Numerical Techniques in Interior Point Methods for Optimization

Jacek Gondzio
Dept. of Maths & Stats, Univ. of Edinburgh
Email: gondzio@maths.ed.ac.uk
URL: http://www.maths.ed.ac.uk/~gondzio

Peschici, September, 2001

Two Parts:

- Interior Point Methods (IPMs) for Linear, Quadratic and Nonlinear Programs.
- Numerical Techniques in IPMs.

- What's wrong with the simplex method?
- Stay in the interior: use logarithmic barriers.
- Proceed against the common sense: use **nonlinear** methodology to solve linear problems.
- Polynomial complexity: $\mathcal{O}(\sqrt{n})$ iterations to reach the optimality.
- Unified view of convex optimization: from LP via QP to NLP.

The Simplex Method

Consider an LP

$$\begin{array}{rl} \text{min} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array}$$

where $c, x \in \mathcal{R}^n, b \in \mathcal{R}^m,\ A \in \mathcal{R}^{m \times n}$.
Matrix A has full row rank, m $(m \leq n)$.

Partition the LP constraint matrix as $A = [B, N]$, where $B \in \mathcal{R}^{m \times m}$ is a nonsingular matrix and $N \in \mathcal{R}^{m \times (n-m)}$.

A vertex in $\mathcal{R}^n$ is defined by a set of n equations:

$$\begin{bmatrix} B & N \\ 0 & I_{n-m} \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Non-basic variables fixed on zero $x_N = 0$.
Basic variables, x_B allowed to be non-zero.

There are

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

bases.
This is a huge number!

What's wrong with the Simplex Method?

The simplex method can make a non-polynomial number of iterations to reach the optimality:
V. Klee and G. Minty gave an example LP the solution of which needs 2^n iterations:
How good is the simplex algorithm, in Inequalities-III, O. Shisha, ed., Acad. Press, 1972, 159–175.

Narendra Karmarkar from AT&T Bell Labs:

"the simplex [method] is complex"

N. Karmarkar:
A New Polynomial–time Algorithm for Linear Programming, *Combinatorica* 4 (1984) 373–395.

What do we need
to derive the **Interior Point Method**?

- logarithmic barriers.
- duality theory:
  Lagrangian function;
  first order optimality conditions.
- Newton method.

Consider the **primal** linear program

$$\begin{array}{rrcl} \min & c^T x & & \\ \text{s.t.} & Ax & = & b, \\ & x \geq 0, & & \end{array} \qquad (1)$$

and the **dual** linear program

$$\begin{array}{rrcccl} \max & b^T y & & & & \\ \text{s.t.} & A^T y & + & s & = & c, \\ & y \text{ free}, & & s \geq 0, & & \end{array} \qquad (2)$$

where $b, y \in \mathcal{R}^m$, $c, x, s \in \mathcal{R}^n$ and $A \in \mathcal{R}^{m \times n}$.

Let $\mathcal{P}$, $\mathcal{D}$ be the feasible sets of (1) and (2)
$\mathcal{P} = \{x \in \mathcal{R}^n \,|\, Ax = b,\, x \geq 0\}$,
$\mathcal{D} = \{y \in \mathcal{R}^m, s \in \mathcal{R}^n \,|\, A^T y + s = c,\, s \geq 0\}$.
Let us introduce the convention that
$\inf_{x \in \mathcal{P}} c^T x = +\infty$, if $\mathcal{P} = \emptyset$; $\sup_{y \in \mathcal{D}} b^T y = -\infty$, if $\mathcal{D} = \emptyset$.

Weak Duality Theorem

$$\inf_{x \in \mathcal{P}} c^T x \geq \sup_{y \in \mathcal{D}} b^T y.$$

Strong Duality Theorem

If either $\mathcal{P} \neq \emptyset$ or $\mathcal{D} \neq \emptyset$ then

$$\inf_{x \in \mathcal{P}} c^T x = \sup_{y \in \mathcal{D}} b^T y.$$

If one of problems (1) and (2) is *solvable* then

$$\min_{x \in \mathcal{P}} c^T x = \max_{y \in \mathcal{D}} b^T y.$$

Consider the **primal-dual pair**:

$$\begin{array}{llll} \text{Primal} & & \text{Dual} & \\ \min & c^T x & \max & b^T y \\ \text{s.t.} & Ax = b, & \text{s.t.} & A^T y + s = c, \\ & x \geq 0; & & s \geq 0. \end{array}$$

Lagrangian

$$L(x, y) = c^T x - y^T (Ax - b).$$

Optimality Conditions in LP

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s & = & c, \\ XSe & = & 0, \\ x, s & \geq & 0, \end{array}$$

where $X = diag\{x_1, \cdots, x_n\}$, $S = diag\{s_1, \cdots, s_n\}$ and $e = (1, 1, \cdots, 1) \in \mathcal{R}^n$.

