Using Controlled Cholesky Factorization in the Normal Equations System Direct Solution from the Interior Point Methods

Luciana Yoshie Tsuchiya

Co-author: Aurelio Ribeiro Leite de Oliveira

IMECC - Unicamp

1st Brasilian Workshop Interior Point Methods

IPM Main Computational Operation

- Solution of a linear system
- Large dimension
- High degree of sparsity
- Most expensive step

Linear Program Problem

Primal

$$\begin{aligned} &\min\ c^t x \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} &\max\ b^T y \\ &A^T y + z = c \\ &z \geq 0 \end{aligned}$$

where $A^{m \times n}$ has full hank.

Otimality Conditions

- $Ax = b$, (primal feasible)
- $A^t y + z = c$,(dual feasible)
- $XZe = 0$, (complementarity condition)
- $(x, z) > 0$

Newton's System

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^t & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix} \tag{1}$$

- (dx, dy, dz) is the search direction to be computed;
- $r_p = b - Ax$, $r_d = c - A^t y - z$, $r_c = -XZe + \sigma\mu e$;
- $\mu = x^t z/n$ (duality gap);
- $\sigma \in [0, 1]$ (centrality parameter).

Normal Equation System - Direct solution

One of the most common approaches to solve the Newton's system reduces it to the normal equations

$$ADA^T dy = \tilde{r}$$

and apply the Cholesky factorization in the system matrix.

Normal Equation System - Direct solution

- The great disadvantage of Cholesky is the number of generated fill-in.

Normal Equation System - Iterative method

- Iterative method: Preconditioned Conjugate Gradient.
- Preconditioning: $(MADA^tM^t)(M^{-t}dy) = Mr$.
- Hybrid preconditioner: Controlled Cholesky factorization in early iterations and the splitting preconditioner in the later iterations.

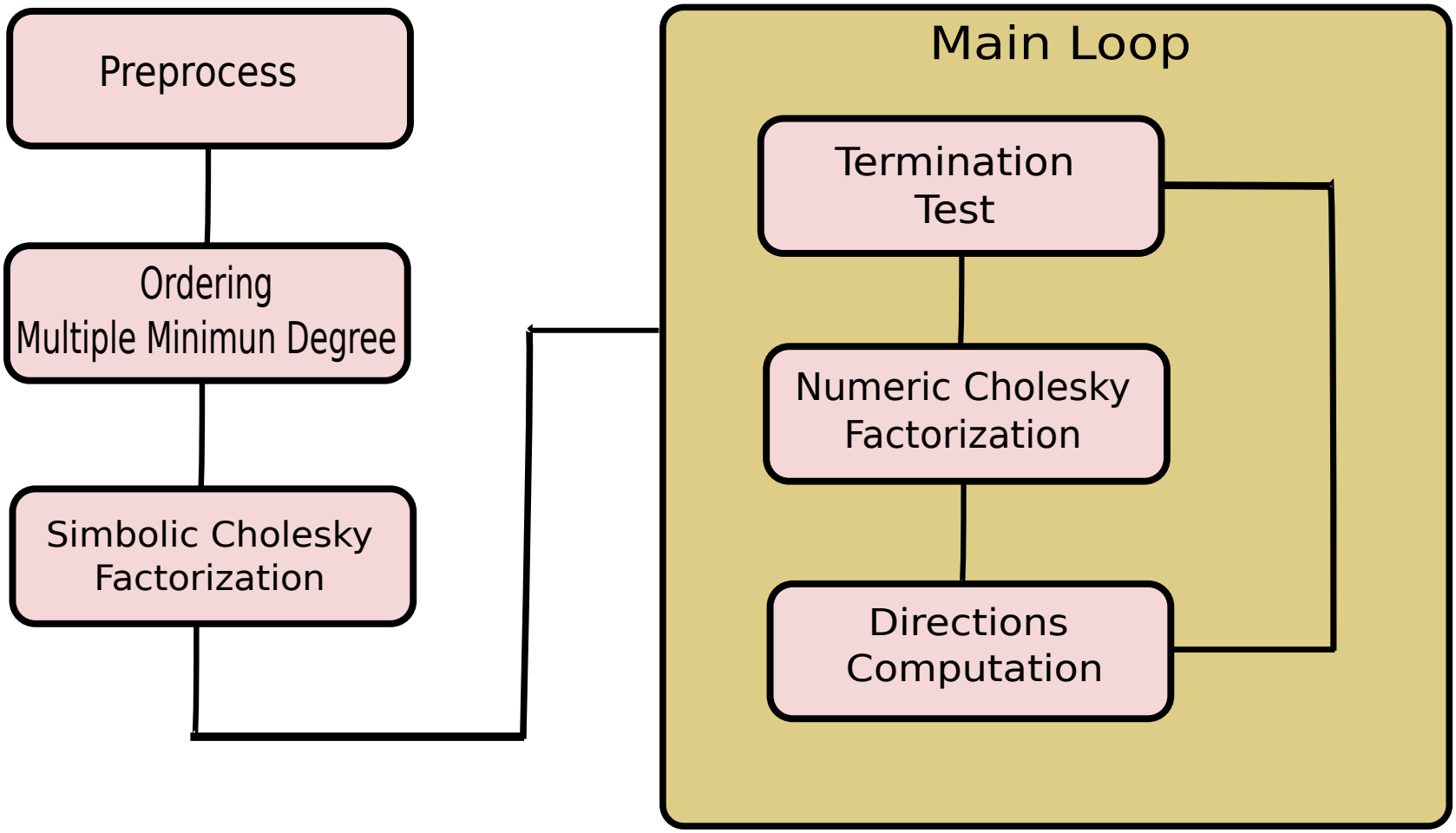
Controlled Cholesky Factorization (CCF)

