12. Primal–dual proximal methods

- primal–dual optimality conditions
- primal–dual hybrid gradient algorithm
- monotone operators
- proximal point algorithm

Primal and dual problem

$$\begin{array}{lll} \text{primal:} & \text{minimize} & f(x) + g(Ax) \\ \text{dual:} & \text{maximize} & -g^*(z) - f^*(-A^T z) \end{array}$$

- f and g are closed convex functions
- dual problem is Lagrange dual of reformulated problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

Optimality (Karush–Kuhn–Tucker) conditions (see pp. 5.21–5.24)

- primal feasibility: $x \in \operatorname{dom} f$ and $Ax = y \in \operatorname{dom} g$
- (x, y) is a minimizer of the Lagrangian $f(x) + g(y) + z^T(Ax - y)$:

$$-A^T z \in \partial f(x), \qquad z \in \partial g(y) \quad \text{(equivalently, } y \in \partial g^*(z)\text{)}$$

Primal–dual optimality conditions

- the optimality conditions can be written symmetrically as

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}$$

- second term on right-hand side denotes the product set

$$\partial f(x) \times \partial g^*(z) = \{(u, v) \mid u \in \partial f(x),\ v \in \partial g^*(z)\}$$

- solutions are saddle points of convex–concave function

$$f(x) - g^*(z) + z^T A x$$

in this lecture we assume that the optimality conditions are solvable

(a sufficient condition is that primal is solvable and Slater's condition holds)

Outline

- primal–dual optimality conditions
- **primal–dual hybrid gradient algorithm**
- monotone operators
- proximal point algorithm

Primal–dual hybrid gradient (PDHG) method

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}$$

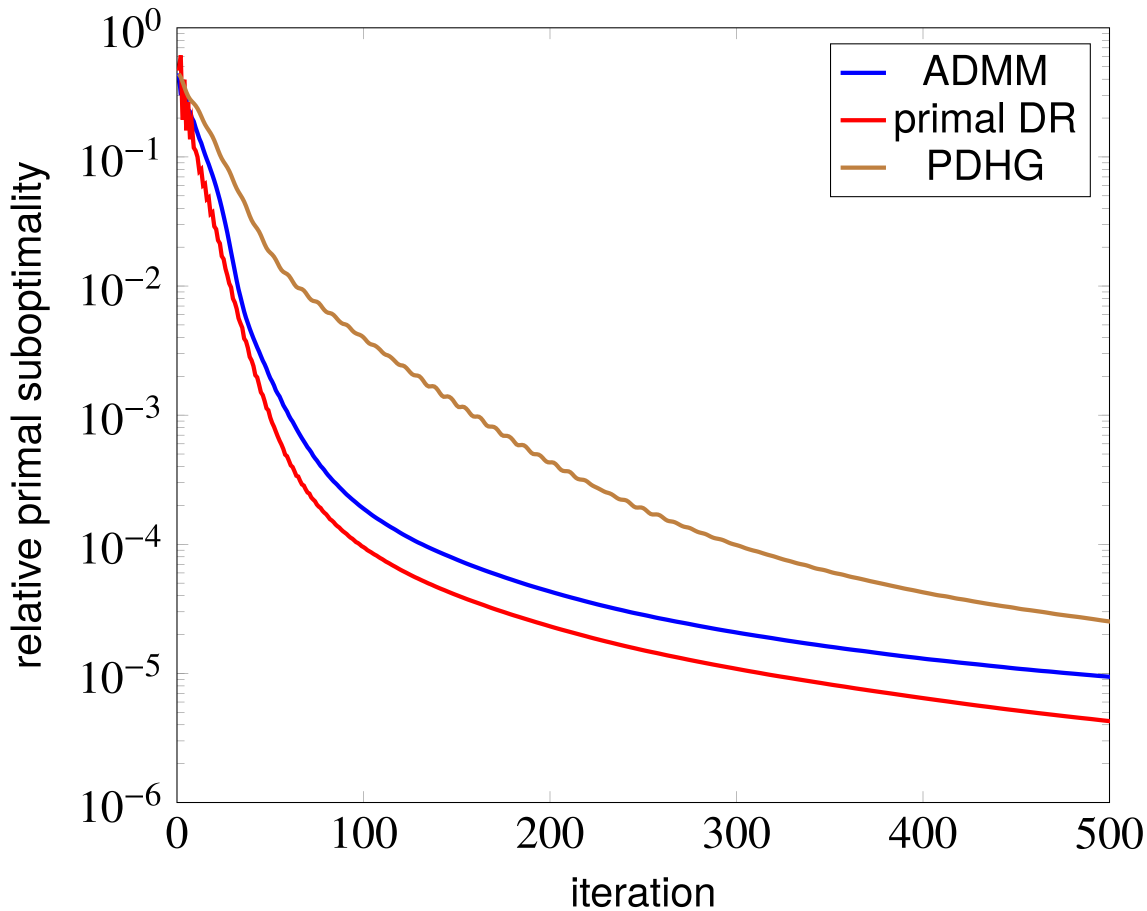
Algorithm

$$\begin{aligned} x_{k+1} &= \operatorname{prox}_{\tau f}(x_k - \tau A^T z_k) \\ z_{k+1} &= \operatorname{prox}_{\sigma g^*}(z_k + \sigma A(2x_{k+1} - x_k)) \end{aligned}$$

