Equality constrained quadratic programming problem

General problem

$$\min f(x) = \frac{1}{2} x^T Q x + c^T x$$

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

- Q symmetric positive semidefinite matrix of dimension $n \times n$
- c, x vectors of dimension $n \times 1$
- A full rank matrix of dimension $m \times n$
- b vector of dimension $m \times 1$

KKT conditions at a minimum x*:

$$\begin{cases} Qx^* + A^T\lambda^* = -c \\ Ax^* = b \end{cases}$$

We can introduce the matrix K and the vectors w*, d

$$K = \begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \qquad w^* = \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} \qquad d = \begin{pmatrix} -c \\ b \end{pmatrix}$$

and rewrite KKT conditions to QP obtaining the linear system $Kw^* = d$

Constrained problem

Considered problem and its constraints

Problem 2

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n-1} x_i x_{i+1} + \sum_{i=1}^{n} x_i \\
\text{s.t.} \quad & \text{the sum } x_1 + x_{1+K} + x_{1+2K} + \ldots \text{ should be } 1 \\
& \text{the sum } x_2 + x_{2+K} + x_{2+2K} + \ldots \text{ should be } 1 \\
& \vdots \\
& \text{the sum } x_K + x_{2K} + x_{3K} + \ldots \text{ should be } 1
\end{aligned}
$$

Q = diag(2) + upper_diag(-1) + lower_diag(-1), everything else 0

C = vector of ones

A = [diag(1) every K columns]

Methods used

Presentation of the methods used to solve the problem

FULL SYSTEM FACTORIZATION

- $K = LDL^T/LU$
- fill-in problem

GMRES

- Iterative solver
- $Kw^* = d$

SCHUR-COMPLEMENT APPROACH

- $\mathbf{Q}=AQ^{-1}A^T$
- Generally more efficient

NULL-SPACE METHOD

- Null-space matrix Z
- Computationally expensive

Evaluation metrics

- Elapsed time
- How close we are to the solution:
  - KKT_gradL_norm = $||Qx^* + A^T\lambda + c||$
- How well we respected the constraints:
  - KKT_eq_norm = $||Ax^* - b||$
- Value of the function in x^*

Disclaimer

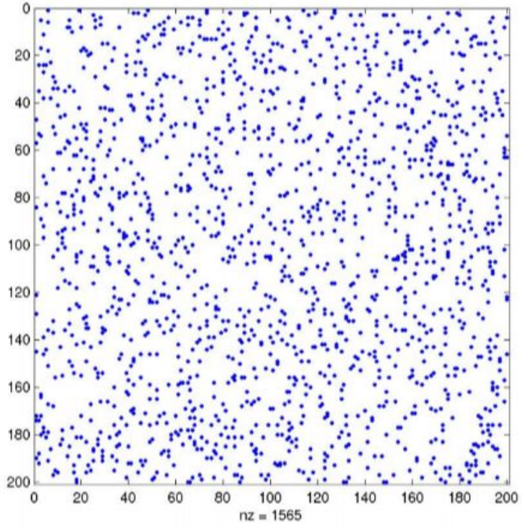
If n=10^5 the matrix is sparse, thus results obtained are not completely comparable with n=10^4.

$$\begin{bmatrix} X & X & X & \cdot & \cdot & \cdot & \cdot \\ X & X & \cdot & X & X & \cdot & \cdot \\ X & \cdot & X & \cdot & X & \cdot & \cdot \\ \cdot & X & \cdot & X & \cdot & X & \cdot \\ \cdot & X & X & \cdot & X & X & X \\ \cdot & \cdot & \cdot & X & X & X & \cdot \\ \cdot & \cdot & \cdot & \cdot & X & \cdot & X \end{bmatrix}$$

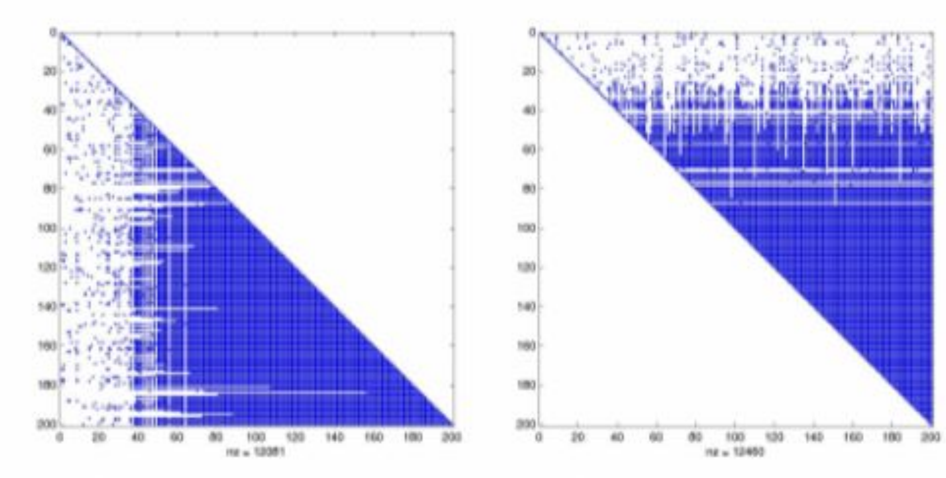
FULL SYSTEM FACTORIZATION

- Symmetric factorization $K=LDL^T/LU$
- Fill-in problem if n+m is very large and K sparse

