School of Mathematics

Massively Parallel Implementation of Interior Point Methods for Very Large Scale Optimization

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

joint work with **Andreas Grothey**

Outline

- Optimality Conditions for LP
- Simplex Method vs Interior Point Method
- IPM Framework: LP, QP, NLP
- Features of Logarithmic Function
- From Sparse to Block-Sparse Problems
- Structured Very Large Optimization Problems
- Object-Oriented Parallel Solver
- Financial Planning Problems: Asset/Liability Models
- Conclusions

Primal-Dual Pair of Linear Programs

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

Lagrangian

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

Optimality Conditions

$$\begin{array}{rcll} Ax & = & b, & \\ A^T y + s & = & c, & \\ XSe & = & 0, & (\text{ i.e., } \ x_j \cdot s_j = 0 \quad \forall j), \\ x & \geq & 0, & \\ s & \geq & 0, & \end{array}$$

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

Complementarity

Recall that the **Simplex Method** works with a partitioned formulation:

LP constraint matrix	$A = [B, N]$,	B is nonsingular
primal variables	$x = (x_B, x_N)$,	
reduced costs	$s = (s_B, s_N)$.	

The simplex method maintains the complementarity of primal and dual solutions:

$$x_j \cdot s_j = 0 \quad \forall j = 1, 2, ..., n.$$

For **basic** variables, $s_B = 0$ and

$$(x_B)_j \cdot (s_B)_j = 0 \quad \forall j \in \mathcal{B}.$$

For **non-basic** variables, $x_N = 0$ hence

$$(x_N)_j \cdot (s_N)_j = 0 \quad \forall j \in \mathcal{N}.$$

What's wrong with the Simplex Method?

A **vertex** is defined by a set of n equations:

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

The linear program with m constraints and n variables $(n \geq m)$ has at most

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

vertices and the simplex method can make a non-polynomial number of iterations to reach the optimality.

V. Klee and G. Minty's example LP: simplex method needs 2^n iterations.

How good is the simplex algorithm,
in: Inequalities-III, O. Shisha, ed., Academic Press, 1972, 159–175.

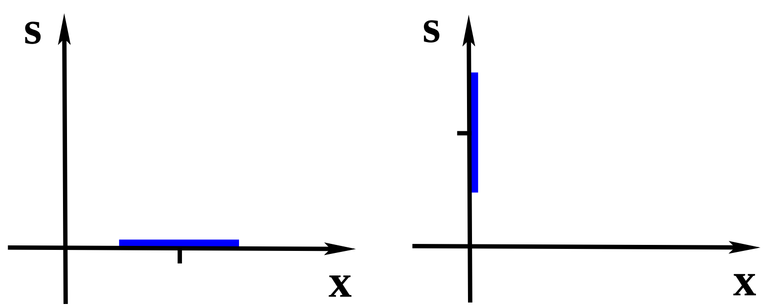
First Order Optimality Conditions

Simplex Method:

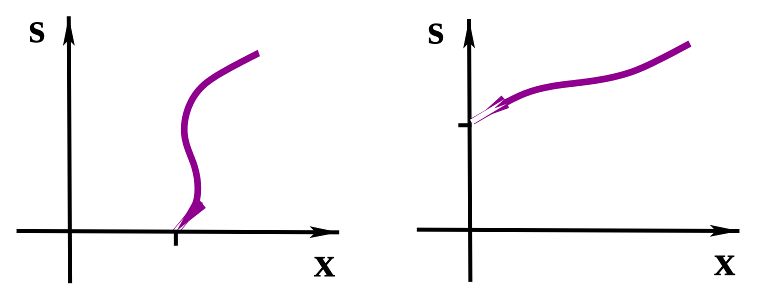
$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= 0 \\ x,\ s &\geq 0. \end{aligned}$$

Interior Point Method:

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



Basic: x > 0, s = 0 Nonbasic: x = 0, s > 0



"Basic": x > 0, s = 0 "Nonbasic": x = 0, s > 0

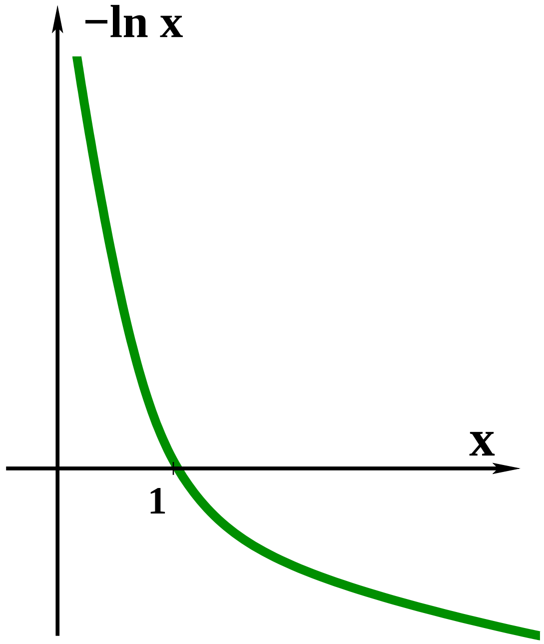
Theory: IPMs converge in $\mathcal{O}(\sqrt{n})$ or $\mathcal{O}(n)$ iterations

Practice: IPMs converge in $\mathcal{O}(\log n)$ iterations

... but one iteration may be expensive!

