**Interior Point Methods**  
for Convex Quadratic 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:** Linear Algebra in IPM  
  - LP case  
  - QP case  
  - Cholesky factorization  
  - exploiting sparsity  
- **Part 3:** Huge Problems: Block-Sparsity  
- Final Comments

---

**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 feasible solution to it,  
then there exists an 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:

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 5 & 4 \\ 4 & 3 \end{bmatrix}, \quad Q_4 = \begin{bmatrix} 5 & -2 \\ -2 & 3 \end{bmatrix}.$$ 

$Q_1$ and $Q_4$ are positive definite (hence $f_1, f_4$ are convex). $Q_2$ and $Q_3$ are indefinite ($f_2, f_3$ are not convex).

Dual Quadratic Program

Consider a quadratic program

$$\min \quad c^T x + \frac{1}{2} x^T Q x$$

$$\text{s.t.} \quad Ax = b,$$

$$x \geq 0,$$

where $c, x \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}, Q \in \mathbb{R}^{n \times n}$.

We associate Lagrange multipliers $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n (s \geq 0)$ with the constraints $Ax = b$ and $x \geq 0$, and write the **Lagrangian**

$$L(x, y, s) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - s^T x.$$ 

Dual QP (cont’d)

To determine the **Lagrangian dual**

$$L_D(y, s) = \min_{x \in X} L(x, y, s)$$

we need stationarity with respect to $x$:

$$\nabla_x L(x, y, s) = c + Q x - A^T y - s = 0.$$ 

Hence

$$L_D(y, s) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - s^T x$$

$$= b^T y + x^T (c + Q x - A^T y - s) - \frac{1}{2} x^T Q x$$

$$= b^T y - \frac{1}{2} x^T Q x,$$

and the **dual** problem has the form:

$$\max \quad b^T y - \frac{1}{2} x^T Q x$$

$$\text{s.t.} \quad A^T y + s - Q x = c,$$

$$x, s \geq 0,$$

where $y \in \mathbb{R}^m$ and $x, s \in \mathbb{R}^n$.  

NATCOR, Edinburgh, June 2014
**QP with IPMs**

Consider the *convex* quadratic programming problem. The **primal**

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

and the **dual**

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

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

---

**First Order Optimality Conditions**

Consider the **primal barrier quadratic program**

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

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

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

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

Let us denote \( s = \mu X^{-1} e, \) i.e. \( XSe = \mu e. \)

The **First Order Optimality Conditions** are:

\[ Ax = b, \]
\[ ATy + s - Qx = c, \]
\[ XSe = \mu e. \]

---

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

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

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

Note that

\[ \nabla F(x, y, s) = \begin{bmatrix} A & 0 & 0 \\ -Q & AT & I \\ S & 0 & X \end{bmatrix}. \]
From LP to QP

QP problem

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

First order conditions (for barrier problem)

\[
\begin{align*}
Ax &= b, \\
A^T y + s - Qx &= c, \\
XS e &= \mu e.
\end{align*}
\]

Linear Algebra of IPM: LP Case

FOC

\[
\begin{align*}
Ax &= b, \\
A^T y + s &= c, \\
XS e &= \mu e.
\end{align*}
\]

Newton direction

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

where

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

Linear Algebra, LP Case (cont’d)

In Newton direction

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

use the third equation to eliminate

\[
\Delta s = X^{-1}(\xi_\mu - S\Delta x) = -X^{-1}S\Delta x + X^{-1}\xi_\mu,
\]

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.
Linear Algebra of IPM: QP Case

**FOC**

\[ Ax = b, \]
\[ A^T y + s - Qx = c, \]
\[ XS = \mu e. \]

**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

\[ \xi_p = b - Ax, \]
\[ \xi_d = c - A^T y - s + Qx, \]
\[ \xi_\mu = \mu e - XSe. \]

**Summary: From LP to QP**

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

\[ \xi_p = b - Ax, \]
\[ \xi_d = c - A^T y - s + Qx, \]
\[ \xi_\mu = \mu e - XSe. \]

**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.

**Linear Algebra, QP Case (cont’d)**

In *Newton direction*

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

use the third equation to eliminate

\[ \Delta s = X^{-1}(\xi_\mu - S\Delta x) = -X^{-1}S\Delta x + X^{-1}\xi_\mu, \]

from the second equation and get

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

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

**IPMs: LP vs QP**

Augmented system in *LP*

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

Eliminate \( \Delta x \) from the first equation and get normal equations

\[ (A\Theta A^T)\Delta y = g. \]
**IPMs: LP vs QP**

Augmented system in QP

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

Eliminate \( \Delta x \) from the first equation and get normal equations

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

One can use normal equations in LP, but not in QP. Normal equations in QP may become almost completely dense even for sparse matrices \( A \) and \( Q \). Thus, in QP, usually the indefinite augmented system form is used.