- each iteration requires evaluations of proximal mappings of f and g^*
- requires multiplications with A and A^T, but no solutions of linear equations
- primal and dual step sizes τ, σ are positive and must satisfy $\sigma\tau\|A\|_2^2 \leq 1$

Example

same problem as on pp. 11.20–11.24



- multiplications with A and A^T require 2-D FFTs
- with periodic boundary conditions, cost/iteration is similar for the three methods

Douglas–Rachford method derived from PDHG

$$\text{minimize} \quad f(x) + g(x)$$

- a special case of the standard problem on page 12.2 with $A = I$
- apply PDHG with $\sigma = \tau = 1$:

$$\begin{aligned} x_{k+1} &= \text{prox}_f(x_k - z_k) \\ z_{k+1} &= \text{prox}_{g^*}(z_k + 2x_{k+1} - x_k) \end{aligned}$$

- this is the primal–dual form of the Douglas–Rachford method on page 11.8

PDHG derived from Douglas–Rachford method

apply the Douglas–Rachford splitting method to a reformulation of the problem:

$$\text{minimize} \quad f(x) + g(Ax) \qquad \longrightarrow \qquad \begin{array}{ll} \text{minimize} & f(x) + g(Ax + By) \\ \text{subject to} & y = 0 \end{array}$$

- B is chosen to satisfy

$$AA^T + BB^T = (1/\alpha)I \qquad \text{where } 1/\alpha \geq \|A\|_2^2$$

  for example, $B = ((1/\alpha)I - AA^T)^{1/2}$

- reformulated problem is equivalent to minimizing $\tilde{f}(x, y) + \tilde{g}(x, y)$ with

$$\tilde{f}(x, y) = f(x) + \delta_{\{0\}}(y), \qquad \tilde{g}(x, y) = g(Ax + By)$$

- after simplifications, DR applied to reformulated problem will reduce to PDHG

Proximal operators for reformulated problem

- proximal operator of $\tilde{f}(x, y) = f(x) + \delta_{\{0\}}(y)$:

$$\mathrm{prox}_{\tau \tilde{f}}(x, y) = \begin{bmatrix} \mathrm{prox}_{\tau f}(x) \\ 0 \end{bmatrix}$$

- proximal operator of $\tilde{g}(x, y) = g(Ax + By)$ follows from page 6.8 and page 6.7:

$$\begin{aligned} \mathrm{prox}_{\tau \tilde{g}}(x, y) &= \begin{bmatrix} x \\ y \end{bmatrix} - \alpha \begin{bmatrix} A^T \\ B^T \end{bmatrix} (Ax + By - \mathrm{prox}_{(\tau/\alpha) g}(Ax + By) \\ &= \begin{bmatrix} x \\ y \end{bmatrix} - \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \mathrm{prox}_{\sigma g^*}(\sigma(Ax + By)) \end{aligned}$$

  where $\sigma = \alpha/\tau$

- proximal operator of $\tilde{g}^*$ follows from Moreau identity (page 6.6)

$$\mathrm{prox}_{(\tau \tilde{g})^*}(x, y) = \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \mathrm{prox}_{\sigma g^*}(\sigma(Ax + By))$$

Douglas–Rachford algorithm applied to reformulated problem

$$\text{minimize} \quad \underbrace{f(x) + \delta_{\{0\}}(y)}_{\tilde{f}(x,y)} + \underbrace{g(Ax + By)}_{\tilde{g}(x,y)}$$

- primal–dual form of Douglas–Rachford algorithm (page 11.8) with step size τ

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \operatorname{prox}_{\tau\tilde{f}}\left(\begin{bmatrix} x_k - p_k \\ y_k - q_k \end{bmatrix}\right)$$

$$\begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} = \operatorname{prox}_{(\tau\tilde{g})^*}\left(\begin{bmatrix} p_k + 2x_{k+1} - x_k \\ q_k + 2y_{k+1} - y_k \end{bmatrix}\right)$$

