School of Mathematics

Linear Algebra in IPMs

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

Outline

- **Linear Algebra in IPMs for LP, QP and NLP**
- **Definite, Indefinite and Quasidefinite Systems**
- **Cholesky factorization**
- **Exploiting Sparsity in Gaussian Elimination**
  - Minimum degree ordering
  - Nested dissection
- **Very Large Scale Optimization**
  - implicit inverse representation
  - from sparsity to block-sparsity
  - structured optimization problems
  - **OOPS**: Object-Oriented Parallel Solver
- **Applications**
  - financial planning problems (nonlinear risk measures)
  - data mining (nonlinear kernels in SVMs)

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

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

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.

IPMs: LP vs QP

Augmented system in **LP**

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

Eliminate Δx from the first equation and get normal equations

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

IPMs: LP vs QP

Augmented system in **QP**

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

Eliminate Δx 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.

KKT systems in IPMs for LP, QP and NLP

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

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

NLP
$$\begin{bmatrix} Q(x,y) + \Theta_P^{-1} & A(x)^T \\ A(x) & -\Theta_D \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

Cholesky factorization

Compute a decomposition

$$LDL^T = A\Theta A^T.$$

where:
L is a lower triangular matrix; and
D is a diagonal matrix.

Cholesky factorization is simply the **Gaussian Elimination** process that exploits two properties of the matrix:

- symmetry;
- positive definiteness.

Use of Cholesky factorization

Replace the **difficult** equation

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

with a sequence of **easy** equations:

$$\begin{aligned} L \cdot u &= g, \\ D \cdot v &= u, \\ L^T \cdot \Delta y &= v. \end{aligned}$$

Note that

$$\begin{aligned} g &= Lu \\ &= L(Dv) \\ &= LD(L^T \Delta y) \\ &= (LDL^T)\Delta y \\ &= (A\Theta A^T)\Delta y. \end{aligned}$$

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:

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

Symmetric GE: Examples

Example 1:

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

Example 2:

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 5 & 7 \\ -1 & 7 & 22 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Existence of LDL^T factorization

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

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

Proof (cont'd):

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

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

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 $d = q - l^T Dl$.
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 & 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 even start the elimination.

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

Definite & Indefinite Systems (cont'd)

IPMs:

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**.

For positive definite *normal equations*

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

one can compute the **Cholesky factorization**.

Major Cholesky

Andre-Louis Cholesky (1875-1918)
Major of French Army,
descendant from the Cholewski family of Polish imigrants.

Read: **M. A. Saunders**, Major Cholesky would feel proud, *ORSA Journal on Computing*, vol 6 (1994) No 1, pp 23–27.

Symmetric Factorization

Two step solution method:

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

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

A symmetric matrix H is **strongly factorizable** if for any permutation matrix P a factorization $PHP^T = LDL^T$ exists.

The general symmetric indefinite matrix **is not factorizable**.

Factoring Indefinite Matrix

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 decomp. $H = LDL^T$ 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.

Quasidefinite (QDF) Matrices

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.

QDF matrices are **strongly factorizable**. For any quasidefinite matrix there exists a **Cholesky-like** factorization

$$K = LDL^T,$$

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

From Indefinite to Quasidefinite

Indefinite matrix

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

in IPMs can be converted to a quasidefinite one.
Regularize indefinite matrix to produce a **quasi-definite** matrix.
Use **dynamic regularization**

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

where $R_p \in \mathcal{R}^{n \times n}$ and $R_d \in \mathcal{R}^{m \times m}$ are the *primal* and *dual* regularizations. 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 and m positive pivots.

Large Problems are Sparse

Suppose a large LP is solved: $m, n \sim 10^4 - 10^6$.
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**.
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.

Exploiting sparsity: Example

$$Ax = \begin{bmatrix} 2 & 1 & 0 & 4 & 0 & 0 \\ 0 & 2 & 0 & -1 & 5 & -1 \\ 3 & 0 & 3 & 8 & 0 & 5 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 5 \\ 0 \\ 0 \\ -2 \end{bmatrix}.$$

It requires computing

$$2 \cdot A_{.1} + 5 \cdot A_{.3} - 2 \cdot A_{.6}$$

and involves only five multiplications and five additions.
We say that this matrix-vector multiplication needs 5 flops.

A **flop** is a *floating point operation*:

$$x := x + a \cdot b.$$

General Sparse Systems

Single step in Gaussian Elimination

$$A = \begin{bmatrix} \mathbf{p} & v^T \\ u & A_1 \end{bmatrix}$$

produces the following Schur complement

$$A_1 - p^{-1} u v^T .$$

Markowitz Pivot Choice

Let r_i and $c_i, i = 1, 2, ..., n$ be numbers of nonzero entries in row and column i, respectively. The elimination of the pivot a_{ij} needs

$$f_{ij} = (r_i - 1)(c_j - 1)$$

flops to be made. This step creates at most f_{ij} new nonzero entries in the Schur complement.

General Sparse Systems

The effect of pivot elimination on the sparsity pattern

$$
\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & \mathbf{p} & & & x & x & & x & x \\
2 & & x & & & & x & x & x \\
3 & x & x & x & & x & & & \\
4 & & & x & x & & x & & \\
5 & x & & & & x & x & & \\
6 & & & & x & & & x & \\
7 & & x & & & x & x & & \\
8 & x & & & & x & & x & x
\end{array}
\qquad
\begin{array}{rl}
\text{pivot}: & \mathbf{p} \\
\text{nonzero}: & x
\end{array}
$$

$$
\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & \mathbf{p} & & & \mathbf{x} & \mathbf{x} & & \mathbf{x} & \mathbf{x} \\
2 & & x & & & & x & x & x \\
3 & \mathbf{x} & x & x & \mathbf{f} & \mathbf{f} & & \mathbf{f} & \mathbf{f} \\
4 & & & x & x & & x & & \\
5 & \mathbf{x} & & & \mathbf{f} & \mathbf{f} & x & \mathbf{f} & \mathbf{f} \\
6 & & & & x & & & x & \\
7 & & x & & & x & x & & \\
8 & \mathbf{x} & & & \mathbf{f} & \mathbf{f} & & \mathbf{f} & \mathbf{f}
\end{array}
\qquad
\begin{array}{rl}
\text{pivot}: & \mathbf{p} \\
\text{nonzero}: & x \\
\text{fill}-\text{in}: & \mathbf{f}
\end{array}
$$

Markowitz Pivot Choice: Example

Markowitz: Choose the pivot with $\min_{i,j} \; f_{ij}$.