Notation:

$$X = diag\{x_1, x_2, \cdots, x_n\} = \begin{bmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{bmatrix}.$$

$X^{-1} = diag\{x_1^{-1}, x_2^{-1}, \cdots, x_n^{-1}\}$.

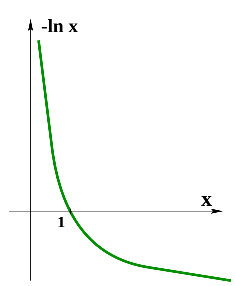
An equation $XSe = \mu e$,
is equivalent to $x_j s_j = \mu$, $\forall j = 1, 2, \cdots, n$.

Log barriers keep the point in the interior

The following **logarithmic barrier**

$$-\ln x_j$$

added to the objective in the optimization problem prevents variable x_j from approaching zero.



In other words, the logarithmic barrier can be used to "replace" the inequality

$$x_j \geq 0.$$

Observe that

$$\min\, e^{-\sum_{j=1}^n \ln x_j} \iff \max \prod_{j=1}^n x_j$$

The minimization of $-\sum_{j=1}^n \ln x_j$ is equivalent to the maximization of the product of distances from all hyperplanes defining the positive orthant: it prevents all x_j from approaching zero.

First Order Optimality Conditions: LP

Replace the **primal LP**

$$\begin{array}{rrcl} \min & c^T x & & \\ \text{s.t.} & Ax & = & b, \\ & x \geq 0, & & \end{array}$$

with the **primal barrier program**

$$\begin{array}{ll} \min & c^T x - \sum_{j=1}^n \ln x_j \\ \text{s.t.} & Ax = b, \end{array}$$

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

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

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:

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s & = & c, \\ XSe & = & \mu e. \end{array}$$

Note that the first order optimality conditions for the barrier problem

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e, \end{aligned}$$

approximate the first order optimality conditions for the linear program

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= 0, \end{aligned}$$

more and more closely as μ approaches zero.

Parameter μ controls the distance to optimality.

$$c^T x - b^T y = c^T x - x^T A^T y = x^T (c - A^T y) = x^T s = n\mu.$$

Analytic center (μ-center): a (unique) point

$$(x(\mu), y(\mu), s(\mu)), \quad x(\mu) > 0, \; s(\mu) > 0$$

that satisfies FOC.

The path

$$\{(x(\mu), y(\mu), s(\mu)) : \mu > 0\}$$

is called the **primal-dual central trajectory**.

We use **Newton Method** to find a stationary point of the barrier problem.

Recall how to use Newton Method to find a root of a nonlinear equation

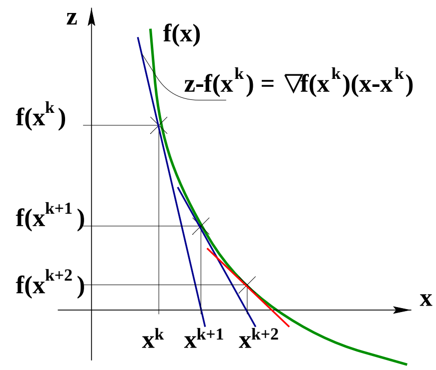
$$f(x) = 0.$$

A tangent line

$$z - f(x^k) = \nabla f(x^k) \cdot (x - x^k)$$

is a local approximation of the graph of the function $f(x)$. Substituting $z = 0$ gives a new point

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



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 $F : \mathcal{R}^{2n+m} \mapsto \mathcal{R}^{2n+m}$ is an application defined as follows:

$$F(x, y, s) = \begin{bmatrix} Ax & - & b \\ A^T y + s & - & c \\ XSe & - & \mu e \end{bmatrix}.$$

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

Note that

$$\nabla F(x, y, s) = \begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix}.$$

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:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - s \\ \mu e - XSe \end{bmatrix}.$$

Interior-Point Framework

We have already gathered all the necessary elements to derive an interior point method.

The **logarithmic barrier**

$$-\ln x_j$$

added to the objective in the optimization problem prevents variable x_j from approaching zero and "replaces" the inequality

$$x_j \geq 0.$$

We derive the **first order optimality conditions** for the primal barrier problem:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e, \end{aligned}$$

and apply **Newton method** to solve this system of nonlinear equations.

Actually, we fix the barrier parameter μ and make only **one** (dumped) Newton step towards the solution of FOC. We do not solve the current FOC exactly. Instead, we immediately reduce the barrier parameter μ (to ensure progress towards optimality) and repeat the process.

