

MACM 316 - Computing Assignment 2

- **Read the *Guidelines for Assignments* first.**
 - Submit a one-page PDF report to Crowdmark and upload your Matlab scripts (as m-files) to Canvas. *Do not use any other file formats.*
 - Keep in mind that Canvas discussions are open forums.
 - You must acknowledge any collaborations/assistance from colleagues, TAs, instructors etc.
-

We are interested in solving $Ax = b$. In this assignment, you will compare the computing time of finding matrix inverses and finding matrix factorizations for *large sparse* matrices.

The following line of code produces a commonly-used tridiagonal matrix of size n (the one-dimensional *discrete Laplacian*):

```
e=ones(n,1); A_n=spdiags([e -2*e e],[-1:1, n, n]);
```

Part A: Complete the following, using the above line of code, to generate sparse $n \times n$ matrices.

- For each n that you are considering, generate a large sparse matrix A_n and measure the average computing time that Matlab's `inv()` algorithm requires to find the inverse of A_n . Be sure to specify how many executions of `inv()` you are averaging over to obtain the average time. In the computing time estimate, do not include the cost of constructing A_n .
- Plot the average computing time vs n for a variety of n to visualize the relationship between n and computing time. Choose appropriate scales for the axes in your plot.
- Analyze the plot to determine how the time for computing `inv()` varies with n for large n . Use big-Oh notation to describe the relationship and explain your findings.

Continued on page 2

MACM 316 - Computing Assignment 2

Part B: In this question, you are asked to repeat the process in Part A but this time using LU factorization:

- For each n you are considering, generate a number of large sparse matrices and measure the average computing time that Matlab's `LU()` algorithm requires to find the decomposition $LU = A_n$. Be sure to specify how many executions of `LU()` you are averaging over. In the computing time estimate, do not include the cost of constructing A_n .
- Plot the average computing time vs n for a variety of n to visualize the relationship between n and computing time. Choose appropriate scales for the axes in your plot.
- Analyze the plot to determine how the time for computing L and U varies with n for large n . Use big-Oh notation to describe the relationship and explain your findings.

Part C: Suppose $n = 100$ and $b = (1:n)'$. In Part C of your report, give the Matlab code to solve $A_n x = b_n$ using

- (1) the inverse of A_n , and
- (2) the L and U that arise from LU decomposition.

You do not need to present the solution x .

Are your numerical solutions x for (1) and (2) exactly the same? Explain.

Notes:

- In Matlab, a matrix can be factored into LU by the LU factorization command `[L,U,P]=lu(A)`.
- The time that a code requires to execute can be measured using the Matlab commands `tic` and `toc`.
- Use a sparse matrix representation for the matrix A_n to ensure that Matlab understands the zero structure of the matrix. The `spdiags([e -2*e e],-1:1,n,n)` command should automatically take care of this for you.

Continued on page 3

MACM 316 - Computing Assignment 2

Your report cannot exceed one page. It should include at least two figures with proper labels. Make sure to choose axis scales appropriate for the data. Discussions should be kept brief and answer all questions asked.

*Submit your 1 page report for this question to Crowdmark in .pdf format according to the Assignment Guidelines described in the syllabus. Your grade is based on your Crowdmark submission. We ask that you do **not** include identifying information (name, ID number, etc) on your submission.*

Submit your Matlab code to Canvas "Computing Assignment 2 - Matlab Code".

After marking, we will post a few exemplary reports as sample solutions. We appreciate your support on this. If you do not wish to have your report posted, please state so at the top of your report.

Grades will be based on 5 criteria:

Figures/Quality of presentation:

2 marks: Well-presented

1 mark: Missing labels, figure is small, inappropriate axis scaling

0 marks: Figures do not present much useful information

Writing / Clarity and conciseness:

2 marks: Writing is clear and concise

1 mark: Contains minor spelling or grammatical errors, too brief or too long, does not convey the main ideas

0 marks: Writing quality is poor and cannot be understood easily or at all

Data:

2 marks: Data is correct and relevant to the report

1 mark: some data is missing, unimportant data is included

0 marks: No data, data is incorrect, data is irrelevant

Parts A-B:

3 marks: Complete and correct analysis and discussion

1-2 marks: Values are given but discussion contains small errors or is incomplete

0 marks: Analysis and discussion missing, incomplete, and/or incorrect

Part C:

1 mark: Correct code and interpretation

0 marks: Code or interpretation incorrect