LECTURE 23

LECTURE OUTLINE

• Interior point methods
• Constrained optimization case - Barrier method
• Conic programming cases
• Linear programming - Path following

All figures are courtesy of Athena Scientific, and are used with permission.
BARRIER METHOD

• Inequality constrained problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in X, \quad g_j(x) \leq 0, \quad j = 1, \ldots, r,
\end{align*}
\]

where \( f \) and \( g_j \) are real-valued convex and \( X \) is closed convex.

• We assume that the interior (relative to \( X \)) set

\[
S = \{x \in X \mid g_j(x) < 0, \quad j = 1, \ldots, r\}
\]

is nonempty.

• Note that because \( S \) is convex, any feasible point can be approached through \( S \) (the Line Segment Principle).

• The barrier method is an approximation method.

• It replaces the indicator function of the constraint set

\[
\delta(x \mid \text{cl}(S))
\]

by a smooth approximation within the relative interior of \( S \).
BARRIER FUNCTIONS

- Consider a barrier function, that is continuous and goes to $\infty$ as any one of the constraints $g_j(x)$ approaches 0 from negative values.

- Examples:

$$B(x) = -\sum_{j=1}^{r} \ln \{-g_j(x)\}, \quad B(x) = -\sum_{j=1}^{r} \frac{1}{g_j(x)}.$$ 

- Barrier method:

$$x^k = \arg \min_{x \in S} \{f(x) + \epsilon_k B(x)\}, \quad k = 0, 1, \ldots,$$

where the parameter sequence $\{\epsilon_k\}$ satisfies $0 < \epsilon_{k+1} < \epsilon_k$ for all $k$ and $\epsilon_k \to 0$. 

![Diagram showing barrier functions and constraints](image-url)
minimize $f(x) = \frac{1}{2} \left( (x^1)^2 + (x^2)^2 \right)$
subject to $2 \leq x^1$,

with optimal solution $x^* = (2, 0)$.

- Logarithmic barrier: $B(x) = -\ln (x^1 - 2)$
- We have $x_k = (1 + \sqrt{1 + \epsilon_k}, 0)$ from
  
  $x_k \in \arg \min \left\{ \frac{1}{2} \left( (x^1)^2 + (x^2)^2 \right) - \epsilon_k \ln (x^1 - 2) \right\}$

- As $\epsilon_k$ is decreased, the unconstrained minimum $x_k$ approaches the constrained minimum $x^* = (2, 0)$.
- As $\epsilon_k \to 0$, computing $x_k$ becomes more difficult because of ill-conditioning (a Newton-like method is essential for solving the approximate problems).
CONVERGENCE

- Every limit point of a sequence \( \{x_k\} \) generated by a barrier method is a minimum of the original constrained problem.

**Proof:** Let \( \{x\} \) be the limit of a subsequence \( \{x_k\}_{k \in K} \). Since \( x_k \in S \) and \( X \) is closed, \( x \) is feasible for the original problem.

If \( x \) is not a minimum, there exists a feasible \( x^* \) such that \( f(x^*) < f(x) \) and therefore also an interior point \( \tilde{x} \in S \) such that \( f(\tilde{x}) < f(x) \). By the definition of \( x_k \),

\[
f(x_k) + \epsilon_k B(x_k) \leq f(\tilde{x}) + \epsilon_k B(\tilde{x}), \quad \forall k,
\]

so by taking limit

\[
f(x) + \liminf_{k \to \infty, k \in K} \epsilon_k B(x_k) \leq f(\tilde{x}) < f(x)
\]

Hence \( \liminf_{k \to \infty, k \in K} \epsilon_k B(x_k) < 0 \).

If \( x \in S \), we have \( \lim_{k \to \infty, k \in K} \epsilon_k B(x_k) = 0 \), while if \( x \) lies on the boundary of \( S \), we have by assumption \( \lim_{k \to \infty, k \in K} B(x_k) = \infty \). Thus

\[
\liminf_{k \to \infty} \epsilon_k B(x_k) \geq 0,
\]

- a contradiction.
SECOND ORDER CONE PROGRAMMING

- Consider the SOCP

\[ \begin{align*}
\text{minimize } & \quad c'x \\
\text{subject to } & \quad A_i x - b_i \in C_i, \quad i = 1, \ldots, m,
\end{align*} \]

where \( x \in \mathbb{R}^n \), \( c \) is a vector in \( \mathbb{R}^n \), and for \( i = 1, \ldots, m \), \( A_i \) is an \( n_i \times n \) matrix, \( b_i \) is a vector in \( \mathbb{R}^{n_i} \), and \( C_i \) is the second order cone of \( \mathbb{R}^{n_i} \).

- We approximate this problem with

\[ \begin{align*}
\text{minimize } & \quad c'x + \epsilon_k \sum_{i=1}^{m} B_i(A_i x - b_i) \\
\text{subject to } & \quad x \in \mathbb{R}^n,
\end{align*} \]

where \( B_i \) is the logarithmic barrier function:

\[ B_i(y) = -\ln \left( y_{n_i}^2 - (y_1^2 + \cdots + y_{n_i-1}^2) \right), \quad y \in \text{int}(C_i), \]

and \( \{\epsilon_k\} \) is a positive sequence with \( \epsilon_k \to 0 \).

