

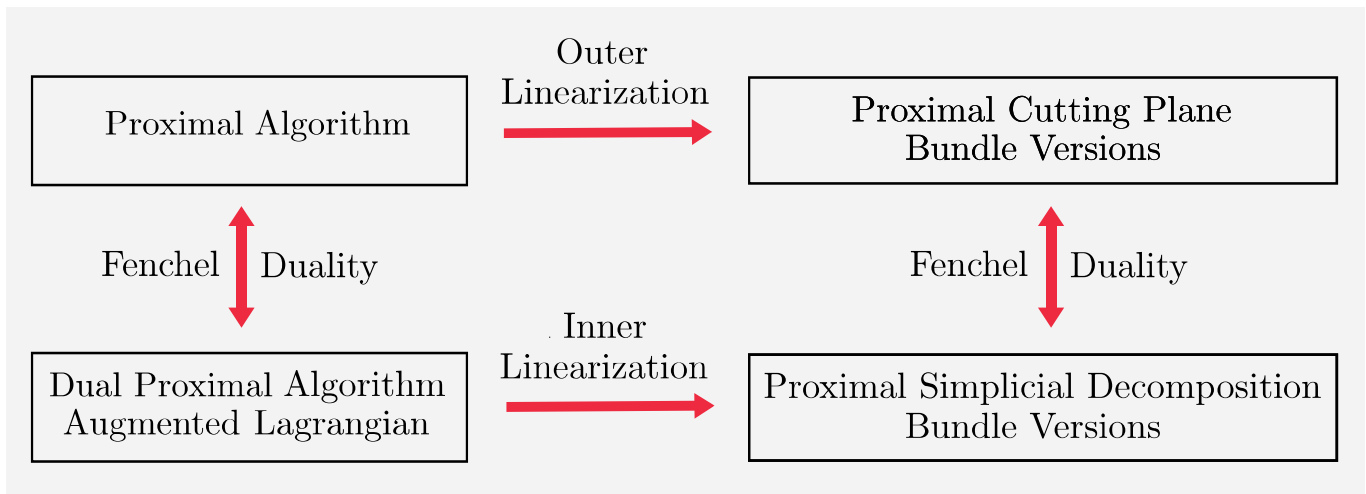
LECTURE 19

LECTURE OUTLINE

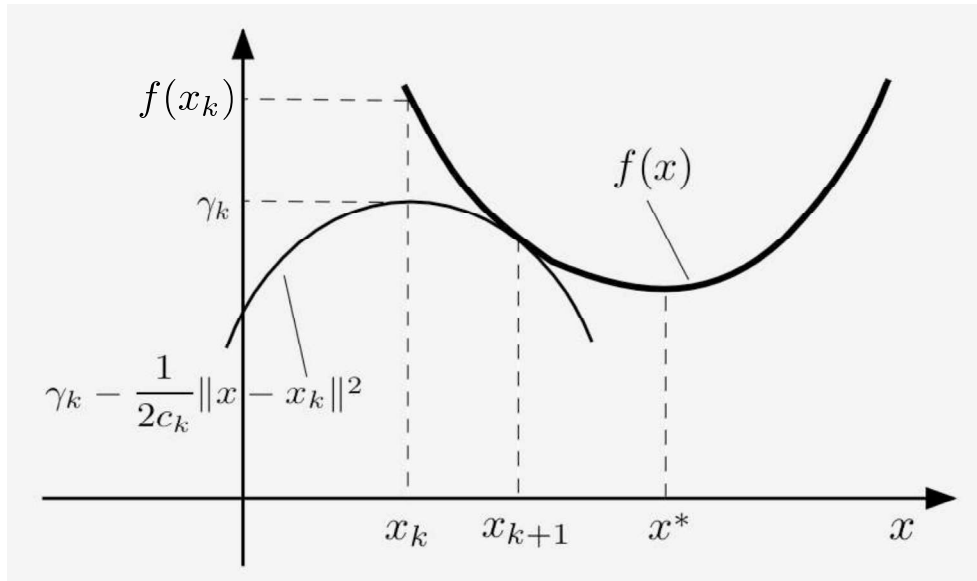
- Review of proximal algorithm
- Dual proximal algorithm
- Augmented Lagrangian methods
- Proximal cutting plane algorithm
- Bundle methods

Start with proximal algorithm and generate other methods via:

- Fenchel duality
- Outer/inner linearization



RECALL PROXIMAL ALGORITHM



- Minimizes closed convex proper f :

$$x_{k+1} = \arg \min_{x \in \mathcal{R}^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}$$

where x_0 is an arbitrary starting point, and $\{c_k\}$ is a positive parameter sequence.