- substitute expressions for proximal operators (with σ defined as $\sigma = \alpha/\tau$)

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \operatorname{prox}_{\tau f}(x_k - p_k) \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} = \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \operatorname{prox}_{\sigma g^*}\left(\sigma \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} p_k + 2x_{k+1} - x_k \\ q_k + 2y_{k+1} - y_k \end{bmatrix}\right)$$

First simplification

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \mathrm{prox}_{\tau f}(x_k - p_k) \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} = \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \mathrm{prox}_{\sigma g^*}\left(\sigma \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} p_k + 2x_{k+1} - x_k \\ q_k + 2y_{k+1} - y_k \end{bmatrix}\right)$$

- from first step, $y_k = 0$ for all k if we start with $y_0 = 0$
- we remove the zero variable y_k:

$$x_{k+1} = \mathrm{prox}_{\tau f}(x_k - p_k)$$

$$\begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} = \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \mathrm{prox}_{\sigma g^*}\left(\sigma \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} p_k \\ q_k \end{bmatrix} + \sigma A(2x_{k+1} - x_k)\right)$$

Second simplification

$$
\begin{aligned}
x_{k+1} &= \operatorname{prox}_{\tau f}(x_k - p_k) \\
\begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} &= \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} \operatorname{prox}_{\sigma g^*}\left(\sigma(Ap_k + Bq_k) + \sigma A(2x_{k+1} - x_k)\right)
\end{aligned}
$$

- from step 2: $\begin{bmatrix} p_k \\ q_k \end{bmatrix} \in \operatorname{range} \begin{bmatrix} A^T \\ B^T \end{bmatrix}$ for all k, if this holds for (p_0, q_0)

- since $AA^T + BB^T = (1/\alpha)I$ and $\sigma = \alpha/\tau$,

$$
\begin{bmatrix} p_k \\ q_k \end{bmatrix} = \tau \begin{bmatrix} A^T \\ B^T \end{bmatrix} z_k \quad \text{for a unique } z_k \text{ given by } z_k = \sigma(Ap_k + Bq_k)
$$

- a change of variables $z_k = \sigma(Ap_k + Bq_k)$ gives

$$
x_{k+1} = \operatorname{prox}_{\tau f}(x_k - \tau A^T z_k), \qquad z_{k+1} = \operatorname{prox}_{\sigma g^*}(z_k + \sigma A(2x_{k+1} - x_k))
$$

this is the PDHG algorithm with $\sigma\tau = \alpha \leq 1/\|A\|_2^2$

PDHG with overrelaxation

$$
\begin{aligned}
\bar{x}_{k+1} &= \operatorname{prox}_{\tau f}(x_k - \tau A^T z_k) \\
\bar{z}_{k+1} &= \operatorname{prox}_{\sigma g^*}(z_k + \sigma A(2\bar{x}_{k+1} - x_k)) \\
\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} &= \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \rho_k \begin{bmatrix} \bar{x}_{k+1} - x_k \\ \bar{z}_{k+1} - z_k \end{bmatrix}
\end{aligned}
$$

- $\rho_k \in (0, 2)$
- convergence follows from convergence of Douglas–Rachford splitting method
- other types of acceleration exist for problems with strongly convex f or g^*

Outline

- primal–dual optimality conditions
- primal–dual hybrid gradient algorithm
- **monotone operators**
- proximal point algorithm

Multivalued (set-valued) operator

Definition: operator F maps vectors $x \in \mathbf{R}^n$ to sets $F(x) \subseteq \mathbf{R}^n$

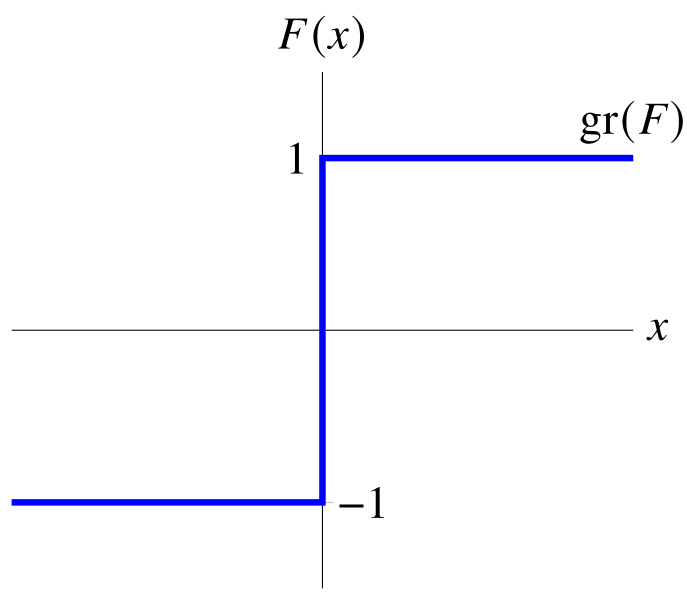
- the domain and graph of F are defined as

$$\begin{aligned} \operatorname{dom} F &= \{x \in \mathbf{R}^n \mid F(x) \neq \emptyset\} \\ \operatorname{gr}(F) &= \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^n \mid x \in \operatorname{dom} F,\ y \in F(x)\} \end{aligned}$$