Assume a primal-dual strictly feasible solution $(x,y,s) \in \mathcal{F}^0$ lying in a neighbourhood of the central path is given; namely (x,y,s) satisfies:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &\approx \mu e. \end{aligned}$$

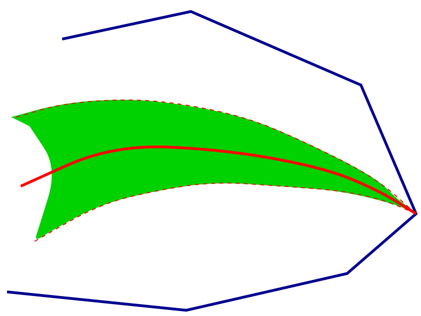
We define a θ-**neighbourhood** of the central path $N_2(\theta)$, a set of primal-dual strictly feasible solutions $(x,y,s) \in \mathcal{F}^0$ that satisfy:

$$\|XSe - \mu e\| \le \theta\mu,$$

where $\theta \in (0,1)$ and the barrier μ satisfies:

$$x^T s = n\mu.$$

Hence $N_2(\theta) = \{(x,y,s) \in \mathcal{F}^0 \mid \|XSe - \mu e\| \le \theta\mu\}$.



$N_2(\theta)$ neighbourhood of the central path

Assume a primal-dual strictly feasible solution $(x,y,s) \in N_2(\theta)$ for some $\theta \in (0,1)$ is given.

The interior point algorithm tries to move from this point to another one that also belongs to a θ-neighbourhood of the central path but corresponds to a smaller μ. The required reduction of μ is small:

$$\mu^{k+1} = \sigma\mu^k,$$

where

$$\sigma = 1 - \beta/\sqrt{n},$$

for some $\beta \in (0,1)$.

Given a new μ-center, the interior point algorithm computes the Newton direction:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - XSe \end{bmatrix},$$

and makes a step in this direction.

The **magic numbers** (will be explained later) are:

$$\theta = 0.1 \quad \text{and} \quad \beta = 0.1.$$

How to prove $\mathcal{O}(\sqrt{n})$ complexity result?

We can prove the following:

- a full step in Newton direction is feasible;
- the new iterate $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$ belongs to a θ-neighbourhood of the new μ-center (with $\mu^{k+1} = \sigma\mu^k$);
- the duality gap is reduced $1 - \beta/\sqrt{n}$ times.

Note that since at one iteration the duality gap is reduced $1 - \beta/\sqrt{n}$ times, after $\sqrt{n}$ iterations the reduction achieves:

$$(1 - \beta/\sqrt{n})^{\sqrt{n}} \approx e^{-\beta}.$$

After $C \cdot \sqrt{n}$ iterations, the reduction is $e^{-C\beta}$. For a sufficiently large constant C the reduction can thus be arbitrarily large (i.e. the duality gap can become arbitrarily small).

Hence this algorithm has complexity $\mathcal{O}(\sqrt{n})$.

Linear Algebra of IPM for LP

First order optimality conditions:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e. \end{aligned}$$

Newton direction:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix},$$

where

$$\begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - s \\ \mu e - XSe \end{bmatrix}.$$

Use the third equation to eliminate

$$\begin{aligned} \Delta s &= X^{-1}(\xi_\mu - S\Delta x) \\ &= -X^{-1}S\Delta x + X^{-1}\xi_\mu, \end{aligned}$$

from the second equation and get

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

where $\Theta = XS^{-1}$ is a diagonal scaling matrix.

Consider the **convex** quadratic programming problem

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array}$$

where $Q \in \mathcal{R}^{n \times n}$ is positive semidefinite matrix.

Apply the *usual* procedure:

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

Replace the **primal QP**

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array}$$

with the **primal barrier QP**

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x - \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} & Ax = b. \end{array}$$

The QP problem:

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t.} & Ax = b, \\ & x \geq 0. \end{array}$$

First order conditions (for barrier problem):

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s - Qx & = & c, \\ XSe & = & \mu e. \end{array}$$

Newton direction:

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

where

$$\begin{array}{rcl} \xi_p & = & b - Ax, \\ \xi_d & = & c - A^T y - s + Qx, \\ \xi_\mu & = & \mu e - XSe. \end{array}$$

Augmented system

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1} \xi_\mu \\ \xi_p \end{bmatrix}.$$

Conclusion:
QP is a natural extension of LP.

Convex Nonlinear Optimization

Consider the nonlinear optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \leq 0, \end{array}$$

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

The vector-valued function $g : \mathcal{R}^n \mapsto \mathcal{R}^m$ has a derivative $A(x) \in \mathcal{R}^{m \times n}$

$$A(x) = \nabla g(x) = \left[\frac{\delta g_i}{\delta x_j} \right]_{i=1..m,\, j=1..n}$$

which is called the **Jacobian** of g.