- We have $f(x_k) \rightarrow f^*$. Also $x_k \rightarrow$ some minimizer of f , provided one exists.
- Finite convergence for polyhedral f .
- Each iteration can be viewed in terms of Fenchel duality.

REVIEW OF FENCHEL DUALITY

- Consider the problem

$$\begin{aligned} & \text{minimize} && f_1(x) + f_2(x) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned}$$

where f_1 and f_2 are closed proper convex.

- **Fenchel Duality Theorem:**

- (a) If f^* is finite and $\text{ri}(\text{dom}(f_1)) \cap \text{ri}(\text{dom}(f_2)) \neq \emptyset$, then strong duality holds and there exists at least one dual optimal solution.
- (b) Strong duality holds, and (x^*, λ^*) is a primal and dual optimal solution pair if and only if

$$x^* \in \arg \min_{x \in \mathfrak{R}^n} \{ f_1(x) - x' \lambda^* \}, \quad x^* \in \arg \min_{x \in \mathfrak{R}^n} \{ f_2(x) + x' \lambda^* \}$$

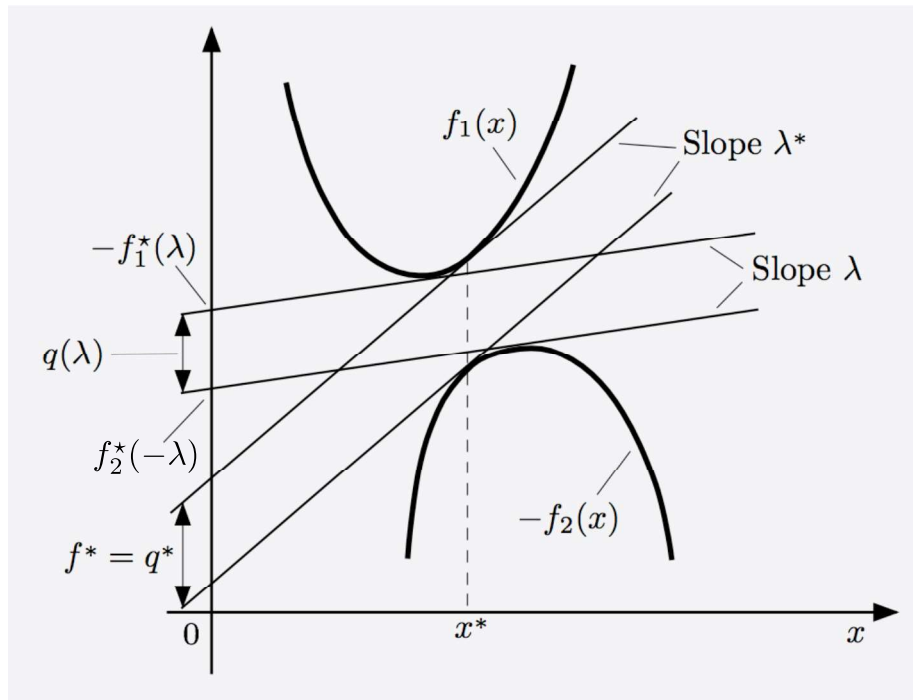
- By conjugate subgradient theorem, the last condition is equivalent to

$$\lambda^* \in \partial f_1(x^*) \quad [\text{or equivalently } x^* \in \partial f_1^*(\lambda^*)]$$

and

$$-\lambda^* \in \partial f_2(x^*) \quad [\text{or equivalently } x^* \in \partial f_2^*(-\lambda^*)]$$

