right),$$

  where $\bar{\mathbf{u}}^k := (k+1)^{-1}\sum_{j=0}^k \mathbf{u}^j$. The sequence $\{\bar{\mathbf{u}}^k\}_{k\geq 0}$ converges to $\mathbf{u}^\star$ at the $\mathcal{O}(1/k^2)$ rate.

- **Assumptions:**
  - f and g^* are strongly convex with a strong convexity parameters $\sigma_f > 0$ and $\sigma_{g^*} > 0$, respectively.
  - Let $\{(\mathbf{u}^k, \mathbf{v}^k)\}_{k\geq 0}$ be the sequence generated by CPA_2.
  - Let

$$c := \frac{1+\theta}{2(1+\sqrt{\sigma_f \sigma_{g^*}}/L)} < 1.$$

  Conclusion: Then $\{(\mathbf{u}^k, \mathbf{v}^k)\}_{k\geq 0}$ converges to $(\mathbf{u}^\star, \mathbf{v}^\star)$ at linear rate $\mathcal{O}(c^k)$.

Example: Image inpainting

Mathematical formulation

Given a damaged image $\mathbf{b} \in \mathbb{R}^{m\times n}$, where the missed pixels b_{ij} are in certain region, i.e., $(i,j) \in \mathcal{M} \subset \mathcal{I} := \{1, \cdots, m\} \times \{1, \cdots, n\}$. The aim is to recover a undamaged image $\mathbf{x}$ by using the total variation operator. This problem can be formulated as:

$$\min_{\mathbf{u}\in\mathbb{R}^{m\times n}} \|\mathbf{K}\mathbf{u}\|_1 + (\rho/2) \sum_{(i,j)\in\mathcal{I}\setminus\mathcal{M}} (u_{ij} - b_{ij})^2 \tag{28}$$

where $\rho > 0$ is a regularization parameter and $\mathbf{K}$ is the total variation linear transform.

Example: Image inpainting

Mathematical formulation

Given a damaged image $\mathbf{b} \in \mathbb{R}^{m \times n}$, where the missed pixels b_{ij} are in certain region, i.e., $(i,j) \in \mathcal{M} \subset \mathcal{I} := \{1, \cdots, m\} \times \{1, \cdots, n\}$. The aim is to recover a undamaged image $\mathbf{x}$ by using the total variation operator. This problem can be formulated as:

$$\min_{\mathbf{u} \in \mathbb{R}^{m \times n}} \|\mathbf{K}\mathbf{u}\|_1 + (\rho/2) \sum_{(i,j) \in \mathcal{I} \setminus \mathcal{M}} (u_{ij} - b_{ij})^2 \tag{28}$$

where $\rho > 0$ is a regularization parameter and $\mathbf{K}$ is the total variation linear transform.

How to apply the **Chambolle-Pock algorithm**?

- This problem is of the form $F^\star := \min_{\mathbf{u} \in \mathbb{R}^{p_1}} \left\{ F(\mathbf{u}) := f(\mathbf{u}) + g(\mathbf{K}\mathbf{u}) \right\}$, where $f(\mathbf{u}) := (\rho/2) \sum_{(i,j) \in \mathcal{I} \setminus \mathcal{M}} (u_{ij} - b_{ij})^2$ and $g(\mathbf{v}) := \|\mathbf{v}\|_1$.
- Both f and g have closed form prox-operators:

$$\begin{aligned} \mathrm{prox}_{\sigma g^*}(\mathbf{v}) &= \mathbf{v}./\max(1, |v|) \\ \mathrm{prox}_{\tau f}(\mathbf{u}) &= \begin{cases} u_{ij} & \text{if} (i,j) \in \mathcal{M} \\ \frac{u_{ij} + \tau\rho b_{ij}}{1 + \tau\rho} & \text{otherwise} \end{cases} \end{aligned}$$

Example: Image inpainting - configuration

We implement Chambolle-Pock's algorithm for solving the inpainting problem (28) using the following configurations:

- **Parameter selection:**
  - $\sigma = 10$, $\tau = 0.01125$ and $\theta = 1$.
  - The initial point $\mathbf{u}^0 := \mathbf{b}$ and $\mathbf{v}^0 := 0$.
  - The tolerance $\epsilon = 10^{-5}$.
- **Stopping criterion:**
$$\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_F \leq \epsilon \|\mathbf{u}^k\|_F.$$
- **Data generating:**
  - We take a real gray image of size 255×255
  - The image is damaged by a mask of 30 lines crossing from the left to the right.
  - The regularization parameter ρ is chosen as $\rho = 1$ and $\rho = 0.75$ for two cases.

Convergence behavior: Left: $\rho = 1$ – Right: $\rho = 0.75$.

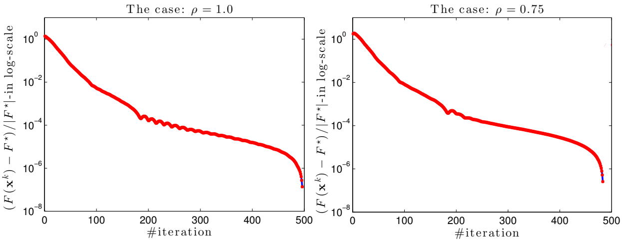


Image inpainting: outputs

Original image

Damaged image

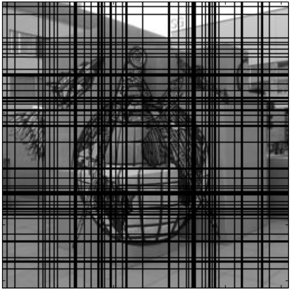
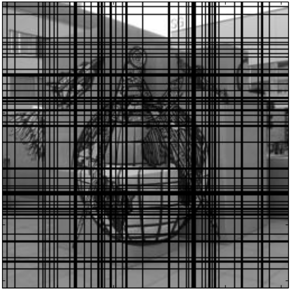


Image inpainting: outputs

Original image

Damaged image



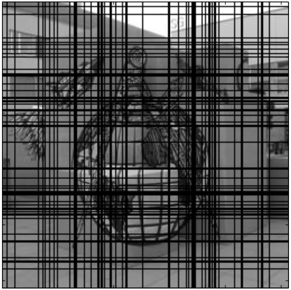
- The objective value: 9.1323×10^3
- Relative error:
  $$\|\mathbf{x}^k - \mathbf{x}^\natural\| / \|\mathbf{x}^\natural\| = 0.076398$$
  where $\mathbf{x}^\natural$ is the original image
- The number of iterations: 497
- The CPU time: 2.864s.

- The objective value: 5.9100×10^3
- Relative error:
  $$\|\mathbf{x}^k - \mathbf{x}^\natural\| / \|\mathbf{x}^\natural\| = 0.078396$$
  where $\mathbf{x}^\natural$ is the original image
- The number of iterations: 484
- The CPU time: 2.707s.

Image inpainting: outputs

Original image

Damaged image



Recovered image ($\rho = 1$)

Recovered image ($\rho = 0.75$)

Outline

- Today
  1. Convex constrained optimization and motivating examples
  2. Optimality condition
  3. Conjugate functions
  4. Monotone inclusion and monotone mixed variational inequality formulations
  5. Chambolle-Pock's primal-dual method
  6. Primal-dual hybrid gradient method
  7. Splitting methods
  8. Model-based excessive gap primal-dual method
- Next week
  1. Disciplined convex programming

Splitting methods

- **Splitting methods** have been widely used to solve monotone inclusions involving the **sum of two** maximal monotone operators.
- They can be used to solve the constrained convex optimization problem (2).

From the first line of (19) we have $\mathbf{v}^\star \in \partial g(\mathbf{K}\mathbf{u}^\star)$. Plug this into the second line of (19) to get

$$\boxed{0 \in \partial f(\mathbf{u}^\star) + \mathbf{K}^T \partial g(\mathbf{K}\mathbf{u}^\star)} \tag{29}$$

Splitting methods

- **Splitting methods** have been widely used to solve monotone inclusions involving the **sum of two** maximal monotone operators.
- They can be used to solve the constrained convex optimization problem (2).

From the first line of (19) we have $\mathbf{v}^\star \in \partial g(\mathbf{K}\mathbf{u}^\star)$. Plug this into the second line of (19) to get

$$\boxed{0 \in \partial f(\mathbf{u}^\star) + \mathbf{K}^T \partial g(\mathbf{K}\mathbf{u}^\star)} \quad (29)$$

Splitting monotone inclusion

Assumptions:

- Define $A(\mathbf{u}) := \mathbf{K}^T \partial g(\mathbf{K}\mathbf{u})$ and $B(\mathbf{u}) := \partial f(\mathbf{u})$.
- Assume that $\mathbf{K}$ is **full rank**.

Conclusion:

- A and B are two maximal monotone operators.
- (29) can be expressed as:

$$\boxed{0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)} \quad (30)$$

Splitting methods

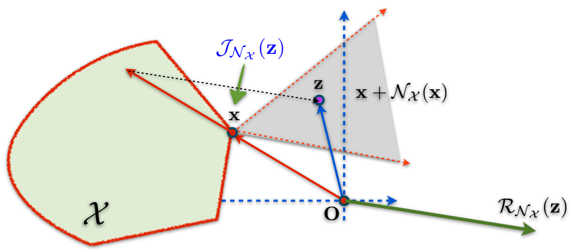
- **Splitting methods** have been widely used to solve monotone inclusions involving the **sum of two** maximal monotone operators.
- They can be used to solve the constrained convex optimization problem (2).

Reflection operator of the **resolvent**

Let $\mathcal{J}_{\mathcal{T}}(\cdot) := (\mathbb{I} + \mathcal{T})^{-1}(\cdot)$ be the **resolvent** of a maximal monotone operator $\mathcal{T}$. We define the reflection operator of $\mathcal{J}_{\mathcal{T}}$ as

$$\boxed{\mathcal{R}_{\mathcal{T}}(\mathbf{z}) := 2\mathcal{J}_{\mathcal{T}}(\mathbf{z}) - \mathbf{z}.}$$

Example (The reflection operator of the normal cone $\mathcal{N}_{\mathcal{X}}$)



Deriving a fixed-point formulation

- **A splitting method** generates an iterative sequence $\{\mathbf{u}^k\}$ by using a fixed-point derivation of the inclusion (30): $0 \in A(\mathbf{u}) + B(\mathbf{u})$.
- In addition, it splits the **computations** such that one can exploit the individual computations of A and B separately.

Deriving a fixed-point formulation

Starting from $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we can rewrite

$$2\mathbf{u} \in (\mathbb{I} + A)(\mathbf{u}) + (\mathbb{I} + B)(\mathbf{u}). \quad (31)$$

Deriving a fixed-point formulation

- **A splitting method** generates an iterative sequence $\{\mathbf{u}^k\}$ by using a fixed-point derivation of the inclusion (30): $0 \in A(\mathbf{u}) + B(\mathbf{u})$.
- In addition, it splits the **computations** such that one can exploit the individual computations of A and B separately.

Deriving a fixed-point formulation

Starting from $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we can rewrite

$$2\mathbf{u} \in (\mathbb{I} + A)(\mathbf{u}) + (\mathbb{I} + B)(\mathbf{u}). \tag{31}$$

Now, let take $\mathbf{z} \in (\mathbb{I} + B)(\mathbf{u})$, we have $\mathbf{u} = \mathcal{J}_B(\mathbf{z})$. The expression (31) leads to

$$2\mathbf{u} - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}) \quad \Leftrightarrow \quad 2\mathcal{J}_B(\mathbf{z}) - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}).$$

Deriving a fixed-point formulation

- **A splitting method** generates an iterative sequence $\{\mathbf{u}^k\}$ by using a fixed-point derivation of the inclusion (30): $0 \in A(\mathbf{u}) + B(\mathbf{u})$.
- In addition, it splits the **computations** such that one can exploit the individual computations of A and B separately.

Deriving a fixed-point formulation

Starting from $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we can rewrite

$$2\mathbf{u} \in (\mathbb{I} + A)(\mathbf{u}) + (\mathbb{I} + B)(\mathbf{u}). \tag{31}$$

Now, let take $\mathbf{z} \in (\mathbb{I} + B)(\mathbf{u})$, we have $\mathbf{u} = \mathcal{J}_B(\mathbf{z})$. The expression (31) leads to

$$2\mathbf{u} - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}) \quad \Leftrightarrow \quad 2\mathcal{J}_B(\mathbf{z}) - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}).$$

Using the resolvent of B, we can express the last inclusion as

$$\mathbf{u} = \mathcal{J}_A(2\mathcal{J}_B(\mathbf{z}) - \mathbf{z}) = \mathcal{J}_A(\mathcal{R}_B(\mathbf{z})).$$

Deriving a fixed-point formulation

- **A splitting method** generates an iterative sequence $\{\mathbf{u}^k\}$ by using a fixed-point derivation of the inclusion (30): $0 \in A(\mathbf{u}) + B(\mathbf{u})$.
- In addition, it splits the **computations** such that one can exploit the individual computations of A and B separately.

Deriving a fixed-point formulation

Starting from $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we can rewrite

$$2\mathbf{u} \in (\mathbb{I} + A)(\mathbf{u}) + (\mathbb{I} + B)(\mathbf{u}). \tag{31}$$

Now, let take $\mathbf{z} \in (\mathbb{I} + B)(\mathbf{u})$, we have $\mathbf{u} = \mathcal{J}_B(\mathbf{z})$. The expression (31) leads to

$$2\mathbf{u} - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}) \quad \Leftrightarrow \quad 2\mathcal{J}_B(\mathbf{z}) - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}).$$

Using the resolvent of B, we can express the last inclusion as

$$\mathbf{u} = \mathcal{J}_A(2\mathcal{J}_B(\mathbf{z}) - \mathbf{z}) = \mathcal{J}_A(\mathcal{R}_B(\mathbf{z})).$$

Hence, by using the reflection operator of B, we can rewrite this equivalently to

$$\mathbf{z} = 2\mathcal{J}_A(\mathcal{R}_B(\mathbf{z})) - (2\mathbf{u} - \mathbf{z}) = 2\mathcal{J}_A(\mathcal{R}_B(\mathbf{z})) - \mathcal{R}_B(\mathbf{z}).$$

Deriving a fixed-point formulation

- **A splitting method** generates an iterative sequence $\{\mathbf{u}^k\}$ by using a fixed-point derivation of the inclusion (30): $0 \in A(\mathbf{u}) + B(\mathbf{u})$.
- In addition, it splits the **computations** such that one can exploit the individual computations of A and B separately.

Deriving a fixed-point formulation

Starting from $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we can rewrite

$$2\mathbf{u} \in (\mathbb{I} + A)(\mathbf{u}) + (\mathbb{I} + B)(\mathbf{u}). \tag{31}$$

Now, let take $\mathbf{z} \in (\mathbb{I} + B)(\mathbf{u})$, we have $\mathbf{u} = \mathcal{J}_B(\mathbf{z})$. The expression (31) leads to

$$2\mathbf{u} - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}) \quad \Leftrightarrow \quad 2\mathcal{J}_B(\mathbf{z}) - \mathbf{z} \in (\mathbb{I} + A)(\mathbf{u}).$$

Using the resolvent of B, we can express the last inclusion as

$$\mathbf{u} = \mathcal{J}_A(2\mathcal{J}_B(\mathbf{z}) - \mathbf{z}) = \mathcal{J}_A(\mathcal{R}_B(\mathbf{z})).$$

Hence, by using the reflection operator of B, we can rewrite this equivalently to

$$\mathbf{z} = 2\mathcal{J}_A(\mathcal{R}_B(\mathbf{z})) - (2\mathbf{u} - \mathbf{z}) = 2\mathcal{J}_A(\mathcal{R}_B(\mathbf{z})) - \mathcal{R}_B(\mathbf{z}).$$

Using the reflection operator of A we finally get

$$\boxed{\mathbf{z} = \mathcal{R}_A(\mathcal{R}_B(\mathbf{z})).}$$

Fixed-point characterization and splitting computation

Fixed-point formulation

- If $\mathbf{u}^\star$ is a solution of (30) (cf. $0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)$) then

$$\boxed{\mathbf{z}^\star = \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{32}$$

- Alternatively, if $\mathbf{u}^\star$ is a solution of (30) then for any $\beta \neq 0$, we have

$$\boxed{\mathbf{z}^\star = (1-\beta)\mathbf{z}^\star + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{33}$$

Fixed-point characterization and splitting computation

Fixed-point formulation

- If $\mathbf{u}^\star$ is a solution of (30) (cf. $0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)$) then

$$\boxed{\mathbf{z}^\star = \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{32}$$

- Alternatively, if $\mathbf{u}^\star$ is a solution of (30) then for any $\beta \neq 0$, we have

$$\boxed{\mathbf{z}^\star = (1-\beta)\mathbf{z}^\star + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{33}$$

Splitting computation

Let assume that our iterative scheme is based on the fixed point formulation (33) as:

$$\mathbf{z}^{k+1} := (1-\beta)\mathbf{z}^k + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right).$$

We can split this formula by using the definition of $\mathcal{R}$ and $\mathcal{J}$.