The Lagrangian associated with the NLP is:

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

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

The first derivatives of the Lagrangian:

$$\begin{array}{rcl} \nabla_x \mathcal{L}(x, y) & = & \nabla f(x) + \nabla g(x)^T y \\ \nabla_y \mathcal{L}(x, y) & = & g(x). \end{array}$$

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

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

IPM for NLP

Add slack variables to nonlinear inequalities:

$$\begin{array}{lll} \min & f(x) & \\ \text{s.t.} & g(x) + z & = 0 \\ & z & \geq 0, \end{array}$$

where $z \in \mathcal{R}^m$. Replace inequality $z \geq 0$ with the logarithmic barrier:

$$\begin{array}{ll} \min & f(x) - \mu \sum_{i=1}^{m} \ln z_i \\ \text{s.t.} & g(x) + z = 0. \end{array}$$

Write out the **Lagrangian**

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

and the conditions for a stationary point

$$\begin{array}{rcrcl} \nabla_x L(x, y, z, \mu) & = & \nabla f(x) + \nabla g(x)^T y & = & 0 \\ \nabla_y L(x, y, z, \mu) & = & g(x) + z & = & 0 \\ \nabla_z L(x, y, z, \mu) & = & y - \mu Z^{-1} e & = & 0, \end{array}$$

where $Z^{-1} = diag\{z_1^{-1}, z_2^{-1}, \cdots, z_m^{-1}\}$.

The **First Order Optimality Conditions** are:

$$\begin{array}{rcl} \nabla f(x) + \nabla g(x)^T y & = & 0, \\ g(x) + z & = & 0, \\ YZe & = & \mu e. \end{array}$$

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

$$F(x, y, z) = 0,$$

where $F : \mathcal{R}^{n+2m} \mapsto \mathcal{R}^{n+2m}$ 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*.)

Note 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 $\mathcal{L}$. They are defined as follows:

$$\begin{aligned} A(x) &= \nabla g(x) && \in \mathcal{R}^{m\times n} \\ Q(x,y) &= \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x) && \in \mathcal{R}^{n\times n} \end{aligned}$$

Newton's direction for **NLP**

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

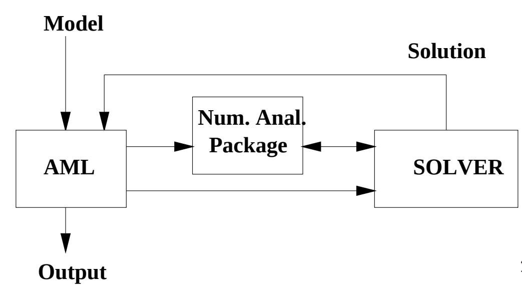
The corresponding augmented system

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

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

$$\begin{aligned} A(x) &= \nabla g(x) \\ Q(x,y) &= \nabla^2 f(x) + \sum_{i=1}^{m} y_i \nabla^2 g_i(x) \end{aligned}$$

Automatic differentiation is very useful: get $Q(x,y)$ and $A(x)$ from the **Algebraic Modeling Language**.



From QP to NLP

Newton's 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

$$\begin{bmatrix} -Q - SX^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

Newton's direction for **NLP**

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

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.

Interior Point Methods

Conclusions:

- Unified framework for convex optimization.
- Polynomial algorithms for LP, QP and NLP.
- Similar linear algebra in LP, QP and NLP.
- Suitable to solve very large problems.

What is supposed to come soon:

- Extension for **nonconvex** optimization.

Are IPMs really new?

- **Lagrange (1788)**
  handling equality constraints - multipliers; minimization with equality constraints replaced with unconstrained minimization
- **Fiacco & McCormick (1968)**
  handling inequality constraints - log barrier; minimization with inequality constraints replaced with a sequence of unconstrained minimizations
- **Newton (1687)**
  solving unconstrained minimization problems.

- Linear Algebra in IPMs.
- Symmetric Systems:
  - Positive Definite vs Indefinite Systems.
  - Quasi-definite Systems.
  - Regularizations.
- Sparsity Issues in Cholesky Decomposition.
- Unavoidable Ill-conditioning.
- Why Ill-conditioning is Benign ?
- Direct vs Iterative Methods.

First order optimality conditions

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &= \mu e. \end{aligned}$$

Newton's direction

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix},$$

where

$$\begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - s \\ \mu e - XSe \end{bmatrix}.$$

Use the third equation to eliminate

$$\begin{aligned} \Delta s &= X^{-1}(\xi_\mu - S\Delta x) \\ &= -X^{-1} S \Delta x + X^{-1}\xi_\mu, \end{aligned}$$

from the second equation and get

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

where $\Theta = XS^{-1}$ is a diagonal scaling matrix.

Weighted Least Squares

Augmented system (symmetric but indefinite)

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix},$$

where

$$\begin{bmatrix} r \\ h \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}.$$

Eliminate

$$\Delta x = \Theta A^T \Delta y - \Theta r,$$

to get **normal equations** (symmetric, positive definite system)

$$(A\Theta A^T)\Delta y = g = A\Theta r + h.$$

Matrix $A\Theta A^T$ has always the same sparsity structure (only Θ changes in subsequent iterations).

Two step solution method:

- factorization to LDL^T form,
- backsolve to compute direction Δy.