Fill-in phenomenon



Matrix A



Matrices L, U

GMRES

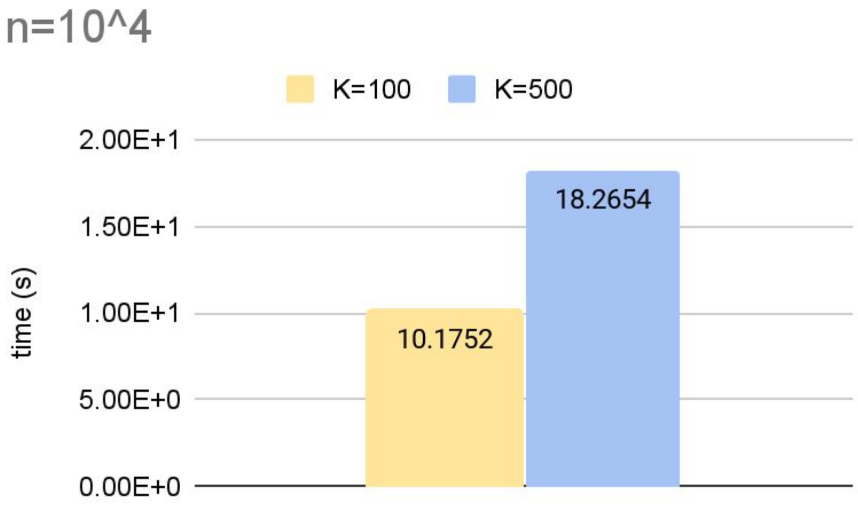
- Iterative solver
- Linearly increasing cost per iterations
- Solve the system Kw*=d

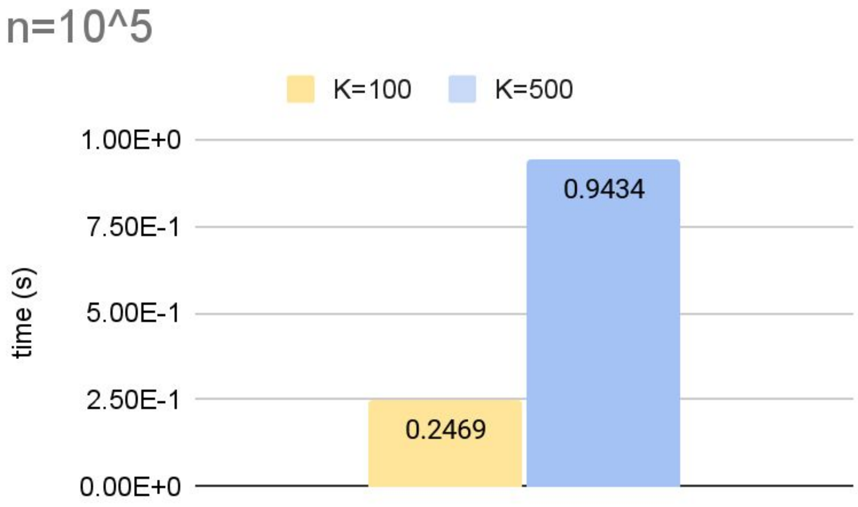
- Parameters used:

tol = 10^{-6}

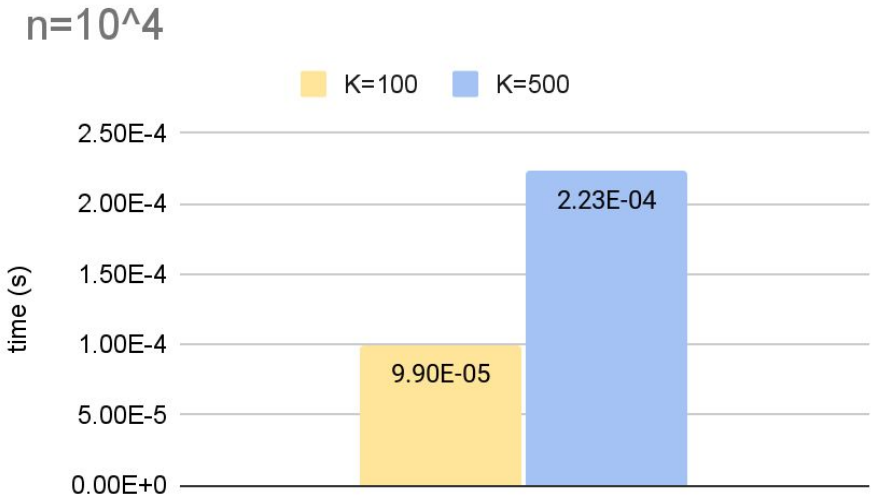
maxit = 200

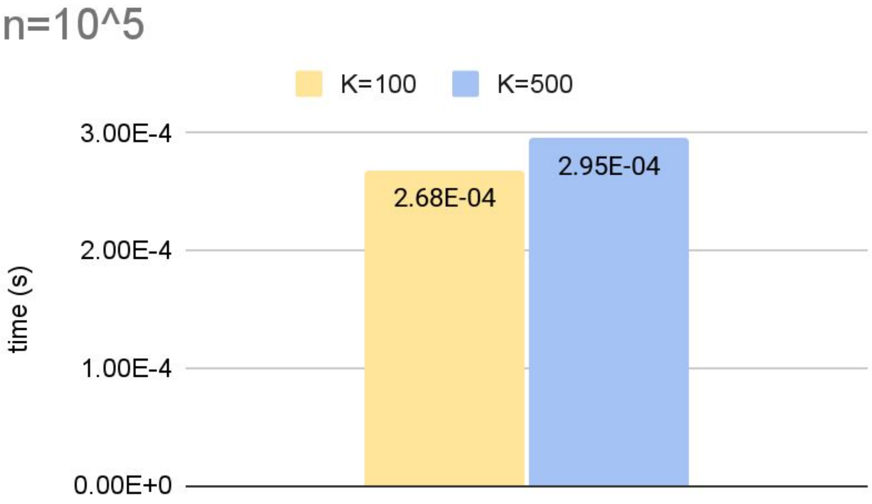
GMRES results: elapsed time



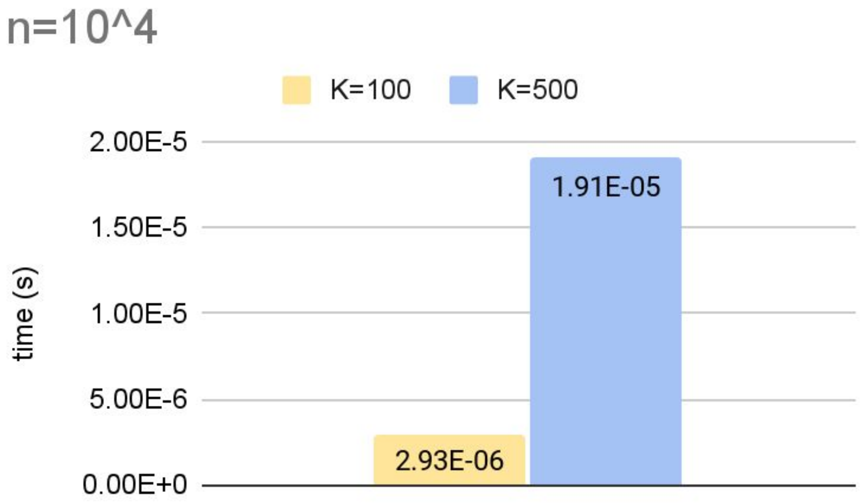


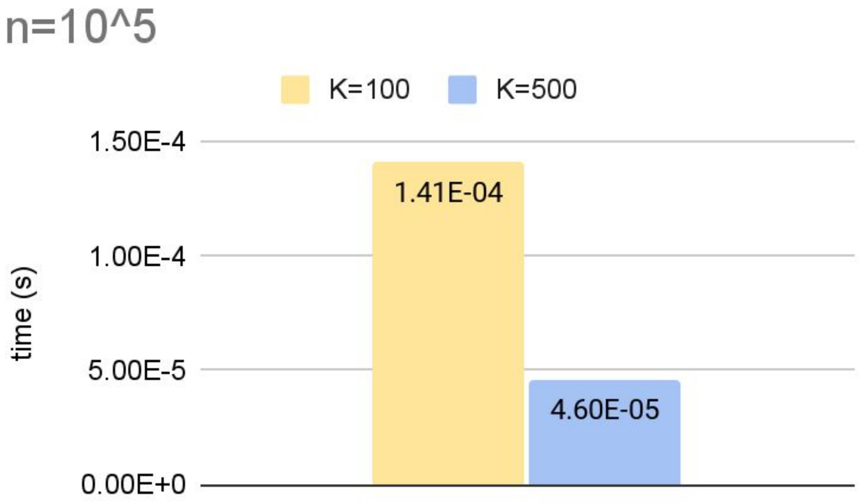
GMRES results: KKT_GradL_norm



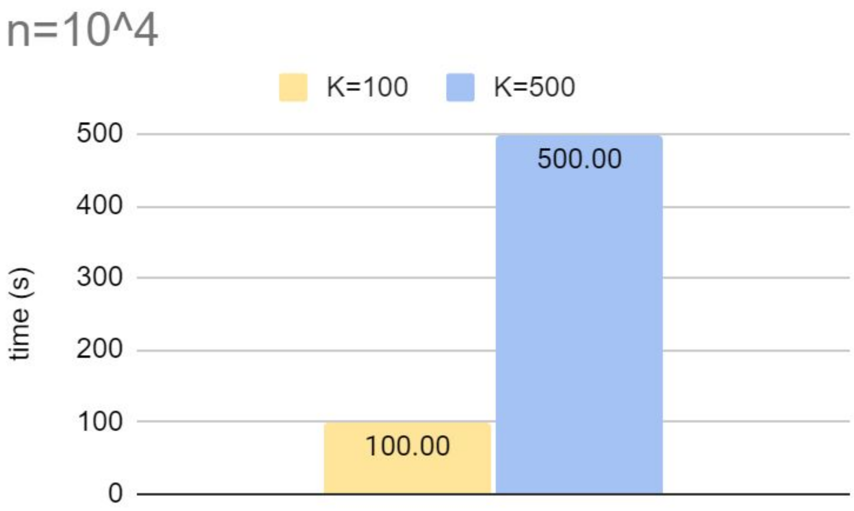


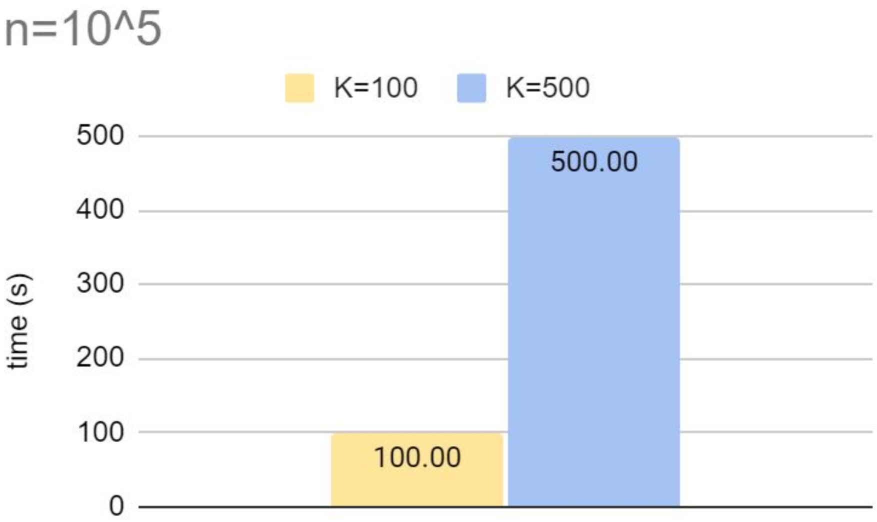
GMRES results: KKT_EQ_norm





GMRES results: f(x*)

