Interior Point Methods
for Convex Quadratic
and Convex Nonlinear Programming

Jacek Gondzio
Email: J.Gondzio@ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio
Outline

- **Part 1: IPM for QP**
  - quadratic forms
  - duality in QP
  - first order optimality conditions
  - primal-dual framework

- **Part 2: IPM for NLP**
  - NLP notation
  - Lagrangian
  - first order optimality conditions
  - primal-dual framework

- Algebraic Modelling Languages

- **Self-concordant barrier**
IPM for Convex QP
Convex Quadratic Programs

The quadratic function

\[ f(x) = x^T Q x \]

is convex if and only if the matrix \( Q \) is positive definite. In such case the quadratic programming problem

\[
\begin{align*}
\min & \quad c^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0,
\end{align*}
\]

is well defined.

If there exists a \textit{feasible} solution to it, then there exists an \textit{optimal} solution.
**QP Background:**

**Def.** A matrix $Q \in \mathbb{R}^{n \times n}$ is positive semidefinite if $x^T Q x \geq 0$ for any $x \neq 0$. We write $Q \succeq 0$.

**Def.** A matrix $Q \in \mathbb{R}^{n \times n}$ is positive definite if $x^T Q x > 0$ for any $x \neq 0$. We write $Q \succ 0$.

**Example:**

Consider quadratic functions $f(x) = x^T Q x$ with the following matrices:

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 5 & 4 \\ 4 & 3 \end{bmatrix}.$$

$Q_1$ is positive definite (hence $f_1$ is convex).

$Q_2$ and $Q_3$ are indefinite ($f_2$, $f_3$ are not convex).
QP with IPMs

Consider the convex quadratic programming problem.

The **primal**

\[
\min \quad c^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} \quad A x = b, \\
\quad x \geq 0,
\]

and the **dual**

\[
\max \quad b^T y - \frac{1}{2} x^T Q x \\
\text{s.t.} \quad A^T y + s - Q x = c, \\
\quad x, s \geq 0.
\]

Apply the usual procedure:

- replace inequalities with log barriers;
- form the Lagrangian;
- write the first order optimality conditions;
- apply Newton method to them.
**QP with IPMs: Log Barriers**

Replace the **primal QP**

\[
\begin{align*}
\min & \quad c^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0,
\end{align*}
\]

with the **primal barrier QP**

\[
\begin{align*}
\min & \quad c^T x + \frac{1}{2} x^T Q x - \sum_{j=1}^{n} \ln x_j \\
\text{s.t.} & \quad Ax = b.
\end{align*}
\]
QP with IPMs: Log Barriers

Replace the **dual QP**

\[
\begin{align*}
\text{max} & \quad b^T y - \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad A^T y + s - Q x = c, \\
& \quad y \text{ free, } s \geq 0,
\end{align*}
\]

with the **dual barrier QP**

\[
\begin{align*}
\text{max} & \quad b^T y - \frac{1}{2} x^T Q x + \sum_{j=1}^{n} \ln s_j \\
\text{s.t.} & \quad A^T y + s - Q x = c.
\end{align*}
\]
First Order Optimality Conditions

Consider the **primal barrier quadratic program**

\[
\min \quad c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^{n} \ln x_j \\
\text{s.t.} \quad Ax = b,
\]

where \( \mu \geq 0 \) is a barrier parameter.

Write out the **Lagrangian**

\[
L(x, y, \mu) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j,
\]
First Order Optimality Conditions (cont’d)

The conditions for a stationary point of the Lagrangian:

\[ L(x, y, \mu) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j, \]

are

\[ \nabla_x L(x, y, \mu) = c - A^T y - \mu X^{-1} e + Q x = 0 \]
\[ \nabla_y L(x, y, \mu) = Ax - b = 0, \]

where \( X^{-1} = diag\{x_1^{-1}, x_2^{-1}, \cdots, x_n^{-1}\} \).