- Let $\mathbf{u}^k := \mathcal{J}_B(\mathbf{z}^k)$. Then $\mathcal{R}_B(\mathbf{z}^k) = 2\mathbf{u}^k - \mathbf{z}^k$.

Fixed-point characterization and splitting computation

Fixed-point formulation

- If $\mathbf{u}^\star$ is a solution of (30) (cf. $0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)$) then

$$\boxed{\mathbf{z}^\star = \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{32}$$

- Alternatively, if $\mathbf{u}^\star$ is a solution of (30) then for any $\beta \neq 0$, we have

$$\boxed{\mathbf{z}^\star = (1-\beta)\mathbf{z}^\star + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{33}$$

Splitting computation

Let assume that our iterative scheme is based on the fixed point formulation (33) as:

$$\mathbf{z}^{k+1} := (1-\beta)\mathbf{z}^k + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right).$$

We can split this formula by using the definition of $\mathcal{R}$ and $\mathcal{J}$.

- Let $\mathbf{u}^k := \mathcal{J}_B(\mathbf{z}^k)$. Then $\mathcal{R}_B(\mathbf{z}^k) = 2\mathbf{u}^k - \mathbf{z}^k$.
- $\mathcal{J}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) = \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) := \mathbf{v}^k$.

Fixed-point characterization and splitting computation

Fixed-point formulation

- If $\mathbf{u}^\star$ is a solution of (30) (cf. $0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)$) then

$$\boxed{\mathbf{z}^\star = \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \text{ and } \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{32}$$

- Alternatively, if $\mathbf{u}^\star$ is a solution of (30) then for any $\beta \neq 0$, we have

$$\boxed{\mathbf{z}^\star = (1-\beta)\mathbf{z}^\star + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \text{ and } \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{33}$$

Splitting computation

Let assume that our iterative scheme is based on the fixed point formulation (33) as:

$$\mathbf{z}^{k+1} := (1-\beta)\mathbf{z}^k + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right).$$

We can split this formula by using the definition of $\mathcal{R}$ and $\mathcal{J}$.

- Let $\mathbf{u}^k := \mathcal{J}_B(\mathbf{z}^k)$. Then $\mathcal{R}_B(\mathbf{z}^k) = 2\mathbf{u}^k - \mathbf{z}^k$.
- $\mathcal{J}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) = \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) := \mathbf{v}^k$.
- $\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) = 2\mathbf{v}^k - (2\mathbf{u}^k - \mathbf{z}^k)$.

Fixed-point characterization and splitting computation

Fixed-point formulation

- If $\mathbf{u}^\star$ is a solution of (30) (cf. $0 \in A(\mathbf{u}^\star) + B(\mathbf{u}^\star)$) then

$$\boxed{\mathbf{z}^\star = \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{32}$$

- Alternatively, if $\mathbf{u}^\star$ is a solution of (30) then for any $\beta \neq 0$, we have

$$\boxed{\mathbf{z}^\star = (1-\beta)\mathbf{z}^\star + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^\star)\right) \quad \text{and} \quad \mathbf{u}^\star = \mathcal{J}_B(\mathbf{z}^\star).} \tag{33}$$

Splitting computation

Let assume that our iterative scheme is based on the fixed point formulation (33) as:

$$\mathbf{z}^{k+1} := (1-\beta)\mathbf{z}^k + \beta\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right).$$

We can split this formula by using the definition of $\mathcal{R}$ and $\mathcal{J}$.

- Let $\mathbf{u}^k := \mathcal{J}_B(\mathbf{z}^k)$. Then $\mathcal{R}_B(\mathbf{z}^k) = 2\mathbf{u}^k - \mathbf{z}^k$.
- $\mathcal{J}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) = \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) := \mathbf{v}^k$.
- $\mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) = 2\mathbf{v}^k - (2\mathbf{u}^k - \mathbf{z}^k)$.
- $\mathbf{z}^{k+1} = (1-\beta)\mathbf{z}^k + \beta(2\mathbf{v}^k - (2\mathbf{u}^k - \mathbf{z}^k)) = \mathbf{z}^k + 2\beta(\mathbf{v}^k - \mathbf{u}^k)$.

Splitting: Douglas-Rachford's method

Fixed-point iteration

The **Douglas-Rachford method** bases on the fixed-point formulation (33) to generate an iterative sequence as:

$$\mathbf{z}^{k+1} := (1-\beta_k)\mathbf{z}^k + \beta_k \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) \quad \text{and} \quad \mathbf{u}^k = \mathcal{J}_B(\mathbf{z}^k).$$

By **splitting the computation** as in the previous slide, we can summarize this scheme as:

$$\begin{cases} \mathbf{u}^k & := \mathcal{J}_B(\mathbf{z}^k) \\ \mathbf{v}^k & := \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) \\ \mathbf{z}^{k+1} & := \mathbf{z}^k + \eta_k(\mathbf{v}^k - \mathbf{u}^k) \end{cases}$$

for $\eta_k := 2\beta_k \neq 0$.

Splitting: Douglas-Rachford's method

Fixed-point iteration

The **Douglas-Rachford method** bases on the fixed-point formulation (33) to generate an iterative sequence as:

$$\mathbf{z}^{k+1} := (1-\beta_k)\mathbf{z}^k + \beta_k \mathcal{R}_A\left(\mathcal{R}_B(\mathbf{z}^k)\right) \quad \text{and} \quad \mathbf{u}^k = \mathcal{J}_B(\mathbf{z}^k).$$

By **splitting the computation** as in the previous slide, we can summarize this scheme as:

$$\begin{cases} \mathbf{u}^k & := \mathcal{J}_B(\mathbf{z}^k) \\ \mathbf{v}^k & := \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) \\ \mathbf{z}^{k+1} & := \mathbf{z}^k + \eta_k(\mathbf{v}^k - \mathbf{u}^k) \end{cases}$$

for $\eta_k := 2\beta_k \neq 0$.

Douglas-Rachford's algorithm (DRA)
1. Given $\mathbf{z}^0 \in \text{dom}(B)$ as an initial point and $\eta_0 \neq 0$.
2. For $k = 0, 1, \cdots$, perform: $\begin{cases} \mathbf{u}^k & := \mathcal{J}_B(\mathbf{z}^k) \\ \mathbf{v}^k & := \mathcal{J}_A(2\mathbf{u}^k - \mathbf{z}^k) \\ \mathbf{z}^{k+1} & := \mathbf{z}^k + \eta_k(\mathbf{v}^k - \mathbf{u}^k) \end{cases}$
and update η_k if required.

Douglas-Rachford method for convex problem (15)

In order to apply DRA for solving (15), we define $A(\mathbf{u}) := \tau\mathbf{K}^T\partial g(\mathbf{K}\mathbf{u})$ and $B(\mathbf{u}) := \tau\partial f(\mathbf{u})$ for a given scaling factor $\tau > 0$.

Douglas-Rachford's method for solving (15)

1. Given $\mathbf{z}^0 \in \mathrm{dom}(f)$ as an initial point. Choose $\tau_0 > 0$ and $\eta_0 > 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{u}^k & := \mathrm{prox}_{\tau_k f}(\mathbf{z}^k), \\ \mathbf{v}^k & := \operatorname*{argmin}_{\mathbf{v}} \left\{ g(\mathbf{K}\mathbf{v}) + (1/(2\tau_k))\|\mathbf{v} - 2\mathbf{u}^k + \mathbf{z}^k\|_2^2 \right\}, \\ \mathbf{z}^{k+1} & := \mathbf{z}^k + \eta_k(\mathbf{v}^k - \mathbf{u}^k). \end{cases}$$

and update τ_k and η_k if necessary.

Douglas-Rachford method for convex problem (15)

In order to apply DRA for solving (15), we define $A(\mathbf{u}) := \tau\mathbf{K}^T\partial g(\mathbf{K}\mathbf{u})$ and $B(\mathbf{u}) := \tau\partial f(\mathbf{u})$ for a given scaling factor $\tau > 0$.

Douglas-Rachford's method for solving (15)

1. Given $\mathbf{z}^0 \in \mathrm{dom}(f)$ as an initial point. Choose $\tau_0 > 0$ and $\eta_0 > 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{u}^k & := \mathrm{prox}_{\tau_k f}(\mathbf{z}^k), \\ \mathbf{v}^k & := \underset{\mathbf{v}}{\mathrm{argmin}}\left\{g(\mathbf{K}\mathbf{v}) + (1/(2\tau_k))\|\mathbf{v} - 2\mathbf{u}^k + \mathbf{z}^k\|_2^2\right\}, \\ \mathbf{z}^{k+1} & := \mathbf{z}^k + \eta_k(\mathbf{v}^k - \mathbf{u}^k). \end{cases}$$

and update τ_k and η_k if necessary.

Remark: Quadratic loss and diagonalizable operator

- **Assumptions:** $g(\mathbf{K}\mathbf{v}) := (1/2)\|\mathbf{K}\mathbf{v} - \mathbf{b}\|_2^2$ and $\mathbf{K}^T\mathbf{K} = \Sigma$, where Σ is diagonal.
- **Conclusion:** $\boxed{\mathbf{v}^k := (\tau_k^{-1}\mathbb{I} + \Sigma)^{-1}(\mathbf{K}^T\mathbf{b} + \tau_k^{-1}\mathbf{s}^k)}$.
  - Let $\mathbf{s}^k := 2\mathbf{u}^k - \mathbf{z}^k$. We can write the optimality condition of $\mathbf{v}^k$ as

$$(\tau_k^{-1}\mathbb{I} + \mathbf{K}^T\mathbf{K})\mathbf{v}^k = \mathbf{K}^T\mathbf{b} + \tau_k^{-1}\mathbf{s}^k.$$

  - Since $\mathbf{K}^T\mathbf{K} = \Sigma$, we can compute $\mathbf{v}^k$ explicitly as

$$\mathbf{v}^k = (\tau_k^{-1}\mathbb{I} + \Sigma)^{-1}(\mathbf{K}^T\mathbf{b} + \tau_k^{-1}\mathbf{s}^k).$$

Convergence of splitting methods

Theorem (Convergence of Douglas-Rachford's method [4])

Assume that the ***solution set*** $\mathcal{U}^\star$ *of* (30) *is nonempty and the sequence* $\{\eta_k\}$ *is chosen such that*

$$\eta_k \in [0,2] \ \text{ and } \ \sum_{k=0}^{+\infty} \eta_k(2-\eta_k) = \infty. \tag{34}$$

Then, the sequence $\{\mathbf{u}^k\}$ *generated by the* ***Douglas-Rachford algorithm*** *converges to a* ***solution*** $\mathbf{u}^\star$ *in* $\mathcal{U}^\star$.

Convergence of splitting methods

Theorem (Convergence of Douglas-Rachford's method [4])

Assume that the ***solution set*** $\mathcal{U}^\star$ *of* (30) *is nonempty and the sequence* $\{\eta_k\}$ *is chosen such that*

$$\eta_k \in [0, 2] \;\; \text{and} \;\; \sum_{k=0}^{+\infty} \eta_k (2 - \eta_k) = \infty. \tag{34}$$

Then, the sequence $\{\mathbf{u}^k\}$ *generated by the* ***Douglas-Rachford algorithm*** *converges to a* ***solution*** $\mathbf{u}^\star$ *in* $\mathcal{U}^\star$.

Remarks

- We can choose η_k as a constant step, e.g., $\eta_k = 1$ for $k \geq 0$.
- If $\eta_k = 1$ for all $k \geq 0$, then **Douglas-Rachford's algorithm** coincides with **Peaceman-Rachford's method** [10].
- When **Douglas-Rachford's algorithm** is applied to solve (15), it coincides with **ADMM** [3].

Alternative derivation

Assumptions

- The **solution set** $\mathcal{U}^\star$ of $0 \in A(\mathbf{u}) + B(\mathbf{u})$ is nonempty
- For simplicity of discussion, we assume that B is single-valued and $\beta = 1/2$.

By using the definition of $\mathcal{R}_A = 2\mathcal{J}_A - \mathbb{I}$ and $\mathcal{R}_B = 2\mathcal{J}_B - \mathbb{I}$, the **Douglas-Rachford** iterative scheme

$$\begin{cases} \mathbf{u}^k & := \mathcal{J}_B(\mathbf{z}^k), \\ \mathbf{z}^{k+1} & := (1/2)\mathbf{z}^k + (1/2)\mathcal{R}_A(\mathcal{R}_B(\mathbf{z}^k)). \end{cases}$$

can be expressed as

$$\mathbf{u}^{k+1} := \mathcal{J}_B\left(\mathcal{J}_A(2\mathcal{J}_B(\mathcal{J}_B^{-1}(\mathbf{u}^k)) - \mathcal{J}_B^{-1}(\mathbf{u}^k)) + \mathcal{J}_B^{-1}(\mathbf{u}^k) - \mathcal{J}_B(\mathcal{J}_B^{-1}(\mathbf{u}^k))\right). \quad (35)$$

- First, it is obvious that $\mathcal{J}_B(\mathcal{J}_B^{-1}(\mathbf{u}^k)) = \mathbf{u}^k$
- Second, by definition of $\mathcal{J}_B$, we also have $\mathcal{J}_B^{-1}(\mathbf{u}^k) = \mathbf{u}^k + B(\mathbf{u}^k)$.

Substituting these relations into (35), we obtain

$$\boxed{\mathbf{u}^{k+1} := \mathcal{J}_B\left(\mathcal{J}_A(\mathbf{u}^k - B(\mathbf{u}^k)) + B(\mathbf{u}^k)\right)} \quad (36)$$

We can rewrite (36) as

$$\boxed{(\mathbb{I} + B)(\mathbf{u}^{k+1}) \in (\mathbb{I} + B)(\mathbf{u}^k) - e(\mathbf{u}^k)} \quad (37)$$

where $e(\mathbf{u}^k) := \mathbf{u}^k - \mathcal{J}_A(\mathbf{u}^k - B(\mathbf{u}^k))$.

Convergence rate

Facts:

- It is obvious that $\mathbf{u}^\star = \mathcal{J}_A(\mathbf{u}^\star - B(\mathbf{u}^\star))$ for any $\mathbf{u}^\star \in \mathcal{U}^\star$.
- We define $e(\mathbf{u}) := \mathbf{u} - \mathcal{J}_A(\mathbf{u} - B(\mathbf{u}))$ the residual operator at $\mathbf{u}$. Then $e(\mathbf{u}^\star) = 0$.
- Let $\mathcal{T} := \mathbb{I} + A$. The **Douglas-Rachford** scheme (37) can be written as

$$\boxed{\mathcal{T}(\mathbf{u}^{k+1}) \in \mathcal{T}(\mathbf{u}^k) - e(\mathbf{u}^k)} \tag{38}$$

Theorem (Convergence rate [7])

*Assume that B is single-valued and the **solution set** $\mathcal{U}^\star$ of $0 \in A(\mathbf{u}) + B(\mathbf{u})$ is nonempty. Let $\{\mathbf{u}^k\}_{k\geq 0}$ be the sequence generated by the **Douglas-Rachford algorithm** with $\eta_k = 1$ for all $k \geq 0$. Then for any **solution** $\mathbf{u}^\star$ of $0 \in A(\mathbf{u}) + B(\mathbf{u})$, we have*

$$\|e(\mathbf{u}^k)\|^2 \leq \frac{1}{k+1}\|(\mathbf{u}^0 - \mathbf{u}^\star) + (B(\mathbf{u}^0) - B(\mathbf{u}^\star))\|^2. \tag{39}$$

The sequence $\{\|e(\mathbf{u}^k)\|^2\}$ converges to zero at the $\mathcal{O}(1/k)$ rate. Consequently, $\{\mathbf{u}^k\}$ converges to a solution $\mathbf{u}^\star$ of $0 \in A(\mathbf{u}) + B(\mathbf{u})$.