$$
\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & x & & & x & x & & x & x \\
2 & & x & & & & x & x & x \\
3 & x & x & x & & x & & & \\
4 & & & x & x & & x & & \\
5 & x & & & & x & x & & \\
6 & & & & \mathbf{p} & & & x & \\
7 & & x & & & x & x & & \\
8 & x & & & & x & & x & x
\end{array}
$$

$$
\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & x & & & \mathbf{x} & x & & \mathbf{f} & x \\
2 & & x & & & & x & x & x \\
3 & x & x & x & & x & & & \\
4 & & & x & \mathbf{x} & & x & \mathbf{f} & \\
5 & x & & & & x & x & & \\
6 & & & & \mathbf{p} & & & \mathbf{x} & \\
7 & & x & & & x & x & & \\
8 & x & & & & x & & x & x
\end{array}
$$

Exploiting Sparsity in Cholesky Factorization

Matrix H and its Cholesky Factor

$$H = \begin{bmatrix} \mathbf{p} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & x & & \\ \mathbf{x} & & x & \\ \mathbf{x} & & & x \end{bmatrix} \Rightarrow L = \begin{bmatrix} x & & & \\ x & x & & \\ x & x & x & \\ x & x & x & x \end{bmatrix}$$

Reordered Matrix H and its Cholesky Factor

$$PHP^T = \begin{bmatrix} x & & & x \\ & x & & x \\ & & x & x \\ x & x & x & x \end{bmatrix} \Rightarrow L = \begin{bmatrix} x & & & \\ & x & & \\ & & x & \\ x & x & x & x \end{bmatrix}$$

From Sparsity to Block-Sparsity:

Sparse Matrix **Block-Sparse Matrix**

$$H = \begin{bmatrix} \mathbf{p} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & x & & \\ \mathbf{x} & & x & \\ \mathbf{x} & & & x \end{bmatrix} \Rightarrow L = \begin{bmatrix} x & & & \\ x & x & & \\ x & x & x & \\ x & x & x & x \end{bmatrix} \qquad \begin{bmatrix} \mathbf{P} & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & & \\ \blacksquare & & \blacksquare & \\ \blacksquare & & & \blacksquare \end{bmatrix} \Rightarrow L = \begin{bmatrix} \blacksquare & & & \\ \blacksquare & \blacksquare & & \\ \blacksquare & \blacksquare & \blacksquare & \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{bmatrix}$$

$$PHP^T = \begin{bmatrix} x & & & x \\ & x & & x \\ & & x & x \\ x & x & x & x \end{bmatrix} \Rightarrow L = \begin{bmatrix} x & & & \\ & x & & \\ & & x & \\ x & x & x & x \end{bmatrix} \qquad \begin{bmatrix} \blacksquare & & & \blacksquare \\ & \blacksquare & & \blacksquare \\ & & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{bmatrix} \Rightarrow L = \begin{bmatrix} \blacksquare & & & \\ & \blacksquare & & \\ & & \blacksquare & \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{bmatrix}$$

Minimum Degree Ordering (MDO)

In symmetric positive definite case:
pivots are chosen from the diagonal and $r_i = c_i$
hence choose the pivot with $\min_i \; r_i$

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

that is, choose a node with the minimum number of neighbours
(a node with the *minimum degree*)
in a graph related to sparsity pattern of the matrix.

Minimum Degree Ordering (MDO)

Sparse Matrix **Pivot h_{11}** **Pivot h_{22}**

$$H = \begin{bmatrix} x & & x & x & x & \\ & x & & & x & \\ x & & x & & & x \\ x & & & x & & x \\ x & x & & & x & \\ & & x & x & & x \end{bmatrix} \qquad \begin{bmatrix} \mathbf{p} & & \mathbf{x} & \mathbf{x} & \mathbf{x} & \\ & x & & & x & \\ \mathbf{x} & & x & \mathbf{f} & \mathbf{f} & x \\ \mathbf{x} & & \mathbf{f} & x & \mathbf{f} & x \\ \mathbf{x} & x & \mathbf{f} & \mathbf{f} & x & \\ & & x & x & & x \end{bmatrix} \qquad \begin{bmatrix} x & & x & x & x & \\ & \mathbf{p} & & & \mathbf{x} & \\ x & & x & & & x \\ x & & & x & & x \\ x & \mathbf{x} & & & x & \\ & & x & x & & x \end{bmatrix}$$

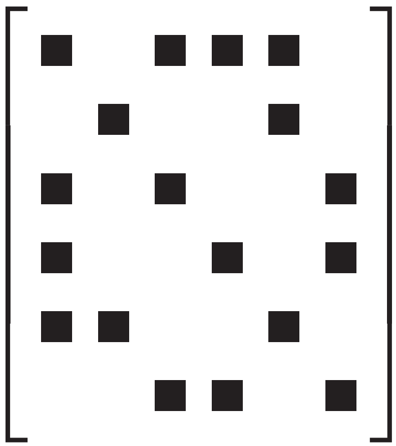
Minimum degree ordering:
choose a diagonal element corresponding to a row with the *minimum* number of nonzeros.
Permute rows and columns of H accordingly.

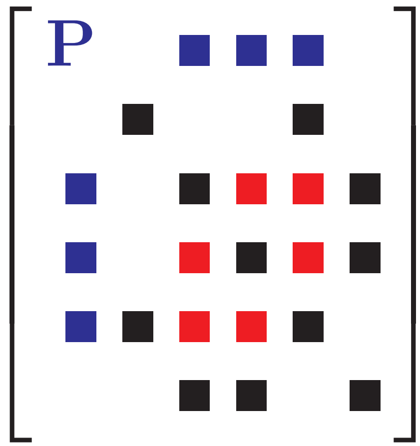
MDO is simply the symmetric version of Markowitz pivot rule.

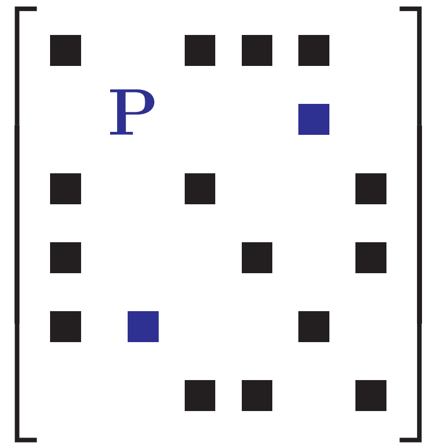
From Sparsity to Block-Sparsity:

Apply minimum degree ordering to **(sparse) blocks**:

Block-Sparse Matrix | **Pivot Block** H_{11} | **Pivot Block** H_{22}

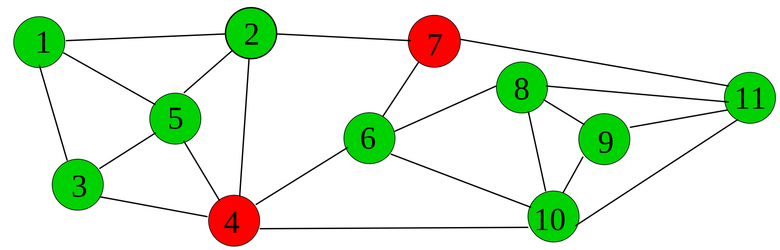
$H =$ 





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.