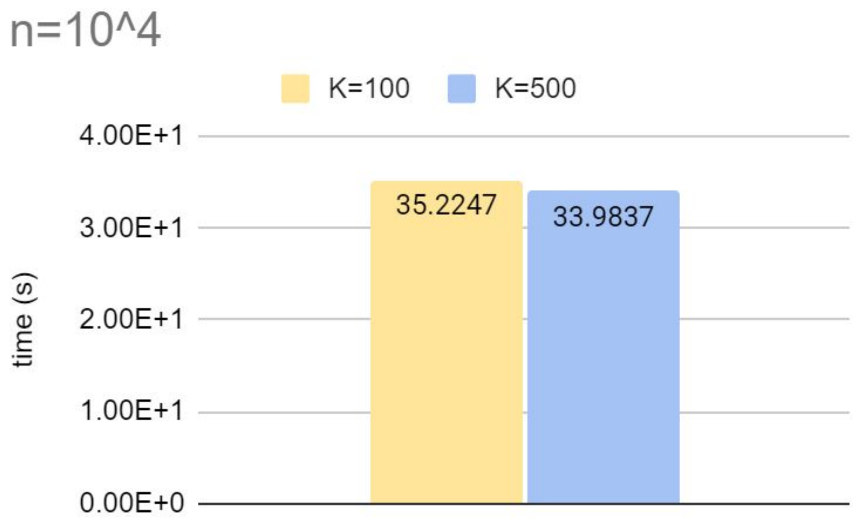


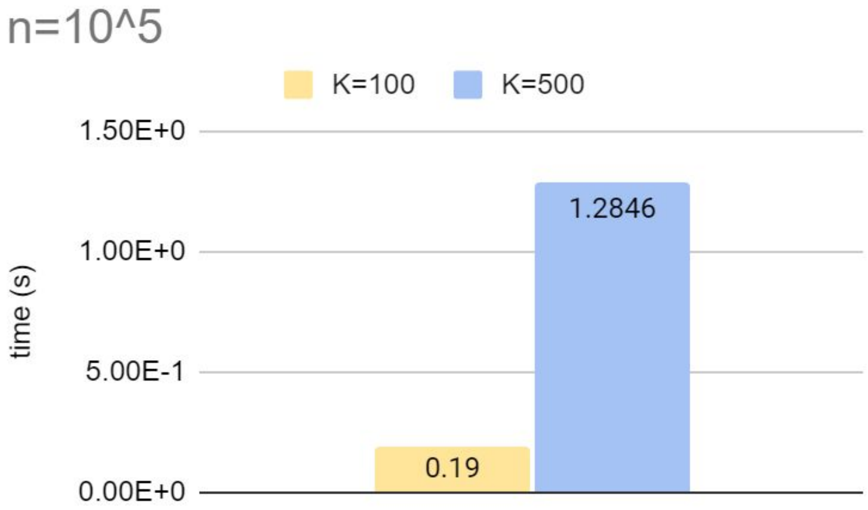


SCHUR-COMPLEMENT APPROACH

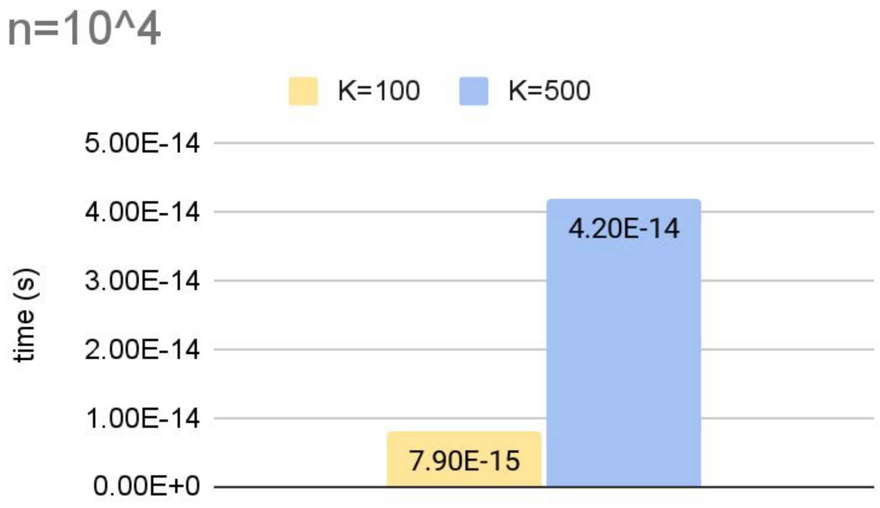
- Q symmetric positive semi-definite
- A full rank
- Efficient method
- $\boldsymbol{Q}_{\boldsymbol{Schur}} = AQ^{-1}A^{T}$