GEOMETRIC INTERPRETATION

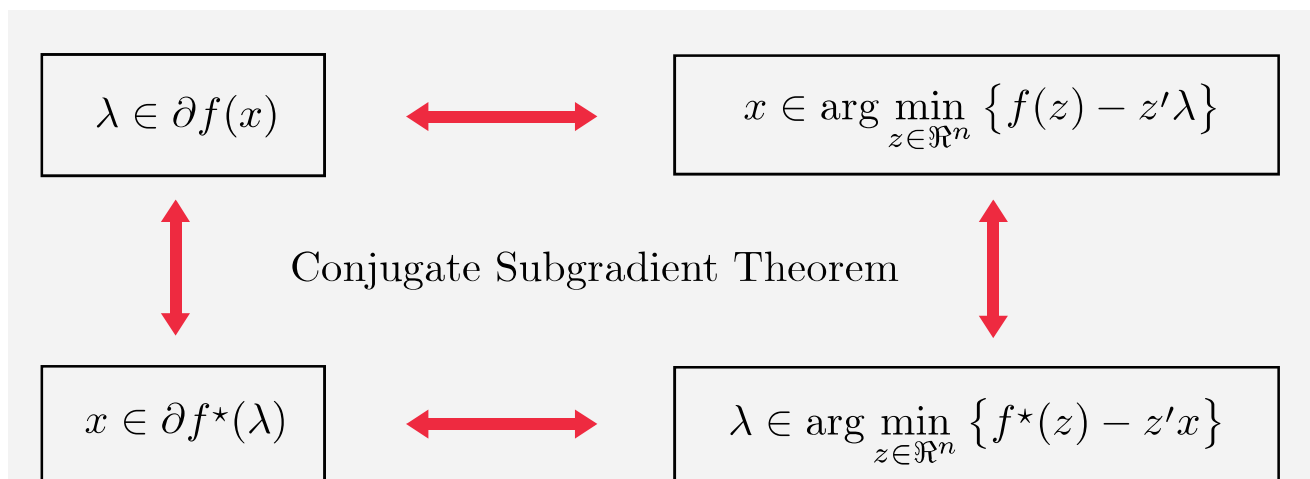


- The optimality condition is equivalent to

$$\lambda^* \in \partial f_1(x^*) \quad \text{and} \quad \lambda^* \in -\partial f_2(x^*); \quad \text{or}$$

$$x^* \in \partial f_1^*(\lambda^*) \quad \text{and} \quad x^* \in \partial f_2^*(-\lambda^*)$$

- More generally: Once we obtain one of x^* or λ^* , we can obtain the other by “differentiation”



DUAL PROXIMAL MINIMIZATION

- The proximal iteration can be written in the Fenchel form: $\min_x \{f_1(x) + f_2(x)\}$ with

$$f_1(x) = f(x), \quad f_2(x) = \frac{1}{2c_k} \|x - x_k\|^2$$

- The Fenchel dual is

$$\begin{aligned} &\text{minimize} && f_1^*(\lambda) + f_2^*(-\lambda) \\ &\text{subject to} && \lambda \in \mathfrak{R}^n \end{aligned}$$

- We have $f_2^*(-\lambda) = -x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2$, so the dual problem is

$$\begin{aligned} &\text{minimize} && f^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \\ &\text{subject to} && \lambda \in \mathfrak{R}^n \end{aligned}$$

where f^* is the conjugate of f .

- f_2 is real-valued, so no duality gap.
- Both primal and dual problems have a unique solution, since they involve a closed, strictly convex, and coercive cost function.