Nested Dissection:



Original Matrix

	1	2	3	4	5	6	7	8	9	10	11
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

Reordered Matrix

	1	2	3	5	6	8	9	10	11	4	7
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**

Structured Problems

Observation:

Truly large scale problems are not only sparse...
$\rightarrow$ such problems are structured

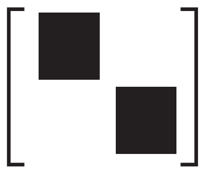
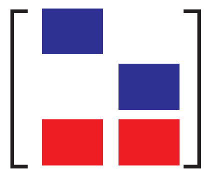
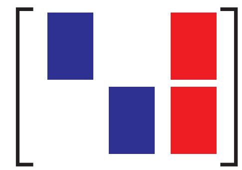
Structure is displayed in:

- **Jacobian matrix** A
- **Hessian matrix** Q

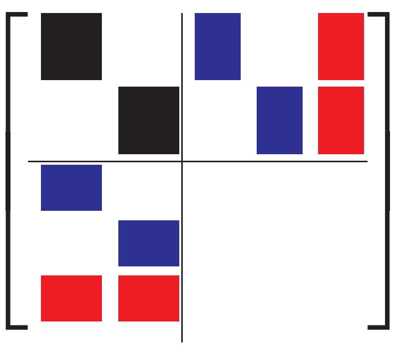
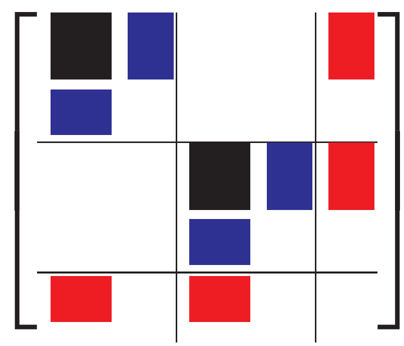
Structure can be exploited in:

- **IPM Algorithm**
- **Linear Algebra of IPM**$\longrightarrow$(focus of this lecture)

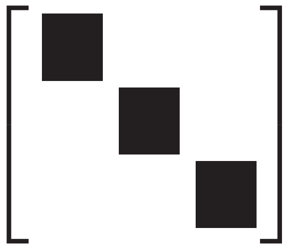
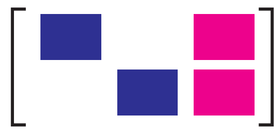
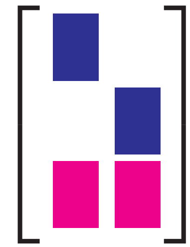
Primal Block-Angular Structure:

$Q =$ , $A =$  *and* $A^T =$ 

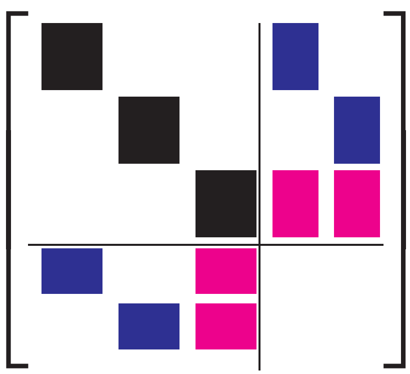
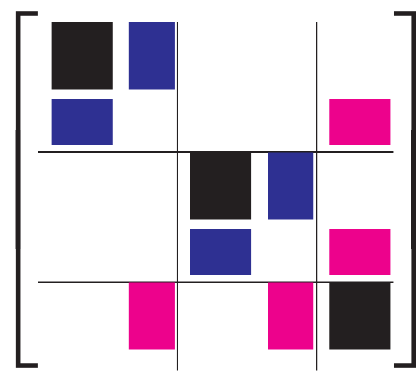
Reorder blocks: $\{1, 3;\ 2, 4;\ 5\}$.

$H =$ , $PHP^T =$ 

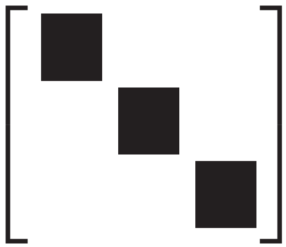
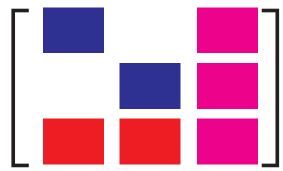
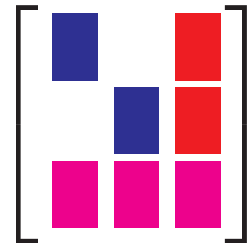
Dual Block-Angular Structure:

$Q =$  , $A =$  *and* $A^T =$ 

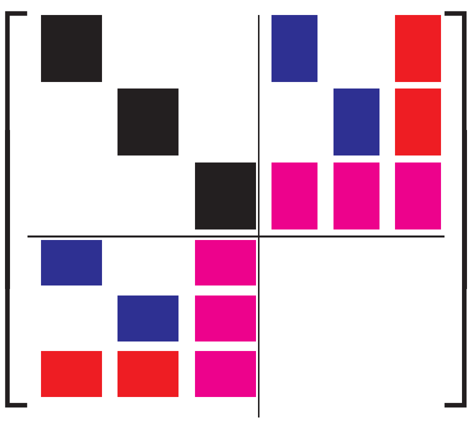
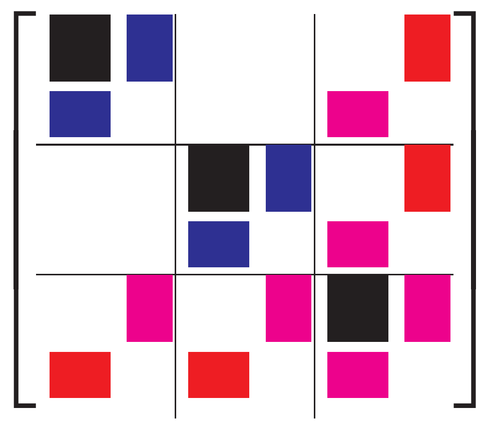
Reorder blocks: $\{1, 4;\ 2, 5;\ 3\}$.

$H =$  , $PHP^T =$ 

Row & Col Bordered Block-Diag Structure:

$Q =$  , $A =$  *and* $A^T =$ 

Reorder blocks: $\{1, 4;\ 2, 5;\ 3, 6\}$.