- if $F(x)$ is a singleton, we write $F(x) = y$ instead of $F(x) = \{y\}$

Example: sign operator

$$F(x) = \begin{cases} -1 & x < 0 \\ [-1, 1] & x = 0 \\ 1 & x > 0 \end{cases}$$



Transformations as operations on graph

Inverse: $F^{-1}(x) = \{y \mid x \in F(y)\}$

$$\mathrm{gr}(F^{-1}) = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \mathrm{gr}(F)$$

Composition with scaling: $(\lambda F)(x) = \lambda F(x)$ and $(F\lambda)(x) = F(\lambda x)$

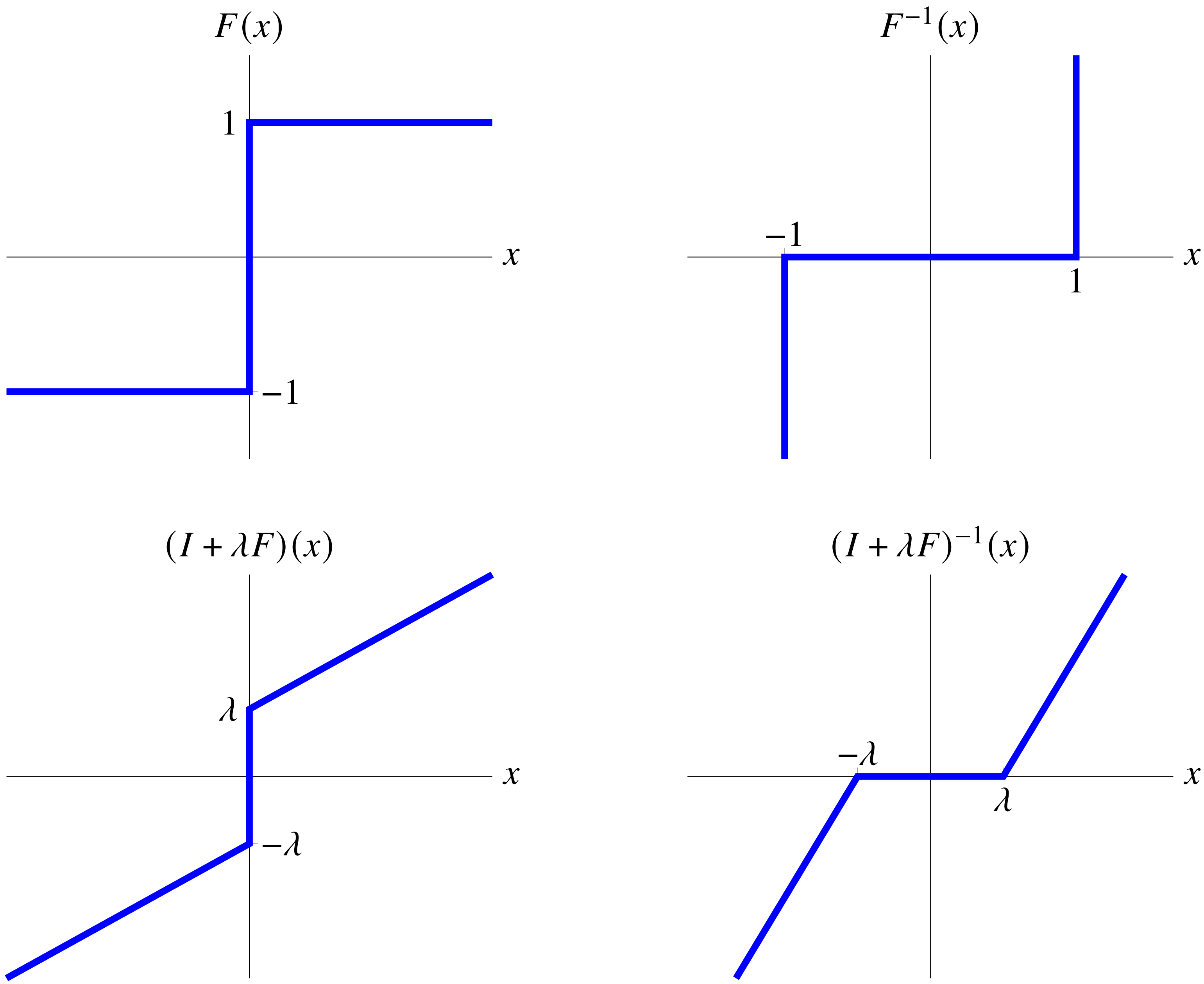
$$\mathrm{gr}(\lambda F) = \begin{bmatrix} I & 0 \\ 0 & \lambda I \end{bmatrix} \mathrm{gr}(F), \qquad \mathrm{gr}(F\lambda) = \begin{bmatrix} (1/\lambda)I & 0 \\ 0 & I \end{bmatrix} \mathrm{gr}(F)$$