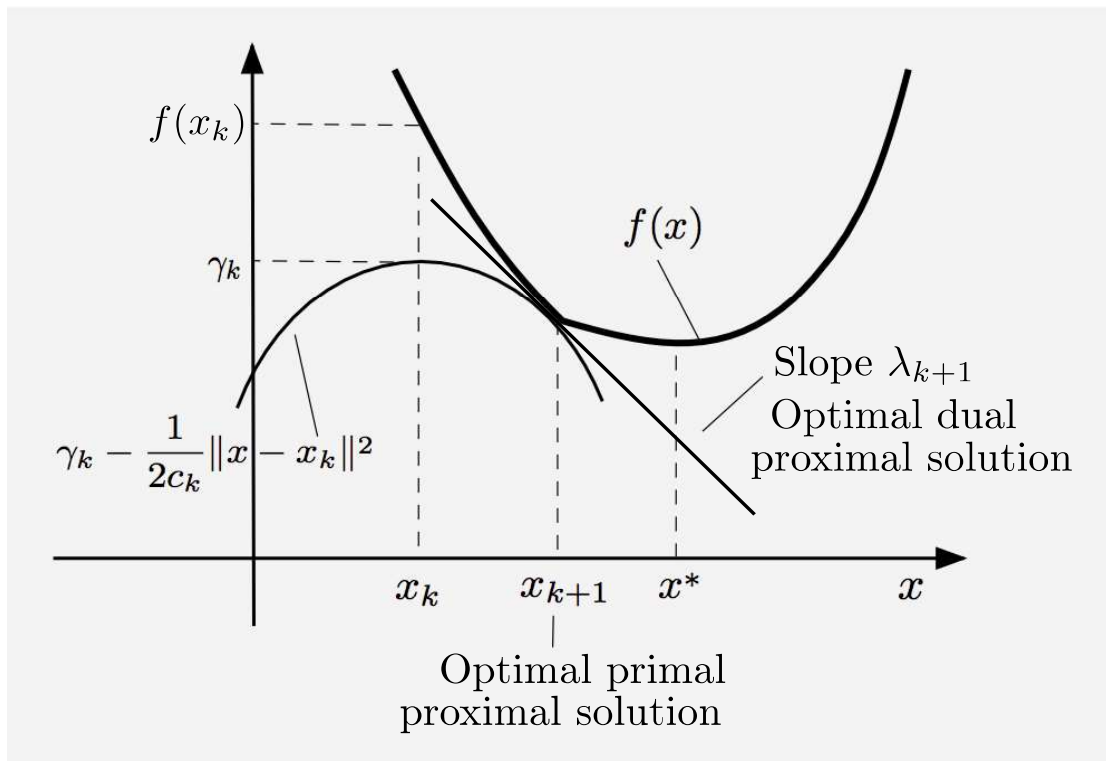
DUAL IMPLEMENTATION

- We can solve the Fenchel-dual problem instead of the primal at each iteration:

$$\lambda_{k+1} = \arg \min_{\lambda \in \mathbb{R}^n} \left\{ f^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \right\}$$

- Primal-dual optimal pair (x_{k+1}, λ_{k+1}) are related by the “differentiation” condition:

$$\lambda_{k+1} = \frac{x_k - x_{k+1}}{c_k}$$



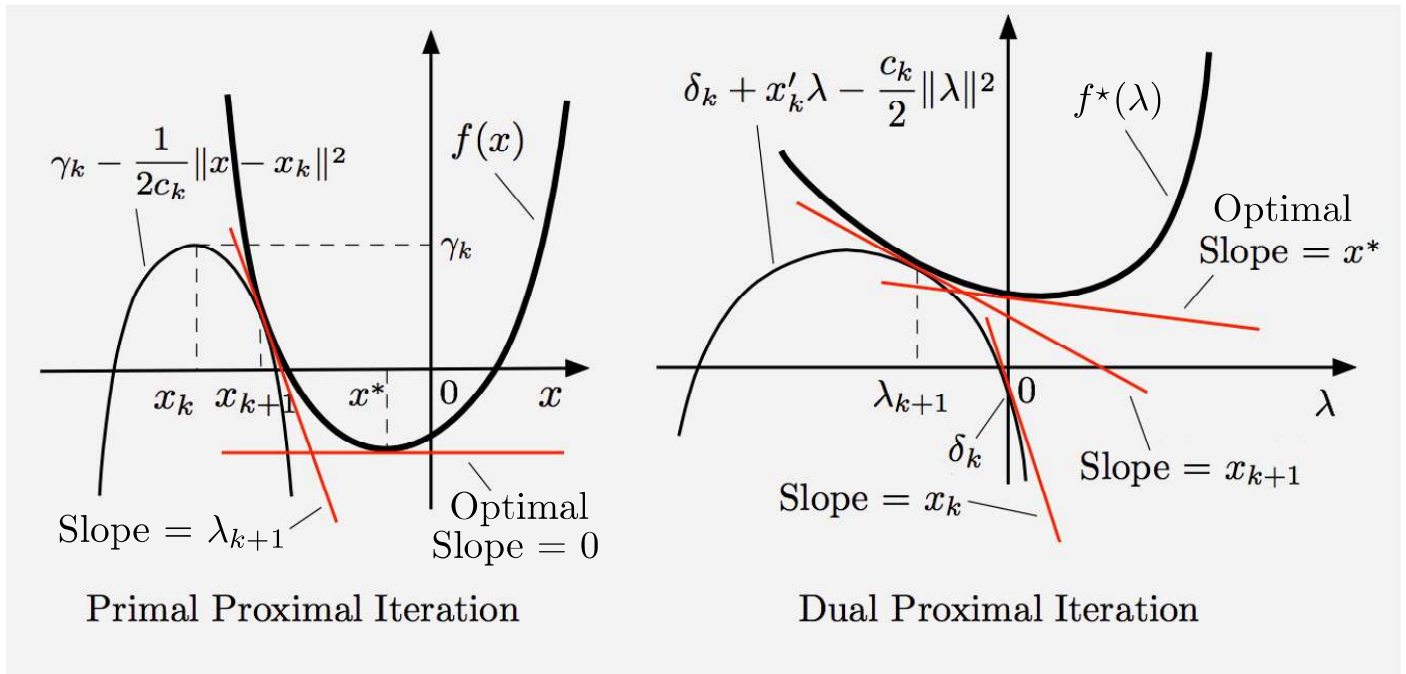
DUAL PROXIMAL ALGORITHM

- Obtain λ_{k+1} and x_{k+1} from

$$\lambda_{k+1} = \arg \min_{\lambda \in \mathbb{R}^n} \left\{ f^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \right\}$$

$$x_{k+1} = x_k - c_k \lambda_{k+1}$$

- As x_k converges to x^* , the dual sequence λ_k converges to 0 (a subgradient of f at x^*).