$H =$  , $PHP^T =$ 

Example: Bordered Block-Diagonal Structure

$$
\underbrace{\begin{pmatrix} \Phi_1 & & & B_1^\top \\ & \ddots & & \vdots \\ & & \Phi_n & B_n^\top \\ B_1 & \dots & B_n & \Phi_0 \end{pmatrix}}_{\Phi} =
$$

$$
= \underbrace{\begin{pmatrix} L_1 & & & \\ & \ddots & & \\ & & L_n & \\ L_{1,0} & \dots & L_{n,0} & L_0 \end{pmatrix}}_{L}
\underbrace{\begin{pmatrix} D_1 & & & \\ & \ddots & & \\ & & D_n & \\ & & & D_0 \end{pmatrix}}_{D}
\underbrace{\begin{pmatrix} L_1^\top & & & L_{1,0}^\top \\ & \ddots & & \vdots \\ & & L_n^\top & L_{n,0}^\top \\ & & & L_0^\top \end{pmatrix}}_{L^\top}
$$

The blocks Φ_i, $i = 0, 1, ..., n$ are KKT systems.

Example: Bordered Block-Diagonal Structure

- Cholesky-like factors obtained by Schur-complement:

$$\begin{aligned} \Phi_i &= L_i D_i L_i^\top \\ L_{i,0} &= B_i L_i^{-\top} D_i^{-1}, \quad i = 1..n \\ C &= \Phi_0 - \textstyle\sum_{i=1}^n L_{i,0} D_i L_{i,0}^\top \;=\; L_0 D_0 L_0^\top \end{aligned}$$

- And the system $\Phi x = b$ is solved by

$$\begin{aligned} z_i &= L_i^{-1} b_i \\ z_0 &= L_0^{-1}(b_0 - \sum L_{i,0} z_i) \\ y_i &= D_i^{-1} z_i \\ x_0 &= L_0^{-\top} y_0 \\ x_i &= L_i^{-\top}(y_i - L_{i,0}^\top x_0) \end{aligned}$$

- Operations (Cholesky, Solve, Product) performed on sub-blocks

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.

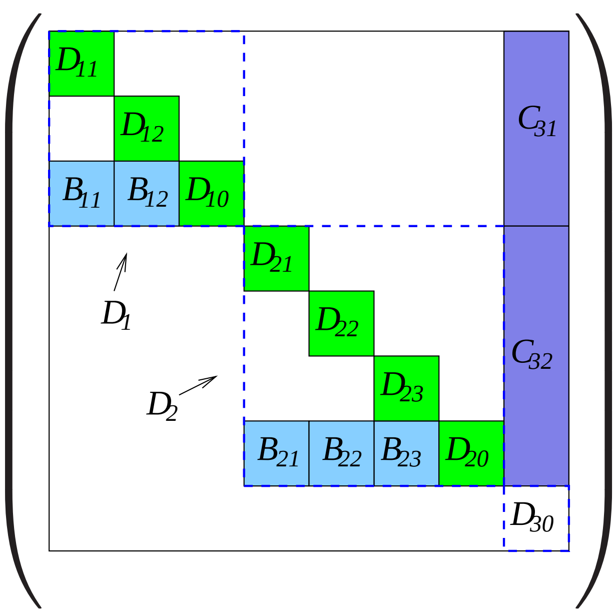
Assumptions:

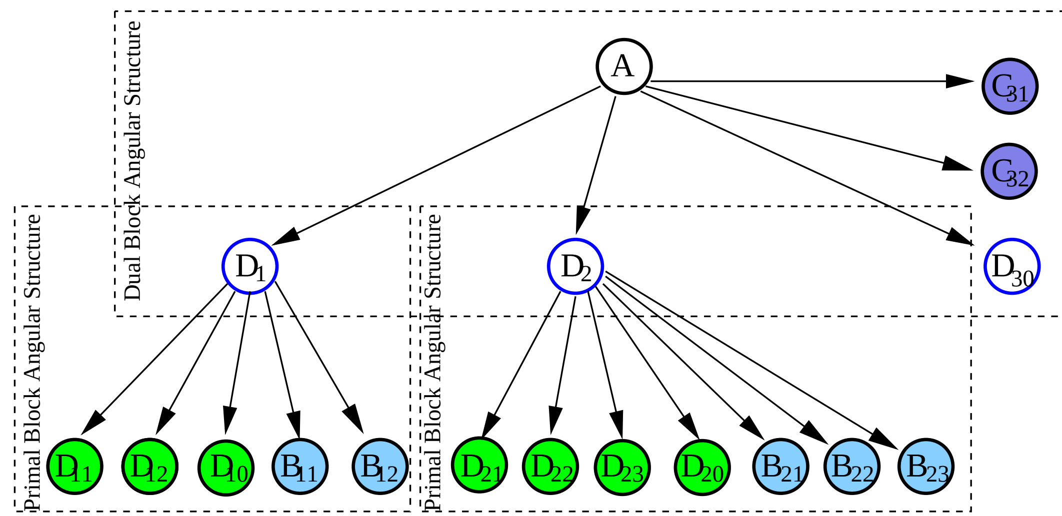
Q and A are block-structured

Linear Algebra of IPMs

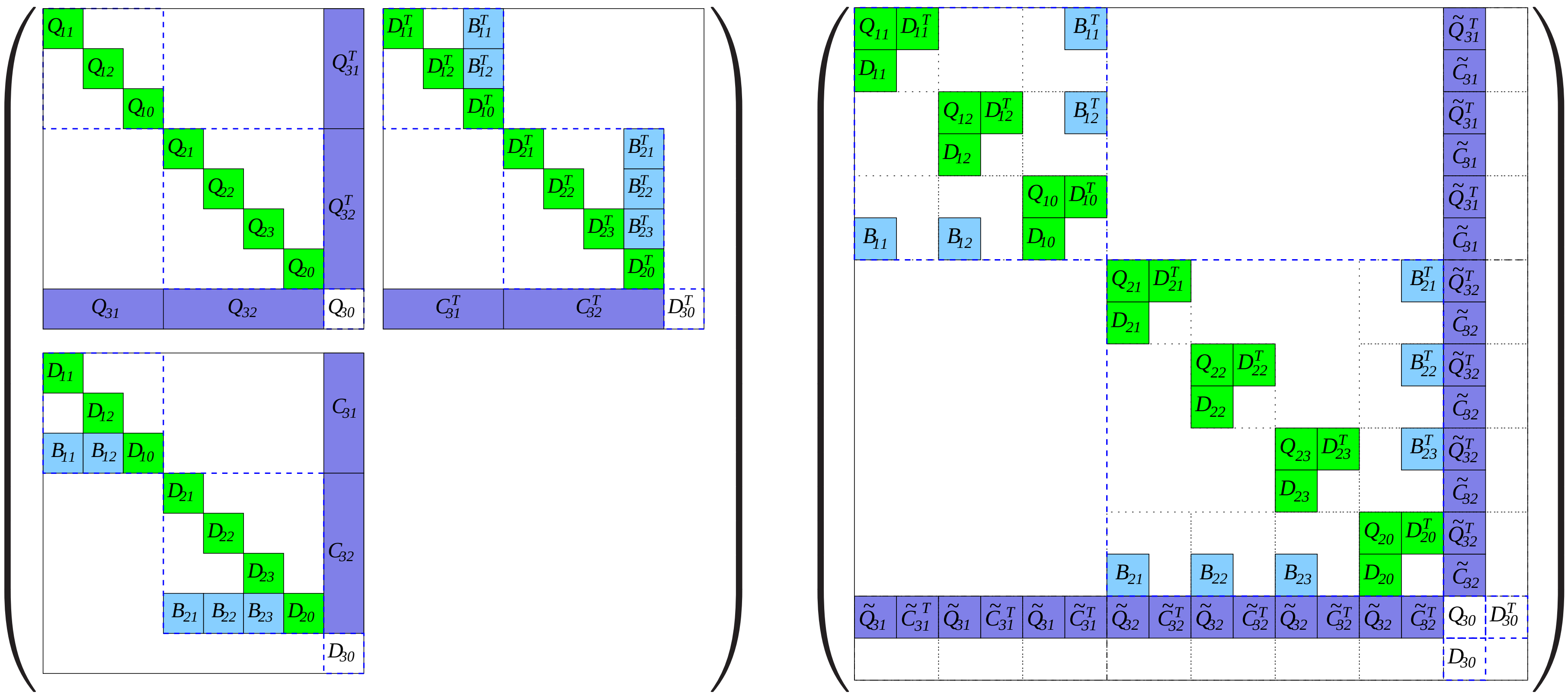
$$\underbrace{\begin{bmatrix} -Q-\Theta_P^{-1} & A^\top \\ A & \Theta_D \end{bmatrix}}_{\Phi\,(NLP)} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix}$$

Tree representation of matrix A:





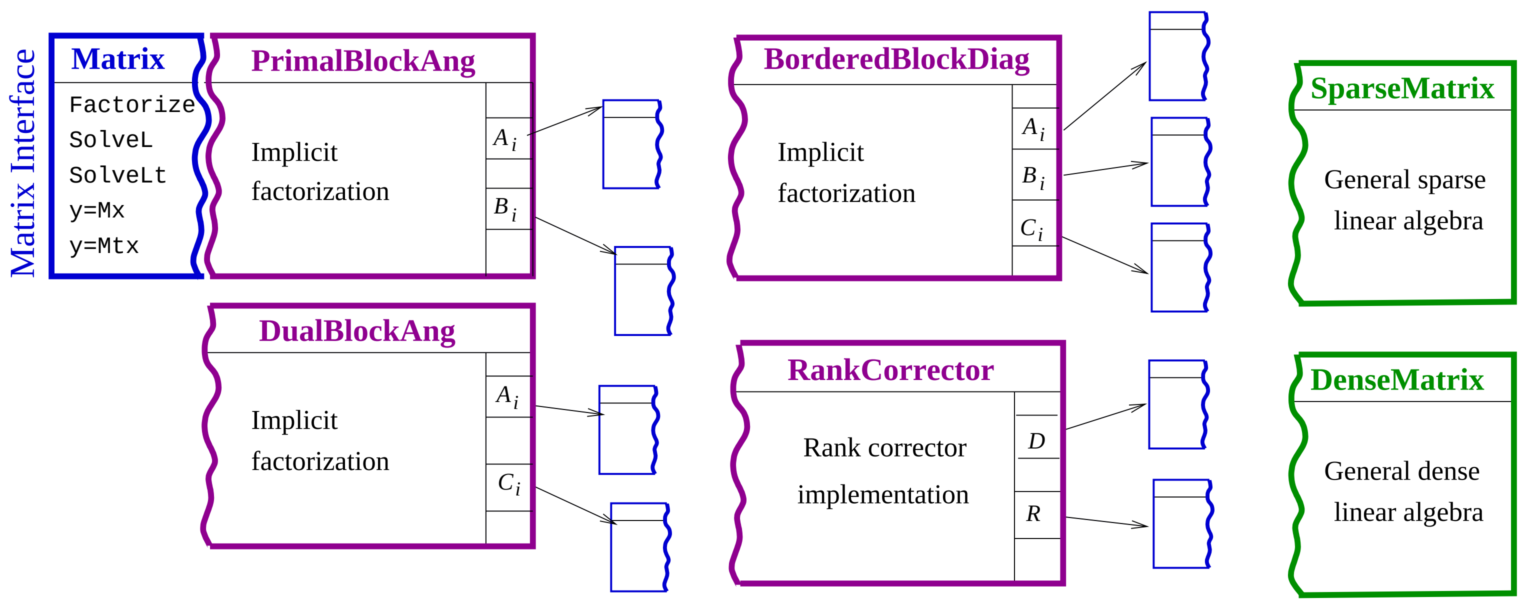
Structures of A and Q imply structure of Φ:



$$\begin{pmatrix} Q & A^\top \\ A & 0 \end{pmatrix} \qquad\qquad P \begin{pmatrix} Q & A^\top \\ A & 0 \end{pmatrix} P^{-1}$$

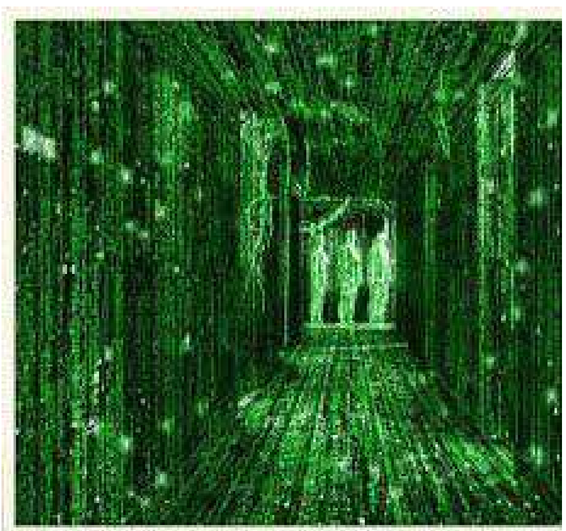
OOPS: Object-oriented linear algebra for IPM