Remarks

- Since we assume that B is single-valued, the right-hand side (39) is bounded.
- In the case B is a set-valued mapping, the **right-hand side** may be no longer bounded. For example, if $B = \mathcal{N}_{\mathcal{Z}}$ the normal cone of a convex set $\mathcal{Z}$.

Example: Image denoising

TV-denoising

Given a noisy image $\mathbf{b} \in \mathbb{R}^{m\times n}$, we want to recover a clean image $\mathbf{x}$ by using the total variation operator. This problem can be formulated as:

$$\boxed{\min_{\mathbf{x}\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathbf{x}-\mathbf{b}\|^2 + \rho\|\mathbf{x}\|_{\mathrm{TV}},}$$

where $\rho > 0$ is a regularization parameter and

$$\|\mathbf{x}\|_{\mathrm{TV}} := \begin{cases} \sum_{i,j} |x_{i,j+1} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| & \text{anisotropic (ATV)} \\ \sum_{i,j} \sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} & \text{isotropic (ITV)} \end{cases}$$

Example: Image denoising

TV-denoising

Given a noisy image $\mathbf{b} \in \mathbb{R}^{m \times n}$, we want to recover a clean image $\mathbf{x}$ by using the total variation operator. This problem can be formulated as:

$$\boxed{\min_{\mathbf{x} \in \mathbb{R}^{m \times n}} \frac{1}{2}\|\mathbf{x} - \mathbf{b}\|^2 + \rho\|\mathbf{x}\|_{\mathrm{TV}},}$$

where $\rho > 0$ is a regularization parameter and

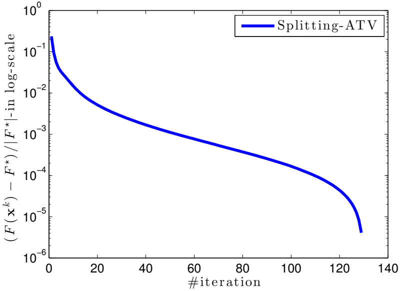
$$\|\mathbf{x}\|_{\mathrm{TV}} := \begin{cases} \sum_{i,j} |x_{i,j+1} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| & \text{anisotropic (ATV)} \\ \sum_{i,j} \sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} & \text{isotropic (ITV)} \end{cases}$$

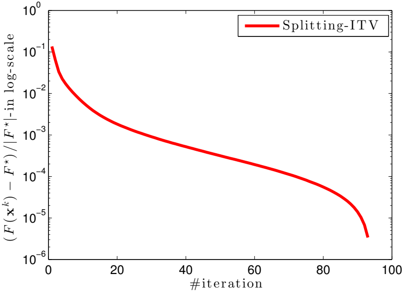
How to apply the **splitting algorithm**?

- By letting $\mathbf{z} = \mathbf{D}\mathbf{x}$, we can convert the TV-denoising problem into (2), where $\mathbf{D}$ is a matrix representing the **total variation**.
- **Splitting algorithm** is now applied to the resulting problem.
- We choose $\tau_k = \eta_k = 1$ in our test.

Example: Image denoising - convergence behavior

- The convergence of the splitting method using ATV and ITV norms on a gray image of size 512×512.
- The regularization parameter $\rho = 20$.





- The objective value: 9.9319×10^7
- Relative error:
$$\|\mathbf{x}^k - \mathbf{x}^\natural\|/\|\mathbf{x}^\natural\| = 0.0727258$$
where $\mathbf{x}^\natural$ is the original image
- The number of iterations: 131
- The CPU time: 30.6s.
- The PSNR $= 23.1983$.

- The objective value: 9.4897×10^7
- Relative error:
$$\|\mathbf{x}^k - \mathbf{x}^\natural\|/\|\mathbf{x}^\natural\| = 0.0676706$$
where $\mathbf{x}^\natural$ is the original image
- The number of iterations: 95
- The CPU time: 26.0s.
- The PSNR $= 23.6192$.

Example: Image denoising

Original

Noisy

Example: Image denoising

Original

Noisy

Anisotropic TV denoising

Isotropic TV denoising

Primal-dual hybrid gradient (PDHG) algorithm

The idea of PDHG

- Originally, PDHG is a combination of a primal and dual proximal-gradient descent step applying to the min-max problem (17) [17]:

$$\min_{\mathbf{u}\in\mathbb{R}^p}\max_{\mathbf{v}\in\mathbb{R}^q}\Big\{\langle\mathbf{K}\mathbf{u},\mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v})\Big\}.$$

- First, PDHG performs a primal proximal-gradient step on the minimization problem w.r.t. $\mathbf{u}$ given $\mathbf{v}^k$ to compute $\mathbf{u}^{k+1}$.
- Second, PDHG performs a dual proximal-gradient step on the maximization problem w.r.t. $\mathbf{v}$ given $\mathbf{u}^{k+1}$ to compute $\mathbf{v}^{k+1}$.
- We can also add an intermediate step $\bar{\mathbf{u}}^{k+1}$ before performing the dual step.

Primal-dual hybrid gradient (PDHG) algorithm

The idea of PDHG

- Originally, PDHG is a combination of a primal and dual proximal-gradient descent step applying to the min-max problem (17) [17]:

$$\min_{\mathbf{u}\in\mathbb{R}^p}\max_{\mathbf{v}\in\mathbb{R}^q}\Big\{\langle\mathbf{K}\mathbf{u},\mathbf{v}\rangle + f(\mathbf{u}) - g^*(\mathbf{v})\Big\}.$$

- First, PDHG performs a primal proximal-gradient step on the minimization problem w.r.t. $\mathbf{u}$ given $\mathbf{v}^k$ to compute $\mathbf{u}^{k+1}$.
- Second, PDHG performs a dual proximal-gradient step on the maximization problem w.r.t. $\mathbf{v}$ given $\mathbf{u}^{k+1}$ to compute $\mathbf{v}^{k+1}$.
- We can also add an intermediate step $\bar{\mathbf{u}}^{k+1}$ before performing the dual step.

Primal-Dual Hybrid Gradient algorithm (PDHG) for (15)

1. Given $\mathbf{u}^0 \in \mathbb{R}^p$ and $\mathbf{v}^0 \in \mathbb{R}^n$ as an initial point.
2. Choose $\tau_0 > 0$, $\sigma_0 > 0$ and $\eta_0 \neq 0$.
3. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{u}^{k+1} & := \mathrm{prox}_{\tau_k f}(\mathbf{u}^k - \tau_k\mathbf{K}^T\mathbf{v}^k), \\ \bar{\mathbf{u}}^{k+1} & := \mathbf{u}^{k+1} + \eta_k(\mathbf{u}^{k+1} - \mathbf{u}^k), \\ \mathbf{v}^{k+1} & := \mathrm{prox}_{\sigma_k g^*}(\mathbf{v}^k + \sigma_k\mathbf{K}\bar{\mathbf{u}}^{k+1}). \end{cases}$$

and update τ_k, σ_k and η_k if necessary.

Connection to Chambolle-Pock's algorithm and enhancement

Connection to Chambolle-Pock's algorithm

- **PDHG** is very similar to **Chambolle-Pock's algorithm**.
  - **Chambolle-Pock's algorithm** performs the dual step on $\mathbf{v}$ and then the primal step on $\mathbf{u}$.
  - **PDHG** performs the primal step on $\mathbf{u}$ and then the dual step on $\mathbf{v}$.
  - Symmetrically, we can say that both methods are equivalent.
- The convergence theory of **Chambolle-Pock's algorithm** is applicable to **PDHG**.

Connection to Chambolle-Pock's algorithm and enhancement

Connection to Chambolle-Pock's algorithm

- **PDHG** is very similar to **Chambolle-Pock's algorithm**.
  - **Chambolle-Pock's algorithm** performs the dual step on $\mathbf{v}$ and then the primal step on $\mathbf{u}$.
  - **PDHG** performs the primal step on $\mathbf{u}$ and then the dual step on $\mathbf{v}$.
  - Symmetrically, we can say that both methods are equivalent.
- The convergence theory of **Chambolle-Pock's algorithm** is applicable to **PDHG**.

Adaptive PDHG

- If we select the parameters τ_k and σ_k based on the convergence theory, PDHG often has poor performance in practice.
- We can enhance the performance of PDHG by adaptively updating τ_k and σ_k [5].
  - Define the primal and dual residuals $(\mathbf{p}^{k+1}, \mathbf{d}^{k+1})$:

$$\begin{cases} \mathbf{p}^{k+1} & := \tau_k^{-1}(\mathbf{u}^k - \mathbf{u}^{k+1}) - \mathbf{K}^T(\mathbf{v}^k - \mathbf{v}^{k+1}), \\ \mathbf{d}^{k+1} & := \sigma_k^{-1}(\mathbf{v}^k - \mathbf{v}^{k+1}) + \mathbf{K}(\mathbf{u}^k - \mathbf{u}^{k+1}). \end{cases}$$

  - Adaptively update τ_k and σ_k by trading-off the primal residual $\|\mathbf{p}^{k+1}\|$ and dual residual $\|\mathbf{d}^{k+1}\|$ at each iteration.

Outline

- Today
  1. Convex constrained optimization and motivating examples
  2. Optimality condition
  3. Conjugate functions
  4. Monotone inclusion and monotone mixed variational inequality formulations
  5. Chambolle-Pock's primal-dual method
  6. Primal-dual hybrid gradient method
  7. Splitting methods
  8. Model-based excessive gap primal-dual method
- Next week
  1. Disciplined convex programming

Primal-dual method using model-based excessive gap technique

Problem restatement

We consider again problem (2) with additional convex constraint as:

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathcal{X} \end{cases} \tag{40}$$

where f, $\mathbf{A}$ and $\mathbf{b}$ are defined as in (2) and $\mathcal{X}$ is a nonempty, closed, convex and bounded set in $\mathbb{R}^p$.

We recall the variational inequality presenting the optimality condition of (40) as

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathcal{X} \times \mathbb{R}^n \tag{41}$$

where $\mathbf{z}^\star := (\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ is a primal-dual solution of (40) and

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix}.$$

Primal-dual method using model-based excessive gap technique

Problem restatement

We consider again problem (2) with additional convex constraint as:

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathcal{X} \end{cases} \tag{40}$$

where f, $\mathbf{A}$ and $\mathbf{b}$ are defined as in (2) and $\mathcal{X}$ is a nonempty, closed, convex and bounded set in $\mathbb{R}^p$.

We recall the variational inequality presenting the optimality condition of (40) as

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathcal{X} \times \mathbb{R}^n \tag{41}$$

where $\mathbf{z}^\star := (\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ is a primal-dual solution of (40) and

$$M(\mathbf{z}) := \begin{bmatrix} \mathbf{A}^T\lambda \\ \mathbf{b} - \mathbf{A}\mathbf{x} \end{bmatrix}.$$

Gap function

The **gap function** of (41) is redefined as (different from (14) at $\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n$):

$$G(\mathbf{z}) := \max_{\hat{\mathbf{z}}\in\mathcal{X}\times\mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\} \tag{42}$$

Smoothed gap function

- Let b be a prox-function of $\mathcal{X}$ with strong convexity parameter $\sigma_b = 1$.
  - i.e., b is a smooth, strongly convex function with strong convexity parameter $\sigma_b = 1$.
- We define ξ the Bregman distance corresponding to b as

$$\xi(\mathbf{x}, \hat{\mathbf{x}}) := b(\mathbf{x}) - b(\hat{\mathbf{x}}) - \nabla b(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}).$$

- For $\mathbf{x}_c \in \mathbb{R}^p$ and two positive parameters γ and β we define

$$\xi_{\gamma\beta}(\mathbf{z}) := \gamma\xi(\mathbf{x}, \mathbf{x}_c) + (\beta/2)\|\lambda\|_2^2 \tag{43}$$

  a smoother for the gap function (42).
- $\xi_{\gamma\beta}$ is strongly convex and satisfies $\xi_{\gamma\beta}(\mathbf{z}) \geq (\gamma/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2 + (\beta/2)\|\lambda\|_2^2$.

Smoothed gap function

We define a smoothed version for the gap function G given by (42) as follows:

Smoothed gap function

- Let b be a prox-function of $\mathcal{X}$ with strong convexity parameter $\sigma_b = 1$.
  - i.e., b is a smooth, strongly convex function with strong convexity parameter $\sigma_b = 1$.
- We define ξ the Bregman distance corresponding to b as

$$\xi(\mathbf{x}, \hat{\mathbf{x}}) := b(\mathbf{x}) - b(\hat{\mathbf{x}}) - \nabla b(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}).$$

- For $\mathbf{x}_c \in \mathbb{R}^p$ and two positive parameters γ and β we define

$$\xi_{\gamma\beta}(\mathbf{z}) := \gamma\xi(\mathbf{x}, \mathbf{x}_c) + (\beta/2)\|\lambda\|_2^2 \tag{43}$$

  a smoother for the gap function (42).
- $\xi_{\gamma\beta}$ is strongly convex and satisfies $\xi_{\gamma\beta}(\mathbf{z}) \geq (\gamma/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2 + (\beta/2)\|\lambda\|_2^2$.

Smoothed gap function

We define a smoothed version for the gap function G given by (42) as follows:

$$G(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\}$$

Smoothed gap function

- Let b be a prox-function of $\mathcal{X}$ with strong convexity parameter $\sigma_b = 1$.
  - i.e., b is a smooth, strongly convex function with strong convexity parameter $\sigma_b = 1$.
- We define ξ the Bregman distance corresponding to b as

$$\xi(\mathbf{x}, \hat{\mathbf{x}}) := b(\mathbf{x}) - b(\hat{\mathbf{x}}) - \nabla b(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}).$$

- For $\mathbf{x}_c \in \mathbb{R}^p$ and two positive parameters γ and β we define

$$\xi_{\gamma\beta}(\mathbf{z}) := \gamma\xi(\mathbf{x}, \mathbf{x}_c) + (\beta/2)\|\lambda\|_2^2 \qquad (43)$$

  a smoother for the gap function (42).
- $\xi_{\gamma\beta}$ is strongly convex and satisfies $\xi_{\gamma\beta}(\mathbf{z}) \geq (\gamma/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2 + (\beta/2)\|\lambda\|_2^2$.

Smoothed gap function

We define a smoothed version for the gap function G given by (42) as follows:

$$G_{\gamma\beta}(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) - \xi_{\gamma\beta}(\hat{\mathbf{z}}) \right\} \qquad (44)$$

Smoothed gap function

- Let b be a prox-function of $\mathcal{X}$ with strong convexity parameter $\sigma_b = 1$.
  - i.e., b is a smooth, strongly convex function with strong convexity parameter $\sigma_b = 1$.
- We define ξ the Bregman distance corresponding to b as

$$\xi(\mathbf{x}, \hat{\mathbf{x}}) := b(\mathbf{x}) - b(\hat{\mathbf{x}}) - \nabla b(\hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}).$$

- For $\mathbf{x}_c \in \mathbb{R}^p$ and two positive parameters γ and β we define

$$\xi_{\gamma\beta}(\mathbf{z}) := \gamma\xi(\mathbf{x}, \mathbf{x}_c) + (\beta/2)\|\lambda\|_2^2 \tag{43}$$

  a smoother for the gap function (42).
- $\xi_{\gamma\beta}$ is strongly convex and satisfies $\xi_{\gamma\beta}(\mathbf{z}) \geq (\gamma/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2 + (\beta/2)\|\lambda\|_2^2$.

Smoothed gap function

We define a smoothed version for the gap function G given by (42) as follows:

$$G_{\gamma\beta}(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) - \xi_{\gamma\beta}(\hat{\mathbf{z}}) \right\} \tag{44}$$

Properties of $G_{\gamma\beta}$

- Evaluating $G_{\gamma\beta}$ and its gradient requires to solve a strongly convex program (42)
- $G_{\gamma\beta}(\mathbf{z}) \to G(\mathbf{z})$ as γ and β go to zero for all $\mathbf{z} \in \mathbb{R}^{p+n}$.

Comments on the gap function G and its smooth version $G_{\gamma\beta}$

- The gap function G defined by (42):

$$G(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\}$$

can be written as

$$G(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathbb{R}^p \times \mathbb{R}^n} \left\{ (f(\mathbf{x}) + \iota_{\mathcal{X}}(\mathbf{x})) - (f(\hat{\mathbf{x}}) + \iota_{\mathcal{X}}(\hat{\mathbf{x}})) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \right\}$$

which is exactly the gap function G in the unconstrained form (14), where $\iota_{\mathcal{X}}$ is the indicator function of $\mathcal{X}$.