**Normal Equations**

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

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

**Two step solution method:**
- factorization to \( LDL^T \) form,
- backsolve to compute direction \( \Delta y \).

**Use of Cholesky factorization**

Replace the difficult equation

\[(A\Theta A^T) \cdot \Delta y = g,\]

with a sequence of easy equations:

\[L \cdot u = g,\]
\[D \cdot v = u,\]
\[L^T \cdot \Delta y = v.\]

Note that

\[g = Lu = L(Dv) = LD(L^T \Delta y) = (LDL^T) \Delta y = (A\Theta A^T) \Delta y.\]
**Existence of \(LDL^T\) factorization**

**Lemma 2:** The decomposition \(H = LDL^T\) with \(d_{ii} > 0, \forall i\) exists if \(H\) is positive definite (PD).

**Proof:**

Part 1 (⇒)

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

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

**Proof (cont’d):**

Part 2 (⇐)

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

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

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 \\ 0 & D \end{bmatrix} \begin{bmatrix} L^T & I \\ I^T & 0 \end{bmatrix}
\]

**Symmetric Gaussian Elimination**

Let \(H \in \mathcal{R}^{m \times m}\) be a symmetric positive definite matrix

\[
H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1m} \\ h_{21} & h_{22} & \cdots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mm} \end{bmatrix}.
\]

By applying Gaussian Elimination to it, we can represent it in the following form:

\[
\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{mm} \end{bmatrix} = \begin{bmatrix} 1 & l_{21} & \cdots & l_{m1} \\ 0 & 1 & \cdots & l_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.
\]

**Example 1:**

\[
\begin{bmatrix} 1 & -1 & 2 \\ -1 & 3 & 0 \\ 2 & 0 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 & 2 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.
\]

**Example 2:**

\[
\begin{bmatrix} 1 & 1 & -1 \\ 1 & 5 & 7 \\ -1 & 7 & 22 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.
\]
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 \( LDL^T \) 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

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

which completes the proof.

---

**Large Problems are Sparse**

Suppose a medium or large LP is solved: \( m, n \sim 10^3 - 10^6 \).

Can all variables be linked at the same time?

No, usually only a subset of them is linked.

There are usually only several nonzeros per row in an LP.

Large problems are always sparse.

Very large problems are often block-sparse.

Exploiting sparsity in computations leads to huge savings.

Exploiting sparsity means mainly avoiding doing useless computations: the computations for which the result is known, as for example multiplications with zero.
Minimum Degree Ordering

**Sparse Matrix**

\[
H = \begin{bmatrix}
x & x & x & x \\
x & x & x & x \\
x & x & x & x \\
x & x & x & x \\
\end{bmatrix}
\]

**Pivot**

\[
\begin{bmatrix}
p & x & x & x \\
x & x & x & x \\
x & x & f & f \\
x & f & x & f \\
\end{bmatrix}
\]

**Pivot**

\[
\begin{bmatrix}
x & x & x & x \\
x & x & x & x \\
x & x & x & x \\
x & x & x & x \\
\end{bmatrix}
\]

Minimum degree ordering:

choose a diagonal element corresponding to a row with the *minimum* number of nonzeros.

Permute rows and columns of \( H \) accordingly.

---

**Cholesky factorization**

\[ LDL^T = A\Theta A^T. \]

Involved preparation step:

- minimum degree ordering (reduces \# of nonzeros of \( L \));
- symbolic factorization (predicts the sparsity structure of \( L \)).

Computational complexity of different steps:

- minimum degree ordering \( O(\sum_i n_i^2) \)
- numerical factorization \( O(\sum_i n_i^2) \)
- symbolic factorization \( O(\sum_i n_i) \)
- backsolve \( O(\sum_i n_i) \)

where \( n_i \) is \# of nonzero entries in \( L_{i,i} \)

---

**Linear Algebra: Simplex Method vs IPM**

Suppose an LP of dimension \( m \times n \) is solved.

**Iterations to reach an optimum:**

<table>
<thead>
<tr>
<th>Simplex Method</th>
<th>IPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory Practice</td>
<td>Theory Practice</td>
</tr>
<tr>
<td>Nonpolynomial ( O(m+n) )</td>
<td>( O(\sqrt{n}) )</td>
</tr>
<tr>
<td>( O(\log_{10} n) )</td>
<td></td>
</tr>
</tbody>
</table>

But one iteration of the simplex method is usually significantly less expensive. Simplex method solves equation with the basis matrix:

\[
\begin{bmatrix}
B & N \\
0 & I_{n-m}
\end{bmatrix}
\begin{bmatrix}
x_B \\
x_N
\end{bmatrix} = \begin{bmatrix}
b \\
0
\end{bmatrix},
\]

which reduces to

\[ Bx_B = b. \]

IPM solves equation with the matrix \( A\Theta A^T \):

\[(A\Theta A^T)\Delta y = g.\]
Structured Problems

Observation:

Truly large scale problems are not only sparse...
→ such problems are structured

Structure is displayed in:
- Jacobian matrix $A$
- Hessian matrix $Q$

Structure can be exploited in:
- IPM Algorithm
- Linear Algebra of IPM (focus of the rest of this lecture)

Part 3:

Huge Problems: Block-Sparsity

... are present everywhere.
Sources of Structure

Dynamics → Staircase structure

\[ x_{t+1} = A_t x_t + B_t u_t \]

Uncertainty → Block-angular structure

\[ T_i^1 x^1 + W_i y_i = b_i \]

Common resource constraint

\[ \sum_{i=1}^k B_i x_i = b \rightarrow \text{Dantzig-Wolfe structure} \]
**Sources of Structure**

(low) rank-corrector

\[ A + VV^T = C \]

and networks, ODE- or PDE-discretizations, etc.