Let us denote

\[ s = \mu X^{-1} e, \quad \text{i.e.} \quad XSe = \mu e. \]

The First Order Optimality Conditions are:

\[ Ax = b, \]
\[ A^T y + s - Q x = c, \]
\[ XSe = \mu e. \]
Apply Newton Method to the FOC

The first order optimality conditions for the barrier problem form a large system of nonlinear equations

\[ F(x, y, s) = 0, \]

where \( F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m} \) is an application defined as follows:

\[
F(x, y, s) = \begin{bmatrix}
Ax & -b \\
ATy + s & -Qx - c \\
XSs & -\mu e
\end{bmatrix}.
\]

Actually, the first two terms of it are *linear*; only the last one, corresponding to the complementarity condition, is *nonlinear*. Note that

\[
\nabla F(x, y, s) = \begin{bmatrix}
A & 0 & 0 \\
-Q & AT & I \\
S & 0 & X
\end{bmatrix}.
\]
Newton Method for the FOC (cont’d)

Thus, for a given point \((x, y, s)\) we find the Newton direction \((\Delta x, \Delta y, \Delta s)\) by solving the system of linear equations:

\[
\begin{bmatrix}
A & 0 & 0 \\
-Q & A^T & I \\
S & 0 & X
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta s
\end{bmatrix}
= \begin{bmatrix}
b - Ax \\
c - A^T y - s + Qx \\
\mu e - XSe
\end{bmatrix}.
\]
Interior-Point QP Algorithm

Initialize
\[ k = 0, \quad (x^0, y^0, s^0) \in \mathcal{F}^0, \quad \mu_0 = \frac{1}{n} \cdot (x^0)^T s^0, \quad \alpha_0 = 0.9995 \]

Repeat until optimality
\[ k = k + 1 \]
\[ \mu_k = \sigma \mu_{k-1}, \text{ where } \sigma \in (0, 1) \]
\[ \Delta = \text{Newton direction towards } \mu\text{-center} \]

**Ratio test:**
\[ \alpha_P := \max \{ \alpha > 0 : x + \alpha \Delta x \geq 0 \}, \]
\[ \alpha_D := \max \{ \alpha > 0 : s + \alpha \Delta s \geq 0 \}. \]

**Make step:**
\[ x^{k+1} = x^k + \alpha_0 \alpha_P \Delta x, \]
\[ y^{k+1} = y^k + \alpha_0 \alpha_D \Delta y, \]
\[ s^{k+1} = s^k + \alpha_0 \alpha_D \Delta s. \]
From LP to QP

QP problem

\[
\begin{align*}
\min & \quad c^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \geq 0.
\end{align*}
\]

First order conditions (for barrier problem)

\[
\begin{align*}
Ax & = b, \\
A^T y + s - Q x & = c, \\
X S e & = \mu e.
\end{align*}
\]
IPMs for Convex NLP
Convex Nonlinear Optimization

Consider the nonlinear optimization problem

$$\min \ f(x)$$
$$\text{s.t.} \ g(x) \leq 0,$$

where $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ are convex, twice differentiable.

**Assumptions:**

- $f$ and $g$ are convex
  - ⇒ If there exists a **local** minimum then it is a **global** one.
- $f$ and $g$ are twice differentiable
  - ⇒ We can use the **second order Taylor approximations**.

Some additional (technical) conditions

- ⇒ We need them to prove that the point which satisfies the first order optimality conditions is the optimum. **We won’t use them in this course.**
Taylor Expansion of $f : \mathbb{R} \mapsto \mathbb{R}$

Let $f : \mathbb{R} \mapsto \mathbb{R}$.
If all derivatives of $f$ are continuously differentiable at $x_0$, then

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k,$$

where $f^{(k)}(x_0)$ is the $k$-th derivative of $f$ at $x_0$.

The first order approximation of the function:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + r_2(x - x_0),$$

where the remainder satisfies:

$$\lim_{x \to x_0} \frac{r_2(x - x_0)}{x - x_0} = 0.$$
Taylor Expansion (cont’d)

The second order approximation:

\[ f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + r_3(x - x_0), \]

where the remainder satisfies:

\[ \lim_{x \to x_0} \frac{r_3(x - x_0)}{(x - x_0)^2} = 0. \]
Derivatives of $f : \mathbb{R}^n \mapsto \mathbb{R}$

The vector

$$(\nabla f(x))^T = \left( \frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right)$$

is called the **gradient** of $f$ at $x$.