Logarithmic barrier $-\ln x_j$

"replaces" the inequality $x_j \geq 0$.



Observe that

$$\min \ \mathrm{e}^{-\sum_{j=1}^{n} \ln x_j} \quad \Longleftrightarrow \quad \max \prod_{j=1}^{n} x_j$$

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.

Use Logarithmic Barrier

Primal Problem

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

Dual Problem

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + s = c, \\ & s \geq 0. \end{array}$$

Primal Barrrier Problem

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

Dual Barrrier Problem

$$\begin{array}{ll} \max & b^T y + \sum_{j=1}^{n} \ln s_j \\ \text{s.t.} & A^T y + s = c, \end{array}$$

Primal Barrier Program:

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

Lagrangian:

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

Stationarity:

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

Denote: $s = \mu X^{-1} e, \quad$ i.e. $\quad XSe = \mu e.$

The **First Order Optimality Conditions** are:

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s & = & c, \\ XSe & = & \mu e \\ (x, s) & > & 0. \end{array}$$

Newton Method

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

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

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

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:

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

IPM for QP

$$\begin{array}{ll} \min & c^T x + \frac{1}{2} x^T \mathbf{Q}\, x \\ \text{s.t.} & Ax = b, \\ & x \geq 0. \end{array} \quad \longrightarrow \quad \begin{array}{ll} \min & c^T x + \frac{1}{2} x^T \mathbf{Q}\, x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} & Ax = b, \end{array}$$

The first order conditions (for the barrier problem)

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

Newton direction

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

Augmented system

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

IPM for NLP

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

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

The first order conditions (for the barrier problem)

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

Newton direction

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

Augmented system

$$\begin{bmatrix} Q(x,y) & A(x)^T \\ A(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - A(x)^T y \\ -g(x) - \mu Y^{-1} e \end{bmatrix} \quad \text{where} \quad \begin{aligned} A(x) &= \nabla g \\ Q(x,y) &= \nabla^2_{xx} L \end{aligned}$$

Optimality Conditions:

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

Newton Direction:

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

Linear Algebra involves an (ill-conditioned) scaling matrix $\Theta = XS^{-1}$.

Augmented System vs Normal Equations

LP

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

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

QP

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

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

NLP

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

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

Direct Methods: Symmetric LDL^T Factorization

Indefinite	Quasidefinite	Positive Definite
$H = \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$	$H = \begin{bmatrix} Q & A^T \\ A & -R \end{bmatrix}$	$H = AQ^{-1}A^T$
2×2 pivots needed	1×1 pivots (any sign)	1×1 pivots (positive)
$\begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & a \\ a & d \end{bmatrix}$	strongly factorizable	easy

Vanderbei, *SIOPT* (1995): 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**:
D has n negative pivots and m positive pivots.

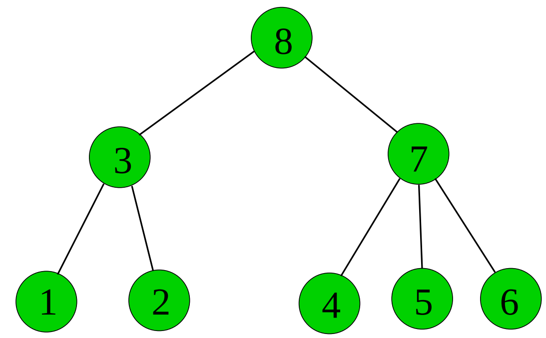
OOPS (Object Oriented Parallel Solver)

- Mantra: "Truly large scale problems are not only sparse but structured" (due to e.g. dynamics, uncertainty, spatial distribution etc.)
- Exploiting structure is key to building efficient IPMs for large problems:
  - Faster linear algebra
  - Reduced memory use
  - Possibility to exploit (massive) parallelism
  - **We assume that structure is known!** $\Rightarrow$ no automatic detection.
- OOPS currently solves LP/QP problems.
- Simple sequential-QP scheme solves nonlinear ALM models

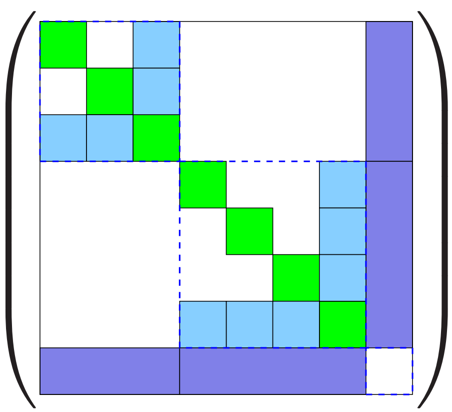
OOPS: (Block) Elimination Trees:

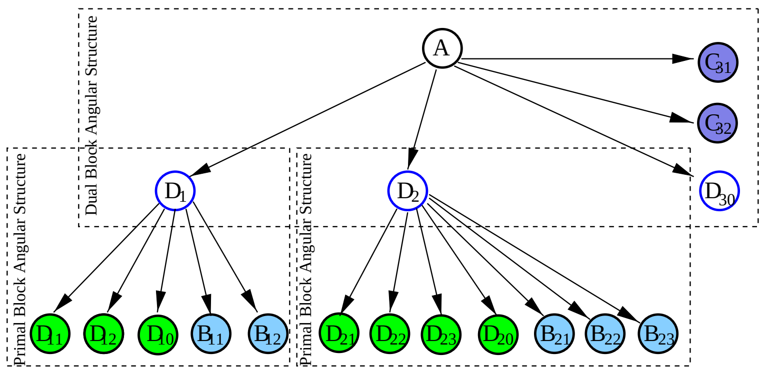
Elimination tree orders rows/columns for elimination with minimum fill-in:

$$\begin{bmatrix} x & & x & & & & & x \\ & x & x & & & & & x \\ x & x & x & & & & & x \\ & & & x & & & x & x \\ & & & & x & & x & x \\ & & & & & x & x & x \\ & & & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{bmatrix}$$



Elimination Tree can be extended to Block Elimination Tree





$\Rightarrow$ Organisation of linear algebra, Parallelism

Minimum Degree Ordering

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

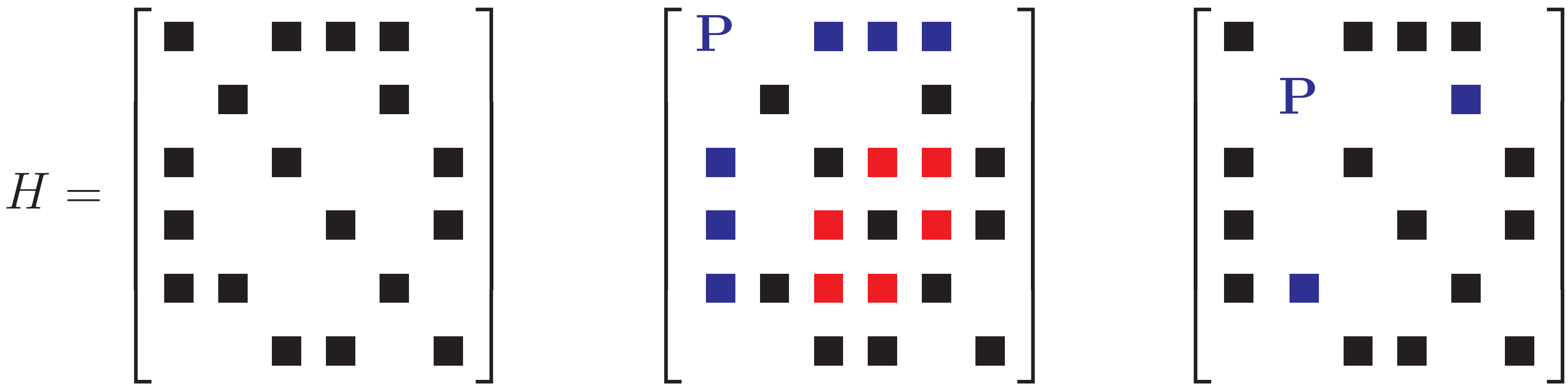
$$H = \begin{bmatrix} x & & x & x & x & \\ & x & & & x & \\ x & & x & & & x \\ x & & & x & & x \\ x & x & & & x & \\ & & x & x & & x \end{bmatrix} \qquad \begin{bmatrix} \mathbf{p} & & \mathbf{x} & \mathbf{x} & \mathbf{x} & \\ & x & & & x & \\ \mathbf{x} & & x & \mathbf{f} & \mathbf{f} & x \\ \mathbf{x}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

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

From Sparsity to Block-Sparsity:

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

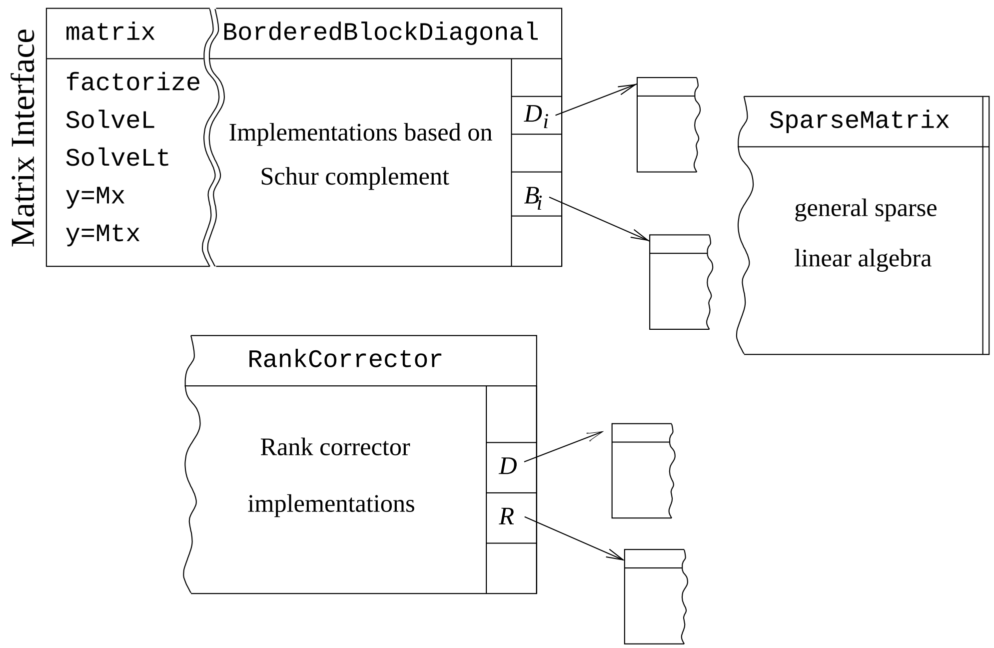
Block-Sparse Matrix **Pivot Block** H_{11} **Pivot Block** H_{22}



choose a diagonal block-pivot corresponding to a block-row with the *min* number of blocks.
Permute block-rows and block-columns of H accordingly.

OOPS: Object-oriented linear algebra implementation

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



⇒ Rebuild *block elimination tree* with matrix interface structures

Application: Asset and Liability Management Problem

- A set of assets $\mathcal{J} = \{1, ..., J\}$ is given (e.g. 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* (but distribution is known).

We have to make investment decisions: what to buy or sell, at which time stage

Objectives:

- maximize the final wealth
- minimize the associated risk

$\Rightarrow$ Mean Variance formulation:

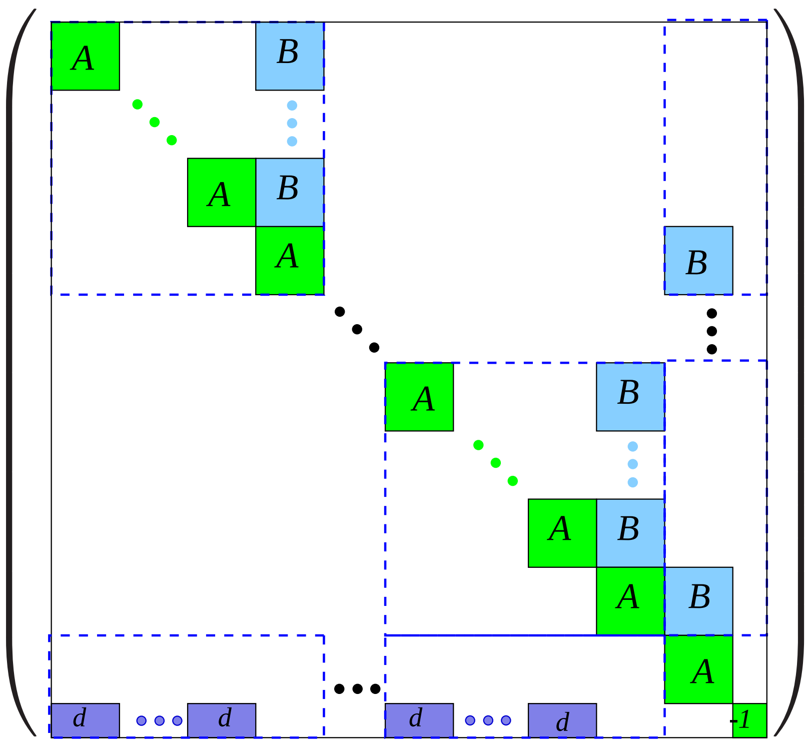
$$\max \mathbb{E}(X) - \rho \text{Var}(X)$$

$\Rightarrow$ Stochastic Program:

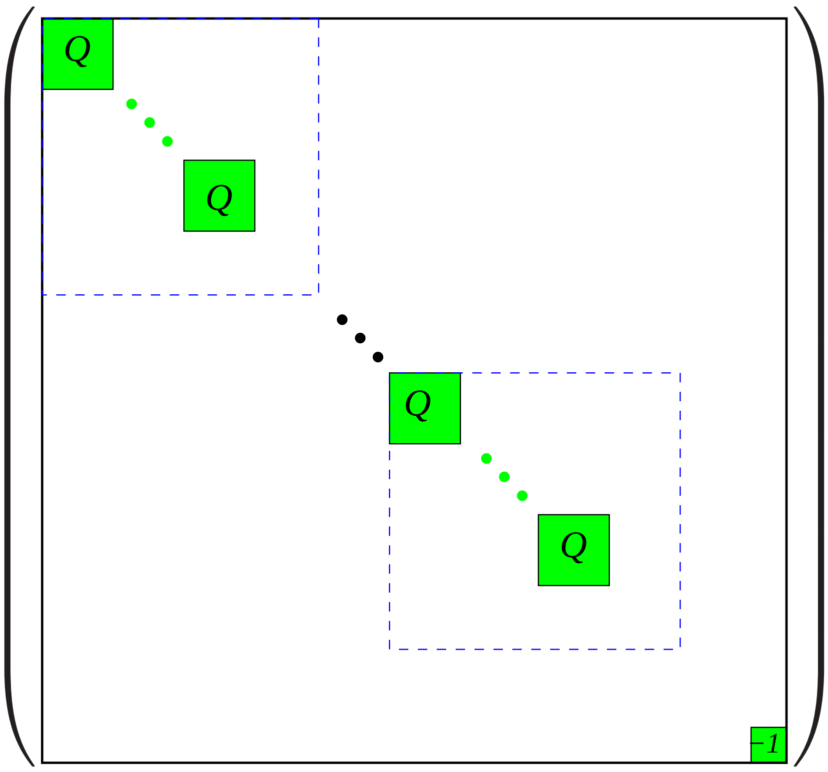
- Can formulate deterministic equivalent problem
- standard QP, but huge

ALM: Structure of matrices A and Q:

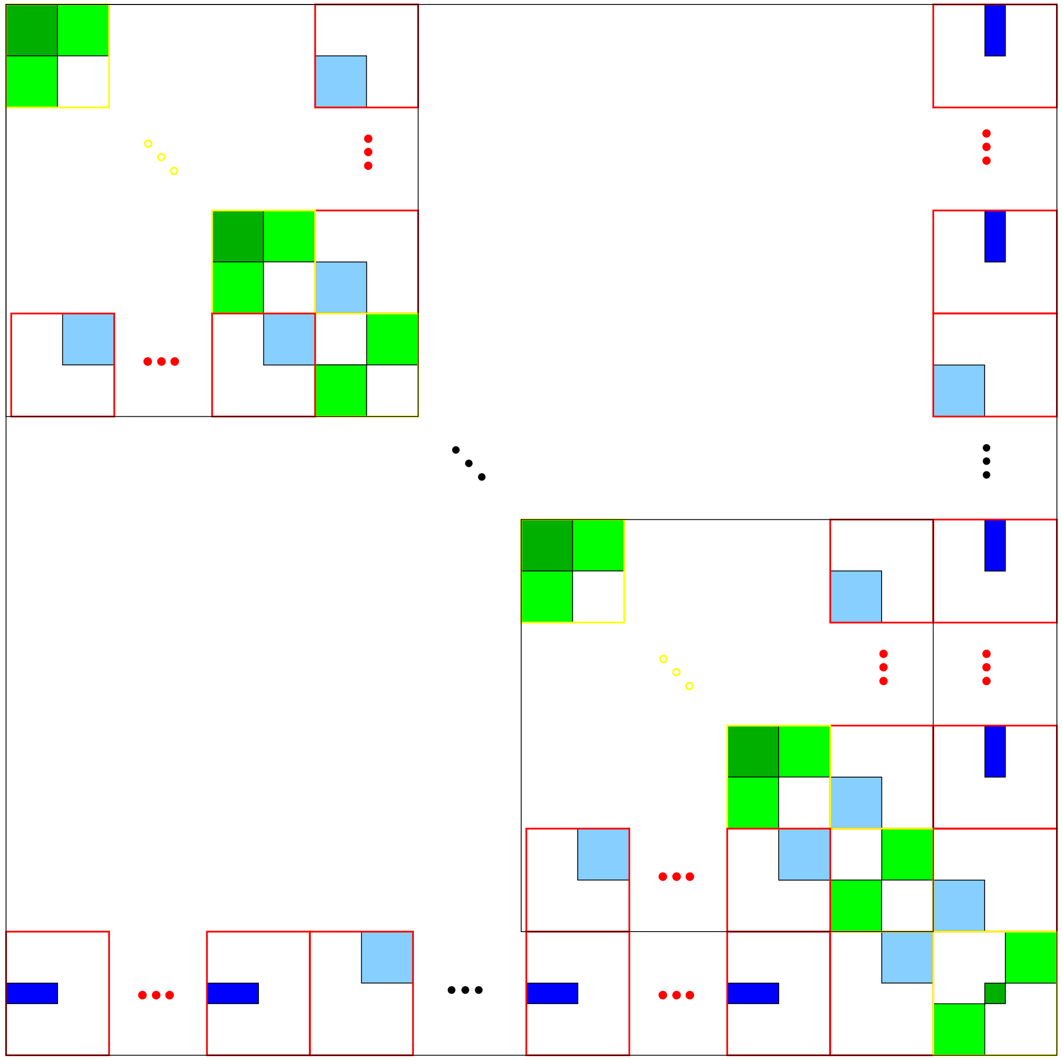
Matrix A



Matrix Q



ALM: Structure of Augmented System matrix:



ALM: Largest Problem Attempted

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

Issues for Massive Parallelism

- Sparsity of multilevel linear Algebra
- Memory Management

BlueGene (Edinburgh, Scotland)

- 2048 Processors
- 0.7GHz, 256Mb
- 4.7 TFlops
- **#64** in `top500.org` list

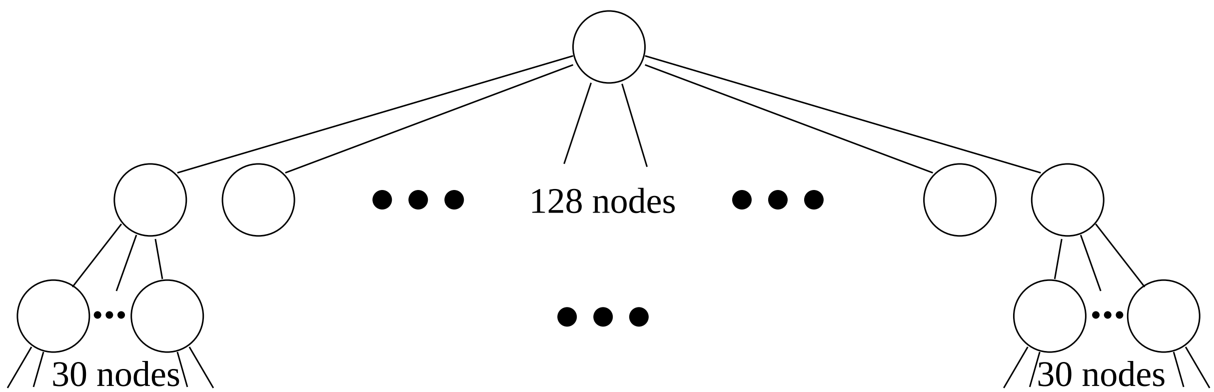
HPCx (Daresbury, England)

- 1600 IBM Power-4 Processors
- 1.7GHz, 800Mb
- 6.2 TFlops
- **#45** in `top500.org` list

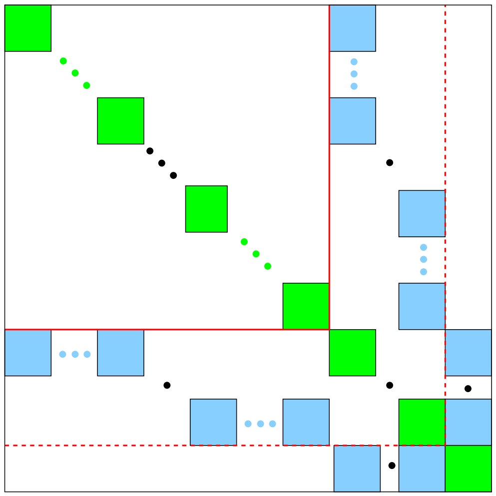
Sparsity of Linear Algebra I

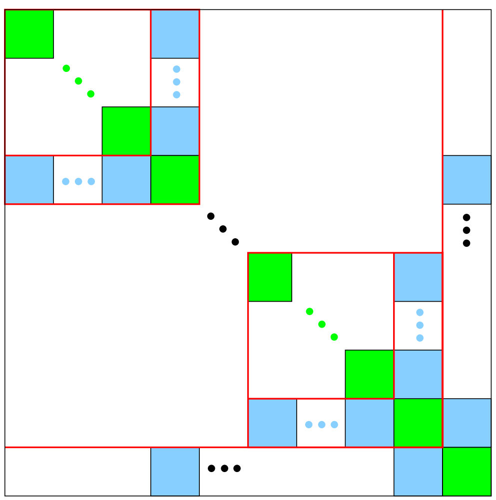
- In ALM problems matrices up to ≈ 500.000 − 1.000.000 variables can be treated as unstructured sparse matrices
- Problem has:
  - 128 first level nodes with 10.000.000 variables each.
  - 3840 second level nodes with 350.000 variables each.

⇒ need to decompose problem at second level
(with 1280 processors ⇒ 3 blocks per processor)



Sparsity of Linear Algebra II

- 
  $\Rightarrow$
  - $63 + 128 \times 63 = 8127$ columns for Schur-complement
  - Prohibitively expensive