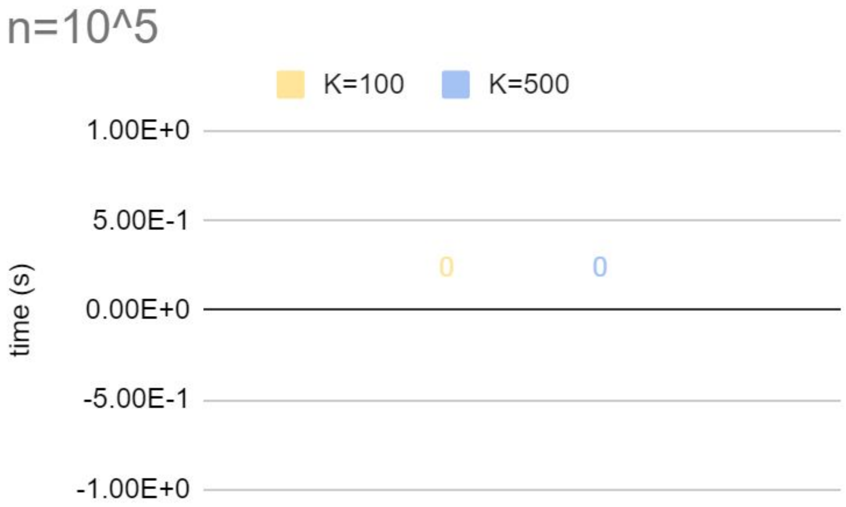
SCHUR results: elapsed time



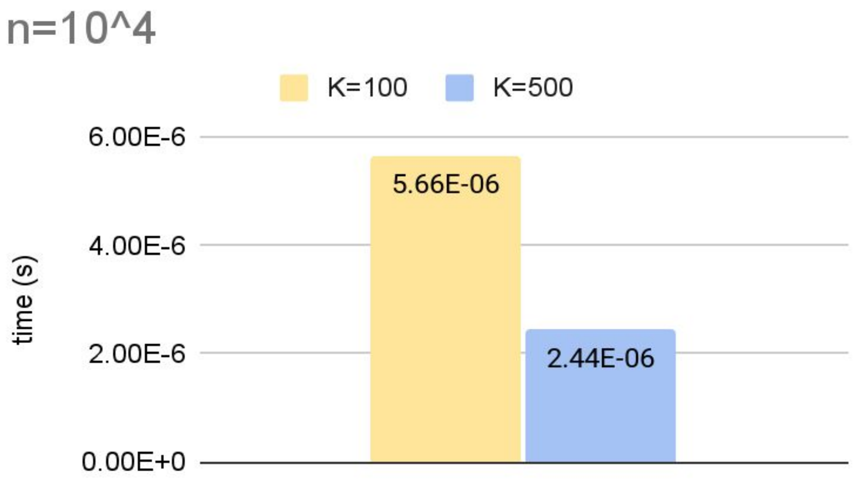


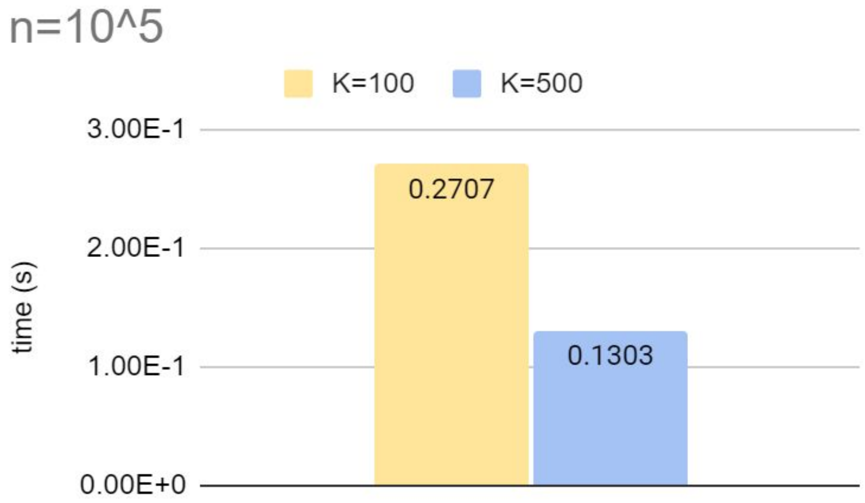
SCHUR results: KKT_GradL_norm



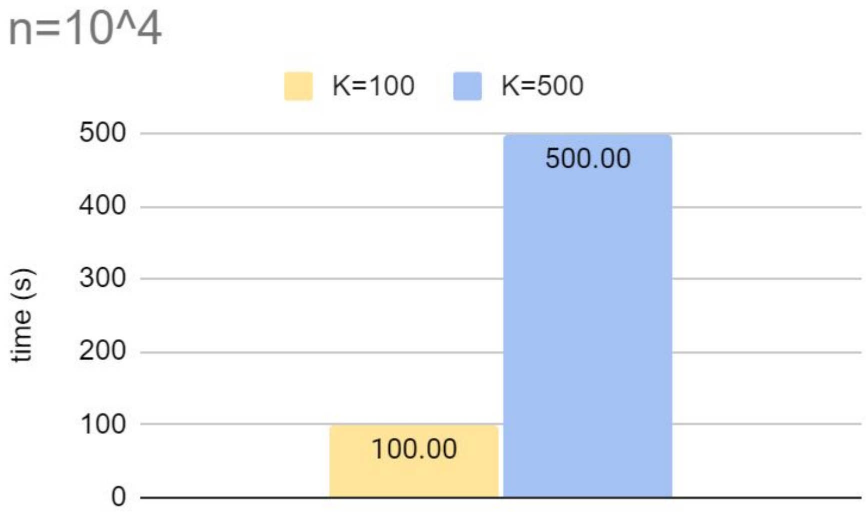


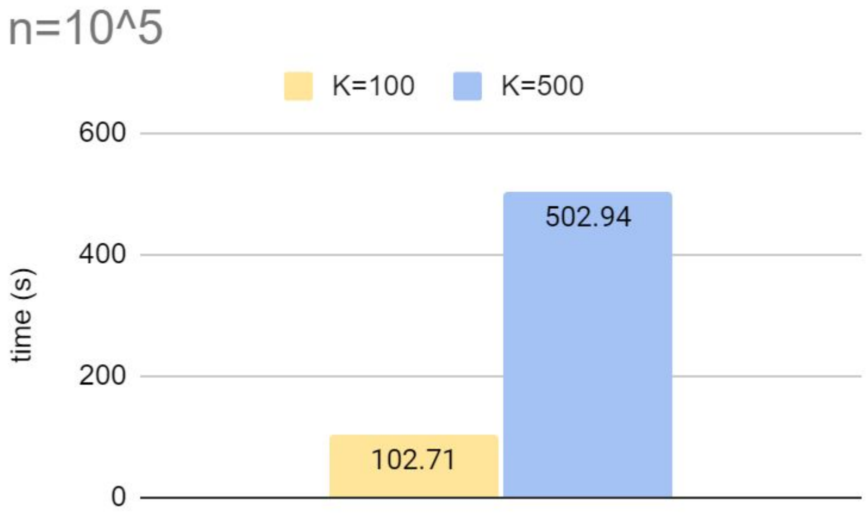
SCHUR results: KKT_EQ_norm