The matrix

$$\nabla^2 f(x) = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\
\frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\
\cdots & \cdots & \cdots & \cdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x)
\end{bmatrix}$$

is called the **Hessian** of $f$ at $x$. 

Paris, January 2018
Taylor Expansion of $f : \mathbb{R}^n \mapsto \mathbb{R}$

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$.
If all derivatives of $f$ are continuously differentiable at $x_0$, then

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k,$$

where $f^{(k)}(x_0)$ is the $k$-th derivative of $f$ at $x_0$.

The first order approximation of the function:

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + r_2(x - x_0),$$

where the remainder satisfies:

$$\lim_{x \to x_0} \frac{r_2(x - x_0)}{\|x - x_0\|} = 0.$$
Taylor Expansion (cont’d)

The second order approximation of the function:

\[ f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0) + r_3(x - x_0), \]

where the remainder satisfies:

\[ \lim_{x \to x_0} \frac{r_3(x - x_0)}{\|x - x_0\|^2} = 0. \]
Convexity: Reminder

Property 1. For any collection \( \{ C_i \mid i \in I \} \) of convex sets, the intersection \( \bigcap_{i \in I} C_i \) is convex.

Property 4. If \( C \) is a convex set and \( f : C \rightarrow \mathbb{R} \) is convex function, the level sets \( \{ x \in C \mid f(x) \leq \alpha \} \) and \( \{ x \in C \mid f(x) < \alpha \} \) are convex for all scalars \( \alpha \).

**Lemma 1:** If \( g : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is a convex function, then the set \( \{ x \in \mathbb{R}^n \mid g(x) \leq 0 \} \) is convex.

**Proof:** Since every function \( g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, ..., m \) is convex, from Property 4, we conclude that every set \( X_i = \{ x \in \mathbb{R}^n \mid g_i(x) \leq 0 \} \) is convex. Next from Property 1, we deduce that the intersection

\[
X = \bigcap_{i=1}^{m} X_i = \{ x \in \mathbb{R}^n \mid g(x) \leq 0 \}
\]

is convex, which completes the proof.

Paris, January 2018
Differentiable Convex Functions

Property 8. Let $C \in \mathbb{R}^n$ be a convex set and $f : C \mapsto \mathbb{R}$ be twice continuously differentiable over $C$.

(a) If $\nabla^2 f(x)$ is positive semidefinite for all $x \in C$, then $f$ is convex.

(b) If $\nabla^2 f(x)$ is positive definite for all $x \in C$, then $f$ is strictly convex.

(c) If $f$ is convex, then $\nabla^2 f(x)$ is positive semidefinite for all $x \in C$.

Let the second order approximation of the function be given:

$$f(x) \approx f(x_0) + c^T(x - x_0) + \frac{1}{2}(x - x_0)^T Q(x - x_0),$$

where $c = \nabla f(x_0)$ and $Q = \nabla^2 f(x_0)$.

From Property 8, it follows that when $f$ is convex and twice differentiable, then $Q$ exists and is a positive semidefinite matrix.

**Conclusion:**

If $f$ is convex and twice differentiable, then optimization of $f(x)$ can (locally) be replaced with the minimization of its quadratic model.
Nonlinear Optimization with IPMs

Nonlinear Optimization via QPs:
Sequential Quadratic Programming (SQP).
Repeat until optimality:
  • approximate NLP (locally) with a QP;
  • solve (approximately) the QP.

Nonlinear Optimization with IPMs:
works similarly to SQP scheme.
However, the (local) QP approximations are not solved to optimality. Instead, only one step in the Newton direction corresponding to a given QP approximation is made and the new QP approximation is computed.
Nonlinear Optimization with IPMs

Derive an IPM for NLP:

- replace inequalities with log barriers;
- form the Lagrangian;
- write the first order optimality conditions;
- apply Newton method to them.
NLP Notation

Consider the nonlinear optimization problem

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,$$

where \(x \in \mathbb{R}^n\), and \(f : \mathbb{R}^n \mapsto \mathbb{R}\) and \(g : \mathbb{R}^n \mapsto \mathbb{R}^m\) are convex, twice differentiable.