IPMs: LP, QP & NLP

Augmented system in **LP**

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}.$$

Eliminate Δx from the first equation and get normal equations

$$(A\Theta A^T)\Delta y = g.$$

Augmented system in **QP**

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}.$$

Eliminate Δx from the first equation and get normal equations

$$(A(Q + \Theta^{-1})^{-1} A^T)\Delta y = g.$$

Augmented system in **NLP**

$$\begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}.$$

Eliminate Δx from the first equation and get normal equations

$$(AQ^{-1}A^T + \Theta^{-1})\Delta y = g.$$

The inverse of the sparse matrix can be dense.
Example: Consider a symmetric 4×4 matrix

$$H = \begin{bmatrix} x & x & & \\ x & x & x & \\ & x & x & x \\ & & x & x \end{bmatrix} \quad \text{and} \quad H^{-1} = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}.$$

Hence the computation of $(Q + \Theta^{-1})^{-1}$ could produce a dense matrix and lead to a loss of efficiency in the linear system:

$$(A(Q + \Theta^{-1})^{-1} A^T)\Delta y \; = \; g.$$

One can use normal equations in LP, but not in QP or NLP. Normal equations produce sometimes excessively dense factors even in LP.

Problem	Dense col	Nonzeros NE	Nonzeros AS	Flops in 10^3 NE	Flops in 10^3 AS
aircraft	751	1437398	19759	361174	53
fit1p	627	206097	10118	42920	115
fit2p	3000	4500000	50583	4×10^9	481
storm8	18	114619	139396	11871	13671
storm27	37	1075443	350139	601859	19845
storm125	135	7000000	1441133	10^{10}	52866

Lemma: The decomposition $H = LDL^T$ with $d_{ii} > 0, \forall i$ exists iff H is positive definite (PD).

Proof:
Part 1 ($\Rightarrow$)
Let $H = LDL^T$ with $d_{ii} > 0$. Take any $x \neq 0$ and let $u = L^T x$. Since L is a unit lower triangular matrix it is nonsingular so $u \neq 0$ and

$$x^T H x = x^T LDL^T x = u^T D u = \sum_{i=1}^{m} d_{ii} u_i^2 > 0.$$

Part 2 ($\Leftarrow$)
Proof by induction on dimension of H.
For $m = 1$. $H = h_{11} = d_{11} > 0$ iff H is PD.

Assume the result is true for $m = k-1 \geq 1$.
Let $H = \begin{bmatrix} W & a \\ a^T & q \end{bmatrix} \in \mathcal{R}^{k \times k}$ be given $k \times k$ positive definite matrix with $W \in \mathcal{R}^{(k-1)\times(k-1)}$, $a \in \mathcal{R}^{k-1}$ and $q \in \mathcal{R}$. Note first that since H is PD, W is also PD. Indeed for any $(x, 0) \in \mathcal{R}^k$ we have

$$[x, 0] \begin{bmatrix} W & a \\ a^T & q \end{bmatrix} \begin{bmatrix} x \\ 0 \end{bmatrix} = x^T W x > 0 \quad \forall x \in \mathcal{R}^{k-1}, x \neq 0.$$

Existence of LDL^T factorization (cont'd)

From inductive hypothesis we know that $W = LDL^T$ with $d_{ii} > 0$. Let

$$\begin{bmatrix} W & a \\ a^T & q \end{bmatrix} = \begin{bmatrix} L & 0 \\ l^T & 1 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} L^T & l \\ 0 & 1 \end{bmatrix},$$

where l is the solution of equation $(LD)l = a$ (it is well defined since L and D are nonsingular) and d is given by $d = q - l^T D l$.

Hence matrix $H = \begin{bmatrix} W & a \\ a^T & q \end{bmatrix}$ has an $\bar{L}\bar{D}\bar{L}^T$ decomposition. It remains to prove that $d > 0$. Consider the vector

$$x = \begin{bmatrix} -L^{-T} l \\ 1 \end{bmatrix}.$$

Since H is positive definite, we get

$$\begin{aligned} 0 &< x^T H x \\ &= [-l^T L^{-1}, 1] \begin{bmatrix} L & 0 \\ l^T & 1 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} L^T & l \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -L^{-T} l \\ 1 \end{bmatrix} \\ &= [0, 1] \begin{bmatrix} D & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = d, \end{aligned}$$

which completes the proof.

Definite and Indefinite Systems

Cholesky factorization fails for indefinite matrix.

Example 1: Negative pivot $d_{22} < 0$.

$$\begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2/3 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -1/3 \end{bmatrix} \begin{bmatrix} 1 & 2/3 \\ 0 & 1 \end{bmatrix}.$$

Example 2: $d_{11} = 0$. Can't start elimination.

$$\begin{bmatrix} 0 & 2 \\ 2 & 5 \end{bmatrix} = ???$$

IPMs:

For positive definite *normal equations*

$$(A\Theta A^T)\Delta y \; = \; g.$$

one can compute the **Cholesky factorization**.