- If we define

$$\begin{aligned} d_\gamma(\lambda) &:= \min_{\mathbf{x} \in \mathcal{X}} \left\{ f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma \xi(\mathbf{x}; \mathbf{x}_c) \right\} \\ f_\beta(\mathbf{x}) &:= f(\mathbf{x}) + \max_{\lambda \in \mathbb{R}^n} \{ \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) - (\beta/2)\|\lambda\|^2 \} \\ &= f(\mathbf{x}) + (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \end{aligned}$$

then:

  - d_γ is an approximation of the **dual function** d
  - f_β is an approximation of the **objective function** f.

Moreover, we have

$$G_{\gamma\beta}(\mathbf{z}) = f_\beta(\mathbf{x}) - d_\gamma(\lambda).$$

Illustration of smooth gap function $G_{\gamma\beta}$

- The **objective function** $f(\mathbf{x})$ does not depend on β
- The **dual function** $d(\lambda)$ does not depend on γ:

$$d(\bar{\lambda}) := \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \bar{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}.$$

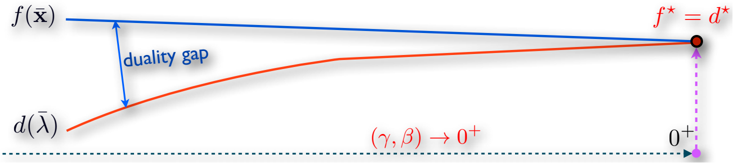
- $f(x)$ is decreasing over $\mathbf{x}$ and $d(\lambda)$ is increasing over λ.

Illustration of smooth gap function $G_{\gamma\beta}$

- The **objective function** $f(\mathbf{x})$ does not depend on β
- The **dual function** $d(\lambda)$ does not depend on γ:

$$d(\bar{\lambda}) := \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \bar{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}.$$

- $f(x)$ is decreasing over $\mathbf{x}$ and $d(\lambda)$ is increasing over λ.



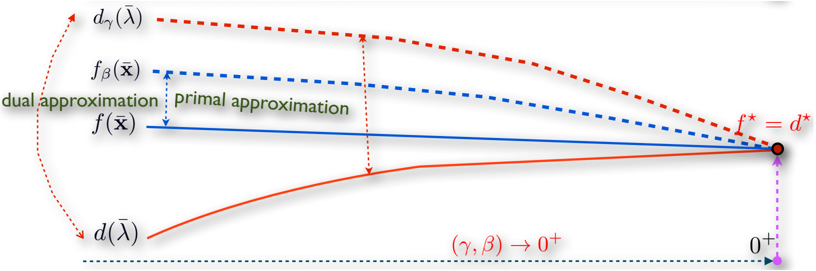
- The duality gap is defined as $G(\bar{\mathbf{z}}) := f(\bar{\mathbf{x}}) - d(\bar{\lambda}) \geq 0$.
- At the optimal solution $\mathbf{z}^\star := (\mathbf{x}^\star, \lambda^\star)$, one has $f(\mathbf{x}^\star) = d(\lambda^\star)$ and $G(\mathbf{z}^\star) = 0$.

Illustration of smooth gap function $G_{\gamma\beta}$

- The **objective function** $f(\mathbf{x})$ does not depend on β
- The **dual function** $d(\lambda)$ does not depend on γ:

$$d(\bar{\lambda}) := \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \bar{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}.$$

- $f(x)$ is decreasing over $\mathbf{x}$ and $d(\lambda)$ is increasing over λ.



- The augmented function f_β approximates f: $f_\beta(\mathbf{x}) = f(\mathbf{x}) + (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$
- The smoothed dual function d_γ approximates d:

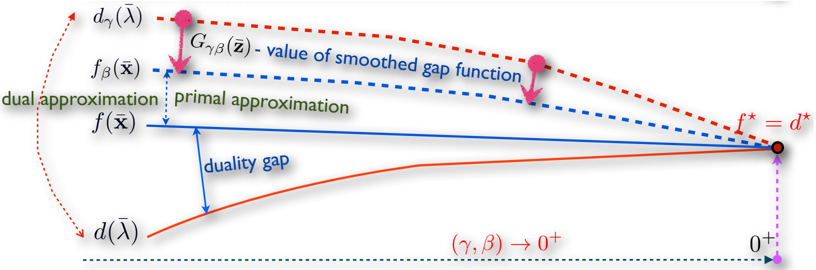
$$d_\gamma(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} \left\{f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma\xi(\mathbf{x}; \mathbf{x}_c)\right\}$$

Illustration of smooth gap function $G_{\gamma\beta}$

- The **objective function** $f(\mathbf{x})$ does not depend on β
- The **dual function** $d(\lambda)$ does not depend on γ:

$$d(\bar{\lambda}) := \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) + \bar{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}.$$

- $f(x)$ is decreasing over $\mathbf{x}$ and $d(\lambda)$ is increasing over λ.



- The smoothed duality gap is defined as $G_{\gamma\beta}(\bar{\mathbf{z}}) := f_\beta(\bar{\mathbf{x}}) - d_\gamma(\bar{\lambda}) \leq 0$.
- At the optimal solution $\mathbf{z}^\star := (\mathbf{x}^\star, \lambda^\star)$, one has $f(\mathbf{x}^\star) = d(\lambda^\star)$ and

$$G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to G(\mathbf{z}^\star) = 0 \text{ as } \gamma_k\beta_k \to 0^+.$$

Model-based excessive gap technique

What is the smoothed gap function used for?

Aim: To generate a primal-dual sequence $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ such that

$$G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to 0^+$$

by controlling γ_k and $\beta_k \to 0^+$.

- When γ_k and β_k go to zero, we have $G_{\gamma_k\beta_k}(\cdot) \to G(\cdot)$.
- **Consequence**: $\boxed{G(\mathbf{z}^k) \to 0^+ \quad \Rightarrow \quad \bar{\mathbf{z}}^k \to \mathbf{z}^\star = (\mathbf{x}^\star, \lambda^\star)}$ (primal-dual solution).

Model-based excessive gap condition

A **sequence** $\{\bar{\mathbf{z}}^k\}_{k\geq 0} \subset \mathcal{X} \times \mathbb{R}^n$ is said to satisfy the model-based excessive gap condition if

$$\boxed{G_{\gamma_{k+1}\beta_{k+1}}(\bar{\mathbf{z}}^{k+1}) \leq (1-\tau_k)G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) - \psi_k} \tag{45}$$

where $\psi_k \geq 0$, $\tau_k \in (0,1)$ and $\gamma_k\beta_{k+1} < \gamma_k\beta_k$ for $k \geq 0$.

Model-based excessive gap technique

What is the smoothed gap function used for?

Aim: To generate a primal-dual sequence $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ such that

$$G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to 0^+$$

by controlling γ_k and $\beta_k \to 0^+$.

- When γ_k and β_k go to zero, we have $G_{\gamma_k\beta_k}(\cdot) \to G(\cdot)$.
- **Consequence**: $\boxed{G(\mathbf{z}^k) \to 0^+ \quad \Rightarrow \quad \bar{\mathbf{z}}^k \to \mathbf{z}^\star = (\mathbf{x}^\star, \lambda^\star)}$ (primal-dual solution).

Model-based excessive gap condition

A **sequence** $\{\bar{\mathbf{z}}^k\}_{k\geq 0} \subset \mathcal{X} \times \mathbb{R}^n$ is said to satisfy the model-based excessive gap condition if

$$\boxed{G_{\gamma_{k+1}\beta_{k+1}}(\bar{\mathbf{z}}^{k+1}) \leq (1-\tau_k)G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) - \psi_k} \tag{45}$$

where $\psi_k \geq 0$, $\tau_k \in (0,1)$ and $\gamma_k\beta_{k+1} < \gamma_k\beta_k$ for $k \geq 0$.

Model-based excessive gap technique

What is the smoothed gap function used for?

Aim: To generate a primal-dual sequence $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ such that

$$G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to 0^+$$

by controlling γ_k and $\beta_k \to 0^+$.

- When γ_k and β_k go to zero, we have $G_{\gamma_k\beta_k}(\cdot) \to G(\cdot)$.
- **Consequence**: $\boxed{G(\mathbf{z}^k) \to 0^+ \quad \Rightarrow \quad \bar{\mathbf{z}}^k \to \mathbf{z}^\star = (\mathbf{x}^\star, \lambda^\star)}$ (primal-dual solution).

Model-based excessive gap condition

A **sequence** $\{\bar{\mathbf{z}}^k\}_{k\geq 0} \subset \mathcal{X} \times \mathbb{R}^n$ is said to satisfy the model-based excessive gap condition if

$$\boxed{G_{\gamma_{k+1}\beta_{k+1}}(\bar{\mathbf{z}}^{k+1}) \leq (1-\tau_k)G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) - \psi_k} \qquad (45)$$

where $\psi_k \geq 0$, $\tau_k \in (0,1)$ and $\gamma_k\beta_{k+1} < \gamma_k\beta_k$ for $k \geq 0$.

Let $\bar{G}_k := G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k)$. By induction, we have

$$\bar{G}_{k+1} \leq \prod_{j=0}^{k}(1-\tau_j)\bar{G}_0 - \left[\psi_0 + \sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)\psi_j\right].$$

$\Rightarrow$ The convergence rate of $\{\bar{G}_k\}$ depends on the convergence rate of $\{\tau_k\}$.

Key estimates

- For a bounded set $\mathcal{X}$ and $\hat{\mathbf{x}} \in \mathcal{X}$, the quality $D_{\mathcal{X}}$ defined below is finite

$$D_{\mathcal{X}} := \max_{\mathbf{x} \in \mathcal{X}} \xi(\mathbf{x}, \hat{\mathbf{x}}) < +\infty.$$

- Denote

$$\omega_k := \prod_{j=0}^{k} (1 - \tau_j) \quad \text{and} \quad \Psi_k := \psi_0 + \sum_{j=1}^{k-1} \prod_{l=0}^{j-1} (1 - \tau_l) \psi_j.$$

Key estimates

- For a bounded set $\mathcal{X}$ and $\hat{\mathbf{x}} \in \mathcal{X}$, the quality $D_{\mathcal{X}}$ defined below is finite

$$D_{\mathcal{X}} := \max_{\mathbf{x}\in\mathcal{X}} \xi(\mathbf{x}, \hat{\mathbf{x}}) < +\infty.$$

- Denote

$$\omega_k := \prod_{j=0}^{k}(1-\tau_j) \quad \text{and} \quad \Psi_k := \psi_0 + \sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)\psi_j.$$

Theorem (Bounds on the objective residual and primal feasibility)

Assume that $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ *is a sequence satisfying* (45). *Then*

$$\begin{cases} -\|\lambda^\star\|\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq f(\bar{\mathbf{x}}^k) - f^\star \leq C_k, \\ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq \beta_k\left[\|\lambda^\star\| + \sqrt{\|\lambda^\star\|^2 + 2\beta_k^{-1}C_k}\right], \end{cases} \tag{46}$$

where $C_k := \omega_{k-1}G_{\gamma_0\beta_0}(\bar{\mathbf{z}}^0) + \gamma_k D_{\mathcal{X}} - \Psi_{k-1}$, *provided that* $\beta_k\|\lambda^\star\| + 2C_k \geq 0$.

Key estimates

- For a bounded set $\mathcal{X}$ and $\hat{\mathbf{x}} \in \mathcal{X}$, the quality $D_{\mathcal{X}}$ defined below is finite

$$D_{\mathcal{X}} := \max_{\mathbf{x}\in\mathcal{X}} \xi(\mathbf{x}, \hat{\mathbf{x}}) < +\infty.$$

- Denote

$$\omega_k := \prod_{j=0}^{k}(1-\tau_j) \quad \text{and} \quad \Psi_k := \psi_0 + \sum_{j=1}^{k-1}\prod_{l=0}^{j-1}(1-\tau_l)\psi_j.$$

Theorem (Bounds on the objective residual and primal feasibility)

Assume that $\{\bar{\mathbf{z}}^k\}_{k\geq 0}$ *is a sequence satisfying* (45)*. Then*

$$\begin{cases} -\|\lambda^\star\|\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq f(\bar{\mathbf{x}}^k) - f^\star \leq C_k, \\ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq \beta_k\Big[\|\lambda^\star\| + \sqrt{\|\lambda^\star\|^2 + 2\beta_k^{-1}C_k}\Big], \end{cases} \tag{46}$$

where $C_k := \omega_{k-1}G_{\gamma_0\beta_0}(\bar{\mathbf{z}}^0) + \gamma_k D_{\mathcal{X}} - \Psi_{k-1}$*, provided that* $\beta_k\|\lambda^\star\| + 2C_k \geq 0$*.*

As a consequence, we have

$$\begin{cases} |f(\bar{\mathbf{x}}^k) - f^\star| \leq \max\Big\{\gamma_k D_{\mathcal{X}}, \Big(2\beta_k D_{\Lambda^\star} + \sqrt{2\gamma_k\beta_k D_{\mathcal{X}}}\Big)D_{\Lambda^\star}\Big\}, \\ \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq 2\beta_k D_{\Lambda^\star} + \sqrt{2\gamma_k\beta_k D_{\mathcal{X}}}, \end{cases} \tag{47}$$

where $D_{\Lambda^\star} := \min\{\|\lambda^\star\| \ : \ \lambda^\star \in \Lambda^\star\}$ the **norm** of the minimum norm solution of the **dual problem**.

Sketch of proof

- From the saddle point inequalities, we have $f^\star = \mathcal{L}(\mathbf{x}^\star, \lambda^\star) \leq \mathcal{L}(\mathbf{x}, \lambda^\star)$. Hence,

$$d(\lambda) \leq f^\star \leq f(\mathbf{x}) + (\mathbf{A}\mathbf{x} - \mathbf{b})^T\lambda^\star \leq f(\mathbf{x}) + \|\mathbf{A}\mathbf{x} - \mathbf{b}\|\|\lambda^\star\|, \ \forall \mathbf{x} \in \mathcal{X}.$$

  We finally get $-\|\lambda^\star\|\|\mathbf{A}\mathbf{x} - \mathbf{b}\| \leq f(\mathbf{x}) - f^\star \leq f(\mathbf{x}) - d(\lambda)$ for all $\mathbf{x} \in \mathcal{X}$.
- Since $\xi(\mathbf{x}, \mathbf{x}_c) \geq 0$ and $d(\lambda) = \min_{\mathbf{x} \in \mathcal{X}} \{f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}$, we have

$$d(\lambda) \leq d_\gamma(\lambda) \leq d(\lambda) + \gamma D_{\mathcal{X}}$$

- Putting things together, we get

$$\begin{aligned} -\|\lambda^\star\|\|\mathbf{A}\mathbf{x} - \mathbf{b}\| &\leq f(\mathbf{x}) - f^\star \leq f(\mathbf{x}) - d(\lambda) \\ &\leq f_\beta(\mathbf{x}) - d_\gamma(\lambda) + \gamma D_{\mathcal{X}} - (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \\ &= G_{\gamma\beta}(\mathbf{z}) + \gamma D_{\mathcal{X}} - (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2. \end{aligned} \quad (48)$$

- Since $G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \leq \omega_{k-1} G_{\gamma_0\beta_0}(\bar{\mathbf{z}}^0) - \Psi_{k-1}$ due to (45), we obtain from (48)

$$-\|\lambda^\star\|\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq f(\bar{\mathbf{x}}^k) - f^\star \leq \omega_{k-1} G_{\gamma_0\beta_0}(\bar{\mathbf{z}}^0) - \Psi_{k-1} + \gamma_k D_{\mathcal{X}} = C_k. \quad (49)$$

  which is the **first inequality of** (46).
- Let $s := \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|$. From (49) and (48) we have $s^2 - 2\beta_k\|\lambda^\star\|s - 2\beta_k C_k \leq 0$. Solving this in equation, we obtain the **second inequality of** (46).