SCHUR results: f(x*)



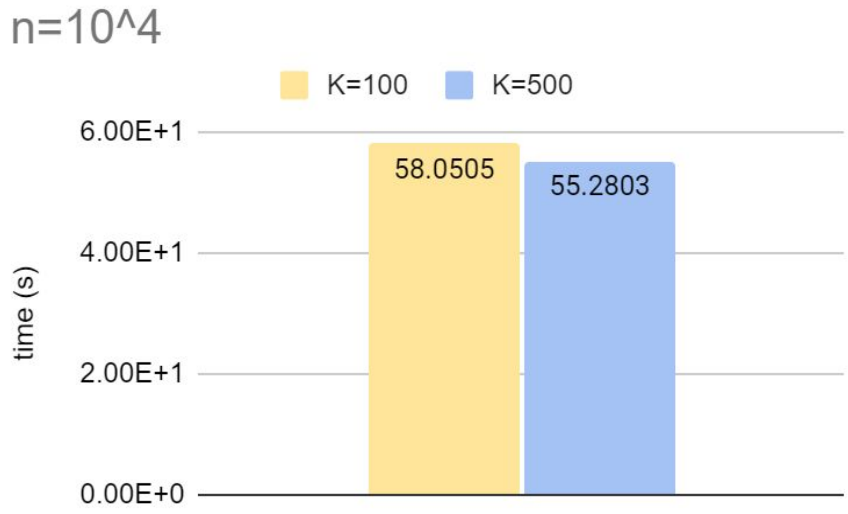


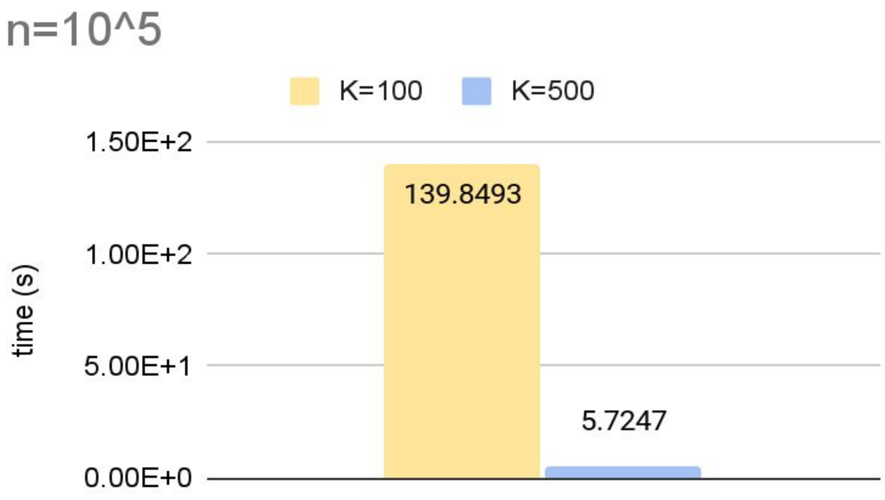
“

NULL-SPACE METHOD

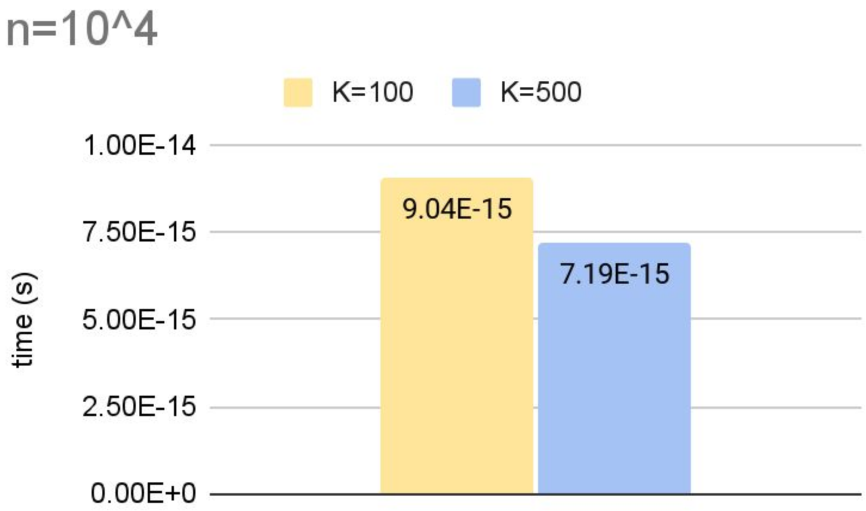
- Does not require non-singularity of Q
- A is a full-rank matrix
- Z has dimension nx(n-m)
- Z^TQZ is positive definite