For indefinite *augmented system*

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}.$$

one needs to use some **special tricks**.

Two options are possible:

1. Replace diagonal matrix D with a block-diagonal one and allow 2×2 (indefinite) pivots

$$\begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & a \\ a & d \end{bmatrix}.$$

Hence obtain a decomposition $H = LDL^T$ with **block-diagonal** D.

2. Regularize indefinite matrix to produce a **quasidefinite** matrix

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix},$$

where
$E \in \mathcal{R}^{n\times n}$ is positive definite,
$F \in \mathcal{R}^{m\times m}$ is positive definite, and
$A \in \mathcal{R}^{m\times n}$ has full row rank.

Symmetric matrix is called **quasidefinite** if

$$K = \begin{bmatrix} -E & A^T \\ A & F \end{bmatrix},$$

where $E \in \mathcal{R}^{n\times n}$ and
$F \in \mathcal{R}^{m\times m}$ are positive definite, and
$A \in \mathcal{R}^{m\times n}$ has full row rank.

Symmetric nonsingular matrix K is *factorizable* if there exists a diagonal matrix D and unit lower triangular matrix L such that $K = LDL^T$.

The symmetric matrix K is *strongly factorizable* if for any permutation matrix P a factorization $PKP^T = LDL^T$ exists.

Vanderbei (1995) proved that
Symmetric QDFM's are strongly factorizable.

For any quasidefinite matrix
there exists a **Cholesky-like** factorization

$$\bar{H} = LDL^T,$$

where
D is **diagonal** but **not positive definite**:
n negative pivots;
m positive pivots.

From Indefinite to Quasidefinite Matrix

Indefinite matrix

$$H = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix}.$$

Vanderbei (1995): replace $Ax = b$ with $Ax \leq b$

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

and eliminate Θ_s^{-1}

$$K = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & \Theta_s \end{bmatrix}.$$

Saunders (1996):

$$H_S = \begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -\gamma^2 I_n & 0 \\ 0 & \delta^2 I_m \end{bmatrix},$$

where

$$\gamma\delta \geq \sqrt{\varepsilon} = 10^{-8}.$$

A & G (1999): use **dynamic regularization**

$$\bar{H} = \begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix},$$

$R_p \in \mathcal{R}^{n\times n}$ is a *primal* regularization
$R_d \in \mathcal{R}^{m\times m}$ is a *dual* regularization.

Primal Regularization

Primal barrier problem

$$\begin{array}{ll} \min & z_P = c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^n (\ln x_j + \ln s_j) \\ \text{s. to} & Ax = b, \\ & x + s = u, \\ & x, s > 0 \end{array}$$

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ h \end{bmatrix}.$$

Primal **regularized** barrier problem

$$\begin{array}{ll} \min & z_P + \frac{1}{2}(x - x_0)^T R_p (x - x_0) \\ \text{s. to} & Ax = b, \\ & x + s = u, \\ & x, s > 0 \end{array}$$

$$\begin{bmatrix} -Q - \Theta^{-1} - R_p & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f' \\ h \end{bmatrix},$$

where

$$f' = f - R_p(x - x_0).$$

Dual barrier problem

$$\begin{array}{ll} \max & z_D = b^T y - u^T w - \frac{1}{2} x^T Q x + \mu \sum_{j=1}^{n} (\ln z_j + \ln w_j) \\ \text{s. to} & A^T y + z - w - Qx = c, \\ & x \geq 0, z, w > 0 \end{array}$$

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ h \end{bmatrix}.$$

Dual **regularized** barrier problem

$$\begin{array}{ll} \max & z_D - \frac{1}{2} (y - y_0)^T R_d (y - y_0) \\ \text{s. to} & A^T y + z - w - Qx = c, \\ & x \geq 0, z, w > 0 \end{array}$$

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & R_d \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ h' \end{bmatrix},$$

where

$$h' = h - R_d (y - y_0).$$

Example: Consider a symmetric 4×4 matrix

$$H = \begin{bmatrix} x & x & x & x \\ x & x & & \\ x & & x & \\ x & & & x \end{bmatrix},$$

where x denotes a *nonzero* and empty spaces denote *zeros*. Direct application of Cholesky factorization would produce a completely dense lower triangular factor

$$L = \begin{bmatrix} x & & & \\ x & x & & \\ x & x & x & \\ x & x & x & x \end{bmatrix}.$$

However, it suffices to reorder (symmetrically) the rows and columns of H

$$\bar{H} = PHP^T = \begin{bmatrix} x & & & x \\ & x & & x \\ & & x & x \\ x & x & x & x \end{bmatrix},$$

to obtain sparse Cholesky factor

$$\bar{L} = \begin{bmatrix} x & & & \\ & x & & \\ & & x & \\ x & x & x & x \end{bmatrix}.$$

Minimum Degree Ordering

How to permute rows and columns of H to get the sparsest possible Cholesky factorization? Difficult problem but *heuristics* can help.