Addition to identity: $(I + \lambda F)(x) = \{x + \lambda y \mid y \in F(x)\}$

$$\mathrm{gr}(I + \lambda F) = \begin{bmatrix} I & 0 \\ I & \lambda I \end{bmatrix} \mathrm{gr}(F)$$

note that these are all *linear* operations on the graph

Example



$F(x)$
1
−1
x
$F^{-1}(x)$
−1
1
x
$(I + \lambda F)(x)$
λ
$-\lambda$
x
$(I + \lambda F)^{-1}(x)$
$-\lambda$
λ
x

Monotone operator

Definition: F is a monotone operator if

$$(y - \hat{y})^T (x - \hat{x}) \geq 0 \qquad \text{for all } x, \hat{x} \in \operatorname{dom} F,\ y \in F(x),\ \hat{y} \in F(\hat{x})$$

in terms of the graph,

$$\begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix}^T \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \geq 0 \qquad \text{for all } (x, y),\ (\hat{x}, \hat{y}) \in \operatorname{gr}(F)$$

Monotone inclusion problem: find $x \in F^{-1}(0)$, *i.e.*, solve

$$0 \in F(x)$$

this covers many equilibrium/optimality conditions as special cases

Examples

we will encounter the following three types of monotone operators

- subdifferentials $\partial f(x)$ of convex functions f
- affine monotone operators: $F(x) = Cx + d$ is monotone if

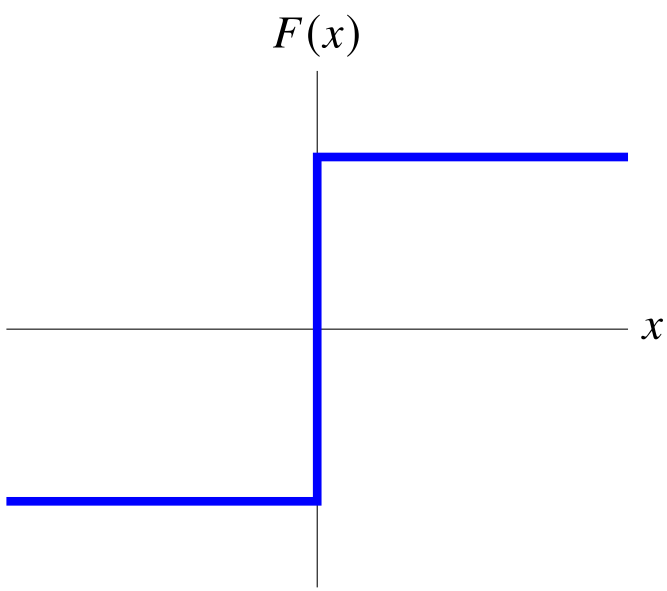
$$C + C^T \succeq 0$$

- sums of the above; in particular,

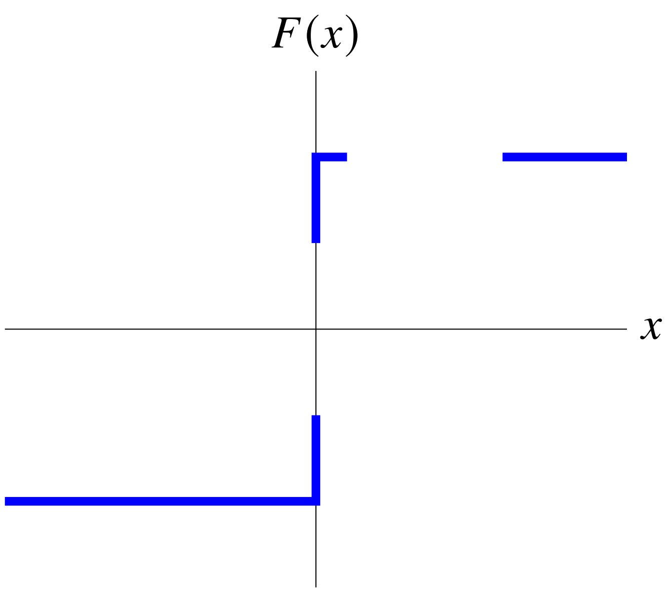
$$F(x, z) = \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}$$