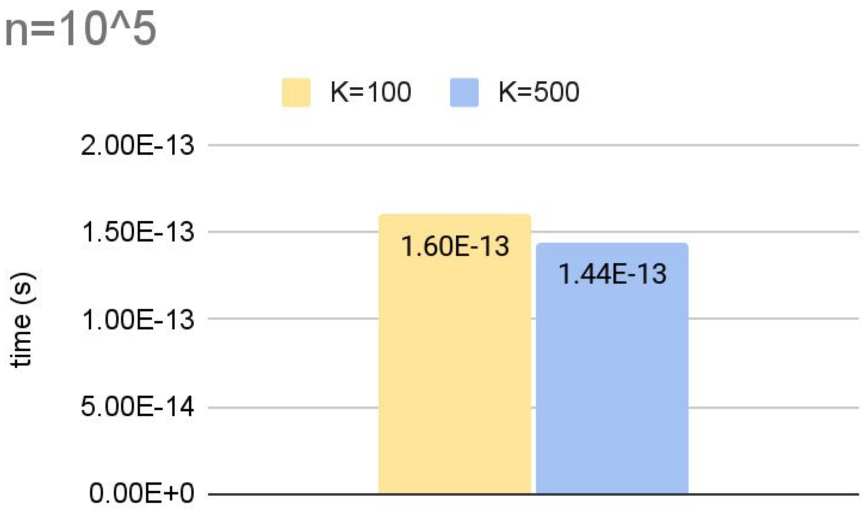
NULL-SPACE results: elapsed time



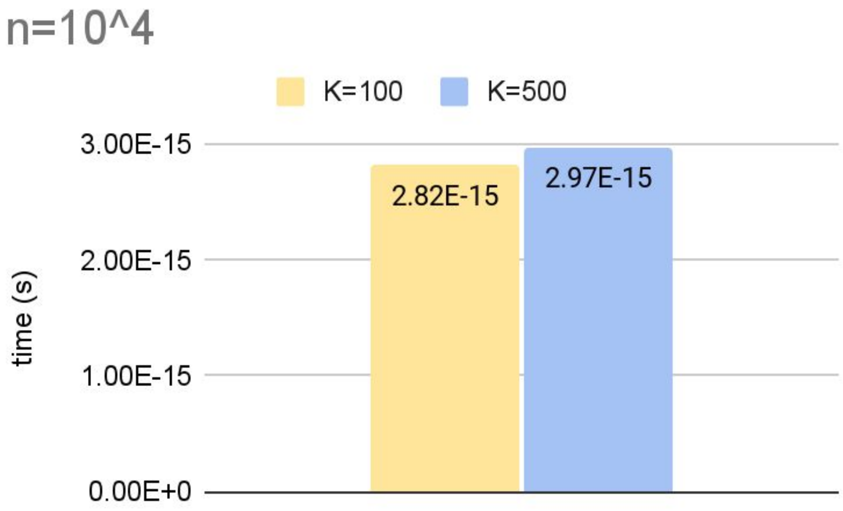


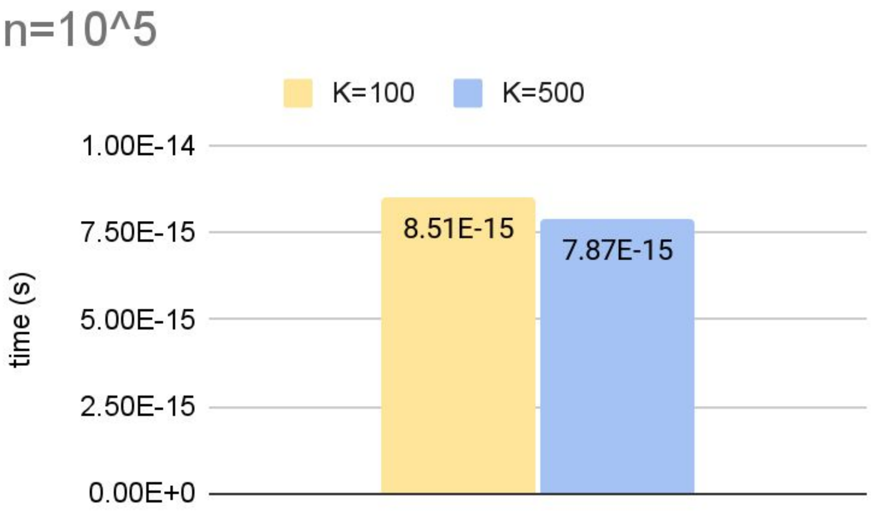
NULL-SPACE results: KKT_gradL_norm



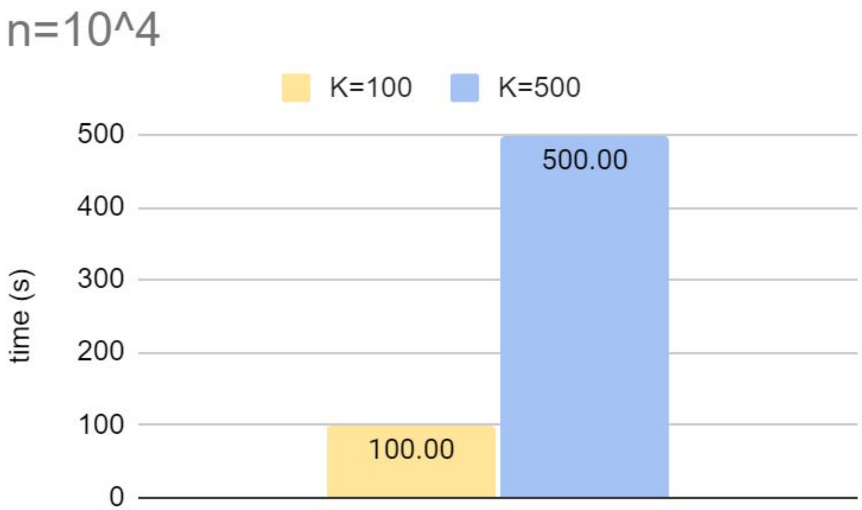


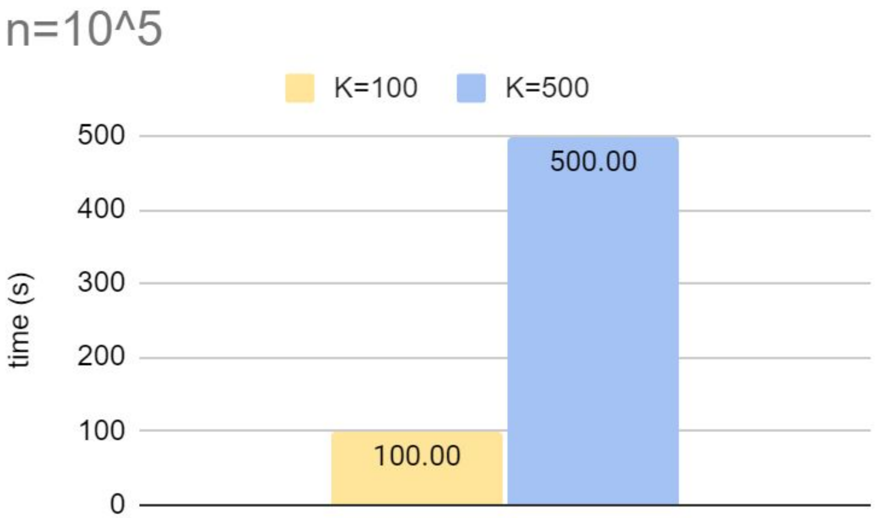
NULL-SPACE results: KKT_EQ_norm





NULL-SPACE results: f(x*)

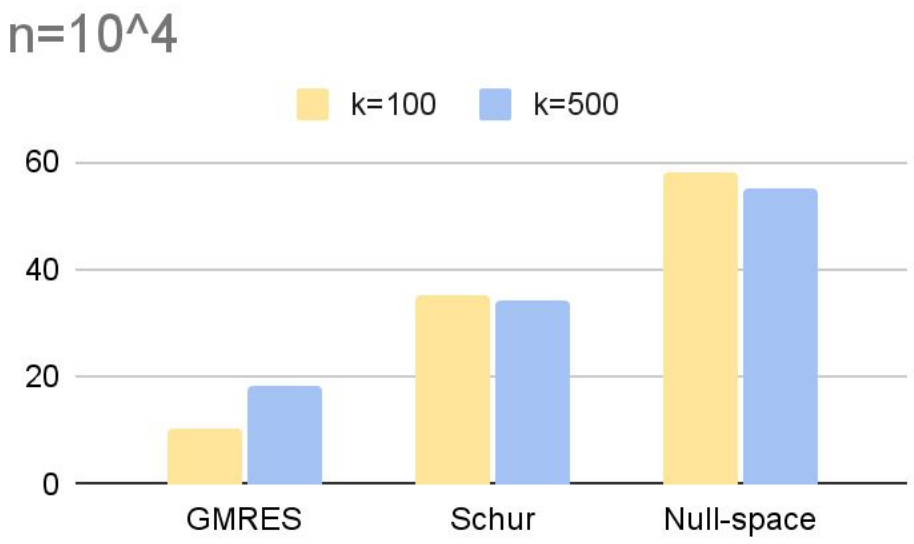


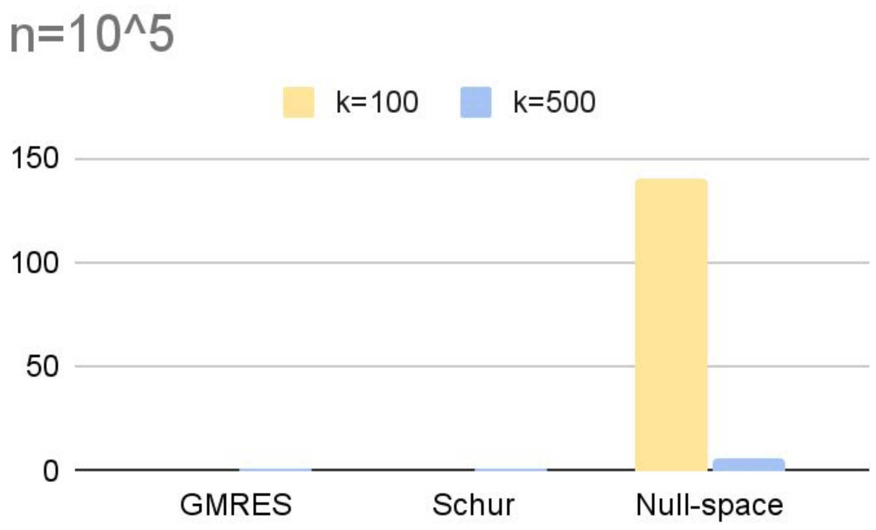


Conclusions

Final considerations and evaluations of the methods in this particular problem

Elapsed time comparison





Conclusions

GMRES

- Lots of iterations
- Lots of computational time to reach high precision

SCHUR

- Computationally efficient
- Fast
- Not completely precise when n is large

NULL-SPACE

- Generally efficient
- Precise
- Can be expensive for some matrix Z

There's not a method always better, it highly depends on the problem faced and the constraints it has to satisfy.

Team Presentation

Florentin-Cristian Udrea
S319029

Sara Rosato
S317547

Thanks for the attention!