The vector-valued function \(g : \mathbb{R}^n \mapsto \mathbb{R}^m\) has a derivative

$$A(x) = \nabla g(x) = \begin{bmatrix} \frac{\partial g_i}{\partial x_j} \end{bmatrix}_{i=1..m, j=1..n} \in \mathbb{R}^{m \times n}$$

which is called the **Jacobian** of \(g\).
NLP Notation (cont’d)

The Lagrangian associated with the NLP is:

\[ \mathcal{L}(x, y) = f(x) + y^T g(x), \]

where \( y \in \mathbb{R}^m, y \geq 0 \) are Lagrange multipliers (dual variables).

The first derivatives of the Lagrangian:

\[
\begin{align*}
\nabla_x \mathcal{L}(x, y) &= \nabla f(x) + \nabla g(x)^T y \\
\nabla_y \mathcal{L}(x, y) &= g(x).
\end{align*}
\]

The Hessian of the Lagrangian, \( Q(x, y) \in \mathbb{R}^{n \times n} \):

\[
Q(x, y) = \nabla^2_{xx} \mathcal{L}(x, y) = \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x).
\]
Convexity in NLP

**Lemma 2:** If $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ are convex, twice differentiable, then the **Hessian** of the Lagrangian

$$Q(x, y) = \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x)$$

is positive semidefinite for any $x$ and any $y \geq 0$. If $f$ is strictly convex, then $Q(x, y)$ is positive definite for any $x$ and any $y \geq 0$.

**Proof:** Using Property 8, the convexity of $f$ implies that $\nabla^2 f(x)$ is positive semidefinite for any $x$. Similarly, the convexity of $g$ implies that for all $i = 1, 2, ..., m$, $\nabla^2 g_i(x)$ is positive semidefinite for any $x$. Since $y_i \geq 0$ for all $i = 1, 2, ..., m$ and $Q(x, y)$ is the sum of positive semidefinite matrices, we conclude that $Q(x, y)$ is positive semidefinite.

If $f$ is strictly convex, then $\nabla^2 f(x)$ is positive definite and so is $Q(x, y)$. 

Paris, January 2018
IPM for NLP

Add slack variables to nonlinear inequalities:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad g(x) + z = 0 \\
& \quad z \geq 0,
\end{align*}
\]

where \( z \in \mathbb{R}^m \). Replace inequality \( z \geq 0 \) with the logarithmic barrier:

\[
\begin{align*}
\min & \quad f(x) - \mu \sum_{i=1}^{m} \ln z_i \\
\text{s.t.} & \quad g(x) + z = 0.
\end{align*}
\]

Write out the **Lagrangian**

\[
L(x, y, z, \mu) = f(x) + y^T(g(x) + z) - \mu \sum_{i=1}^{m} \ln z_i,
\]
IPM for NLP

For the Lagrangian

\[ L(x, y, z, \mu) = f(x) + y^T(g(x) + z) - \mu \sum_{i=1}^{m} \ln z_i, \]

write the conditions for a stationary point

\[ \nabla_x L(x, y, z, \mu) = \nabla f(x) + \nabla g(x)^T y = 0 \]
\[ \nabla_y L(x, y, z, \mu) = g(x) + z = 0 \]
\[ \nabla_z L(x, y, z, \mu) = y - \mu Z^{-1} e = 0, \]

where \( Z^{-1} = diag\{z_1^{-1}, z_2^{-1}, \ldots, z_m^{-1}\} \).