Evaluating the smoothed gap function $G_{\gamma\beta}$

Evaluation of $G_{\gamma\beta}$

In order to evaluate $G_{\gamma\beta}$, we need to solve the maximization problem:

$$G_{\gamma\beta}(\mathbf{z}) := \max_{\hat{\mathbf{z}} \in \mathcal{X} \times \mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) - d_{\gamma\beta}(\hat{\mathbf{z}}) \right\}$$

The solution $\mathbf{z}^\star_{\gamma\beta}(\mathbf{z}) := (\mathbf{x}^\star_\gamma(\lambda), \lambda^\star_\beta(\mathbf{x}))$ of this problem is given as

$$\begin{cases} \mathbf{x}^\star_\gamma(\lambda) & := \arg\min_{\mathbf{x} \in \mathcal{X}} \left\{ f(\mathbf{x}) + (\mathbf{A}^T\lambda)^T\mathbf{x} + \gamma d(\mathbf{x}, \mathbf{x}_c) \right\} \\ \lambda^\star_\beta(\mathbf{x}) & := \beta^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b}). \end{cases} \tag{50}$$

Evaluating the smoothed gap function $G_{\gamma\beta}$

Evaluation of $G_{\gamma\beta}$

In order to evaluate $G_{\gamma\beta}$, we need to solve the maximization problem:

$$G_{\gamma\beta}(\mathbf{z}) := \max_{\hat{\mathbf{z}}\in\mathcal{X}\times\mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) - d_{\gamma\beta}(\hat{\mathbf{z}}) \right\}$$

The solution $\mathbf{z}^\star_{\gamma\beta}(\mathbf{z}) := (\mathbf{x}^\star_\gamma(\lambda), \lambda^\star_\beta(\mathbf{x}))$ of this problem is given as

$$\begin{cases} \mathbf{x}^\star_\gamma(\lambda) & := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + (\mathbf{A}^T\lambda)^T\mathbf{x} + \gamma d(\mathbf{x}, \mathbf{x}_c) \right\} \\ \lambda^\star_\beta(\mathbf{x}) & := \beta^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b}). \end{cases} \quad (50)$$

Given $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ and (γ_k, β_k). The **idea of the algorithms** is to:

- Update $\bar{\mathbf{z}}^{k+1} := (\bar{\mathbf{x}}^{k+1}, \bar{\lambda}^{k+1})$ from $\bar{\mathbf{z}}^k$ and $\mathbf{z}^\star_{\gamma_k\beta_k}(\mathbf{z})$.
- Decrease the parameters $(\gamma_{k+1}, \beta_{k+1})$ such that $\gamma_{k+1}\beta_{k+1} < \gamma_k\beta_k$.

Evaluating the smoothed gap function $G_{\gamma\beta}$

Evaluation of $G_{\gamma\beta}$

In order to evaluate $G_{\gamma\beta}$, we need to solve the maximization problem:

$$G_{\gamma\beta}(\mathbf{z}) := \max_{\hat{\mathbf{z}}\in\mathcal{X}\times\mathbb{R}^n} \left\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z}-\hat{\mathbf{z}}) - d_{\gamma\beta}(\hat{\mathbf{z}}) \right\}$$

The solution $\mathbf{z}^\star_{\gamma\beta}(\mathbf{z}) := (\mathbf{x}^\star_\gamma(\lambda), \lambda^\star_\beta(\mathbf{x}))$ of this problem is given as

$$\begin{cases} \mathbf{x}^\star_\gamma(\lambda) & := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + (\mathbf{A}^T\lambda)^T\mathbf{x} + \gamma d(\mathbf{x},\mathbf{x}_c) \right\} \\ \lambda^\star_\beta(\mathbf{x}) & := \beta^{-1}(\mathbf{A}\mathbf{x}-\mathbf{b}). \end{cases} \tag{50}$$

Given $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ and (γ_k, β_k). The **idea of the algorithms** is to:

- Update $\bar{\mathbf{z}}^{k+1} := (\bar{\mathbf{x}}^{k+1}, \bar{\lambda}^{k+1})$ from $\bar{\mathbf{z}}^k$ and $\mathbf{z}^\star_{\gamma_k\beta_k}(\mathbf{z})$.
- Decrease the parameters $(\gamma_{k+1}, \beta_{k+1})$ such that $\gamma_{k+1}\beta_{k+1} < \gamma_k\beta_k$.

Proximal-gradient step

In our algorithms, we need to compute $\bar{\mathbf{x}}^{k+1}$ using the following mapping:

$$\text{prox}_{\beta f}(\mathbf{x},\lambda) := \arg\min_{\hat{\mathbf{x}}\in\mathcal{X}} \left\{ f(\hat{\mathbf{x}}) + (\mathbf{A}^T\lambda)^T\hat{\mathbf{x}} + (\|\mathbf{A}\|^2/(2\beta))\|\hat{\mathbf{x}}-\mathbf{x}\|_2^2 \right\} \tag{51}$$

How can we generate $\{\bar{\mathbf{z}}^k\}$?

The main idea of generating the **sequence** $\{\bar{\mathbf{z}}^k\}$ such that $\{G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k)\}$ decreases come from the following **observations**:

- Since $G_{\gamma\beta}(\mathbf{z}) = f_\beta(\mathbf{x}) - d_\gamma(\lambda)$, then $G_{\gamma\beta}(\mathbf{z})$ decreases if at least $f_\beta(\mathbf{x})$ decrease or $d_\gamma(\lambda)$ increases.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

The main idea of generating the **sequence** $\{\bar{\mathbf{z}}^k\}$ such that $\{G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k)\}$ decreases come from the following **observations**:

- Since $G_{\gamma\beta}(\mathbf{z}) = f_\beta(\mathbf{x}) - d_\gamma(\lambda)$, then $G_{\gamma\beta}(\mathbf{z})$ decreases if at least $f_\beta(\mathbf{x})$ decrease or $d_\gamma(\lambda)$ increases.
- We note on the one hand that:
  - d_γ is a smoothed version of the **dual function** d.
  - this function is concave and has Lipschitz gradient ∇d_γ.

  In order to increase d_γ as much as possible, one can implement a Nesterov's accelerated gradient ascent scheme.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

The main idea of generating the **sequence** $\{\bar{\mathbf{z}}^k\}$ such that $\{G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k)\}$ decreases come from the following **observations**:

- Since $G_{\gamma\beta}(\mathbf{z}) = f_\beta(\mathbf{x}) - d_\gamma(\lambda)$, then $G_{\gamma\beta}(\mathbf{z})$ decreases if at least $f_\beta(\mathbf{x})$ decrease or $d_\gamma(\lambda)$ increases.
- We note on the one hand that:
  - d_γ is a smoothed version of the **dual function** d.
  - this function is concave and has Lipschitz gradient ∇d_γ.

  In order to increase d_γ as much as possible, one can implement a Nesterov's accelerated gradient ascent scheme.
- On the other hand, $f_\beta(\mathbf{x}) = f(\mathbf{x}) + (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$:
  - f_β is the sum of a convex function f and $g(\mathbf{x}) := (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$.
  - g is convex and has Lipschitz gradient, while f is just convex.

  In order to decrease f_β, we can implement a Nesterov's accelerated proximal-gradient scheme.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

The main idea of generating the **sequence** $\{\bar{\mathbf{z}}^k\}$ such that $\{G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k)\}$ decreases come from the following **observations**:

- Since $G_{\gamma\beta}(\mathbf{z}) = f_\beta(\mathbf{x}) - d_\gamma(\lambda)$, then $G_{\gamma\beta}(\mathbf{z})$ decreases if at least $f_\beta(\mathbf{x})$ decrease or $d_\gamma(\lambda)$ increases.
- We note on the one hand that:
  - d_γ is a smoothed version of the **dual function** d.
  - this function is concave and has Lipschitz gradient ∇d_γ.

  In order to increase d_γ as much as possible, one can implement a Nesterov's accelerated gradient ascent scheme.
- On the other hand, $f_\beta(\mathbf{x}) = f(\mathbf{x}) + (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$:
  - f_β is the sum of a convex function f and $g(\mathbf{x}) := (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$.
  - g is convex and has Lipschitz gradient, while f is just convex.

  In order to decrease f_β, we can implement a Nesterov's accelerated proximal-gradient scheme.
- Every iteration, we can perform one scheme or both in order to decrease $G_{\gamma\beta}(\mathbf{z})$, while simultaneously decrease the **product** $\gamma\beta$.

This observations lead to the following two update schemes in the next slides.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

Expression of the idea

We denote by $\mathbf{s}$ a variable standing for either $\mathbf{x}$ or λ. The **main step** of the algorithmic scheme consists of

- For given $\mathbf{s}$, $\mathbf{s}^\star$ and $\tau \in (0, 1]$ at the current iteration. We perform one interpolation step:

$$\hat{\mathbf{s}} := (1 - \tau)\mathbf{s} + \tau\mathbf{s}^\star.$$

- When $\hat{\mathbf{s}}$ is available, we perform a proximal-gradient or gradient step to compute the next iteration $\mathbf{s}^+$:

$$\mathbf{s}^+ := \mathrm{prox}_\psi(\hat{\mathbf{s}} - (1/L)\nabla\varphi(\hat{\mathbf{s}})).$$

  where
  - $\varphi := d_\gamma$ the smoothed dual function and $\psi = f$ in the primal step
  - $\varphi := (1/(2\beta))\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ and $\psi = 0$ and in the dual scheme.
  - L is the Lipschitz constant of $\nabla\varphi$.

Remarks

- The above scheme looks very similar to FISTA [1] in the context of Nesterov's accelerating method.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

We propose two schemes to generate the sequence $\{\bar{\mathbf{z}}^k\}$

- The **primal-dual** scheme with two primal steps and one dual step (2P1D);
- The **primal-dual** scheme with one primal step and two dual steps (1P2D).

How can we generate $\{\bar{\mathbf{z}}^k\}$?

We propose two schemes to generate the sequence $\{\bar{\mathbf{z}}^k\}$

- The **primal-dual** scheme with two primal steps and one dual step (2P1D);
- The **primal-dual** scheme with one primal step and two dual steps (1P2D).

Primal-dual schemes

- The (2P1D) scheme generates $\{\bar{z}^k\}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ as follows:

$$\begin{cases} \hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\bar{\lambda}^k) \\ \bar{\mathbf{x}}^{k+1} & := \mathrm{prox}_{\beta_{k+1} f}(\hat{\mathbf{x}}^k, \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k)) \\ \bar{\lambda}^{k+1} & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k) \end{cases} \qquad \text{(2P1D)}$$

How can we generate $\{\bar{\mathbf{z}}^k\}$?

We propose two schemes to generate the sequence $\{\bar{\mathbf{z}}^k\}$

- The **primal-dual** scheme with two primal steps and one dual step (2P1D);
- The **primal-dual** scheme with one primal step and two dual steps (1P2D).

Primal-dual schemes

- The (2P1D) scheme generates $\{\bar{z}^k\}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ as follows:

$$\begin{cases} \hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\bar{\lambda}^k) \\ \bar{\mathbf{x}}^{k+1} & := \mathrm{prox}_{\beta_{k+1} f}(\hat{\mathbf{x}}^k, \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k)) \\ \bar{\lambda}^{k+1} & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k) \end{cases} \tag{2P1D}$$

- Symmetrically, the (1P2D) scheme generates $\{\bar{z}^k\}$ as follows:

$$\begin{cases} \hat{\lambda}^k & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_k}(\bar{\mathbf{x}}^k) \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k) \\ \bar{\lambda}^{k+1} & := \hat{\lambda}^k + \alpha_k(\mathbf{A}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k) - \mathbf{b}) \end{cases} \tag{1P2D}$$

where $\alpha_k := \gamma_{k+1}\|\mathbf{A}\|^{-2}$.

How can we generate $\{\bar{\mathbf{z}}^k\}$?

We propose two schemes to generate the sequence $\{\bar{\mathbf{z}}^k\}$

- The **primal-dual** scheme with two primal steps and one dual step (2P1D);
- The **primal-dual** scheme with one primal step and two dual steps (1P2D).

Primal-dual schemes

- The (2P1D) scheme generates $\{\bar{z}^k\}$ with $\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)$ as follows:

$$\begin{cases} \hat{\mathbf{x}}^k & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_k}(\bar{\lambda}^k) \\ \bar{\mathbf{x}}^{k+1} & := \mathrm{prox}_{\beta_{k+1} f}(\hat{\mathbf{x}}^k, \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k)) \\ \bar{\lambda}^{k+1} & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k) \end{cases} \qquad \text{(2P1D)}$$

- Symmetrically, the (1P2D) scheme generates $\{\bar{z}^k\}$ as follows:

$$\begin{cases} \hat{\lambda}^k & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_k}(\bar{\mathbf{x}}^k) \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k) \\ \bar{\lambda}^{k+1} & := \hat{\lambda}^k + \alpha_k(\mathbf{A}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k) - \mathbf{b}) \end{cases} \qquad \text{(1P2D)}$$

  where $\alpha_k := \gamma_{k+1}\|\mathbf{A}\|^{-2}$.
- The parameters β_k and γ_k are updated as ($c_k \in (-1, 1]$ given):

$$\gamma_{k+1} := (1 - c_k\tau_k)\gamma_k \quad \text{and} \quad \beta_{k+1} = (1-\tau_k)\beta_k \qquad (52)$$

Remarks on the computational complexity of both schemes

- (2P1D) requires two primal steps: **one** to compute $\mathbf{x}^\star_{\gamma_k}(\bar{\lambda}^k)$ and **one** to compute $\bar{\mathbf{x}}^{k+1}$.
  - The first step corresponds to solving
  $$\mathbf{x}^\star_{\gamma_k}(\bar{\lambda}^k) := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + (\mathbf{A}^T\bar{\lambda}^k)^T\mathbf{x} + \gamma_k d(\mathbf{x}, \mathbf{x}_c) \right\}.$$
  - The second step corresponds to solving
  $$\bar{\mathbf{x}}^{k+1} := \arg\min_{\hat{\mathbf{x}}\in\mathcal{X}} \left\{ f(\hat{\mathbf{x}}) + (\mathbf{A}^T\lambda^\star_{\beta_{k+1}}(\hat{\mathbf{x}}^k))^T\hat{\mathbf{x}} + (\|\mathbf{A}\|^2/(2\beta))\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^k\|_2^2 \right\}$$
  - If b is a quadratic prox-function and $\mathcal{X}$ is absent, then solving both problems corresponds to computing the proximal operator of f.
- (1P2D) only requires one primal step to compute $\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k)$.
- (1P2D) requires two dual steps corresponding to two matrix-vector multiplications $\mathbf{A}\bar{\mathbf{x}}^k$ and $\mathbf{A}\mathbf{x}^\star_{\gamma_{k+1}}(\hat{\lambda}^k)$.
- (2P1D) requires only one $\mathbf{A}\hat{\mathbf{x}}^k$.

Updating step-size

The key point in both schemes (1P2D) and (2P1D) is to update the step size τ_k:

- The model-based excessive gap condition (45) shows that $G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to 0^+$.
- The convergence rate of $\{\beta_k\}$ and $\{\gamma_k\}$ depends on the convergence rate of $\{\tau_k\}$ and hence, the convergence rate of the algorithms.

Updating step-size

The key point in both schemes (1P2D) and (2P1D) is to update the step size τ_k:

- The model-based excessive gap condition (45) shows that $G_{\gamma_k \beta_k}(\bar{\mathbf{z}}^k) \to 0^+$.
- The convergence rate of $\{\beta_k\}$ and $\{\gamma_k\}$ depends on the convergence rate of $\{\tau_k\}$ and hence, the convergence rate of the algorithms.

Theorem (Key condition)

Let $\{\bar{\mathbf{z}}^k\}$ be the sequence generated by either (1P2D) or (2P1D) and $G_{\gamma_k \beta_k}(\bar{\mathbf{z}}^k) \leq 0$. Then, under the condition:

$$\tau^2 \|\mathbf{A}\|^2 \leq \gamma_{k+1}\beta_{k+1} \tag{53}$$

we have $G_{\gamma_{k+1}\beta_{k+1}}(\bar{\mathbf{z}}^{k+1}) \leq 0$.

Condition (53) and the update rules $\gamma_{k+1} := (1 - c_k\tau_k)\gamma_k$ and $\beta_{k+1} = (1 - \tau_k)\beta_k$ allow us to derive the update rule for τ_k:

Updating step-size

The key point in both schemes (1P2D) and (2P1D) is to update the step size τ_k:

- The model-based excessive gap condition (45) shows that $G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \to 0^+$.
- The convergence rate of $\{\beta_k\}$ and $\{\gamma_k\}$ depends on the convergence rate of $\{\tau_k\}$ and hence, the convergence rate of the algorithms.