**From Sparsity to Block-Sparsity:**

Apply minimum degree ordering to (sparse) blocks:

Block-Sparse Matrix \( H \)

\[
H = \begin{bmatrix}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\end{bmatrix}
\]

Pivot Block \( H_{11} \)

\[
Pivot Block \quad \begin{bmatrix}
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\end{bmatrix}
\]

Choose a diagonal block-pivot corresponding to a block-row with the minimum number of blocks.

Permute block-rows and block-columns of \( H \) accordingly.

**Abstract Linear Algebra for IPMs**

Execute the operation

“solve (reduced) KKT system”

in IPMs for LP, QP and NLP.

It works like the “backslash” operator in MATLAB.

**Assumptions:**

Q and A are block-structured
OOPS: Object-oriented linear algebra for IPM

- Every node in the block elimination tree has its own linear algebra implementation (depending on its type).
- Each implementation is a realisation of an abstract linear algebra interface.
- Different implementations are available for different structures.

⇒ Rebuild block elimination tree with matrix interface structures

Example: Financial Planning Problems (ALM)

- A set of assets \( J = \{1, \ldots, J\} \) given (bonds, stock, real estate).
- At every stage \( t = 0, \ldots, T-1 \) we can buy or sell different assets.
- The return of asset \( j \) at stage \( t \) is uncertain.

Investment decisions: what to buy or sell, at which time stage.

Objectives:

- Maximize the final wealth → Mean Variance formulation: \( \max \mathbb{E}(X) - \rho \text{Var}(X) \)
- Minimize the associated risk

⇒ Stochastic Program: ⇒ formulate deterministic equivalent

- Standard QP, but huge
- Extentions: nonlinear risk measures (log utility, skewness)

Sparsity of Linear Algebra

⇒ 63 + 128 \times 63 = 8127 columns for Schur-complement

- Prohibitively expensive

⇒ Need facility to exploit nested structure
- Need to be careful that Schur-complement calculations stay sparse on second level.
J. Gondzio  

IPMs for QP

**Results** (ALM: Mean-Variance QP formulation):

<table>
<thead>
<tr>
<th>Prob</th>
<th>Stgs</th>
<th>Asts</th>
<th>Scen</th>
<th>Rows</th>
<th>Cols</th>
<th>iter</th>
<th>time</th>
<th>procs</th>
<th>machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALM8</td>
<td>7</td>
<td>6</td>
<td>13M</td>
<td>64M</td>
<td>154M</td>
<td>42</td>
<td>3923</td>
<td>512</td>
<td>BlueGene</td>
</tr>
<tr>
<td>ALM9</td>
<td>7</td>
<td>14</td>
<td>6M</td>
<td>96M</td>
<td>269M</td>
<td>39</td>
<td>4692</td>
<td>512</td>
<td>BlueGene</td>
</tr>
<tr>
<td>ALM10</td>
<td>7</td>
<td>13</td>
<td>12M</td>
<td>180M</td>
<td>500M</td>
<td>45</td>
<td>6089</td>
<td>1024</td>
<td>BlueGene</td>
</tr>
<tr>
<td>ALM11</td>
<td>7</td>
<td>21</td>
<td>16M</td>
<td>353M</td>
<td>1.011M</td>
<td>53</td>
<td>3020</td>
<td>1280</td>
<td>HPCx</td>
</tr>
</tbody>
</table>

The QP problem with

- **353 million of constraints**
- **1 billion of variables**

was solved in 50 minutes using 1280 procs (May 2005).

Equation systems of dimension **1.363 billion** were solved with the direct (implicit) factorization.

→ One IPM iteration takes less than a minute.

NATCOR, Edinburgh, June 2014

57

J. Gondzio  

IPMs for QP

**References**


Papers available: [http://www.maths.ed.ac.uk/~gondzio/](http://www.maths.ed.ac.uk/~gondzio/)

**OOps: Object-Oriented Parallel Solver**


NATCOR, Edinburgh, June 2014

58

--

**Interior Point Methods:**

- Unified view of optimization → from LP via QP to NLP
- Predictable behaviour → small number of iterations
- Unequalled efficiency
  - competitive for small problems \((n \leq 10^6)\)
  - beyond competition for large problems \((n \geq 10^6)\)

**Use IPMs in your research!**

NATCOR, Edinburgh, June 2014