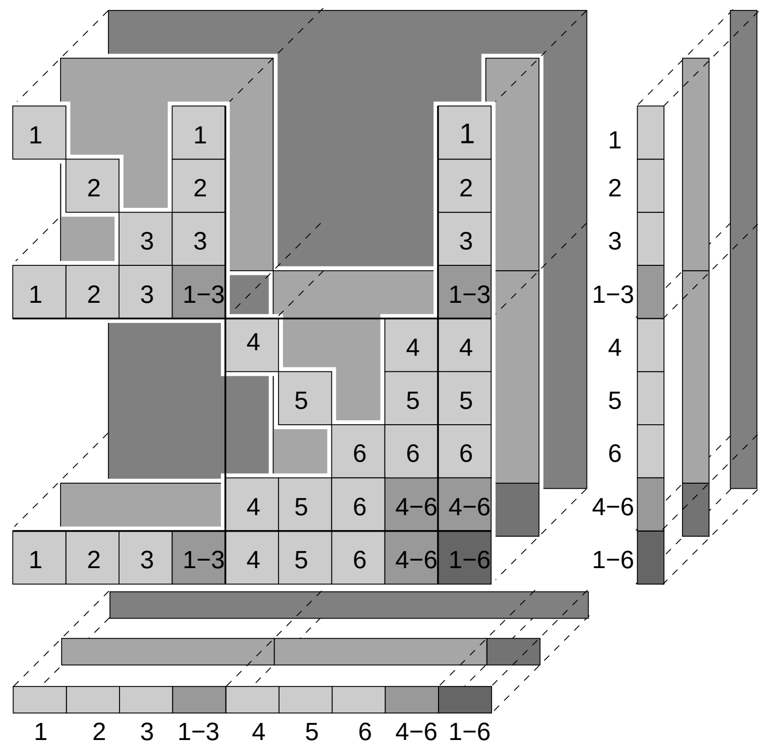
- 
  $\Rightarrow$
  - Need facility to exploit nested structure
  - Need to be careful that Schur-complement calculations stay sparse on second level

Memory Management

- Data for problem requires > 1GB of memory.
  ⇒ need to split information between processors
- To each node in block-elimination tree a set of processors is assigned
- Linear Algebra is implemented so that processors communicate when needed

Distribution of leading matrix blocks among processors implies

- Distribution of subordinate blocks
- Distribution of row/column vector contributions



Results (ALM: Mean-Variance QP formulation):

Problem	Stages	Blk	Assets	Scenarios	Constraints	Variables	iter	time	procs	machine
ALM8	7	128	6	12.831.873	64.159.366	153.982.477	42	3923	512	BlueGene
ALM9	7	64	14	6.415.937	96.239.056	269.469.355	39	4692	512	BlueGene
ALM10	7	128	13	12.831.873	179.646.223	500.443.048	45	6089	1024	BlueGene
ALM11	7	128	21	16.039.809	352.875.799	1.010.507.968	53	3020	1280	HPCx

The problem with

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

solved in 50 minutes using 1280 procs.

Object-Oriented Parallel Solver (OOPS):

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

References:

- J. Gondzio and R. Sarkissian, *Parallel interior point solver for structured linear programs*, **Mathematical Programming** 96 (2003) pp 561–584.
- J. Gondzio and A. Grothey, *Reoptimization with the primal-dual interior point method*, **SIAM J. on Optimization** 13 (2003) pp 842-864.
- J. Gondzio and A. Grothey, *Parallel interior point solver for structured quadratic programs: application to financial planning problems*, Tech. Rep. MS-03-001, School of Maths, University of Edinburgh, April 2003 (to appear in **Annals of OR**).
- J. Gondzio and A. Grothey, *Solving nonlinear portfolio optimization problems with the primal-dual interior point method*, Tech. Rep. MS-04-001, School of Maths, University of Edinburgh, May 2004.
- J. Gondzio and A. Grothey, *Exploiting structure in parallel implementation of interior point methods for optimization*, Tech. Rep. MS-04-004, School of Maths, University of Edinburgh, December 2004.

Conclusions:

- Interior Point Methods are the key optimization technique.
- The theory of IPMs is well understood.
- IPMs demonstrate spectacular efficiency.
- Today IPMs can solve problems of dimension $\mathbf{10^9}$.

IPMs are well-suited to exploit parallelism

Thank you for your attention!

Interior-Point Framework

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

$$x_j \geq 0.$$

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** (damped) 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.

Central Trajectory

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 the **first order optimality conditions**.

The path

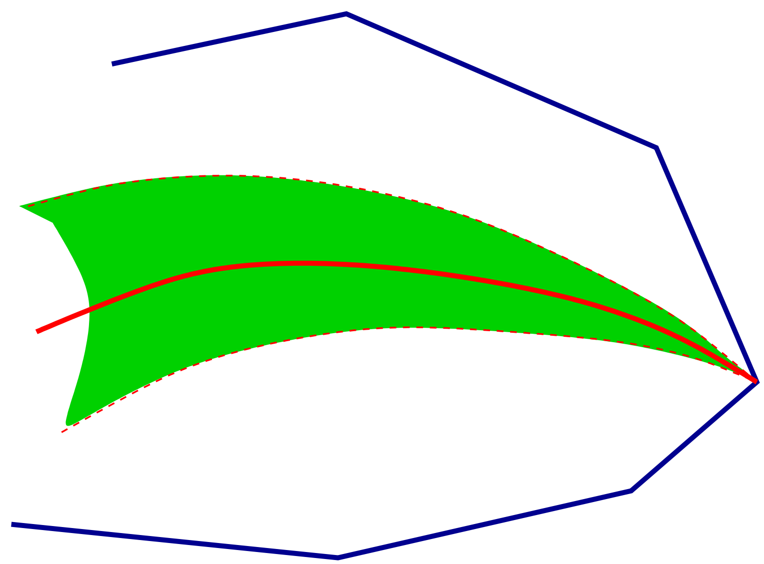
$$\{(x(\mu), y(\mu), s(\mu)) : \mu > 0\}$$

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

Follow the Central Path

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XSe &\approx \mu e, \qquad \text{i.e.} \quad \|XSe - \mu e\| \leq \theta \mu, \end{aligned}$$

where $\theta \in (0,1)$ and the barrier μ satisfies $x^T s = n\mu$.