The First Order Optimality Conditions are:

\[ \nabla f(x) + \nabla g(x)^T y = 0, \]
\[ g(x) + z = 0, \]
\[ Y Z e = \mu e. \]
Newton Method for the FOC

The first order optimality conditions for the barrier problem form a large system of nonlinear equations

\[ F(x, y, z) = 0, \]

where \( F : \mathbb{R}^{n+2m} \rightarrow \mathbb{R}^{n+2m} \) is an application defined as follows:

\[
F(x, y, z) = \begin{bmatrix}
\nabla f(x) + \nabla g(x)^T y \\
g(x) + z \\
YZe - \mu e
\end{bmatrix}.
\]

Note that all three terms of it are nonlinear.
(In LP and QP the first two terms were linear.)
Newton Method for the FOC

Observe that

$$\nabla F(x, y, z) = \begin{bmatrix} Q(x, y) & A(x)^T & 0 \\ A(x) & 0 & I \\ 0 & Z & Y \end{bmatrix},$$

where $A(x)$ is the Jacobian of $g$ and $Q(x, y)$ is the Hessian of $L$.

They are defined as follows:

\[ A(x) = \nabla g(x) \in \mathbb{R}^{m \times n} \]
\[ Q(x, y) = \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x) \in \mathbb{R}^{n \times n} \]
Newton Method (cont’d)

For a given point \((x, y, z)\) we find the Newton direction \((\Delta x, \Delta y, \Delta z)\) by solving the system of linear equations:

\[
\begin{bmatrix}
Q(x, y) & A(x)^T & 0 \\
A(x) & 0 & I \\
0 & Z & Y
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{bmatrix}
= 
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - z \\
\mu e - Y Ze
\end{bmatrix}.
\]

Using the third equation we eliminate

\[
\Delta z = \mu Y^{-1} e - Ze - ZY^{-1} \Delta y,
\]

from the second equation and get

\[
\begin{bmatrix}
Q(x, y) & A(x)^T \\
A(x) & -ZY^{-1}
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
= 
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - \mu Y^{-1} e
\end{bmatrix}.
\]
Interior-Point NLP Algorithm

Initialize
\[ k = 0 \]
\[ (x^0, y^0, z^0) \text{ such that } y^0 > 0 \text{ and } z^0 > 0, \quad \mu_0 = \frac{1}{m} \cdot (y^0)^T z^0 \]

Repeat until optimality
\[ k = k + 1 \]
\[ \mu_k = \sigma \mu_{k-1}, \text{ where } \sigma \in (0, 1) \]
Compute \( A(x) \text{ and } Q(x,y) \)
\( \Delta = \text{Newton direction towards } \mu\text{-center} \)

Ratio test:
\[ \alpha_1 := \max \{ \alpha > 0 : y + \alpha \Delta y \geq 0 \}, \]
\[ \alpha_2 := \max \{ \alpha > 0 : z + \alpha \Delta z \geq 0 \}. \]

Choose the step: (use trust region or line search) \( \alpha \leq \min \{ \alpha_1, \alpha_2 \}. \)

Make step:
\[ x^{k+1} = x^k + \alpha \Delta x, \]
\[ y^{k+1} = y^k + \alpha \Delta y, \]
\[ z^{k+1} = z^k + \alpha \Delta z. \]
From QP to NLP

Newton direction for QP

\[
\begin{bmatrix}
-Q & A^T & I \\
A & 0 & 0 \\
S & 0 & X \\
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta s \\
\end{bmatrix}
= 
\begin{bmatrix}
\xi_d \\
\xi_p \\
\xi_\mu \\
\end{bmatrix}.
\]

Augmented system for QP

\[
\begin{bmatrix}
-Q - SX^{-1} & A^T \\
A & 0 \\
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\end{bmatrix}
= 
\begin{bmatrix}
\xi_d - X^{-1}\xi_\mu \\
\xi_p \\
\end{bmatrix}.
\]
From QP to NLP

Newton direction for NLP

\[
\begin{bmatrix}
Q(x, y) & A(x)^T & 0 \\
A(x) & 0 & I \\
0 & Z & Y
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{bmatrix}
= 
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - z \\
\mu e - Y Ze
\end{bmatrix}.
\]

Augmented system for NLP

\[
\begin{bmatrix}
Q(x, y) & A(x)^T \\
A(x) & -ZY^{-1}
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
= 
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - \mu Y^{-1} e
\end{bmatrix}.
\]

Conclusion:
NLP is a natural extension of QP.
Linear Algebra in IPM for NLP

Newton direction for NLP

\[
\begin{bmatrix}
Q(x, y) & A(x)^T & 0 \\
A(x) & 0 & I \\
0 & Z & Y
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{bmatrix}
=
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - z \\
\mu e - Y Z e
\end{bmatrix}.
\]