Theorem (Key condition)

Let $\{\bar{\mathbf{z}}^k\}$ be the sequence generated by either (1P2D) or (2P1D) and $G_{\gamma_k\beta_k}(\bar{\mathbf{z}}^k) \leq 0$. Then, under the condition:

$$\tau^2\|\mathbf{A}\|^2 \leq \gamma_{k+1}\beta_{k+1} \tag{53}$$

we have $G_{\gamma_{k+1}\beta_{k+1}}(\bar{\mathbf{z}}^{k+1}) \leq 0$.

Condition (53) and the update rules $\gamma_{k+1} := (1 - c_k\tau_k)\gamma_k$ and $\beta_{k+1} = (1 - \tau_k)\beta_k$ allow us to derive the update rule for τ_k:

Update the step-size τ_k

- **Initialization:** $\tau_0 := a_0^{-1}$, where $a_0 := \left(1 + c_0 + [4(1 - c_0) + (1 + c_0)^2]^{1/2}\right)/2$.
- **Update:** τ_{k+1} is updated from τ_k as

$$\tau_k = a_k^{-1}, \quad a_{k+1} := \left(1 + c_{k+1} + \sqrt{4a_k^2 + (1 - c_{k+1})^2}\right)/2$$

Primal-dual framework using model-based excessive gap technique

Putting all ingredients together, we can describe the complete algorithm as below:

Primal-dual method using model-based excessive gap technique (PDM)
Initialization
1.1. Given $\gamma_0 > 0$, $c_0 \in (-1, 1]$ and $\bar{L}_d := \|\mathbf{A}\|^2$.
1.2. $a_0 := (1 + c_0 + \sqrt{4(1 - c_0) + (1 + c_0)^2})/2$, $\tau_0 := a_0^{-1}$ and $\beta_0 := \bar{L}_g \gamma^{-1}$.
1.3. Compute a starting point $\bar{\mathbf{z}}^0 := (\bar{\mathbf{x}}^0, \bar{\lambda}^0)$.

Primal-dual framework using model-based excessive gap technique

Putting all ingredients together, we can describe the complete algorithm as below:

Primal-dual method using model-based excessive gap technique (PDM)

Initialization

1.1. Given $\gamma_0 > 0$, $c_0 \in (-1, 1]$ and $\bar{L}_d := \|\mathbf{A}\|^2$.

1.2. $a_0 := (1 + c_0 + \sqrt{4(1 - c_0) + (1 + c_0)^2})/2$, $\tau_0 := a_0^{-1}$ and $\beta_0 := \bar{L}_g \gamma^{-1}$.

1.3. Compute a starting point $\bar{\mathbf{z}}^0 := (\bar{\mathbf{x}}^0, \bar{\lambda}^0)$.

Iterations: **For $k = 0, 1, \cdots, K$, perform:**

2.1. Given $(\bar{\mathbf{x}}^k, \bar{\lambda}^k)$, compute $(\bar{\mathbf{x}}^{k+1}, \bar{\lambda}^{k+1})$ by either (2P1D) or (1P2D).

2.2. Update $\gamma_{k+1} := (1 - c_k \tau_k)\gamma_k$ and $\beta_{k+1} := (1 - \tau_k)\beta_k$

2.3. Update c_{k+1} from c_k if necessary (optional).

2.4. Update $a_{k+1} := \left(1 + c_{k+1} + \sqrt{4a_k^2 + (1 - c_{k+1})^2}\right)/2$ and $\tau_{k+1} = a_{k+1}^{-1}$.

Convergence guarantee

Theorem (Convergence)

Let $\{\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)\}$ *be the sequence generated by* ***PDM*** *after* $k \geq 1$ *iterations.*

a) *If* (2P1D) *is used and* $\gamma_0 := \|\mathbf{A}\|$ *and* $c_k = 1$ *for all* $k \geq 0$*, then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{\|\mathbf{A}\|(2D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}})}{k+1}, \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^k) - f^\star & \leq \frac{2\|\mathbf{A}\|D_{\mathcal{X}}}{k+1}. \end{cases}$$

Convergence guarantee

Theorem (Convergence)

Let $\{\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)\}$ *be the sequence generated by* ***PDM*** *after* $k \geq 1$ *iterations.*

a) *If* (2P1D) *is used and* $\gamma_0 := \|\mathbf{A}\|$ *and* $c_k = 1$ *for all* $k \geq 0$*, then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{\|\mathbf{A}\|(2D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}})}{k+1}, \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^k) - f^\star & \leq \frac{2\|\mathbf{A}\|D_{\mathcal{X}}}{k+1}. \end{cases}$$

b) *If* (1P2D) *is used and* $\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+1}$ *and* $c_k = 0$ *for all* $0 \leq k \leq K$*, then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq \frac{2\sqrt{2}\|\mathbf{A}\|(D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}})}{K+1}, \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^K) - f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|D_{\mathcal{X}}}{K+1}. \end{cases}$$

The worst-case complexity of ***PDM*** *to reach an* ϵ*-solution* $\mathbf{x}^\star$ *of* (40) *is* $\mathcal{O}\left(\frac{\|\mathbf{A}\|R}{\epsilon}\right)$*, where* $R := \max\{D_{\mathcal{X}}, D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}}\}$.

Convergence guarantee

Theorem (Convergence)

Let $\{\bar{\mathbf{z}}^k := (\bar{\mathbf{x}}^k, \bar{\lambda}^k)\}$ *be the sequence generated by* ***PDM*** *after* $k \geq 1$ *iterations.*

a) *If (2P1D) is used and* $\gamma_0 := \|\mathbf{A}\|$ *and* $c_k = 1$ *for all* $k \geq 0$*, then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{\|\mathbf{A}\|(2D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}})}{k+1}, \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^k) - f^\star & \leq \frac{2\|\mathbf{A}\|D_{\mathcal{X}}}{k+1}. \end{cases}$$

b) *If (1P2D) is used and* $\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+1}$ *and* $c_k = 0$ *for all* $0 \leq k \leq K$*, then*

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq \frac{2\sqrt{2}\|\mathbf{A}\|(D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}})}{K+1}, \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^K) - f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|D_{\mathcal{X}}}{K+1}. \end{cases}$$

The worst-case complexity of ***PDM*** *to reach an* ϵ*-solution* $\mathbf{x}^\star$ *of* (40) *is* $\mathcal{O}\left(\frac{\|\mathbf{A}\|R}{\epsilon}\right)$*, where* $R := \max\{D_{\mathcal{X}}, D_{\Lambda^\star} + \sqrt{D_{\mathcal{X}}}\}$.

Remarks:

- If (1P2D) is used, then $\gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+1}$, which requires to fix the number of iterations priori

Strongly convex case

PDM can be accelerated from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$ if f is strongly convex.

Assumption A.2.

The objective function f is strongly convex with the convexity parameter $\mu_f > 0$.

Strongly convex case

PDM can be accelerated from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$ if f is strongly convex.

Assumption A.2.

The objective function f is strongly convex with the convexity parameter $\mu_f > 0$.

Dual function

We define the **dual function** of (40) as

$$d(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} \Big\{ f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \Big\}. \tag{54}$$

- Let $\mathbf{x}^\star(\lambda)$ be the solution of (54)
- $\mathbf{x}^\star(\lambda)$ exists and is unique.

Strongly convex case

PDM can be accelerated from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$ if f is strongly convex.

Assumption A.2.

The objective function f is strongly convex with the convexity parameter $\mu_f > 0$.

Dual function

We define the **dual function** of (40) as

$$d(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} \left\{ f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \right\}. \tag{54}$$

- Let $\mathbf{x}^\star(\lambda)$ be the solution of (54)
- $\mathbf{x}^\star(\lambda)$ exists and is unique.

Properties of d

- d is concave and smooth.
- Gradient of d is given by $\nabla d(\lambda) := \mathbf{A}\mathbf{x}^\star(\lambda) - \mathbf{b}$.
- ∇d is Lipschitz continuous with a Lipschitz constant $\bar{L}_d := \frac{\|\mathbf{A}\|^2}{\mu_f}$.

Algorithm

When specifying **PDM** to solve the strongly convex case, some steps in the algorithm are changed:

- Only one smoothness parameter β_k is updated.
- The update rule of τ_k is simplified.

Algorithm

When specifying **PDM** to solve the strongly convex case, some steps in the algorithm are changed:

- Only one smoothness parameter β_k is updated.
- The update rule of τ_k is simplified.

Primal-dual method for strongly convex case (PDM$_\mu$)

Initialization:

1.1. Compute $\bar{L}_d := \mu_f^{-1}\|\mathbf{A}\|^2$, $\tau_0 := (\sqrt{5}-1)/2$ and $\beta_0 := \sqrt{\bar{L}_d}$.

1.2. Compute a starting point $\bar{\mathbf{z}}^0 := (\bar{\mathbf{x}}^0, \bar{\lambda}^0)$ as:

$$\bar{\mathbf{x}}^0 := \mathbf{x}^\star(0^n) \quad \text{and} \quad \bar{\lambda}^0 := \beta_0^{-1}(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}).$$

Algorithm

When specifying **PDM** to solve the strongly convex case, some steps in the algorithm are changed:

- Only one smoothness parameter β_k is updated.
- The update rule of τ_k is simplified.

Primal-dual method for strongly convex case (PDM$_\mu$)
Initialization:
1.1. Compute $\bar{L}_d := \mu_f^{-1}\|\mathbf{A}\|^2$, $\tau_0 := (\sqrt{5}-1)/2$ and $\beta_0 := \sqrt{\bar{L}_d}$.
1.2. Compute a starting point $\bar{\mathbf{z}}^0 := (\bar{\mathbf{x}}^0, \bar{\lambda}^0)$ as: $$\bar{\mathbf{x}}^0 := \mathbf{x}^\star(0^n) \;\text{ and }\; \bar{\lambda}^0 := \beta_0^{-1}(\mathbf{A}\bar{\mathbf{x}}^0 - \mathbf{b}).$$
Iterations: **For** $k = 0, 1, \cdots, K$, **perform:**
2.1. Given $(\bar{\mathbf{x}}^k, \bar{\lambda}^k)$, compute $(\bar{\mathbf{x}}^{k+1}, \bar{\lambda}^{k+1})$ as $$\begin{cases} \hat{\lambda}^k & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_k}(\bar{\mathbf{x}}^k) \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \mathbf{x}^\star(\hat{\lambda}^k) \\ \bar{\lambda}^{k+1} & := \hat{\lambda}^k + \bar{L}_d^{-1}(\mathbf{A}\mathbf{x}^\star(\hat{\lambda}^k) - \mathbf{b}). \end{cases} \quad (1\text{P}2\text{D}_\mu)$$
2.2. Update $\beta_{k+1} := (1-\tau_k)\beta_k$ and $\tau_{k+1} := \tau_k(\sqrt{\tau_k^2+4} - \tau_k)/2$.

Convergence guarantee

Theorem (Convergence guarantee)

Assumptions:

- *f is strongly convex with a strong convexity parameter $\mu_f > 0$.*
- *$\{\bar{\mathbf{z}}^k\}$ is generated by PDM$_\mu$.*

Conclusions:

- *We have estimates:*

$$\begin{cases} -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^k) - f^\star \quad \leq 0 \\ & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \quad \leq \frac{\|\mathbf{A}\|^2}{(k+2)^2\mu_f} D_{\Lambda^\star} \\ & \|\bar{\mathbf{x}}^k - \mathbf{x}^\star\| \quad \leq \frac{\|\mathbf{A}\|}{(k+2)^2\mu_f} D_{\Lambda^\star} \end{cases}$$

- *The bounds do not depend on $\mathcal{D}_{\mathcal{X}}$ the prox-diameter of $\mathcal{X}$.*
- *$\{\mathbf{x}^k\}$ converges to the unique solution $\mathbf{x}^\star$ of (40) at $\mathcal{O}(1/k^2)$ rate.*

Convergence guarantee

Theorem (Convergence guarantee)

Assumptions:

- *f is strongly convex with a strong convexity parameter $\mu_f > 0$.*
- *$\{\bar{\mathbf{z}}^k\}$ is generated by PDM$_\mu$.*

Conclusions:

- *We have estimates:*

$$\begin{cases} -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^k) - f^\star & \leq 0 \\ & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{\|\mathbf{A}\|^2}{(k+2)^2\mu_f} D_{\Lambda^\star} \\ & \|\bar{\mathbf{x}}^k - \mathbf{x}^\star\| & \leq \frac{\|\mathbf{A}\|}{(k+2)^2\mu_f} D_{\Lambda^\star} \end{cases}$$

- *The bounds do not depend on $\mathcal{D}_{\mathcal{X}}$ the prox-diameter of $\mathcal{X}$.*
- *$\{\mathbf{x}^k\}$ converges to the unique solution $\mathbf{x}^\star$ of (40) at $\mathcal{O}(1/k^2)$ rate.*

Remarks:

- We always have $f(\bar{\mathbf{x}}^k) \leq f^\star$ in PDM$_\mu$, which is different from the unconstrained case, i.e. $f(\mathbf{x}^k) \geq f^\star$.
- The convergence rate is **optimal** in the sense of black-box first order methods.

ADMM variant

ADMM was originally developed to solve a special case of (40):

$$f^\star := \min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\},$$

where f and $\mathcal{X}$ is decomposable with $g = 2$.

Problem setting: When f and $\mathcal{X}$ are 2-decomposable

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}:=(\mathbf{x}_1,\mathbf{x}_2)} & \Big\{ f(\mathbf{x}) := f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \Big\}, \\ \text{s.t.} & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = b, \\ & \mathbf{x}_1 \in \mathcal{X}_1, \ \mathbf{x}_2 \in \mathcal{X}_2. \end{cases} \quad (55)$$

ADMM variant

ADMM was originally developed to solve a special case of (40):

$$f^\star := \min_{\mathbf{x}\in\mathcal{X}} \{ f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} = \mathbf{b} \},$$

where f and $\mathcal{X}$ is decomposable with $g = 2$.

Problem setting: When f and $\mathcal{X}$ are 2-decomposable

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}:=(\mathbf{x}_1,\mathbf{x}_2)} & \left\{ f(\mathbf{x}) := f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \right\}, \\ \text{s.t.} & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = b, \\ & \mathbf{x}_1 \in \mathcal{X}_1, \ \mathbf{x}_2 \in \mathcal{X}_2. \end{cases} \tag{55}$$

Augmented Lagrangian smoother

- When $\tilde{\mathbf{x}}_1^k$ and $\tilde{\mathbf{x}}_2^{k+1}$ are available, we use

$$d_{\gamma\beta}(\mathbf{z}) := \frac{\gamma}{2}\left[\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\tilde{\mathbf{x}}_2^k - \mathbf{b}\|^2 + \|\mathbf{A}_1\tilde{\mathbf{x}}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|^2 \right] + \frac{\beta}{2}\|\lambda\|^2.$$

  to smooth the gap function $G(\mathbf{z}) := \max_{\hat{\mathbf{z}}\in\mathcal{X}\times\mathbb{R}^n}\{ f(\mathbf{x}) - f(\hat{\mathbf{x}}) + M(\mathbf{z})^T(\mathbf{z} - \hat{\mathbf{z}}) \}$.
- Modify the (1P2D) scheme to obtain a new variant of ADMM by alternating the computation of $\mathbf{x}^\star_\gamma(\lambda)$.