Example: Consider a symmetric matrix

$$H = \begin{bmatrix} x & & x & x & x & \\ & x & & & x & \\ x & & x & & & x \\ x & & & x & & x \\ x & x & & & x & \\ & & x & x & & x \end{bmatrix}.$$

Suppose h_{11} is the first pivot

$$H = \begin{bmatrix} x & & x & x & x & \\ & x & & & x & \\ x & & x & f & f & x \\ x & & f & x & f & x \\ x & x & f & f & x & \\ & & x & x & & x \end{bmatrix}.$$

Suppose h_{22} is the first pivot. Replace rows 1 and 2 and columns 1 and 2. The elimination of the first pivot does not create any fill-in.

$$\bar{H} = \begin{bmatrix} x & & & & x & \\ & x & x & x & x & \\ & x & x & & & x \\ & x & & x & & x \\ x & x & & & x & \\ & & x & x & & x \end{bmatrix}.$$

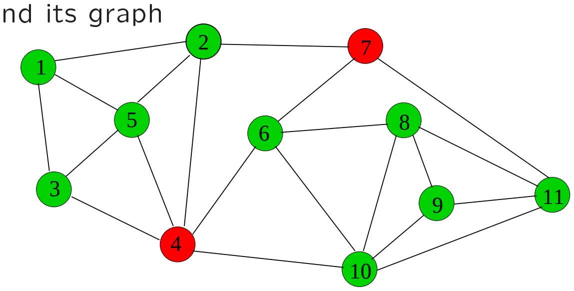
Minimum degree ordering: choose an element with the minimum number of nonzeros in a row.

Nested Dissection

Remove *few* nodes to *disconnect* the graph. Consider a matrix

$$H = \begin{array}{c|ccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \hline 1 & x & x & x & & x & & & & & & \\ 2 & x & x & & x & x & & x & & & & \\ 3 & x & & x & x & x & & & & & & \\ 4 & & x & x & x & x & x & & & & x & \\ 5 & x & x & x & x & x & & & & & & \\ 6 & & & & x & & x & x & x & & x & \\ 7 & & x & & & & x & x & & & & x \\ 8 & & & & & & x & & x & x & x & x \\ 9 & & & & & & & & x & x & x & x \\ 10 & & & & x & & x & & x & x & x & x \\ 11 & & & & & & & x & x & x & x & x \end{array}$$

and its graph



Remove (permute) nodes 4 and 7.

$$PHP^T = \begin{array}{c|ccccccccccc} & 1 & 2 & 3 & 5 & 6 & 8 & 9 & 10 & 11 & 4 & 7 \\ \hline 1 & x & x & x & x & & & & & & & \\ 2 & x & x & & x & & & & & & x & x \\ 3 & x & & x & x & & & & & & x & \\ 5 & x & x & x & x & & & & & & x & \\ 6 & & & & & x & x & & x & & x & x \\ 8 & & & & & x & x & x & x & x & & \\ 9 & & & & & & x & x & x & x & & \\ 10 & & & & & x & x & x & x & x & x & \\ 11 & & & & & & x & x & x & x & & x \\ 4 & & x & x & x & x & & & x & & x & \\ 7 & & x & & & x & & & & x & & x \end{array}$$

Lemma:
If $f : \mathcal{R}^n \mapsto \mathcal{R}$ and $g : \mathcal{R}^n \mapsto \mathcal{R}^m$ 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**.

Lemma:
If $f : \mathcal{R}^n \mapsto \mathcal{R}$ is strictly convex, and $g : \mathcal{R}^n \mapsto \mathcal{R}^m$ is convex, both f and g are twice differentiable, and $A(x)$ has full row rank for any x, then the augmented system matrix

$$H = \begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix}$$

is **quasidefinite** for any x and any $z, y > 0$.

What if f or g are nonconvex?

Augmented system in **NLP**

$$\begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix}.$$

If f or g are convex, then a lot of flexibility is available in the pivot choice (QDF Matrix).
In particular, one can eliminate Δx from the first equation and get

$$(AQ^{-1}A^T + \Theta^{-1})\Delta y \;=\; g.$$

If f and/or g are nonconvex, then $AQ^{-1}A^T$ can be indefinite so do not compute $(AQ^{-1}A^T + \Theta^{-1})$.

Matrix $A\Theta A^T$ is certainly positive definite so eliminate Δy from the second equation and get reduced system

$$(Q + A\Theta A^T)\Delta x \;=\; r'.$$

The "degree" of indefiniteness can be reduced.