- The primal and dual algorithms generate identical sequences $\{x_k, \lambda_k\}$. Which one is preferable depends on whether f or its conjugate f^* has more convenient structure.
- **Special case:** The augmented Lagrangian method.

AUGMENTED LAGRANGIAN METHOD

- Consider the convex constrained problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in X, \quad Ax = b \end{aligned}$$

- Primal and dual functions:

$$p(u) = \inf_{\substack{x \in X \\ Ax - b = u}} f(x), \quad q(\lambda) = \inf_{x \in X} \{f(x) + \lambda'(Ax - b)\}$$

- Assume p : closed, so (q, p) are “conjugate” pair.
- **Primal and dual prox. algorithms for $\max_{\lambda} q(\lambda)$:**

$$\lambda_{k+1} = \arg \max_{\lambda \in \mathbb{R}^m} \left\{ q(\lambda) - \frac{1}{2c_k} \|\lambda - \lambda_k\|^2 \right\}$$

$$u_{k+1} = \arg \min_{u \in \mathbb{R}^m} \left\{ p(u) + \lambda_k' u + \frac{c_k}{2} \|u\|^2 \right\}$$

Dual update: $\lambda_{k+1} = \lambda_k + c_k u_{k+1}$

- Implementation:

$$u_{k+1} = Ax_{k+1} - b, \quad x_{k+1} \in \arg \min_{x \in X} L_{c_k}(x, \lambda_k)$$

where L_c is the **Augmented Lagrangian** function

$$L_c(x, \lambda) = f(x) + \lambda'(Ax - b) + \frac{c}{2} \|Ax - b\|^2$$

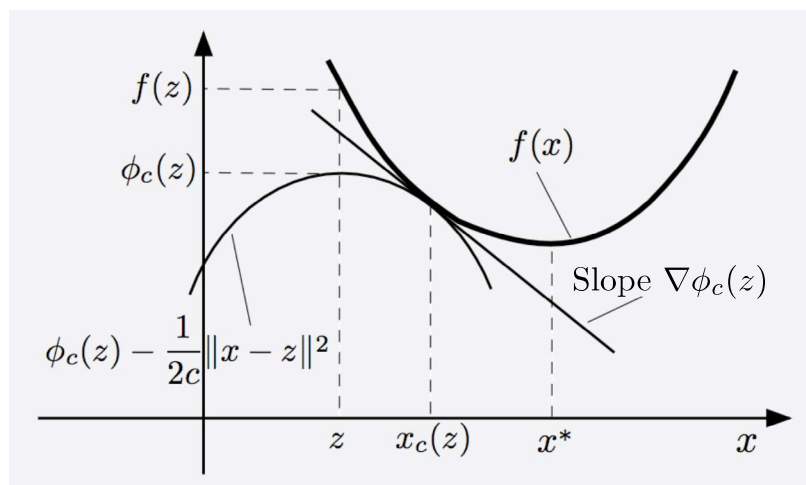
GRADIENT INTERPRETATION

- Back to the dual proximal algorithm and the dual update $\lambda_{k+1} = \frac{x_k - x_{k+1}}{c_k}$
- **Proposition:** λ_{k+1} can be viewed as a gradient,

$$\lambda_{k+1} = \frac{x_k - x_{k+1}}{c_k} = \nabla \phi_{c_k}(x_k),$$

where

$$\phi_c(z) = \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c} \|x - z\|^2 \right\}$$



- So the dual update $x_{k+1} = x_k - c_k \lambda_{k+1}$ can be viewed as a gradient iteration for minimizing $\phi_c(z)$ (which has the same minima as f).
- The gradient is calculated by the dual proximal minimization. Possibilities for faster methods (e.g., Newton, Quasi-Newton). Useful in augmented Lagrangian methods.