The corresponding augmented system

\[
\begin{bmatrix}
Q(x, y) & A(x)^T \\
A(x) & -Z Y^{-1}
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
=
\begin{bmatrix}
-\nabla f(x) - A(x)^T y \\
-g(x) - \mu Y^{-1} e
\end{bmatrix}.
\]

where \( A(x) \in \mathbb{R}^{m \times n} \) is the Jacobian of \( g \)
and \( Q(x, y) \in \mathbb{R}^{n \times n} \) is the Hessian of \( \mathcal{L} \)

\[
A(x) = \nabla g(x)
\]

\[
Q(x, y) = \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x)
\]
Automatic differentiation is very useful ... get $Q(x, y)$ and $A(x)$ from Algebraic Modeling Language.
Automatic Differentiation

AD in the Internet:

- ADIFOR (FORTRAN code for AD):
  http://www-unix.mcs.anl.gov/autodiff/ADIFOR/

- ADOL-C (C/C++ code for AD):

- AD page at Cornell:
  http://www.tc.cornell.edu/~averma/AD/
**IPMs: Remarks**

- Interior Point Methods provide the unified framework for convex optimization.
- IPMs provide polynomial algorithms for LP, QP and NLP.
- The linear algebra in LP, QP and NLP is very similar.
- Use IPMs to solve very large problems.

**Further Extensions:**
- **Nonconvex** optimization.

**IPMs in the Internet:**
- LP FAQ (Frequently Asked Questions):
- Interior Point Methods On-Line:
- NEOS (Network Enabled Optimization Services):
Newton Method
and Self-concordant Barriers
Another View of Newton M. for Optimization

Newton Method for Optimization

Let \( f : \mathbb{R}^n \mapsto \mathbb{R} \) be a twice continuously differentiable function. Suppose we build a quadratic model \( \tilde{f} \) of \( f \) around a given point \( x^k \), i.e., we define \( \Delta x = x - x^k \) and write:

\[
\tilde{f}(x) = f(x^k) + \nabla f(x^k)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^k) \Delta x
\]

Now we **optimize the model** \( \tilde{f} \) instead of **optimizing** \( f \). A minimum (or, more generally, a stationary point) of the quadratic model satisfies:

\[
\nabla \tilde{f}(x) = \nabla f(x^k) + \nabla^2 f(x^k) \Delta x = 0,
\]

i.e.

\[
\Delta x = x - x^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k),
\]

which reduces to the usual equation:

\[
x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k).
\]
Consider the **primal barrier linear program**

\[
\begin{align*}
\min & \quad c^T x - \mu \sum_{j=1}^{n} \ln x_j \\
\text{s.t.} & \quad Ax = b,
\end{align*}
\]

where \( \mu \geq 0 \) is a barrier parameter.

Write out the **Lagrangian**

\[
L(x, y, \mu) = c^T x - y^T (Ax - b) - \mu \sum_{j=1}^{n} \ln x_j,
\]

and the conditions for a stationary point

\[
\begin{align*}
\nabla_x L(x, y, \mu) &= c - A^T y - \mu X^{-1} e = 0 \\
\nabla_y L(x, y, \mu) &= Ax - b = 0,
\end{align*}
\]

where \( X^{-1} = diag\{x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\} \).
\( -\log x \) Barrier Function (cont’d)

Let us denote

\[
s = \mu X^{-1}e, \quad \text{i.e.} \quad XSe = \mu e.\]

The First Order Optimality Conditions are:

\[
Ax = b, \\
A^T y + s = c, \\
XSe = \mu e.\]
**- log x**  \textbf{bf: Newton Method}

The first order optimality conditions for the barrier problem form a large system of nonlinear equations

\[ F(x, y, s) = 0, \]

where \( F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m} \) is an application defined as follows:

\[
F(x, y, s) = \begin{bmatrix}
A x - b \\
A^T y + s - c \\
X Se - \mu e
\end{bmatrix}.
\]

Actually, the first two terms of it are \textit{linear}; only the last one, corresponding to the complementarity condition, is \textit{nonlinear}.