"Easy" Nonconvex NLP

Example:
Consider a nonconvex problem

$$\begin{array}{rl} \min & 2x_1^2 - x_2^2 \\ \text{s. to} & x_1 + x_2 = 2, \\ & x_1, x_2 > 0. \end{array}$$

Eliminate $x_2 = 2 - x_1$ to get

$$\begin{array}{rl} \min & x_1^2 + 4x_1 - 4 \\ \text{s. to} & 0 \leq x_1 \leq 2. \end{array}$$

Remark:
By restricting Δx to be in the null space of A, we can reduce the "degree" of indefiniteness in the NLP problem:
Matrix

$$Q + A\Theta A^T$$

may have fewer negative eigenvalues than Q.

Ill-conditioning

Assume **Normal Equations** are used in LP and the feasible IPM is used ($\xi_b = 0$ and $\xi_c = 0$)

$$(A\Theta A^T)\Delta y \;=\; A\Theta r,$$

where $\Theta = XS^{-1}$ and $r = -X^{-1}\xi_\mu$.

Optimal Partition:

$$\begin{array}{rll} \text{Basic variables} & x_B \to x_B^* > 0 & s_B \to s_B^* = 0 \\ \text{Non-basic variables} & x_N \to x_N^* = 0 & s_N \to s_N^* > 0 \end{array}$$

For **basic** variables: $\Theta_j = x_j/s_j \to \infty$;
For **non-basic** variables: $\Theta_j = x_j/s_j \to 0$.

Hence

$$A\Theta A^T = \sum_{j\in\mathcal{B}} \theta_j a_{.j} a_{.j}^T + \sum_{j\in\mathcal{N}} \theta_j a_{.j} a_{.j}^T \to \sum_{j\in\mathcal{B}} \theta_j a_{.j} a_{.j}^T.$$

The matrix $H = A\Theta A^T$ has usually a huge condition number $\kappa(H)$. Although $\kappa(H) \gg 1/\epsilon$, where ϵ is the relative precision of the computer (e.g. $\epsilon = 10^{-16}$), IPMs nicely converge.

Theorem: (Dikin, 1967)
Let $A \in \mathcal{R}^{m \times n}$ be a full row rank matrix;
g be a vector of dimension n; and
D_+ be the set of $n \times n$ diagonal positive definite matrices.
Then

$$\sup_{D \in D_+} \|(ADA^T)^{-1}ADg\| = \max_{\mathcal{J} \in \mathcal{J}(A)} \|A_{\mathcal{J}}^{-T} g_{\mathcal{J}}\|$$
$$\sup_{D \in D_+} \|(ADA^T)^{-1}AD\| = \max_{\mathcal{J} \in \mathcal{J}(A)} \|A_{\mathcal{J}}^{-T}\|$$

where $\mathcal{J}(A)$ is the set of column indices associated with nonsingular $m \times m$ submatrices of A.

Corollary:
The linear system arising in IPMs for LP

$$(A\Theta A^T)\Delta y = A\Theta r,$$

produces more accurate solutions than those one could have expected from a "classical" worst-case analysis.

Dikin's result applies to *diagonal* positive definite weight matrices D.

Forsgren (1996) generalized it to diagonally dominant weight matrices W.

Lemma:
Let $A \in \mathcal{R}^{m \times n}$ be a full row rank matrix;
g be a vector of dimension n; and
W_+ be the set of $n \times n$ diagonally dominant weight matrices.
Then

$$\sup_{W \in W_+} \|(AWA^T)^{-1}AWg\|$$
$$\sup_{W \in W_+} \|(AWA^T)^{-1}AW\|$$

are bounded.

This Lemma extends Dikin's result to quadratic and nonlinear optimization.

Iterative Methods

Optimal Partition:

For **basic** variables:	$\Theta_j = x_j/s_j \to \infty$;
For **non-basic** variables:	$\Theta_j = x_j/s_j \to 0$.

The spread of elements in Θ causes that $A\Theta A^T$ is very ill-conditioned. Consequently, any iterative method suffers from slow convergence. Yet, with the right preconditioners iterative methods have some promise.
There have been successful implementations of PCG for structured problems.

Idea: Split A into basic and non-basic parts and adjust preconditioner accordingly

$$H = \begin{bmatrix} -\Theta_B^{-1} & 0 & B^T \\ 0 & -\Theta_N^{-1} & N^T \\ B & N & 0 \end{bmatrix}$$

Block-projection Methods are another promise.

Interior Point Methods

This lecture treated:

- IPMs for LP;
- extension from LP to convex QP;
- extension from convex QP to convex NLP.

Conclusions:

- Interior Point Methods provide the unified framework for convex optimization.
- Interior Point Methods 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.

Very Active Area These Days:

- **Nonconvex** optimization with IPMs.

IPMs in the Internet:

- LP FAQ (Frequently Asked Questions):
  `http://www-unix.mcs.anl.gov/otc/Guide/faq/`
- Interior Point Methods On-Line:
  `http://www-unix.mcs.anl.gov/otc/InteriorPoint/`
- NEOS (Network Enabled Opt. Services):
  `http://www-neos.mcs.anl.gov/`