- Incomplete Cholesky factorization
- $ADA^t = LL^t - R$
- L is less dense than the Cholesky factor.
- The fill-in column is controlled by a parameter η.
- Given the L j column, only the $k_j = \eta + nn_j$ largest nonzeros entries will be kept in the column, where nn_j is the number of the ADA^t matrix j column nonzero entries.

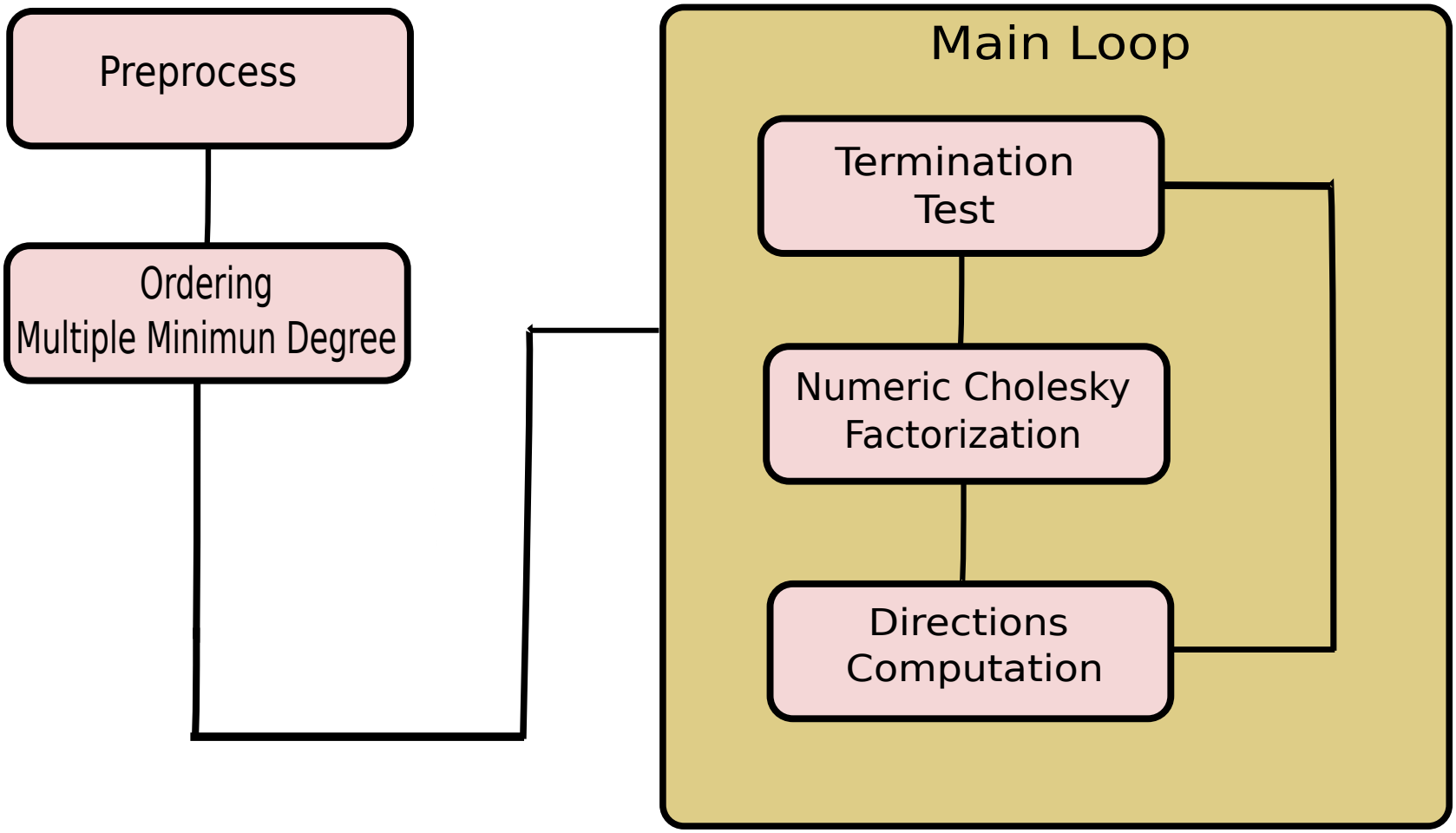
- When $\eta < m$ the CCF computation requires less storage than the Cholesky factorization computation.
- If η is not very large, the CCF computation is faster than the Cholesky computation.