- Essential to use Newton’s method to solve the approximating problems.

- Interesting complexity analysis
SEMIDEFINITE PROGRAMMING

• Consider the dual SDP

\[
\text{maximize} \quad b'\lambda \\
\text{subject to} \quad C - (\lambda_1 A_1 + \cdots + \lambda_m A_m) \in D,
\]

where \( D \) is the cone of positive semidefinite matrices.

• The logarithmic barrier method uses approximating problems of the form

\[
\text{maximize} \quad b'\lambda + \epsilon_k \ln (\det(C - \lambda_1 A_1 - \cdots - \lambda_m A_m))
\]

over all \( \lambda \in \mathbb{R}^m \) such that \( C - (\lambda_1 A_1 + \cdots + \lambda_m A_m) \) is positive definite.

• Here \( \epsilon_k > 0 \) and \( \epsilon_k \to 0 \).

• Furthermore, we should use a starting point such that \( C - \lambda_1 A_1 - \cdots - \lambda_m A_m \) is positive definite, and Newton’s method should ensure that the iterates keep \( C - \lambda_1 A_1 - \cdots - \lambda_m A_m \) within the positive definite cone.
Apply logarithmic barrier to the linear program

\[
\begin{align*}
\text{minimize} & \quad c'x \\
\text{subject to} & \quad Ax = b, \quad x \geq 0,
\end{align*}
\]

(LP)

The method finds for various \( \epsilon > 0 \),

\[
x(\epsilon) = \arg\min_{x \in S} F_\epsilon(x) = \arg\min_{x \in S} \left\{ c'x - \epsilon \sum_{i=1}^{n} \ln x_i \right\},
\]

where \( S = \{ x \mid Ax = b, x > 0 \} \). We assume that \( S \) is nonempty and bounded.

- As \( \epsilon \to 0 \), \( x(\epsilon) \) follows the central path

- All central paths start at the analytic center

\[
x_\infty = \arg\min_{x \in S} \left\{ -\sum_{i=1}^{n} \ln x_i \right\},
\]

and end at optimal solutions of (LP).
Newton’s method for minimizing $F_\epsilon$:

$$\tilde{x} = x + \alpha(x - x),$$

where $x$ is the pure Newton iterate

$$x = \arg \min_{Az=b} \left\{ \nabla F_\epsilon(x)'(z - x) + \frac{1}{2}(z - x)'^{\top}\nabla^2 F_\epsilon(x)(z - x) \right\}$$

By straightforward calculation

$$x = x - Xq(x,\epsilon),$$

$$q(x,\epsilon) = \frac{Xz}{\epsilon} - e, \quad e = (1 \ldots 1)', \quad z = c - A'\lambda,$$

$$\lambda = (AX^2A')^{-1}AX(Xc - \epsilon e),$$

and $X$ is the diagonal matrix with $x_i$, $i = 1, \ldots, n$ along the diagonal.

- View $q(x,\epsilon)$ as a “normalized” Newton increment [the Newton increment $(x - x)$ transformed by $X^{-1}$ that maps $x$ into $e$].

- Consider $\|q(x,\epsilon)\|$ as a proximity measure of the current point to the point $x(\epsilon)$ on the central path.
KEY RESULTS

- It is sufficient to minimize $F_\epsilon$ approximately, up to where $\|q(x,\epsilon)\| < 1$.

- **Fact 1:** If $x > 0$, $Ax = b$, and $\|q(x,\epsilon)\| < 1$,

\[
c'x - \min_{Ay = b, y \geq 0} c'y \leq \epsilon(n + \sqrt{n}).
\]

Defines a “tube of convergence”.

- **Fact 2:** The “termination set” $\{x \mid \|q(x,\epsilon)\| < 1\}$ is part of the region of quadratic convergence.

- **Fact 2:** If $\|q(x,\epsilon)\| < 1$, then the pure Newton iterate $x$ satisfies

\[
\|q(x,\epsilon)\| \leq \|q(x,\epsilon)\|^2 < 1.
\]
**SHORT STEP METHODS**

- **Idea:** Use a **single** Newton step before changing \( \epsilon \) (a little bit, so the next point stays within the “tube of convergence”).

**Proposition** Let \( x > 0, \ Ax = b \), and suppose that for some \( \gamma < 1 \) we have \( \|q(x, \epsilon)\| \leq \gamma \). Then if \( \epsilon = (1 - \delta n^{-1/2})\epsilon \) for some \( \delta > 0 \),

\[
\|q(x, \epsilon)\| \leq \frac{\gamma^2 + \delta}{1 - \delta n^{-1/2}}.
\]

In particular, if

\[
\delta \leq \gamma (1 - \gamma)(1 + \gamma)^{-1},
\]

we have \( \|q(x, \epsilon)\| \leq \gamma \).

- Can be used to establish nice complexity results; but \( \epsilon \) must be reduced **VERY** slowly.
LONG STEP METHODS

• Main features:
  − Decrease $\epsilon$ faster than dictated by complexity analysis.
  − Use more than one Newton step per (approximate) minimization.
  − Use line search as in unconstrained Newton’s method.
  − Require much smaller number of (approximate) minimizations.

The methodology generalizes to quadratic programming and convex programming.