ADMM variant

By alternating the step of $\mathbf{x}^\star_\gamma(\lambda)$, the main step of the **new ADMM variant** becomes:

New ADMM scheme (ADMM$_1$)

$$\begin{cases} \hat{\lambda}^k & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_k}(\bar{\mathbf{x}}^k) \\ \tilde{\mathbf{x}}_1^{k+1} & := \operatorname*{argmin}_{\mathbf{x}_1 \in \mathcal{X}_1} \left\{ f_1(\mathbf{x}_1) + (\mathbf{A}_1^T\hat{\lambda}^k)^T\mathbf{x}_1 + (\gamma_k/2)\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\tilde{\mathbf{x}}_2^k - \mathbf{b}\|^2 \right\} \\ \tilde{\mathbf{x}}_2^{k+1} & := \operatorname*{argmin}_{\mathbf{x}_2 \in \mathcal{X}_2} \left\{ f_2(\mathbf{x}_2) + (\mathbf{A}_2^T\hat{\lambda}^k)^T\mathbf{x}_2 + (\gamma_k/2)\|\mathbf{A}_1\tilde{\mathbf{x}}_1^{k+1} + \mathbf{A}_2\bar{\mathbf{x}}_2 - \mathbf{b}\|^2 \right\} \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k\tilde{\mathbf{x}}^{k+1}, \quad \text{where} \quad \tilde{\mathbf{x}}^{k+1} := (\tilde{\mathbf{x}}_1^{k+1}, \tilde{\mathbf{x}}_2^{k+1}) \\ \bar{\lambda}^{k+1} & := \hat{\lambda}^k + (\gamma_k/2)(\mathbf{A}\tilde{\mathbf{x}}^{k+1} - \mathbf{b}). \end{cases}$$

ADMM variant

By alternating the step of $\mathbf{x}^\star_\gamma(\lambda)$, the main step of the **new ADMM variant** becomes:

New ADMM scheme (ADMM_1)

$$\begin{cases} \hat{\lambda}^k & := (1-\tau_k)\bar{\lambda}^k + \tau_k \lambda^\star_{\beta_k}(\bar{\mathbf{x}}^k) \\ \tilde{\mathbf{x}}_1^{k+1} & := \underset{\mathbf{x}_1 \in \mathcal{X}_1}{\mathrm{argmin}} \left\{ f_1(\mathbf{x}_1) + (\mathbf{A}_1^T \hat{\lambda}^k)^T \mathbf{x}_1 + (\gamma_k/2)\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2 \tilde{\mathbf{x}}_2^k - \mathbf{b}\|^2 \right\} \\ \tilde{\mathbf{x}}_2^{k+1} & := \underset{\mathbf{x}_2 \in \mathcal{X}_2}{\mathrm{argmin}} \left\{ f_2(\mathbf{x}_2) + (\mathbf{A}_2^T \hat{\lambda}^k)^T \mathbf{x}_2 + (\gamma_k/2)\|\mathbf{A}_1\tilde{\mathbf{x}}_1^{k+1} + \mathbf{A}_2 \bar{\mathbf{x}}_2 - \mathbf{b}\|^2 \right\} \\ \bar{\mathbf{x}}^{k+1} & := (1-\tau_k)\bar{\mathbf{x}}^k + \tau_k \tilde{\mathbf{x}}^{k+1}, \quad \text{where } \tilde{\mathbf{x}}^{k+1} := (\tilde{\mathbf{x}}_1^{k+1}, \tilde{\mathbf{x}}_2^{k+1}) \\ \bar{\lambda}^{k+1} & := \hat{\lambda}^k + (\gamma_k/2)(\mathbf{A}\tilde{\mathbf{x}}^{k+1} - \mathbf{b}). \end{cases}$$

Convergence of ADMM_1 [14]

Assumptions:

- Let $\{(\mathbf{x}^k, \lambda^k)\}_{k\geq 0}$ be the sequence generated by **PDM** using ADMM_1.
- Let $\gamma_k = \gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+3}$ and $\beta_{k+1} := (1-\tau_k)\beta_k$ for $k = 0, \cdots, K$.

Conclusion:

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq \frac{2\sqrt{2}\|\mathbf{A}\|(D_{\Lambda^\star} + \bar{D}_{\mathcal{X}})}{K+3} \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^K) - f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|}{K+3}(\bar{D}_{\mathcal{X}})^2 \end{cases}$$

where $\bar{D}_{\mathcal{X}} := 2 \max_{\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}} \|\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}})\|$.

Preconditioned ADMM variant

When f_1 and f_2 are proximally tractable and $\mathcal{X}_1$ and $\mathcal{X}_2$ are absent

- We can linearize the quadratic terms in lines 2 and 3 of ADMM_1.
- Then, by using the gradient step, we obtain a preconditioned ADMM variant.

Preconditioned ADMM variant

When f_1 and f_2 are proximally tractable and $\mathcal{X}_1$ and $\mathcal{X}_2$ are absent

- We can linearize the quadratic terms in lines 2 and 3 of ADMM$_1$.
- Then, by using the gradient step, we obtain a preconditioned ADMM variant.

Preconditioned ADMM variant (PADMM$_1$)

$$\begin{cases} \tilde{\mathbf{x}}_1^{k+1} & := \mathrm{prox}_{\gamma_k^{-1}\alpha_{1k}f_1}\left(\mathbf{g}_1^k + \gamma_k^{-1}\mathbf{A}_1^T\lambda^k\right) \\ \tilde{\mathbf{x}}_2^{k+1} & := \mathrm{prox}_{\gamma_k^{-1}\alpha_{2k}f_2}\left(\mathbf{g}_2^k + \gamma_k^{-1}\mathbf{A}_2^T\lambda^k\right) \end{cases}$$

where g_1^k and g_2^k are the gradient step of the quadratic term computed as

$$\begin{cases} \mathbf{g}_1^k & := \tilde{\mathbf{x}}_1^k - \alpha_{1k}\mathbf{A}_1^T(\mathbf{A}_1\tilde{\mathbf{x}}_1^k + \mathbf{A}_2\tilde{\mathbf{x}}_2^k - \mathbf{b}) \\ \mathbf{g}_2^k & := \tilde{\mathbf{x}}_2^k - \alpha_{2k}\mathbf{A}_2^T(\mathbf{A}_1\tilde{\mathbf{x}}_1^{k+1} + \mathbf{A}_2\tilde{\mathbf{x}}_2^k - \mathbf{b}). \end{cases}$$

Here α_{1k} and α_{2k} are given step-sizes.

There are at least two ways of computing the step-sizes:

- **Constant step size**: We can take $\alpha_{1k} := \|\mathbf{A}_1\|^{-1}$ and $\alpha_{2k} := \|\mathbf{A}_2\|^{-1}$.
- **Adaptive step-size**: α_{1k} and α_{2k} are computed from the exact line-search condition of the form:

$$\alpha := \arg\min_{\alpha > 0} \xi(\mathbf{u}^k - \alpha\nabla\xi(\mathbf{u}^k))$$

where $\mathbf{u}$ can be $\mathbf{x}_1$ or $\mathbf{x}_2$, and ξ is the quadratic function of $\mathbf{x}_1$ or $\mathbf{x}_2$ in PADMM$_1$.

Convergence of PAMMM

Convergence of PADMM$_1$ [14]

Assumptions:

- Let $\{(\mathbf{x}^k, \lambda^k)\}_{k\geq 0}$ be the sequence generated by **PDM** using PADMM$_1$.
- Let $\gamma_k = \gamma_0 := \frac{2\sqrt{2}\|\mathbf{A}\|}{K+3}$ and $\beta_{k+1} := (1-\tau_k)\beta_k$ for $k = 0, \cdots, K$.

Conclusion:

$$\begin{cases} & \|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq \frac{2\sqrt{2}\|\mathbf{A}\|(D_{\Lambda^\star} + \bar{D}_{\mathcal{X}})}{K+3} \\ -D_{\Lambda^\star}\|\mathbf{A}\bar{\mathbf{x}}^K - \mathbf{b}\| & \leq f(\bar{\mathbf{x}}^K) - f^\star & \leq \frac{2\sqrt{2}\|\mathbf{A}\|}{K+3}(\bar{D}_{\mathcal{X}})^2 \end{cases}$$

where $\bar{D}_{\mathcal{X}} := 4 \max_{\mathbf{x},\hat{\mathbf{x}}\in\mathcal{X}} \|\mathbf{x} - \hat{\mathbf{x}}\|$.

Enhancements of the PDM algorithm:

- There is a freedom of choosing the center point $\mathbf{x}_c$ for computing $\mathbf{x}^\star_\gamma(\lambda)$.
  - $\mathbf{x}_c$ can be fixed for all the iterations.
  - One can choose $\mathbf{x}_c$ as the previous iteration, i.e., $\mathbf{x}_c := \mathbf{x}^\star_{\gamma_k}(\lambda^{k-1})$.
  - Or choose $\mathbf{x}_c$ adaptively as in the PADMM variant.
- The smoothness parameter γ can be increased as long as the objective values does not increase substantially.
  - When $\mathbf{x}_c$ is adaptively chosen, we can slightly increase γ_k as $\gamma_{k+1} := c\gamma_k$, for e.g., $c_k := 1.05$.

Comparison

We summarize the convergence rate of 5 different methods and the assumptions where the methods use in the following table:

- The average sequence $\{\widehat{\mathbf{x}}^k\}$ is computed as $\widehat{\mathbf{x}}^k := (k+1)^{-1}\sum_{j=0}^{k}\mathbf{x}^j$.
- Convergence guarantee using this sequence is referred to as an ergodic convergence.

Method name	Assumptions	Convergence	References
ADMM	$\leq$ 2-decomposable	$\mathcal{O}(1/k)$ on the joint $(\mathbf{x}^k, \mathbf{y}^k)$ using a gap function	[2, 8, 9]
[Fast] ADMM	$\leq$ 2-decomposable and f_1 or $f_2 \in \mathcal{F}_\mu$	$[\mathcal{O}(1/k^2)]$ $\mathcal{O}(1/k)$ on the dual-objective	[6]
Decomposition methods with 1P2D and 2P1D	p-decomposable	$\lvert f(\mathbf{x}^k) - f^\star\rvert \leq \mathcal{O}(1/k)$ and $\lVert\mathbf{A}\mathbf{x}^k - \mathbf{b}\rVert_2 \leq \mathcal{O}(1/k)$ (*non-ergodic*)	[14]
	p-decomposable and $f_i \in \mathcal{F}_\mu$	$\lvert f(\mathbf{x}^k) - f^\star\rvert \leq \mathcal{O}(1/k^2)$, $\lVert\mathbf{A}\mathbf{x}^k - \mathbf{b}\rVert_2 \leq \mathcal{O}(1/k^2)$, and $\lVert\mathbf{x}^k - \mathbf{x}^\star\rVert_2 \leq \mathcal{O}(1/k)$ (*non-ergodic*)	
New ADMM	$\leq$ 2-decomposable	$\lvert f(\mathbf{x}^k) - f^\star\rvert \leq \mathcal{O}(1/k)$ and $\lVert\mathbf{A}\mathbf{x}^k - \mathbf{b}\rVert_2 \leq \mathcal{O}(1/k)$ (*non-ergodic*)	[14]
New preconditioned ADMM	$\leq$ 2-decomposable	$\lvert f(\mathbf{x}^k) - f^\star\rvert \leq \mathcal{O}(1/k)$ and $\lVert\mathbf{A}\mathbf{x}^k - \mathbf{b}\rVert_2 \leq \mathcal{O}(1/k)$ (*non-ergodic*)	[14]

Example 1: Group sparse recovery

Sparse recovery

- Let $\mathcal{I} := \{1, \ldots, p\}$ be the set of indices. Let $\mathfrak{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_g\}$ be the set of g groups $\mathcal{G}_i \subseteq \mathcal{I}$ and $\mathcal{I} \subseteq \cup_{i=1}^{g} \mathcal{U}_i$.
- For given group $\mathcal{G}_i$, and a vector $\mathbf{x} \in \mathbb{R}^p$, we use $\mathbf{x}_{\mathcal{G}_i} = \{x_j \ : \ j \in \mathcal{G}_i\}$.
- For fixed group structure $\mathfrak{G}$, $\mathbf{x} \in \mathbb{R}^p$ is called group sparse vector if the number of groups in $\mathcal{G}$ is **small**.
- Given a **linear operator** $\mathbf{A}$ and an **observed/measurement** vector $\mathbf{b} \in \mathbb{R}^n$. We want to recover the **group sparse** input vector $\mathbf{x} \in \mathbb{R}^p$ such that $\mathbf{b} = \mathbf{A}\mathbf{x}$.

Example 1: Group sparse recovery

Sparse recovery

- Let $\mathcal{I} := \{1, \ldots, p\}$ be the set of indices. Let $\mathfrak{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_g\}$ be the set of g groups $\mathcal{G}_i \subseteq \mathcal{I}$ and $\mathcal{I} \subseteq \cup_{i=1}^{g} \mathcal{U}_i$.
- For given group $\mathcal{G}_i$, and a vector $\mathbf{x} \in \mathbb{R}^p$, we use $\mathbf{x}_{\mathcal{G}_i} = \{x_j \ : \ j \in \mathcal{G}_i\}$.
- For fixed group structure $\mathfrak{G}$, $\mathbf{x} \in \mathbb{R}^p$ is called group sparse vector if the number of groups in $\mathcal{G}$ is **small**.
- Given a **linear operator** $\mathbf{A}$ and an **observed/measurement** vector $\mathbf{b} \in \mathbb{R}^n$. We want to recover the **group sparse** input vector $\mathbf{x} \in \mathbb{R}^p$ such that $\mathbf{b} = \mathbf{A}\mathbf{x}$.

Optimization formulation

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \mathbb{R}^p} & \sum_{\mathcal{G}_i \in \mathfrak{G}} \|\mathbf{x}_{\mathcal{G}_i}\|_2 \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{array} \tag{56}$$

Here, $f(\mathbf{x}) := \sum_{\mathcal{G}_i \in \mathfrak{G}} \|\mathbf{x}_{\mathcal{G}_i}\|_2$ and $\mathcal{X} := \mathbb{R}^p$. This problem possesses two common structures: decomposability and tractable proximity.

When $g = p$ and $\mathcal{G}_i = \{i\}$, (56) reduces to the well-known linear sparse recovery problem (basis pursuit):

$$\min_{\mathbf{x} \in \mathbb{R}^p} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{57}$$

Example 1: Group sparse recovery - Numerical results

Algorithm configuration:

- Assume that (56) is constrained by a boxed constraint $\mathbf{x} \in \mathcal{X} := [\mathbf{l}, \mathbf{u}]$.
- The Bregman distance is chosen as $d(\mathbf{x}, \mathbf{x}_c) := (1/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$ and $\mathbf{x}_c = 0 \in [\mathbf{l}, \mathbf{u}]$.
- $\beta_0 = \gamma_0 = \|\mathbf{A}\|$ in 2P1D and $\gamma_0 := 2\sqrt{2}\|\mathbf{A}\|/(K+1)$ with $K = 10^4$.

Data generation:

- $p = 1024$, $n = 341$ and $g = 128$.
- $\mathbf{A}$ is a random matrix generated via the standard Gaussian distribution.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural$, where $\mathbf{x}^\natural$ is a 128-group sparse vector.
- The group $\mathfrak{G}$ is also generated randomly.
- $\mathbf{l} := \min(\mathbf{x}^\natural)$ and $\mathbf{u} := \max(\mathbf{x}^\natural)$.

Example 1: Group sparse recovery - Numerical results

Algorithm configuration:

- Assume that (56) is constrained by a boxed constraint $\mathbf{x} \in \mathcal{X} := [\mathbf{l}, \mathbf{u}]$.
- The Bregman distance is chosen as $d(\mathbf{x}, \mathbf{x}_c) := (1/2)\|\mathbf{x} - \mathbf{x}_c\|_2^2$ and $\mathbf{x}_c = 0 \in [\mathbf{l}, \mathbf{u}]$.
- $\beta_0 = \gamma_0 = \|\mathbf{A}\|$ in 2P1D and $\gamma_0 := 2\sqrt{2}\|\mathbf{A}\|/(K+1)$ with $K = 10^4$.

Data generation:

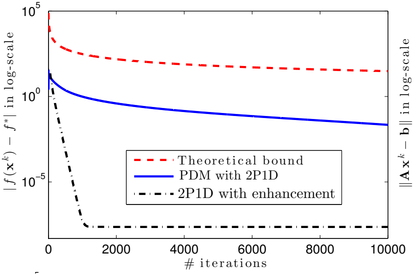
- $p = 1024$, $n = 341$ and $g = 128$.
- $\mathbf{A}$ is a random matrix generated via the standard Gaussian distribution.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural$, where $\mathbf{x}^\natural$ is a 128-group sparse vector.
- The group $\mathfrak{G}$ is also generated randomly.
- $\mathbf{l} := \min(\mathbf{x}^\natural)$ and $\mathbf{u} := \max(\mathbf{x}^\natural)$.

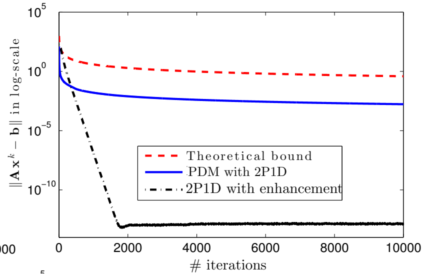
Comparison We compare the following three quantities:

- The theoretical bounds given in the right-hand side of the convergence theorem
- The PDM algorithm with 2P1D or 1P2D - i.e., follow the theory.
- The 2P1D or 1P2D with enhancement - i.e., updating the parameter γ_k by $\gamma_{k+1} := 1.05\gamma_k$ and using the adaptive center point $\mathbf{x}_c$ as in **PADMM**.