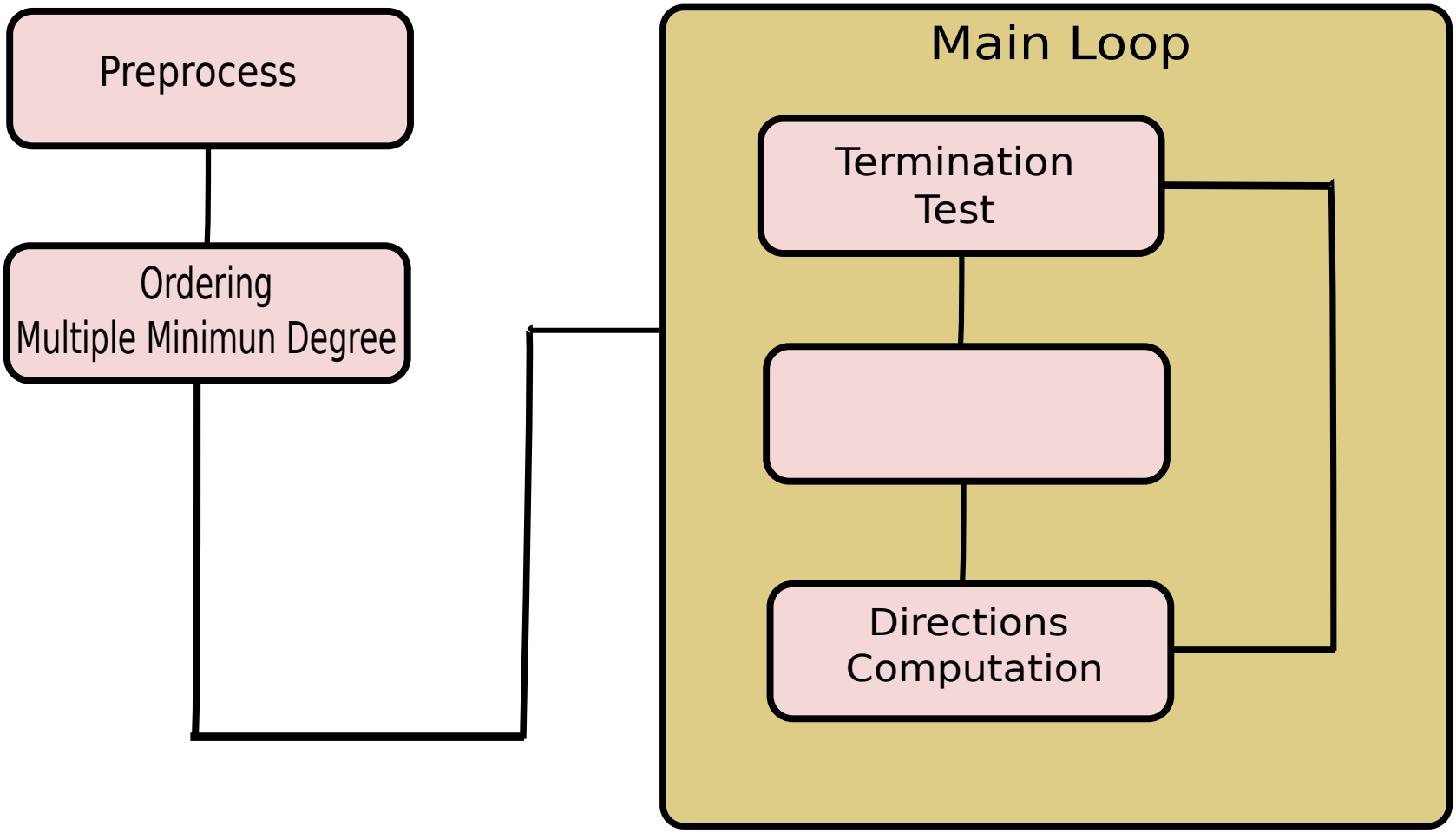
New approach

- Improve the IPM Predictor-Corrector with the normal equations Cholesky factorization, reducing processing time and/or the storage required at each iteration.
- Replacing the Cholesky factorization by CCF.
- In early iterations we adopting a CCF parameter value such that the matrix obtained in factorization is very sparse.
- In later iterations we compute a CCF factorizations closer to the Cholesky factorization.