- 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

OOPS: Matrix Revolutions, Matrix Reloaded

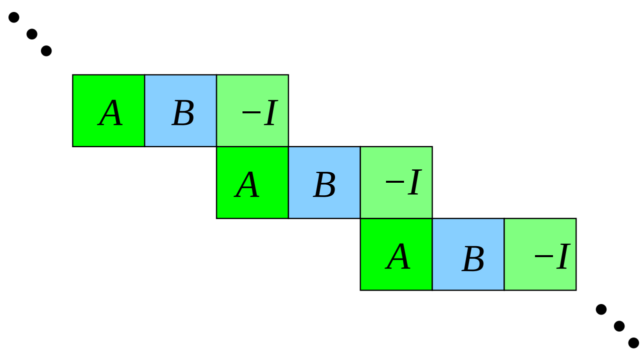


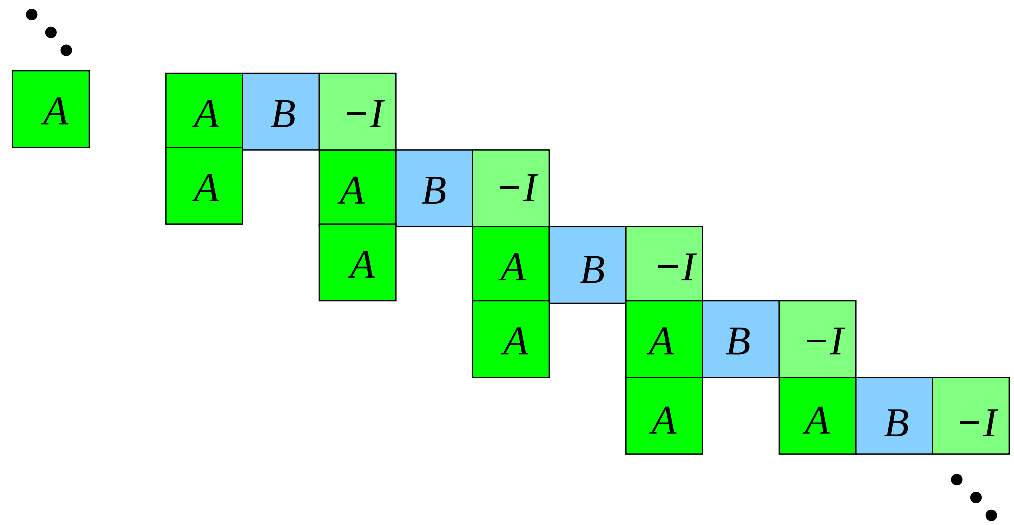
Structured Problems

... are present everywhere.

Sources of Structure

Dynamics → Staircase structure



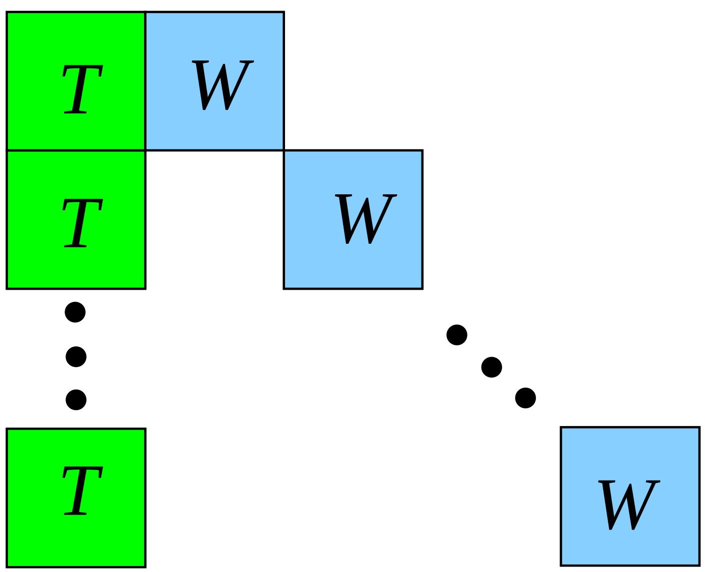


$$x_{t+1} = A_t x_t + B_t u_t$$

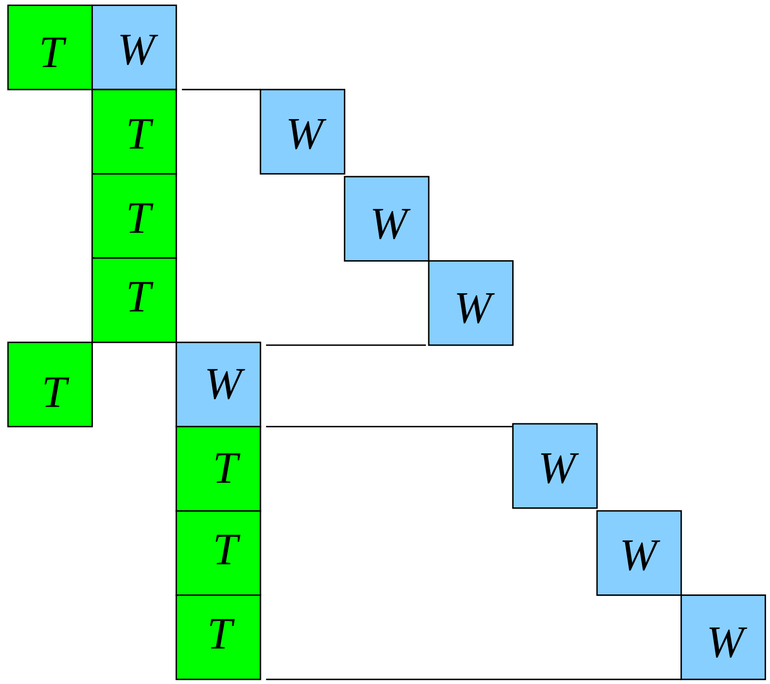
$$x_{t+1} = A_t^{t+1} x_t + \ldots + A_{t-p}^{t+1} x_{t-p} + B_t u_t$$