Note that

\[
\nabla F(x, y, s) = \begin{bmatrix}
A & 0 & 0 \\
0 & A^T & I \\
S & 0 & X
\end{bmatrix}.
\]
Thus, for a given point \((x, y, s)\) we find the Newton direction \((\Delta x, \Delta y, \Delta s)\) by solving the system of linear equations:

\[
\begin{bmatrix}
A & 0 & 0 \\
0 & A^T & I \\
S & 0 & X \\
\end{bmatrix} \cdot \begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta s \\
\end{bmatrix} = \begin{bmatrix}
b - Ax \\
c - A^Ty - s \\
\mu e - XSe \\
\end{bmatrix}.
\]
1/$x^\alpha$, $\alpha > 0$  \textbf{Barrier Function}

Consider the \textbf{primal barrier linear program}

$$\min c^T x - \mu \sum_{j=1}^{n} \frac{1}{x_j^\alpha} \quad \text{s.t.} \quad Ax = b,$$

where $\mu \geq 0$ is a barrier parameter and $\alpha > 0$.

Write out the \textbf{Lagrangian}

$$L(x, y, \mu) = c^T x - y^T (Ax - b) + \mu \sum_{j=1}^{n} \frac{1}{x_j^\alpha},$$

and the conditions for a stationary point

$$\nabla_x L(x, y, \mu) = c - A^T y - \mu \alpha X^{-\alpha-1} e = 0,$$
$$\nabla_y L(x, y, \mu) = Ax - b = 0,$$

where $X^{-\alpha-1} = diag\{x_1^{-\alpha-1}, x_2^{-\alpha-1}, \ldots, x_n^{-\alpha-1}\}$. 

Paris, January 2018
Let us denote

$$s = \mu \alpha X^{-\alpha-1}e,$$

i.e.

$$X^{\alpha+1}Se = \mu \alpha e.$$

The **First Order Optimality Conditions** are:

$$Ax = b,$$

$$A^Ty + s = c,$$

$$X^{\alpha+1}Se = \mu \alpha e.$$
The first order optimality conditions for the barrier problem are

\[ F(x, y, s) = 0, \]

where \( F : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m} \) is an application defined as follows:

\[
F(x, y, s) = \begin{bmatrix}
Ax - b \\
ATy + s - c \\
X^{\alpha+1}Se - \mu \alpha e
\end{bmatrix}.
\]

As before, only the last term, corresponding to the complementarity condition, is \textit{nonlinear}.

Note that

\[
\nabla F(x, y, s) = \begin{bmatrix}
A & 0 & 0 \\
0 & A^T & I \\
(\alpha + 1)X^{\alpha}S & 0 & X^{\alpha+1}
\end{bmatrix}.
\]
Thus, for a given point \((x, y, s)\) we find the Newton direction \((\Delta x, \Delta y, \Delta s)\) by solving the system of linear equations:

\[
\begin{bmatrix}
A & 0 & 0 \\
0 & A^T & I \\
(\alpha + 1)XX^\alpha & 0 & X^{\alpha+1}
\end{bmatrix}
\cdot
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta s
\end{bmatrix}
=
\begin{bmatrix}
b - Ax \\
c - A^Ty - s \\
\mu\alpha e - X^{\alpha+1}Se
\end{bmatrix}.
\]
$e^{1/x}$ Barrier Function

Consider the **primal barrier linear program**

$$
\min c^T x - \mu \sum_{j=1}^{n} e^{1/x_j} \quad \text{s.t.} \quad Ax = b,
$$

where $\mu \geq 0$ is a barrier parameter.

Write out the **Lagrangian**

$$
L(x, y, \mu) = c^T x - y^T (Ax - b) + \mu \sum_{j=1}^{n} e^{1/x_j},
$$

and the conditions for a stationary point

$$
\nabla_x L(x, y, \mu) = c - A^T y - \mu X^{-2} \exp(X^{-1}) e = 0
$$

$$
\nabla_y L(x, y, \mu) = Ax - b = 0,
$$

where $\exp(X^{-1}) = \text{diag}\{e^{1/x_1}, e^{1/x_2}, \ldots, e^{1/x_n}\}$. 