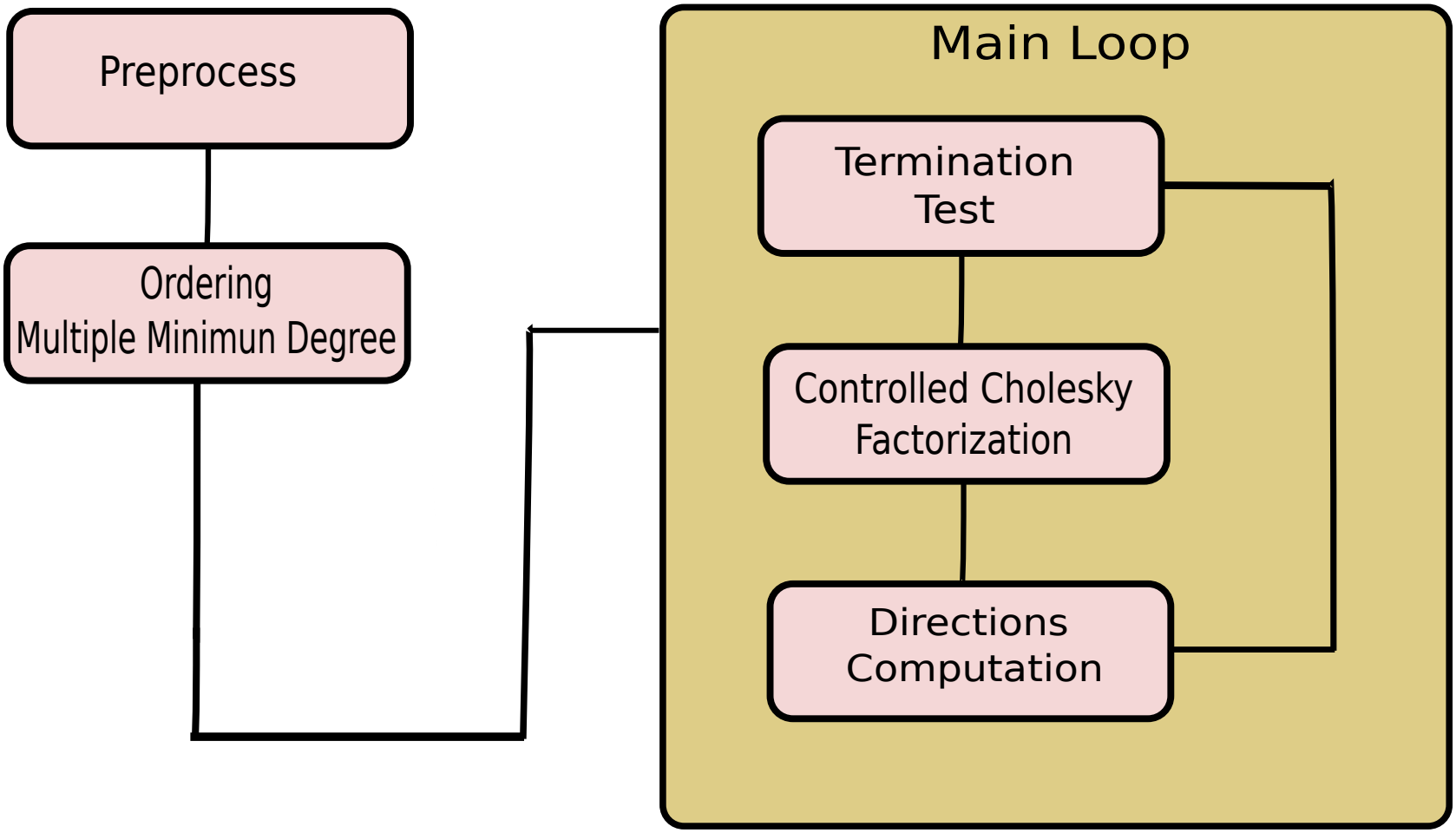

Preprocess
Ordering
Multiple Minimun Degree
Simbolic Cholesky
Factorization
Main Loop
Termination
Test
Numeric Cholesky
Factorization
Directions
Computation

PCx Code with the CCF




Preprocess
Ordering
Multiple Minimun Degree
Main Loop
Termination
Test
Directions
Computation

PCx Code with the CCF



Modifications in the CCF

- Computation of the need storage
- The largest column entries selection
- Treatment of the diagonal failure.

Storage

- *Compressed Column Storage* format.
- L is stored in a vector with maximum size

$$Dim(L) = nnz(AA^t) + \eta \cdot m.$$

When the CCF is used for the normal system direct solution, its maximum dimension will be given by

$$Dim_{max}L = min\{nnz(G), \delta_{max}\},$$

where $nnz(G)$ is the number of Cholesky factor nonzeros entries and

$$\delta_{max} = nnz(AA^t) + m\eta - \left(\frac{\eta^2}{2} - \frac{\eta}{2}\right)$$

and is computed by

$$nnz(AA^t) + \eta \cdot (m - \eta) + \eta + (\eta - 1) + (\eta - 2) \cdots (\eta - (\eta - 1)).$$

The largest columns entries selection

The k largest column entries selection is done through two sorting:

- sorting of the k largest column nonzero entries in decreasing order;
- sorting of the corresponding index in increase order.

Selection sort

To make the two sorts, CCF uses the selection sort algorithm which has complexity:

- $\mathcal{O}(n^2)$ to sort a list of length n,
- $\mathcal{O}(k^3)$ to sort only the k larger entries from a n length list when $k < n - k$.

Heapsort

In the context of precondicioner, the sort used in the CCF don't has strong influence in the processing time.