Maximal monotone operator

graph is not properly contained in the graph of another monotone operator



maximal monotone



monotone, but not maximal monotone

Conditions for maximal monotonicity

- the subdifferential of a closed convex function is maximal monotone
- affine monotone operators are maximal monotone
- (Minty's theorem) a monotone operator F is maximal monotone if and only if

$$\operatorname{im}(I + F) := \bigcup_{x \in \operatorname{dom} F} (x + F(x)) = \mathbf{R}^n$$

  i.e., for every $y \in \mathbf{R}^n$, there exists an x such that $y \in x + F(x)$

- sums $F + G$ of maximal monotone operators are not necessarily maximal

  (sufficient condition: $\operatorname{int} \operatorname{dom} F \cap \operatorname{dom} G \neq \emptyset$)

Coercivity (strong monotonicity)

F is **coercive** with parameter $\mu > 0$ if

$$(y - \hat{y})^T (x - \hat{x}) \geq \mu \|x - \hat{x}\|_2^2 \qquad \text{for all } x, \hat{x} \in \operatorname{dom} F,\ y \in F(x),\ \hat{y} \in F(\hat{x})$$

- $F - \mu I$ is a monotone operator
- equivalently,

$$\begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix}^T \begin{bmatrix} -2\mu I & I \\ I & 0 \end{bmatrix} \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \geq 0 \qquad \text{for all } (x, y),\ (\hat{x}, \hat{y}) \in \operatorname{gr}(F)$$

Examples

- subdifferential of strongly convex function
- affine operator $F(x) = Ax + b$ if $A + A^T \succ 0$ (with $\mu = \lambda_{\min}(A + A^T)/2$)

Co-coercivity

F is **co-coercive** with parameter $\gamma > 0$ if F^{-1} is coercive

- $F(x)$ is single-valued for $x \in \operatorname{dom} F$ and

$$(F(x) - F(\hat{x}))^T (x - \hat{x}) \geq \gamma \|F(x) - F(\hat{x})\|_2^2 \qquad \text{for all } x, \hat{x} \in \operatorname{dom} F$$

- equivalently,

$$\begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix}^T \begin{bmatrix} 0 & I \\ I & -2\gamma I \end{bmatrix} \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \geq 0 \qquad \text{for all } (x, y),\ (\hat{x}, \hat{y}) \in \operatorname{gr}(F)$$

- F is **firmly nonexpansive** if it is co-coercive with $\gamma = 1$

Example: affine operator $F(x) = Ax + b$ with

$$A + A^T \succeq 2\gamma A^T A \qquad \Longleftrightarrow \qquad \|2\gamma A - I\|_2 \leq 1$$

for symmetric positive definite A, this means $\lambda_{\max}(A) \leq 1/\gamma$

Lipschitz continuity

- F is **Lipschitz continuous** with parameter L if it is single-valued on $\operatorname{dom} F$ and

$$\|F(x) - F(\hat{x})\|_2 \leq L\|x - \hat{x}\|_2 \qquad \text{for all } x, \hat{x} \in \operatorname{dom} F$$

- F is **nonexpansive** if it is Lipschitz continuous with $L = 1$

Example: any affine $F(x) = Ax + b$ (parameter $L = \|A\|_2$)

Relation to co-coercivity

- co-coercivity implies Lipschitz continuity (with $L = 1/\gamma$)
- Lipschitz continuity does not imply co-coercivity
- properties are equivalent for gradient of closed convex functions (page 1.15)

Resolvent

the **resolvent** of an operator F is the operator

$$(I + \lambda F)^{-1} \qquad \text{(with } \lambda > 0\text{)}$$

- inverse denotes the operator inverse:

$$y \in (I + \lambda F)^{-1}(x) \qquad \Longleftrightarrow \qquad x - y \in \lambda F(y)$$

- graph of resolvent is a linear transformation of graph of F:

$$\mathrm{gr}((I + \lambda F)^{-1}) = \begin{bmatrix} I & \lambda I \\ I & 0 \end{bmatrix} \mathrm{gr}(F)$$

Examples

Subdifferential: resolvent is proximal mapping

$$(I + \lambda \partial f)^{-1}(x) = \mathrm{prox}_{\lambda f}(x)$$

follows from subgradient characteriation of $\mathrm{prox}_{\lambda f}$ (page 4.7)

$$y = \mathrm{prox}_{\lambda f}(x) \quad \Longleftrightarrow \quad x - y \in \lambda \partial f(y)$$