Sources of Structure

Uncertainty $\rightarrow$ Block-angular structure



$$T_i x^1 + W_i\, y_i = b_i$$

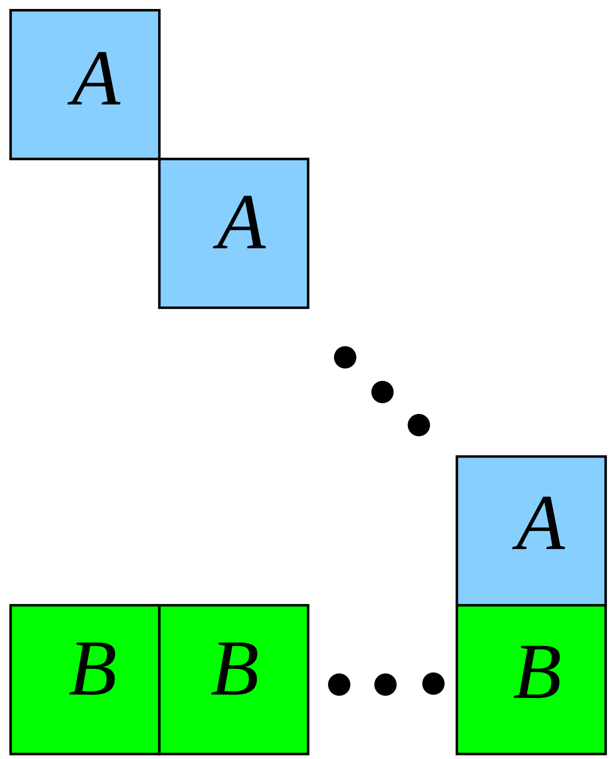


$$T_{l_t} x_{a(l_t)} + W_{l_t}\, x_{l_t} = b_{l_t}$$

Sources of Structure

Common resource constraint

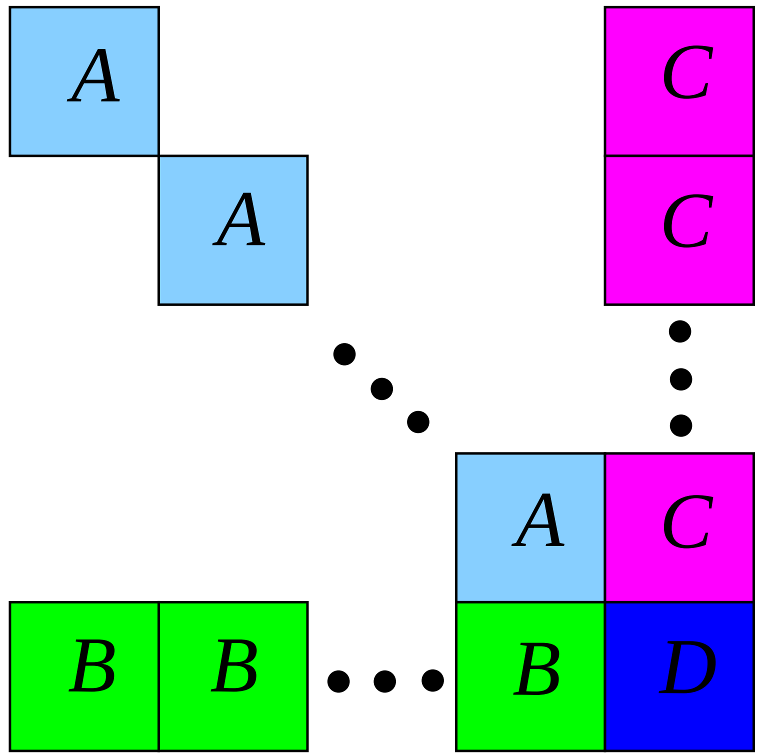
$\sum_{i=1}^{k} B_i x_i = b$ → **Dantzig-Wolfe structure**



Sources of Structure

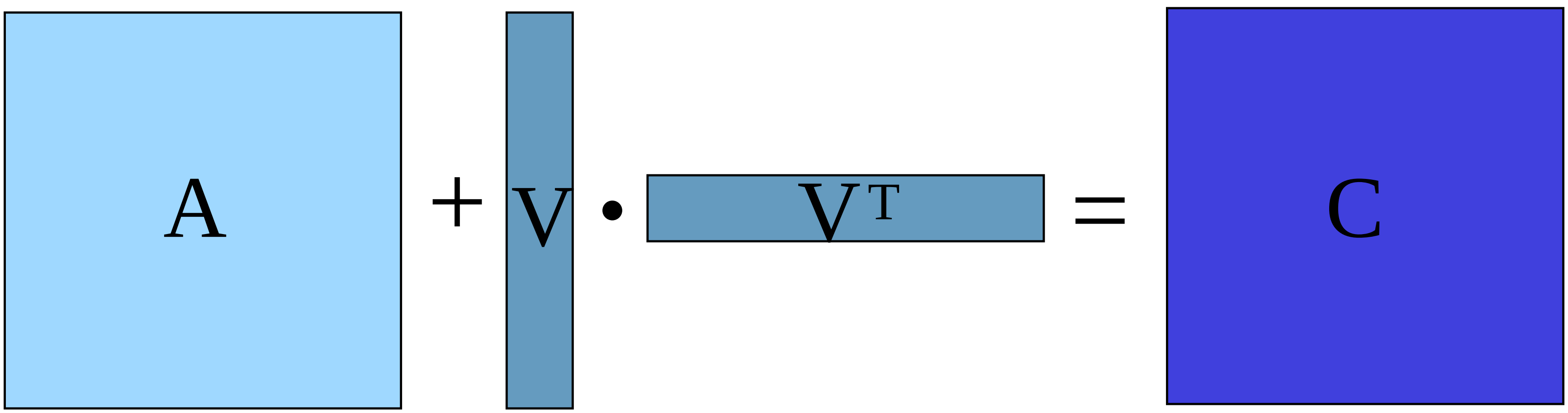
Other types of **near-separability**

→ **Row and column bordered block-diagonal structure**



Sources of Structure

(low) **rank-corrector**
$A + VV^T = C$



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

Example Applications:

- **financial planning problems (nonlinear risk measures)**
- **machine learning (nonlinear kernels in SVMs)**

Financial Planning Problems (ALM)

- A set of assets $\mathcal{J} = \{1..J\}$ given (bonds, stock, real estate)
- At every stage $t = 0..T-1$ 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
- minimize the associated risk

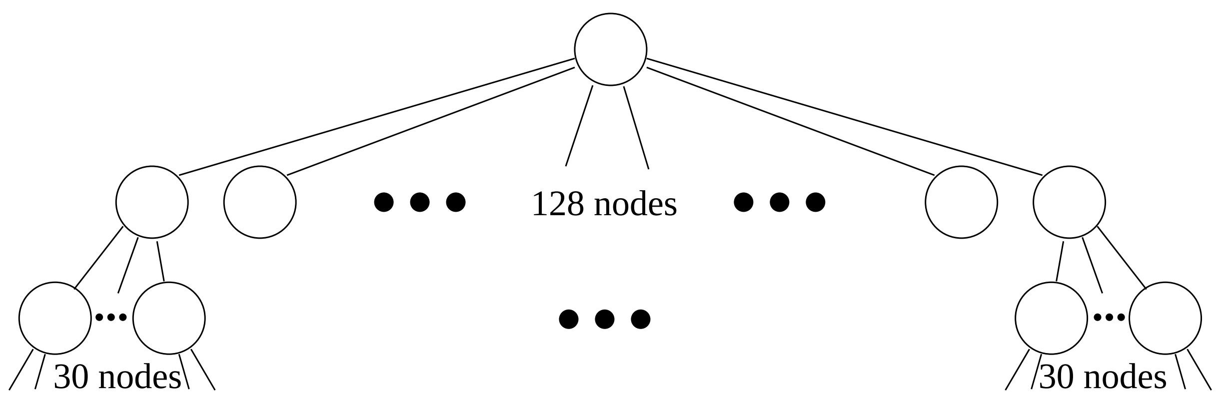
$\Rightarrow$ Mean Variance formulation: $\max \mathbb{E}(X) - \rho \mathrm{Var}(X)$