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

Progress to optimality

Reduce the barrier: $\mu^{k+1} = \sigma\mu^k$, where $\sigma = 1 - \beta/\sqrt{n}$ for some $\beta \in (0,1)$.
Compute 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 make step.

At the new iterate $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$
duality gap is reduced $1 - \beta/\sqrt{n}$ times.

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

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

After $C \cdot \sqrt{n}$ iterations, the reduction is $e^{-C\beta}$.
For sufficiently large constant C the duality gap becomes arbitrarily small.
Hence this algorithm has complexity $\mathcal{O}(\sqrt{n})$.

$\mathcal{O}(\sqrt{n})$ Complexity Result

Theorem

Given $\epsilon > 0$, suppose that a feasible starting point $(x^0, y^0, s^0) \in N_2(0.1)$ satisfies

$$(x^0)^T s^0 = n\mu^0, \text{ where } \mu^0 \leq 1/\epsilon^\kappa,$$

for some positive constant κ.

Then there exists an index K with $K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ such that

$$\mu^k \leq \epsilon, \quad \forall k \geq K.$$

Interior Point Methods

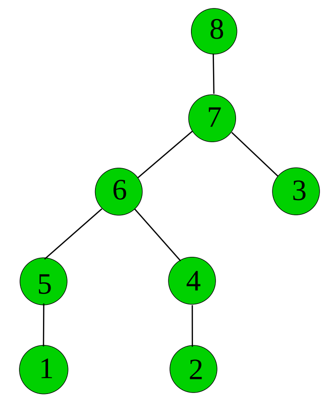
- **Fiacco & McCormick (1968)**
  handling inequality constraints - logarithmic barrier;
  minimization with inequality constraints
  replaced by a sequence of unconstrained minimizations
- **Lagrange (1788)**
  handling equality constraints - multipliers;
  minimization with equality constraints
  replaced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

Marsten, Subramanian, Saltzman, Lustig and Shanno:
"Interior point methods for linear programming:
Just call Newton, Lagrange, and Fiacco and McCormick!",
Interfaces 20 (1990) No 4, pp. 105–116.

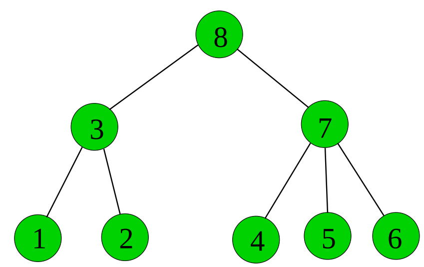
Elimination Tree:

Matrix **Cholesky Factor** **Elimination Tree**

$$\begin{bmatrix} x & & & & x & & x & \\ & x & & x & & x & & \\ & & x & & & & x & x \\ & x & & x & & x & & \\ x & & & & x & x & x & \\ & x & & x & x & x & & \\ x & & x & & x & & x & \\ & & x & & & & & x \end{bmatrix} \quad \begin{bmatrix} x & & & & & & & \\ & x & & & & & & \\ & & x & & & & & \\ & x & & x & & & & \\ x & & & & x & & & \\ & x & & & x & x & & \\ x & & x & & x & f & x & \\ & & x & x & & & f & x \end{bmatrix}$$



$$\begin{bmatrix} x & & x & & & & & x \\ & x & x & & & & & x \\ x & x & x & & & & & x \\ & & & x & & & x & x \\ & & & & x & & x & x \\ & & & & & x & x & x \\ & & & x & x & x & x & x \\ x & x & x & x & x & x & x & x \end{bmatrix} \quad \begin{bmatrix} x & & & & & & & \\ & x & & & & & & \\ x & x & x & & & & & \\ & & & x & & & & \\ & & & & x & & & \\ & & & & & x & & \\ & & & x & x & x & x & \\ x & x & x & x & x & x & x & x \end{bmatrix}$$



From Sparsity to Block-Sparsity:

Sparse Matrix **Block-Sparse Matrix**

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

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

Object-Oriented Parallel Solver $\Rightarrow$ problems of size $\mathbf{10^6}, \mathbf{10^7}, \mathbf{10^8}, \mathbf{10^9}, \ldots$

G. & Sarkissian, *MP* 96 (2003) 561-584.
G. & Grothey, *SIOPT* 13 (2003) 842-864.
G. & Grothey, *AOR* (to appear).