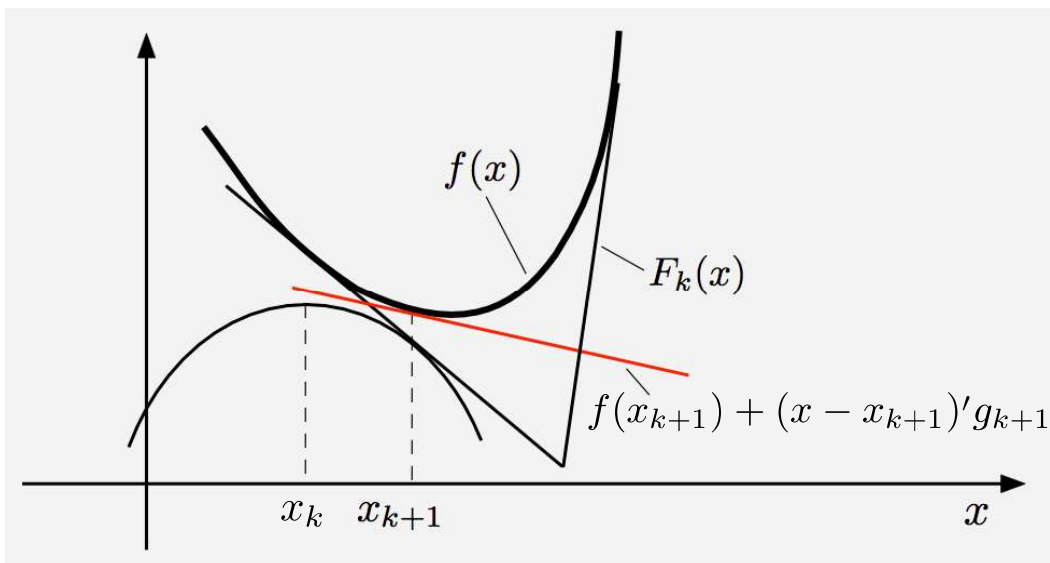
PROXIMAL CUTTING PLANE METHODS

- Same as proximal algorithm, but f is replaced by a cutting plane approximation F_k :

$$x_{k+1} \in \arg \min_{x \in X} \left\{ F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}$$

where

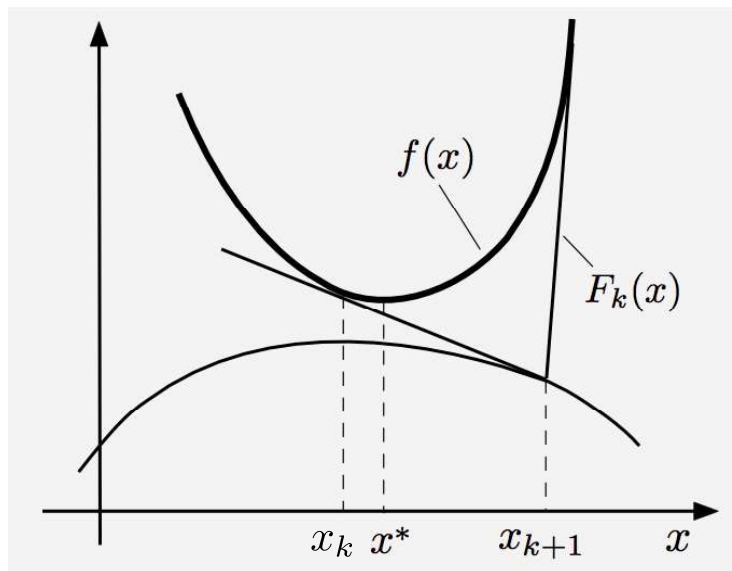
$$F_k(x) = \max \{ f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k \}$$



- Main objective is to reduce instability ... but there are issues to contend with.

DRAWBACKS

- **Stability issue:**
 - For large enough c_k and polyhedral X , x_{k+1} is the exact minimum of F_k over X in a single minimization, so it is identical to the ordinary cutting plane method.



- For small c_k convergence is slow.
- **The number of subgradients used in F_k may become very large;** the quadratic program may become very time-consuming.
- These drawbacks motivate algorithmic variants, called **bundle methods**.

BUNDLE METHODS I

- Replace f with a cutting plane approx. and **change quadratic regularization more conservatively**.
- A general form:

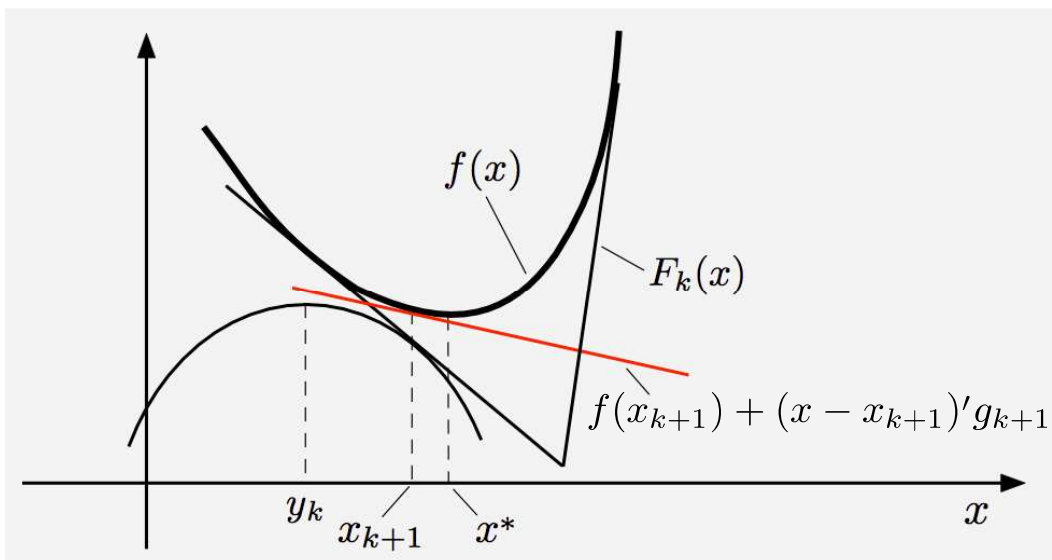
$$x_{k+1} \in \arg \min_{x \in X} \{ F_k(x) + p_k(x) \}$$

$$F_k(x) = \max \{ f(x_0) + (x - x_0)' g_0, \dots, f(x_k) + (x - x_k)' g_k \}$$

$$p_k(x) = \frac{1}{2c_k} \|x - y_k\|^2$$

where c_k is a positive scalar parameter.

- We refer to $p_k(x)$ as the **proximal term**, and to its center y_k as the **proximal center**.



Change y_k in different ways \Rightarrow different methods.

BUNDLE METHODS II

- Allow a proximal center $y_k \neq x_k$:

$$x_{k+1} \in \arg \min_{x \in X} \{ F_k(x) + p_k(x) \}$$

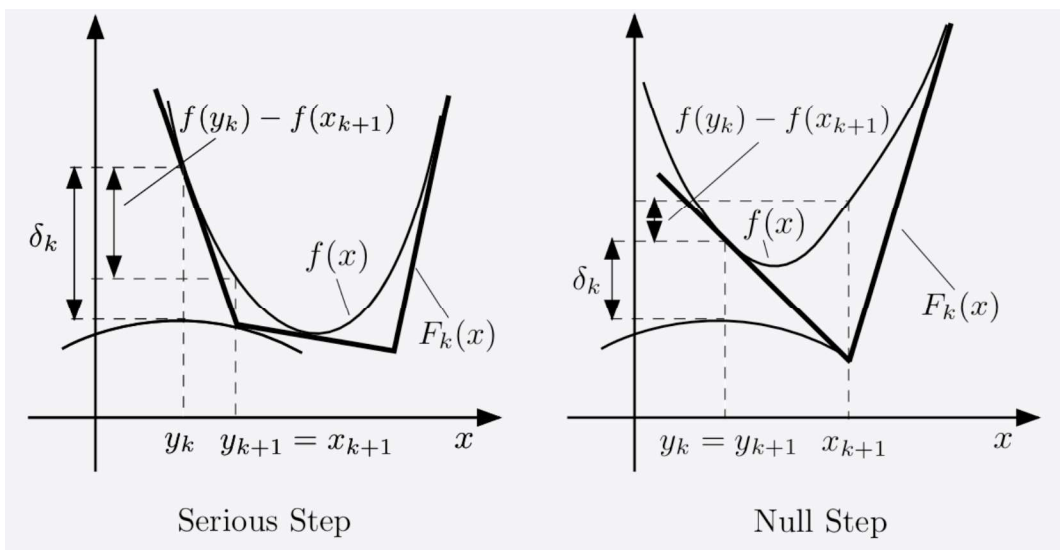
$$F_k(x) = \max \{ f(x_0) + (x - x_0)' g_0, \dots, f(x_k) + (x - x_k)' g_k \}$$

$$p_k(x) = \frac{1}{2c_k} \|x - y_k\|^2$$

- **Null/Serious test** for changing y_k
- **Compare true cost f and proximal cost $F_k + p_k$ reduction in moving from y_k to x_{k+1} , i.e., for some fixed $\beta \in (0, 1)$**

