Nicholas Garrett

Professor Zhu

MATH 4441

11/3/2021

Homework 9

1.

I wrote a MATLAB script: poissonBuilderV3 to build this matrix based on the conditions stated in the problem. However, when I run this script, the values of x and Ax do not converge. So, I am going to continue by following the assumption that my program does not have a bug in it.

1.a.
u = does not converge
b = does not converge

A =

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| -1 | 4 | -1 | 0 | -1 | 0 | 0 | 0 | 0 |
| 0 | -1 | 4 | 0 | 0 | -1 | 0 | 0 | 0 |
| -1 | 0 | 0 | 4 | -1 | 0 | -1 | 0 | 0 |
| 0 | -1 | 0 | -1 | 4 | -1 | 0 | -1 | 0 |
| 0 | 0 | -1 | 0 | -1 | 4 | 0 | 0 | -1 |
| 0 | 0 | 0 | -1 | 0 | 0 | 4 | -1 | 0 |
| 0 | 0 | 0 | 0 | -1 | 0 | -1 | 4 | -1 |
| 0 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 4 |

1.b.
u = does not converge
b = does not converge

A=
A is still the same.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | -1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| -1 | 4 | -1 | 0 | -1 | 0 | 0 | 0 | 0 |
| 0 | -1 | 4 | 0 | 0 | -1 | 0 | 0 | 0 |
| -1 | 0 | 0 | 4 | -1 | 0 | -1 | 0 | 0 |
| 0 | -1 | 0 | -1 | 4 | -1 | 0 | -1 | 0 |
| 0 | 0 | -1 | 0 | -1 | 4 | 0 | 0 | -1 |
| 0 | 0 | 0 | -1 | 0 | 0 | 4 | -1 | 0 |
| 0 | 0 | 0 | 0 | -1 | 0 | -1 | 4 | -1 |
| 0 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 4 |

.

2.
2.a
see code PoissonSolver
2.b
see code PoissonSolver
2.c:
See code PoissonSolver

Let me just say, this part of the problem took a very long time to finish and I am quite pleased with myself for being able to finally get it to work. I checked the returns of my function against the provided MATLAB functions you provided in lecture, so I know they are correct.

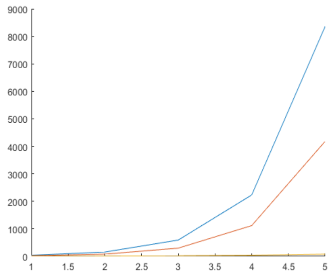
| Type of Iteration: | Jacobi | Gauss Seidel | Conjugate Gradient |
|---|---|---|---|
| n(l = 2): | 38 | 21 | 3 |
| n(l = 3): | 155 | 79 | 9 |
| n(l = 4): | 592 | 298 | 20 |
| n(l = 5): | 2232 | 1118 | 39 |
| n(l = 6): | 8361 | 4183 | 79 |

I am going to continue under the assumption that there was simply a typo in the problem and you want us to determine the relation between l and iteration count. Well, the higher l is, the higher the iterations required to solve it become as well.

Jacobi is the least efficient, experiencing a very significant spike in the number of iterations required at higher l values (blue).

Gauss-Seidel is the next most efficient, following what looks like an exponential growth as well, though accelerates less steeply than the Jacobi iterative method. (red)

Finally, the Conjugate Gradient iterative method is the most efficient, outpacing the two others by a very large margin. From the table iteration data, this too seems to be an exponentially growing function, though it is accelerating much more slowly than the other two. (yellow)


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1
1.5
2
2.5
3
3.5
4
4.5
5