NUMERICAL AND SYMBOLIC ALGORITHM MODELING

A PROJECT REPORT

APPROXIMATED SOLUTION OF LINEAR SYSTEMS

Harimbola Ranto Ranaivo

December 2022

Whenever a matrix A has size $m \times n$ with $m \geq n$, even if it has full rank, a solution of the linear system $Ax = b$ may not exist. The least-square method, then, tries to minimize the error for the Euclidean norm. That is find x' such that $||Ax' - b|| 2$ is minimal. The QR decomposition factors A into QR with Q unitary and R upper triangular.

1 Implementation

This program contains 8 functions :

```
void print_matrix(double* m, int row, int col)
double* transpose(double *m, int row, int col)
double* mult_mm(double *a, int row_a, int col_a, double *b, int row_b, int col_b)
double* create_matrix(int row, int col)
double* create_identity(int row, int col)
double* create_random_matrix(int row, int col)
double* g_rotation(int i, int j, double* r)
double* givens(double *a, double **b)
```

- *mult_mm*() : matrix multiplication of A and B and return a new matrix C
- *create_matrix*() : create a matrix with null coefficient
- *create_identity*() create an identity matrix
- *create_random_matrix*() : create a matrix with random coefficients
- *g_rotation*() : do a rotation according the Given's classic algorithm
- *givens*() : make a QR decomposition following Given's algorithm, return a new matrix Q and the matrix A become R

The *main*() funciton execute the QR decomposition of the input matrix. The number of rows and columns are given parameters at the start of the program. In this program, the execution time of Given's algorithm is also measured.

The program seems to run without any errors. However, in double precision, instead of being null, some coefficients take a negative value infinitely small as observed bellow :

```
R :

 5.83095          1.20049          2.91548          2.40098
 2.64686e-17      4.06925          3.07182          0.766147
-3.21536e-16     -2.51812e-17      0.25287         -1.39781
-3.05171e-16     -1.16739e-17      0         0.833333
```

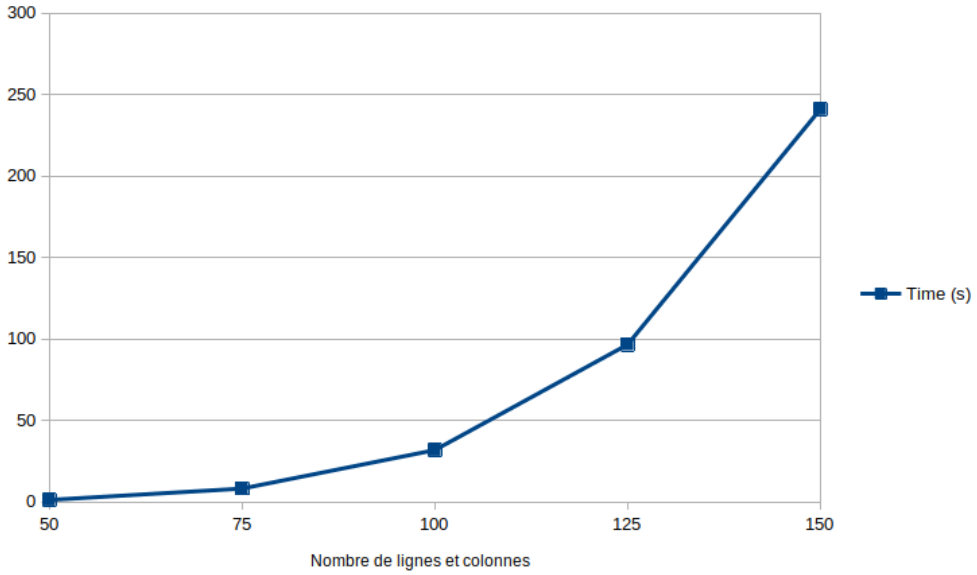
The matrix R is supposed to an upper triangular matrix, with zeroes under the diagonal. Here, there are really small negative numbers under the diagonal.

2 Tests

All the tests have been made with double precision.

Executing the program with different size of matrices we obtain the following results :

Number of rows	50	75	100	125	150
Number of columns	50	75	100	125	150
Time(s)	1,062921	8,043062	31,710457	96,536258	241,175844



We observe that it is not linear at all. The time increase in cubic time and it is cause by the naive multiplication of matrices which is done in $O(n^3)$