$$y_{k+1} = \begin{cases} x_{k+1} & \text{if } f(y_k) - f(x_{k+1}) \geq \beta \delta_k, \\ y_k & \text{if } f(y_k) - f(x_{k+1}) < \beta \delta_k, \end{cases}$$

$$\delta_k = f(y_k) - (F_k(x_{k+1}) + p_k(x_{k+1})) > 0$$



PROXIMAL LINEAR APPROXIMATION

- **Convex problem:** Min $f : \mathfrak{R}^n \mapsto \mathfrak{R}$ over X .
- **Proximal cutting plane method:** Same as proximal algorithm, but f is replaced by a cutting plane approximation F_k :

$$x_{k+1} \in \arg \min_{x \in \mathfrak{R}^n} \left\{ F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}$$

$$\lambda_{k+1} = \frac{x_k - x_{k+1}}{c_k}$$

where $g_i \in \partial f(x_i)$ for $i \leq k$ and

$$F_k(x) = \max \left\{ f(x_0) + (x - x_0)' g_0, \dots, f(x_k) + (x - x_k)' g_k \right\} + \delta_X(x)$$

- **Proximal simplicial decomposition method** (dual proximal implementation): Let F_k^* be the conjugate of F_k . Set

$$\lambda_{k+1} \in \arg \min_{\lambda \in \mathfrak{R}^n} \left\{ F_k^*(\lambda) - x_k' \lambda + \frac{c_k}{2} \|\lambda\|^2 \right\}$$

$$x_{k+1} = x_k - c_k \lambda_{k+1}$$

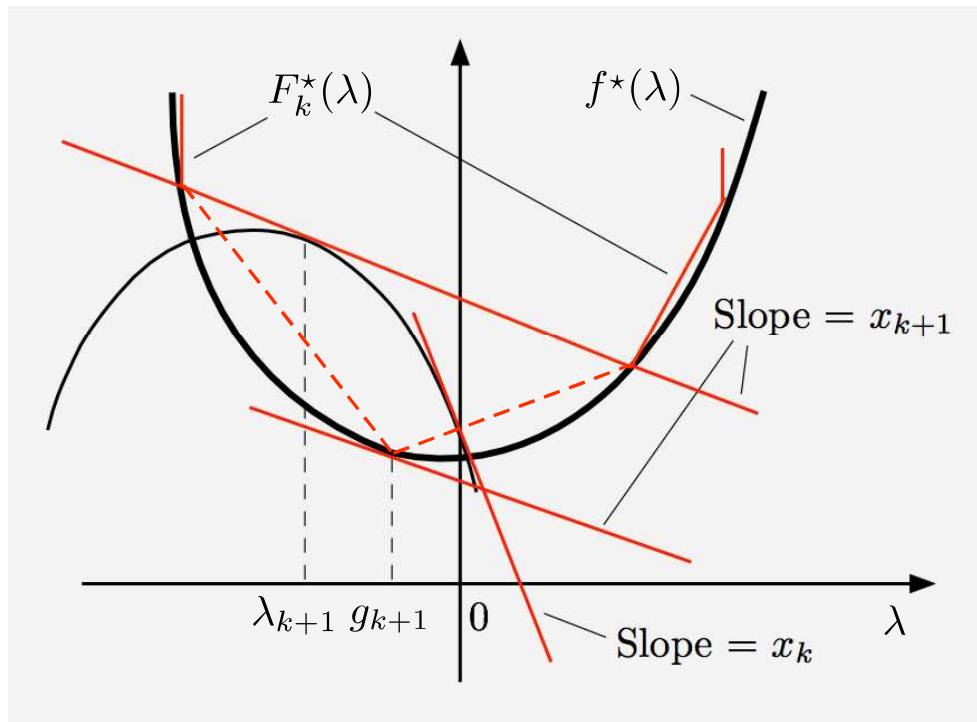
Obtain $g_{k+1} \in \partial f(x_{k+1})$, either directly or via

$$g_{k+1} \in \arg \max_{\lambda \in \mathfrak{R}^n} \left\{ x_{k+1}' \lambda - f^*(\lambda) \right\}$$

- Add g_{k+1} to the outer linearization, or x_{k+1} to the inner linearization, and continue.

PROXIMAL SIMPLICIAL DECOMPOSITION

- It is a mathematical equivalent dual to the proximal cutting plane method.



- Here we use the conjugacy relation between outer and inner linearization.
- Versions of these methods where the proximal center is changed only after some “algorithmic progress” is made:
 - The outer linearization version is the (standard) bundle method.
 - The inner linearization version is an **inner approximation version of a bundle method**.