Computational Physics

Problem Set 7, October 14, 2025

Due: Monday, October 20, 2025 **by 11:59 PM**

Link to join GitHub classroom to submit homework solution: **Click here**.

Submit to the TA a link to the repository checked into your GitHub account containing a Jupyter Notebook including solutions for homework problems. The directory tree of the repository should include a directory for "homework" with subdirectories for each individual homework assignment.

You *must* label all axes of all plots, including the units if applicable.

1 Code LU Solver (40%)

In this problem you will make a code that takes an LU-decomposed matrix and solves the equation $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$, $\mathbf{L}$ is a lower-triangle matrix and $\mathbf{U}$ is an upper-triangle matrix. You will then use this code to make a routine that computes the inverse of $\mathbf{A}$. The algorithms are given in the lecture notes. The point of this problem is just to practice your coding skills and see the efficiency of LU decomposition. Note that you are not required to code a routine to *perform* the LU decomposition.

A) Consider the upper-triangle 5×5 matrix $\mathbf{U}_{\text{test}}$, the lower-triangle 5×5 matrix $\mathbf{L}_{\text{test}}$, and the 5-element vector $\mathbf{b}_{\text{test}}$ for testing, given by

$$
\begin{aligned}
\mathbf{U}_{\text{test}} &= \begin{pmatrix} -5 & 2 & 0 & -2 & -3 \\ 0 & 1 & 0 & -5 & 4 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 4 & -2 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \\
\mathbf{L}_{\text{test}} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 2 & 2 & -4 & 1 & 0 \\ 5 & -4 & -3 & -2 & 1 \end{pmatrix} \\
\mathbf{b}_{\text{test}} &= \begin{pmatrix} 2 \\ -3 \\ -3 \\ 2 \\ 3 \end{pmatrix} .
\end{aligned}
\tag{1}
$$

Find the corresponding matrix $\mathbf{A}_{\text{test}} = \mathbf{L}_{\text{test}} \cdot \mathbf{U}_{\text{test}}$ and solve for $\mathbf{x}_{\text{test}}$ using the numpy.linalg.solve function.

B) Write a routine that, given any $\mathbf{L}$, $\mathbf{U}$, and $\mathbf{b}$, solves for $\mathbf{x}$. The code should allow any size N. DO NOT use any linalg functions from python (numpy or scipy); just use the algorithm presented in class. Test this code on your test matrices from part A to make sure it works properly.

C) Using this code, write another routine to invert **A**, given **L** and **U**. Again, DO NOT use any python linalg functions. Test this code as well.

2 Circuits of Resistors (60%)

A) Exercise (6.1) in *Newman* (pg. 220).

B) Exercise (6.7) in *Newman* (pg. 240).