Monotone affine mapping: resolvent of $F(x) = Ax + b$ is

$$(I + \lambda F)^{-1}(x) = (I + \lambda A)^{-1}(x - \lambda b)$$

- inverse on right-hand side is standard matrix inverse
- $I + \lambda A$ is nonsingular for all $\lambda \geq 0$ because $A + A^T \geq 0$

Monotonicity properties

- an operator is monotone if and only if its resolvent is firmly nonexpansive:

  this follows from the matrix identity

$$\lambda \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} = \begin{bmatrix} I & I \\ \lambda I & 0 \end{bmatrix} \begin{bmatrix} 0 & I \\ I & -2I \end{bmatrix} \begin{bmatrix} I & \lambda I \\ I & 0 \end{bmatrix}$$

  and the expression of the graph of the resolvent on page 12.23

- a monotone operator F is *maximal* monotone if and only

$$\mathrm{dom}((I + \lambda F)^{-1}) = \mathbf{R}^n$$

  follows from Minty's theorem on page 12.19

Outline

- primal–dual optimality conditions
- primal–dual hybrid gradient algorithm
- monotone operators
- **proximal point algorithm**

Proximal point algorithm

Monotone inclusion problem: given maximal monotone F, find x such that

$$0 \in F(x)$$

this is equivalent to finding a fixed point of the resolvent $R_t = (I + tF)^{-1}$ of F:

$$x = R_t(x) \quad \Longleftrightarrow \quad x \in (I + tF)(x) \quad \Longleftrightarrow \quad 0 \in F(x)$$

Proximal point algorithm: fixed point iteration

$$x_{k+1} = R_{t_k}(x_k)$$

Proximal point algorithm with relaxation (relaxation parameter $\rho_k \in (0, 2)$):

$$x_{k+1} = x_k + \rho_k(R_{t_k}(x_k) - x_k)$$

Convergence

if $F^{-1}(0) \neq \emptyset$, proximal point algorithm converges

- with constant $t_k = t > 0$ and $\rho_k = \rho \in (0, 2)$
- with t_k, ρ_k varying and bounded away from their limits, *i.e.*,

$$t_k \geq t_{\min} > 0, \qquad 0 < \rho_{\min} \leq \rho_k \leq \rho_{\max} < 2 \qquad \text{for all } k$$

proof relies on firm nonexpansiveness of resolvent

Linear change of variables

make a change of variables $x = Ay$, with A nonsingular:

$$G(y) = A^T F(Ay)$$

- graph of G is

$$\mathrm{gr}\,(G) = \begin{bmatrix} A^{-1} & 0 \\ 0 & A^T \end{bmatrix} \mathrm{gr}\,(F)$$

- (maximal) monotonicity of G follows from (maximal) monotonicity of F and

$$\begin{bmatrix} A^{-1} & 0 \\ 0 & A^T \end{bmatrix}^T \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & A^T \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

“Preconditioned” proximal point algorithm

$$y_{k+1} = (I + t_k G)^{-1}(y_k)$$

- y_{k+1} is the solution y of the inclusion problem

$$\frac{1}{t_k}(y_k - y) \in A^T F(Ay)$$

- in the original coordinates $x = Ay$, this can be written as

$$\frac{1}{t_k}H(x_k - x) \in F(x)$$

  where $H = A^{-T}A^{-1}$ and $x_k = Ay_k$

- we obtain a generalized proximal point update, with $H \succ 0$ substituted for I:

$$x_{k+1} = (H + t_k F)^{-1}(Hx_k)$$

the purpose is often to make the resolvents cheaper, not preconditioning

Proximal method of multipliers

the proximal point algorithm applied to

$$F(x,z) = \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}$$

is known as the proximal method of multipliers

- basic iteration (without relaxation) is

$$(x_{k+1}, z_{k+1}) = (I + tF)^{-1}(x_k, z_k)$$

- (x_{k+1}, z_{k+1}) is the solution of the monotone inclusion with variables x, z

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix} + \frac{1}{t} \begin{bmatrix} x - x_k \\ z - z_k \end{bmatrix}$$

Evaluation of the resolvent

- equivalent inclusion problem

$$0 \in \begin{bmatrix} 0 & 0 & A^T \\ 0 & 0 & -I \\ -A & I & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g(y) \\ 0 \end{bmatrix} + \frac{1}{t} \begin{bmatrix} x - x_k \\ 0 \\ z - z_k \end{bmatrix}$$

- this is the optimality condition of the optimization problem (variables x, y)

$$\text{minimize} \quad f(x) + g(y) + \frac{t}{2}\|Ax - y + (1/t)z_k\|_2^2 + \frac{1}{2t}\|x - x_k\|_2^2$$

