MACM 203 Assignment 2
Spring 2025

This assignment is due Tuesday February 4th at 10pm. Upload your solutions to Crowdmark. Write your solutions as a single Matlab Live Script and export the script to PDF. Write the course number and assignment number as the title of the Matlab Live Script, followed by the table of contents, and then create a section for each part of the question.

Keep in mind that your assignment, including the source code, is a document that will be read in order to be marked. It has to be very clear and properly formatted. Figures need to be self-explanatory, with the following elements: a general title that describes what the figures represent, labels for both axes that describe clearly what they represent (with the proper units), and if more than one graph is represented in a figure, each needs to be labeled with legend.

Assignments should be written individually. You can discuss in groups, but you have to write your assignment yourself. In case of plagiarism SFU policies will be applied.

Preamble

This week's assignment focuses on comparison of solvers for linear systems available in Matlab.

Question 1 (15 marks)

Let A be an $n \times n$ matrix and b a column vector with n entries. Consider solving the linear system $Ax = b$. If only *one* vector b is given, then the fastest method for solving the system in Matlab is `x=A\b`. Note that \ is a Matlab operator.

However the situation is different if we need to solve *many* linear systems $Ax = b$ with *the same* matrix A and different right-hand side vectors b. You will compare the *efficiency* of the following *two methods* in case of solving many systems $Ax = b$ with the same matrix A:

- $x = A\backslash b$. You can refer to this method as the "built-in solver."
- Compute the LU decomposition of $A = LU$. (Of course, this is done only once for a given matrix A.) Now the system has the form $LUx = b$. Denote $z = Ux$. Then solve the system $Ax = b$ in two steps: First solve $Lz = b$ and then solve $Ux = z$. In each step use the built-in solver.

You are going to look at the efficiency (in terms of computation time) of these two methods, using random linear systems of large size.

Below are the precise instructions of what needs to be done:

- We want to consider values of n ranging from $n = 100$ to $n = 1000$, with increments of 50. Requirement: use the construction `a:b:c` to generate this range of values of n and use a `for` loop to iterate through this range of values of n.

- For each value of n we will generate one $n \times n$ matrix A and 100 right-hand side vectors b. The right-hand sides will be stored as columns of $n \times 100$ matrix B. Both A and B will be filled with random floating point numbers between 0 and 1. Use the command `rand` for this.

- For each n this describes 100 linear systems $Ax = b$. Solve the systems with the two methods described above (built-in solver and LU decomposition). For each method record the time it took to solve the 100 linear systems. To record the time, use the commands `tic` and `toc`.

- For each method plot a graph of how long the computation takes as a function of n. Plot the total time for the 100 systems (not the average time). The Matlab command to plot these graphs is `plot`. To give a title and labels to the axes, use `title, xlabel, ylabel`. Plot the two graphs in the same figure, and use `legend` to label each of them. Use solid and dashed lines for the two graphs.

- Once you have the results and have plotted them, please comment briefly in your document about how the two methods behave in terms of computation time.