Example 1: Theoretical bounds vs actual performance

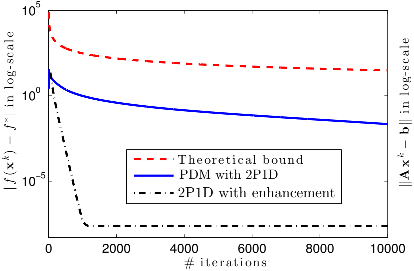
The performance of two variants of PDM using 2P1D

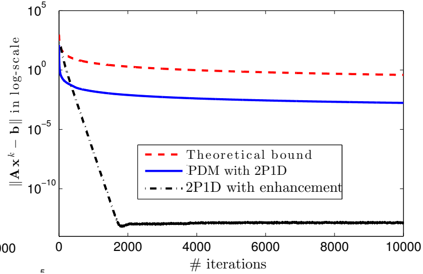




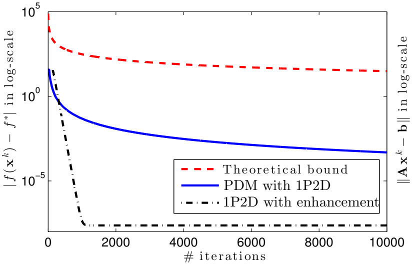
Example 1: Theoretical bounds vs actual performance

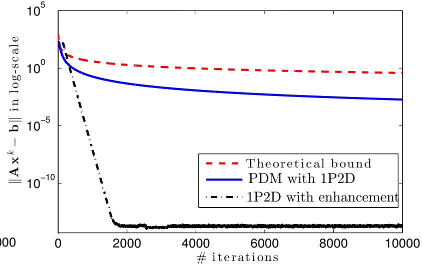
The performance of two variants of PDM using 2P1D





The performance of two variants of PDM using 1P2D





Example 2: Image processing

Problem (Imaging denoising/deblurring)

Our goal is to obtain a clean image $\mathbf{x}$ *given "dirty" observations* $\mathbf{b} \in \mathbb{R}^{n \times 1}$ *via* $\mathbf{b} = \mathcal{A}(\mathbf{x}) + \mathbf{w}$*, where* $\mathcal{A}$ *is a linear operator, which, e.g., captures camera blur as well as image subsampling, and* $\mathbf{w}$ *models Gaussian perturbations.*

Optimization formulation

$$\boxed{\text{Gaussian}: \quad \min_{\mathbf{z} \in \mathcal{Z}} \left\{ (1/2)\|\mathcal{A}(\mathbf{z}) - \mathbf{b}\|_2^2 + \rho\|\mathbf{z}\|_{\text{TV}} \right\}} \tag{58}$$

where $\|\mathbf{z}\|_{\text{TV}} := \sum_{i,j} |z_{i,j+1} - z_{i,j}| + |z_{i+1,j} - z_{i,j}|$, $\rho > 0$ is a regularization parameter and $\mathcal{Z} := [0, 255]^{n \times p}$.

Example 2: Image processing

Problem (Imaging denoising/deblurring)

Our goal is to obtain a clean image $\mathbf{x}$ *given "dirty" observations* $\mathbf{b} \in \mathbb{R}^{n \times 1}$ *via* $\mathbf{b} = \mathcal{A}(\mathbf{x}) + \mathbf{w}$, *where* $\mathcal{A}$ *is a linear operator, which, e.g., captures camera blur as well as image subsampling, and* $\mathbf{w}$ *models Gaussian perturbations.*

Optimization formulation

$$\boxed{\text{Gaussian}: \quad \min_{\mathbf{z} \in \mathcal{Z}} \left\{ (1/2)\|\mathcal{A}(\mathbf{z}) - \mathbf{b}\|_2^2 + \rho\|\mathbf{z}\|_{\mathrm{TV}} \right\}} \tag{58}$$

where $\|\mathbf{z}\|_{\mathrm{TV}} := \sum_{i,j} |z_{i,j+1} - z_{i,j}| + |z_{i+1,j} - z_{i,j}|$, $\rho > 0$ is a regularization parameter and $\mathcal{Z} := [0, 255]^{n \times p}$.

Reformulation: Since $\|\mathbf{z}\|_{\mathrm{TV}} = \|\mathbf{D}\mathbf{z}\|_1$ for a given matrix $\mathbf{D}$. By letting $\mathbf{r} = \mathbf{D}\mathbf{z}$, we can reformulate (58) as

$$\begin{aligned} \min_{\mathbf{z} \in \mathcal{Z}, \mathbf{r}} \quad & \left\{ (1/2)\|\mathcal{A}(\mathbf{z}) - \mathbf{b}\|_2^2 + \rho\|\mathbf{r}\|_1 \right\} \\ \text{s.t.} \quad & \mathbf{D}\mathbf{z} - \mathbf{r} = 0. \end{aligned}$$

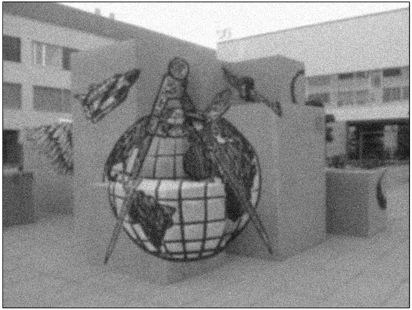
This problem is a constrained convex minimization problem with 2-decomposable objective $f(\mathbf{x}) := (1/2)\|\mathcal{A}(\mathbf{z}) - \mathbf{b}\|_2^2 + \rho\|\mathbf{r}\|_1$ and $\mathbf{x} := (\mathbf{z}, \mathbf{r})$.

Example 2: Image processing - Input data

The original image and Gaussian noise image

original image

input: noise image



Data generation:

- The original image is filtered with a multidimensional filter H of size 9×9 (circulant).
- 0.5% Gaussian noise is added to the output.

Parameter configuration:

- The number of iterations: 200 and the relative tolerance: 10^{-8}.

Example 2: Image processing - Numerical results

The performance of the new ADMM variant of PDM for two values of ρ.

output: 1P2D with $\rho = 100$

output: 1P2D with $\rho = 50$

- $f(\mathbf{x}^\natural) = 138097.919259$ and 77284.828237, where $\mathbf{x}^\natural$ is the original image.
- The **objective values**: $f(\mathbf{x}^k) := 122557.13880$ and $f(\mathbf{x}^k) := 64456.44963$
- **Relative error** between original image to clean image: 0.089152 and 0.089167
- PSNR: 26.011 and 25.994.

Example 3: Binary linear support vector machine

Problem (**Binary classification**)

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ *and a binary class label vector* $\mathbf{b} \in \{-1, +1\}^n$*.*
The goal is to find a separating hyperplane $\varphi(\mathbf{a}, \mathbf{z}) := \mathbf{a}^T\mathbf{z} + \mu$ *such that*

$$b_i = \begin{cases} +1 & \textit{if } \varphi(\mathbf{a}, \mathbf{z}) \geq 0 \\ -1 & \textit{otherwise} \end{cases}$$

where $\mathbf{z} \in \mathbb{R}^p$ *is a weight vector,* μ *is called a bias.*

Example 3: Binary linear support vector machine

Problem (**Binary classification**)

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ *and a binary class label vector* $\mathbf{b} \in \{-1, +1\}^n$.
The goal is to find a separating hyperplane $\varphi(\mathbf{a}, \mathbf{z}) := \mathbf{a}^T\mathbf{z} + \mu$ *such that*

$$b_i = \begin{cases} +1 & \text{if } \varphi(\mathbf{a}, \mathbf{z}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

where $\mathbf{z} \in \mathbb{R}^p$ *is a weight vector,* μ *is called a bias.*

Optimization formulation

$$\min_{\mathbf{z} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathcal{H}(b_i, \mathbf{a}_i^T \mathbf{z} + \mu) + \rho \|\mathbf{z}\|_1 \right\} \tag{59}$$

where $\mathbf{a}_i$ is the i-th row of the observed data matrix $\mathbf{A}$ in $\mathbb{R}^{n \times p}$, $\rho > 0$ is a regularization parameter, and $\mathcal{H}$ is the Hingle loss function $\mathcal{H}(s, \tau) := \max\{0, 1 - s\tau\}$.

Constrained reformulation: By introducing a slack variable $\mathbf{r} := \mathbf{A}\mathbf{z} + \mu$, we have

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^p, \mathbf{r}} \quad & \left\{ \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \mathcal{H}(b_i, r_i) + \rho \|\mathbf{z}\|_1 \right\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{z} + \mu - \mathbf{r} = 0. \end{aligned}$$

Example 3: Binary linear support vector machine

Testing data

- **Test problems**: Two real-world problems `a1a` and `news20` from
  `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`
- The data size of `a1a`: $p = 119$ features and $n = 1605$ data points
- The data size of `news20`: $p = 1355191$ features and $n = 19996$ data points
- The parameter ρ changes from $\rho^{-1} = 10^{-3}$ to $\rho^{-1} = 10^{3}$.

Comparison: We compare the new PADMM variant with **LibSVM**.

Example 3: Binary linear support vector machine

Testing data

- **Test problems**: Two real-world problems `a1a` and `news20` from
  `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`
- The data size of `a1a`: $p = 119$ features and $n = 1605$ data points
- The data size of `news20`: $p = 1355191$ features and $n = 19996$ data points
- The parameter ρ changes from $\rho^{-1} = 10^{-3}$ to $\rho^{-1} = 10^{3}$.

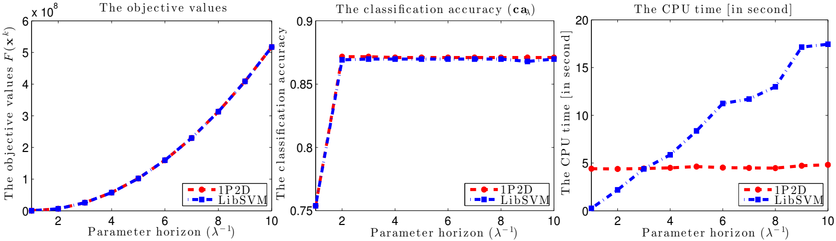
Comparison: We compare the new PADMM variant with **LibSVM**.

Problem	The parameter values									
λ^{-1}	10^{-3}	111.1	222.2	333.3	444.4	555.6	666.7	777.8	888.9	10^{3}
The accuracy of problem `a1a`										
(1P2D)	0.7539	0.8717	0.8717	0.8710	0.8710	0.8710	0.8710	0.8710	0.8710	0.8710
`LibSVM`	0.7539	0.8692	0.8698	0.8698	0.8698	0.8698	0.8698	0.8698	0.8679	0.8698
The CPU time [in second] of problem `a1a`										
(1P2D)	4.4045	4.3769	4.4246	4.4941	4.6238	4.5175	4.4836	4.4719	4.7179	4.8097
`LibSVM`	0.2549	2.1909	4.3884	5.8583	8.3662	11.2350	11.7036	12.9832	17.1424	17.4362
The accuracy of problem `news20`										
(1P2D)	0.5001	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987	0.9987
`LibSVM`	0.5001	0.9987	0.9987	0.9987	0.9987	0.9988	0.9988	0.9988	0.9988	0.9988
The CPU time [in second] of problem `news20`										
(1P2D)	762.31	1023.22	994.64	1043.06	984.24	989.70	1064.33	1073.94	984.47	1018.35
`LibSVM`	890.26	1440.28	1449.23	1439.77	1434.27	1518.56	1560.38	1557.48	1535.19	1530.71

- The **accuracy** is computed as $\mathrm{ca}_{\rho} := 1 - n^{-1}\sum_{j=1}^{n}[\mathrm{sign}(\mathbf{A}\mathbf{z}^k + \mu)_i \neq b_i)]$.

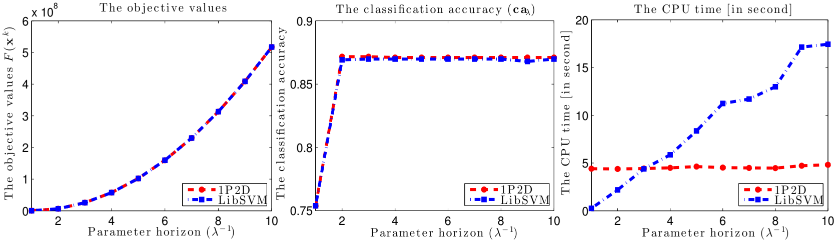
Example 3: Binary linear support vector machine

The results of two algorithms on the real-world problem a1a

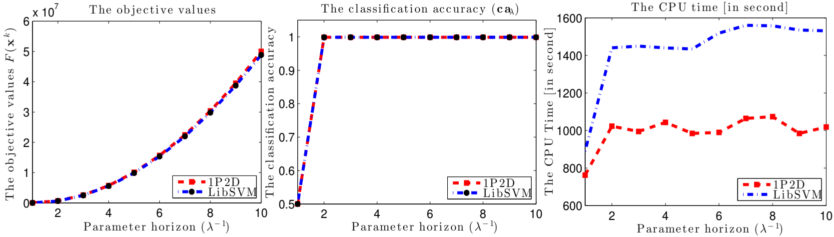


Example 3: Binary linear support vector machine

The results of two algorithms on the real-world problem `a1a`



The results of two algorithms on the real-world problem `news20`



References

[1] A. Beck and M. Teboulle.
A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.
SIAM J. Imaging Sciences, 2(1):183–202, 2009.

[2] A. Chambolle and T. Pock.
A first-order primal-dual algorithm for convex problems with applications to imaging.
Journal of Mathematical Imaging and Vision, 40(1):120–145, 2011.

[3] J. Eckstein and D. Bertsekas.
On the Douglas - Rachford splitting method and the proximal point algorithm for maximal monotone operators.
Math. Program., 55:293–318, 1992.

[4] F. Facchinei and J.-S. Pang.
Finite-dimensional variational inequalities and complementarity problems, volume 1-2.
Springer-Verlag, 2003.

[5] T. Goldstein, E. Esser, and R. Baraniuk.
Adaptive Primal-Dual Hybrid Gradient Methods for Saddle Point Problems.
Tech. Report., http://arxiv.org/pdf/1305.0546v1.pdf:1–26, 2013.

References

[6] T. Goldstein, B. ODonoghue, and S. Setzer.
Fast Alternating Direction Optimization Methods.
Tech. report., Department of Mathematics, University of California, Los Angeles, USA, May 2012.

[7] B. He and X. Yuan.
On convergence rate of the douglas-rachford operator splitting method.
Tech. Report. (submitted to Math. Program.).

[8] B. He and X. Yuan.
On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers.
(submitted for publication), 2012.

[9] B.S. He and X.M. Yuan.
On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method.
SIAM J. Numer. Anal., 50:700–709, 2012.

[10] P. L. Lions and B. Mercier.
Splitting algorithms for the sum of two nonlinear operators.
SIAM J. Num. Anal., 16:964–979, 1979.

References

[11] Y. Ouyang, Y. Chen, G. LanG. Lan., and E. JR. Pasiliao.
An accelerated linearized alternating direction method of multiplier.
Tech, 2014.

[12] R.T. Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM Journal on Control and Optimization, 14:877–898, 1976.

[13] R. Shefi and M. Teboulle.
Rate of Convergence Analysis of Decomposition Methods Based on the Proximal Method of Multipliers for Convex Minimization.
SIAM J. O, 24(1):269–297, 2014.

[14] Q. Tran-Dinh and V. Cevher.
A primal-dual algorithmic framework for constrained convex minimization.
Tech. Report., LIONS, pages 1–54, 2014.

[15] Q. Tran-Dinh, Y. H. Li, and V. Cevher.
Barrier smoothing for nonsmooth convex minimization.
In *Proc. of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*, 2014.

[16] H. Wang and A. Banerjee.
Bregman Alternating Direction Method of Multipliers.
http://arxiv.org/pdf/1306.3203v1.pdf, pages 1–18, 2013.

References

[17] M. Zhu and T. Chan.
An efficient primal-dual hybrid gradient algorithm for total variation image restoration.
UCLA CAM technical report, 08–34, 2008.