Paris, January 2018
$e^{1/x}$  **Barrier Function**

Let us denote

$$s = \mu X^{-2} \exp(X^{-1})e, \text{ i.e. } X^2 \exp(-X^{-1})Se = \mu e.$$  

The **First Order Optimality Conditions** are:

$$Ax = b,$$

$$A^T y + s = c,$$

$$X^2 \exp(-X^{-1})Se = \mu e.$$
The first order optimality conditions are
\[ F(x, y, s) = 0, \]
where \( F : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m} \) is defined as follows:
\[
F(x, y, s) = \begin{bmatrix}
Ax - b \\
ATy + s - c \\
X^2\exp(-X^{-1})Se - \mu e
\end{bmatrix}.
\]
As before, only the last term, corresponding to the complementarity condition, is nonlinear.

Note that
\[
\nabla F(x, y, s) = \begin{bmatrix}
A \\
0 \\
(2X + I)\exp(-X^{-1})
\end{bmatrix} \begin{bmatrix}
0 \\
AT \\
0 \exp(-X^{-1})
\end{bmatrix}.
\]
Newton direction \((\Delta x, \Delta y, \Delta s)\) solves the following system of linear equations:

\[
\begin{bmatrix}
A & 0 & 0 \\
0 & A^T & I \\
(2X + I)\exp(-X^{-1})S & 0 & X^2\exp(-X^{-1})
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta s
\end{bmatrix}
= \begin{bmatrix}
b - Ax \\
c - A^T y - s \\
\mu e - X^2\exp(-X^{-1})S e
\end{bmatrix}.
\]
Why Log Barrier is the Best?

The First Order Optimality Conditions:

- \( - \log x : \ XSe = \mu e, \)
- \( 1/x^\alpha : \ X^{\alpha+1}Se = \mu \alpha e, \)
- \( e^{1/x} : \ X^2 \exp(-X^{-1})Se = \mu e. \)

Log Barrier ensures the symmetry between the primal and the dual.

3rd row in the Newton Equation System:

- \( - \log x : \nabla F_3 = [S, 0, X], \)
- \( 1/x^\alpha : \nabla F_3 = [(\alpha + 1)X^\alpha S, 0, X^{\alpha+1}] \)
- \( e^{1/x} : \nabla F_3 = [(2X+I)\exp(-X^{-1})S, 0, X^2 \exp(-X^{-1})] \)

Log Barrier produces 'the weakest nonlinearity'.

Paris, January 2018
Self-concordant Functions

There is a nice property of the function that is responsible for a good behaviour of the Newton method.

**Def** Let $C \in \mathbb{R}^n$ be an open nonempty convex set.

Let $f : C \mapsto \mathbb{R}$ be a three times continuously differentiable convex function.

A function $f$ is called **self-concordant** if there exists a constant $p > 0$ such that

$$|\nabla^3 f(x)[h, h, h]| \leq 2p^{-1/2}(\nabla^2 f(x)[h, h])^{3/2},$$

$\forall x \in C$, $\forall h : x + h \in C$.

(We then say that $f$ is $p$-self-concordant).

Note that a self-concordant function is always well approximated by the quadratic model because the error of such an approximation can be bounded by the $3/2$ power of $\nabla^2 f(x)[h, h]$. 
Self-concordant Barriers

Lemma
The barrier function \(- \log x\) is self-concordant on \(\mathcal{R}_+\).

Proof Consider \(f(x) = - \log x\).
Compute
\[ f'(x) = -x^{-1}, \quad f''(x) = x^{-2} \quad \text{and} \quad f'''(x) = -2x^{-3} \]
and check that the self-concordance condition is satisfied for \(p = 1\).

Lemma
The barrier function \(1/x^\alpha\), with \(\alpha \in (0, \infty)\) is not self-concordant on \(\mathcal{R}_+\).

Lemma
The barrier function \(e^{1/x}\) is not self-concordant on \(\mathcal{R}_+\).

Use self-concordant barriers in optimization