$\Rightarrow$ Stochastic Program: $\Rightarrow$ formulate deterministic equivalent

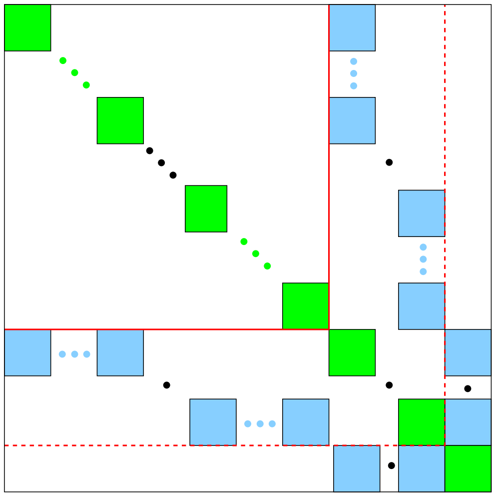
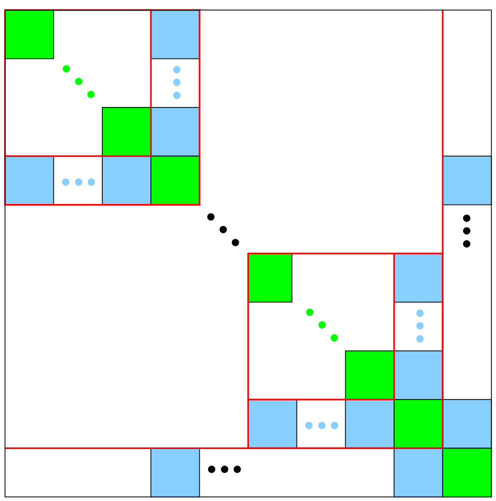
- standard QP, but huge
- extentions: **nonlinear risk measures** (log utility, skewness)

ALM: Largest Problem Attempted

- Optimization of 21 assets (stock market indices) 7 time stages.
- Using multistage stochastic programming
  Scenario tree geometry: 128-30-16-10-5-4 $\Rightarrow$ 16M scenarios.
- 3840 second level nodes with 350.000 variables each.
- Scenario Tree generated using geometric Brownian motion.
- $\Rightarrow$ 1.01 billion variables, 353 million constraints



Sparsity of Linear Algebra

-  $\Rightarrow$
  - $63 + 128 \times 63 = 8127$ columns for Schur-complement
  - Prohibitively expensive
-  $\Rightarrow$
  - Need facility to exploit nested structure
  - Need to be careful that Schur-complement calculations stay sparse on second level

Results (ALM: Mean-Variance QP formulation):

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

The problem with

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

was solved in 50 minutes using 1280 procs.

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

$\longrightarrow$ One IPM iteration takes less than a minute.

Support Vector Machines:

Formulated as the (dual) quadratic program:

$$\begin{array}{ll} \min & -e^T y + \frac{1}{2} y^T K y, \\ \text{s.t.} & d^T y = 0, \\ & 0 \le y \le \lambda e. \end{array}$$

Ferris & Munson, *SIOPT* 13 (2003) 783-804.

Kernel function $K(x,z) = \langle \phi(x), \phi(z) \rangle$,
where ϕ is a (nonlinear) mapping from X to feature space F
Matrix K: $K_{ij} = K(x_i, x_j)$

Linear Kernel $K(x,z) = x^T z.$
Polynomial Kernel $K(x,z) = (x^T z + 1)^d.$
Gaussian Kernel $K(x,z) = e^{-\gamma \|x-z\|^2}.$

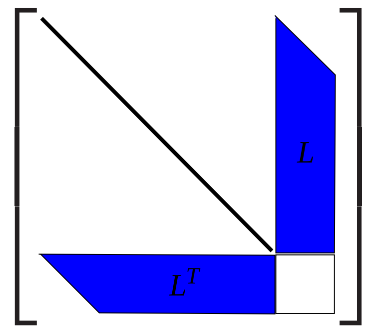
SVMs with Nonlinear Kernels:

K is very large and dense!
Approximate:

$$K \approx LL^T \quad \text{or} \quad K \approx LL^T + D$$

Introduce $v = L^T y$ and get a separable QP:

$$\begin{array}{ll} \min & -e^T y + \frac{1}{2} v^T v + \frac{1}{2} y^T D y, \\ \text{s.t.} & d^T y = 0, \\ & v - L^T y = 0, \\ & 0 \le y \le \lambda e. \end{array}$$

$H =$ 

Structure can be exploited in:

- **Linear Algebra of IPM**
  Kristian Woodsend: PhD Thesis, Edinburgh 2009.

References

- **Gondzio and Sarkissian**, Parallel interior point solver for structured linear programs, *Math Prog* 96 (2003) 561-584.
- **Gondzio and Grothey**, Reoptimization with the primal-dual IPM, *SIAM J. on Optimization* 13 (2003) 842-864.
- **Gondzio and Grothey**, Parallel IPM solver for structured QPs: application to financial planning problems, *Annals of Oper Res* 152 (2007) 319-339.
- **Woodsend and Gondzio**, Hybrid MPI/OpenMP parallel linear support vector machine training, *J. of Machine Learning Research* 20 (2009) 1937-1953.

Papers available: `http://www.maths.ed.ac.uk/~gondzio/`

OOPS: Object-Oriented Parallel Solver

`http://www.maths.ed.ac.uk/~gondzio/parallel/solver.html`

Conclusions:

Interior Point Methods

→ **are well-suited to Large Scale Optimization**

Direct Methods

→ **are well-suited to structure exploitation**

Use IPMs in your research!

Implementation of IPMs

Andersen, Gondzio, Mészáros and Xu
Implementation of IPMs for large scale LP,
in: *Interior Point Methods in Mathematical Programming*,
T. Terlaky (ed.), Kluwer Academic Publishers, 1996, pp. 189–252.

Altman and Gondzio
Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization, *Optimization Methods and Software*, 11-12 (1999), pp 275–302.

Recent Survey on IPMs (easy reading)

Gondzio
Interior point methods 25 years later,
European J. of Operational Research 218 (2012) 587–601.
`http://www.maths.ed.ac.uk/~gondzio/reports/ipmXXV.html`