For the new approach this type of sorting can be inefficient. So we used Heapsort algorithm.

The Heapsort has complexity:

- $\mathcal{O}(nlogn)$ to build a heap of length n,
- $\mathcal{O}(klogn)$, to sort only the k larger entries from a heap of length n.

Prob	Row(m)	Col
PDS-06	9156	61120
PDS-10	15648	48763
PDS-20	32276	106180

Prob	η_1	η_2	η_3	it
PDS-06	36 for $it \leq 11$	m for $it > 11$		35
PDS-10	31 for $it \leq 11$	62 for $12 \leq it \leq 34$	m for $it > 34$	48
PDS-20	64 for $it \leq 19$	128 for $20 \leq it \leq 24$	256 for $25 \leq it \leq 29$	51

Table : η Values and CCF iterations number

Selecting sort×Heapsort

Problem	Selecting sort	Heapsort in the 2^{th} sort	Heapsort
PDS-06	9.67	8.38	8.31
PDS-10	34.94	30.15	28.75
PDS-20	390.21	301.38	290.28

Table : Time in seconds

Treatment of the diagonal failure

- Exponential increase technique: Given an entry very small or negative in the diagonal, restart the factorization after adding to the diagonal entries of ADA^t, the value $\alpha_i = \lambda 2^i$, where $\lambda = 5 \cdot 10^{-4}$ and $i = 1, 2, ..., I_{max}$, being I_{max} the maximum number of restarts allowed .

- The exponential increase technique causes an increase in the processing time of the corresponding iteration and an even greater change in the matrix.

- To avoid restarting, in our approach when a pivot undesired is computed, it is replaced by the value 10^{128} and the factorization continues from there.

CCF Parameter

- **What will be the initial η value?**
- **How will be the η increase along the iterations?**

To determine the η value we can consider:

- the problem dimension;
- the ADA^t matrix Cholesky factor density;
- the progress at each iteration;
- ...

Tests with η increments in PDS-10 problem

- **The same η value is used in several iterations.**

We used $\eta = 31$ until 11^{th} iteration, $\eta = 62$ until 34^{o} iteration and the complete Cholesky was used in the remaining iteration.

- **Increasing successive increments of η are made along the iterations**

We used $\eta = 3$ in the first iteration and increments of λk , where k is the iteration number and $\lambda = 2.1$.

Tests with increments of η in PDS-10 problem

Comparison of time and the number of iterations between the two η increments approaches and complete Cholesky.

η increment	interation number	time
several iteration with same η	48	28.75
successive increments of η	51	46.40
complete Cholesky	40	57.02

Considerations

- The CCF algorithm is used to compute the complete Cholesky factorization, but it isn't efficient for this purpose.
- Use another specific complete factorization implementation, as the Ng-Peyton code.
- Use the preconditioned conjugate gradient method with the splitting preconditioner together with the proposed approach.
- Study if the symbolic factorization, could help the CCF to be more efficient.
- Find a good heuristic to determine the η value at each iteration

Bibliography I

S. J. Wright
Primal-Dual Interior-Point Methods
SIAM Publications, SIAM, Philadelphia, PA, USA, 1996.

F. F. Campos, Analysis of Conjugate Gradients - type methods for solving linear equations, PhD thesis, Oxford University Computing Laboratory, Oxford, 1995.

S. Bocanegra, F. F. Campos, and A. R. L. Oliveira
Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods
International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Scientific and Industrial Applications - SIAM, (2005), pp. 2-4.

Bibliography II


F. F. Campos
Analysis of Conjugate Gradients - type methods for solving linear equations
PhD thesis, Oxford University Computing Laboratory, Oxford, 1995.


S. Mehrotra
On the implementation of a primal-dual interior point method
SIAM Journal on Optimization, 2 (1992), pp. 575-601.


J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright
PCx an interior point code for linear programming
Optimization Methods e Software,11-2 (1999), pp. 397-430.


D. S. Malik
Data Structures using C++
Course Technology, Cengage Learning, Boston, 2010.

Thanks!