(the augmented Lagrangian with an extra quadratic penalty term on x)

- from the minimizer $(\hat{x}, \hat{y})$, we make the update

$$x_{k+1} = \hat{x}, \qquad z_{k+1} = z_k + t(A\hat{x} - \hat{y})$$

PDHG and proximal point algorithm

apply "preconditioned" proximal point algorithm of page 12.29 with $t_k = \tau$ and

$$H = \begin{bmatrix} I & -\tau A^T \\ -\tau A & (\tau/\sigma) I \end{bmatrix}$$

- H is positive definite for $\sigma\tau\|A\|_2^2 < 1$
- x_{k+1} and z_{k+1} are the solution x, z of

$$\frac{1}{\tau}\begin{bmatrix} I & -\tau A^T \\ -\tau A & (\tau/\sigma) I \end{bmatrix}\begin{bmatrix} x_k - x \\ z_k - z \end{bmatrix} \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix}\begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}$$

- this simplifies to

$$\begin{aligned} 0 &\in \partial f(x) + \frac{1}{\tau}(x - x_k + \tau A^T z_k) \\ 0 &\in \partial g^*(z) + \frac{1}{\sigma}(z - z_k - \sigma A(2x - x_k)) \end{aligned}$$

can solve 1st inclusion for x; substitute solution in 2nd inclusion and solve for z

PDHG and proximal point algorithm

$$
\begin{array}{rcl}
0 & \in & \partial f(x) + \dfrac{1}{\tau}(x - x_k + \tau A^T z_k) \\
0 & \in & \partial g^*(z) + \dfrac{1}{\sigma}(z - z_k - \sigma A(2x - x_k))
\end{array}
$$

- solution of the two inclusions is

$$
\begin{array}{rcl}
x_{k+1} & = & (I + \tau \partial f)^{-1}(x_k - \tau A^T z_k) \\
z_{k+1} & = & (I + \sigma \partial g^*)^{-1}(z_k + \sigma A(2x_{k+1} - x_k))
\end{array}
$$

- writing the solution in terms of prox operators gives the PDHG algorithm

$$
\begin{array}{rcl}
x_{k+1} & = & \mathrm{prox}_{\tau f}(x_k - \tau A^T z_k) \\
z_{k+1} & = & \mathrm{prox}_{\sigma g^*}(z_k + \sigma A(2x_{k+1} - x_k))
\end{array}
$$

References

Primal–dual hybrid gradient algorithm

- E. Esser, X. Zhang, T. Chan, *A general framework for a class of first order primal–dual algorithms for convex optimization in imaging sciences*, SIAM J. Imaging Sciences (2010).
- T. Pock, D.Cremers, H. Bischof, A. Chambolle, *An algorithm for minimizing the Mumford-Shah functional*, ICCV (2009).
- A. Chambolle and T. Pock, *A first-order primal–dual algorithm for convex problems with applications to imaging,* Journal of Mathematical Imaging and Vision (2011).
- A. Chambolle and T. Pock, *An introduction to continuous optimization for imaging*, Acta Numerica (2016).
- B. He and X. Yuan, *Convergence analysis of primal–dual algorithms for a saddle-point problem: from contraction perspective,* SIAM J. Imaging Sciences (2012).

  The proximal-point algorithm interpretation on pp. 12.32–12.33.
- D. O'Connor and L. Vandenberghe, *On the equivalence of the primal–dual hybrid gradient method and Douglas–Rachford splitting*, Math. Prog. (2020).

  The Douglas–Rachford interpretation on pp.12.7–12.11.

References

Extensions of PDHG

- L. Condat, *A primal–dual splitting method for convex optimization involving Lipschitzian, proximable, and linear composite terms,* JOTA (2013).
- B. C. Vũ, *A splitting algorithm for dual monotone inclusions involving cocoercive operators*, Advances in Computational Mathematics (2013).
- D. Davis and W. Yin, *A three-operator splitting scheme and its optimization applications,* Set-Valued and Variational Analysis (2017).
- M. Yan, *A new primal-dual algorithm for minimizing the sum of three functions with a linear operator*, Journal of Scientific Computing (2018).

Monotone operators and the proximal point algorithm

- H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (2017).
- E. K. Ryu and W. Yin, *Large-Scale Convex Optimization via Monotone Operators* (2022).
- E. K. Ryu and S. Boyd, *A primer on monotone operator methods*, Appl. Comput. Math. (2016).
- R. T. Rockafellar, *Monotone operators and the proximal point algorithm,* SIAM J. Control and Opt. (1976).
- J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,* Mathematical Programming (1992).

  The convergence result on page 12.27 is Theorem 3 of this paper.