Test title

Linus Falk

October 29, 2022

Introduction

some fancy words about iterations or doda

Problem B1

Method	**Iteration**	**Time**
Jacobi	11	0.004164
Gauss-Seidel	6	0.004300
CG	9	0.000377
myownLU		0.009256
Ab		0.000056
Matlab LU		0.000062

Table 1: example

Problem B2

Method	**Iterations**	**100**	**500**	**1000**
Jacobi	NaN/NaN/NaN	NaN	NaN	NaN
Gauss-Seidel	NaN/NaN/NaN	NaN	NaN	NaN
CG	NaN/NaN/NaN	NaN	NaN	NaN
myownLU		1.73474	259.189	2425.89
Ab		0.00023	0.01010	0.02241
MATLAB LU		0.00021	0.00351	0.02076

Table 2: w = 1

The matrix becomes Diagonal dominant row wise when w=100 and N=100, in all other cases are the matrix not diagonal dominant. This explains why the Jacobi method fails except for this case.

For myownLU was the Doolittle algorithm implemented to solve the system of equation. The method is one of the easier to understand and do with pen and paper but lacks the speed for be useful when solving larger systems as can be seen in table 2.

Method	Iterations	100	500	1000
Jacobi	NaN/NaN/NaN	NaN	NaN	NaN
Gauss-Seidel	110/NaN/NaN	0.00297	NaN	NaN
CG	138/NaN/NaN	0.00245	NaN	NaN
myownLU				
Ab		.00033	0.00982	0.01947
MATLAB LU		0.00020	0.00936	0.01541

Table 3: w = 5

Method	Iterations	100	500	1000
Jacobi	NaN/NaN/NaN	NaN	NaN	NaN
Gauss-Seidel	38/609/NaN	0.01202	19.2424	NaN
CG	42/374/1952	0.004506	0.15488	1.94158
myownLU				
Ab		0.00024	0.009006	0.02136
MATLAB LU		0.00036	0.00426	0.01410

Table 4: w = 10

Method	Iterations	100	500	1000
Jacobi	22/NaN/NaN	0.00120	NaN	NaN
Gauss-Seidel	10/27/57	0.00351	0.8681	10.2448
CG	8/24/44	0.001852	0.019314	0.0588300
myownLU				
Ab		0.00030	0.00724	0.02376
MATLAB LU		0.00025	0.00534	0.01787

Table 5: w = 100

Conjugate gradient method only works for symmetric positive definite matrices and will therefore not be useful for the cases with w = 1. When N and w becomes larger the matrix is ???

Problem B3

In this section is a large sparse matrix solved using the methods from earlier. When trying to solve this system was some of the methods not practical to use: myownLU which was proven to slow previously was therefore excluded.

Even though the Conjugate Gradient method is very useful for large sparse matrices like this, it became clear that the improvement of each iteration for $\alpha = 0$, and 0.00001 was very slow and the test was only finished with $\alpha = 0.1$ and 0.0001. This is Because the improvement of each iteration of the Conjugate Gradient method depends on the condition number. The Conjugate gradient method would finish in the other cases also if we let it run and if the round off errors wouldn't become to big.

For the other two methods (Jacobi and Gauss-Seidel) it was also clear that the condition number had an effect on speed of convergence, making it unpractical to time them for the ill conditioned systems where $\alpha = 0, 0.001$ and 0.00001.

- for $\alpha = 0.1$ cond(A) ≈ 41
- for $\alpha = 0.001$ cond(A) ≈ 4000
- for $\alpha = 0.00001$ cond(A) ≈ 39600
- for $\alpha = 0$ cond(A) ≈ 4053700

Method	**Iteration**	a = 0	a = 0.1	a = 0.001	a = 0.0001
Jacobi	NaN/298/NaN/NaN		507.30		
Gauss-Seidel	NaN/260/NaN/Nan		142.36		
CG	NaN/260/25116/NaN	NaN	0.0974	7.2487	NaN
myownLU					
Ab		0.0003	0.0003	0.0002	0.0004
Matlab LU		0.0014	0.0016	0.0014	0.0017

Table 6: Result Problem B3