MATH 3351-1 | **Numerical Analysis I** | **Fall 2025**

Homework 3

Due Date: September 16

Grace Etzel

Complete the assignment in Google Colab and typeset your wite-up in LaTeX. Show all work and code, and provide clear justifications for your answers. Submissions that are messy or poorly organized may be returned with a grade of zero.

Link to Google Colab: link

Problem 1. Understanding operations counts and computational costs. Remember to show all steps; no coding required.

(a) The approximate operation count for the Gaussian elimination is $\frac{2n^3}{3}$. Estimate how much longer it would take to solve a system if the size n is quadrupled. Express your answer as a multiple of the original time.

Solution. We denote the quadrupled form as $4n$. We plug this into the elimination to get

$$\frac{2(4n)^3}{3} = 64\left(\frac{2n^3}{3}\right).$$

That is, the cost increases by 64 units.

(b) Suppose your computer performs a back substitution for a 5000×5000 upper-triangular system in 0.005 seconds. Use the estimates

$$\text{Back substitution: } n^2, \quad \text{Elimination: } \frac{2n^3}{3}.$$

to estimate how long it would take to perform a full Gaussian elimination for a 5000×5000 system. Round your answer to the nearest second.

Solution. Denote C_1 as the numerical cost for backward substitution. We are given

$$\begin{aligned} C_1 &= n^2 \\ &= 5000^2 \\ &= 25,000,000. \end{aligned}$$

Next, we want to compute for a constant that allows to interchange between time units and cost units. So, let α denote this constant so that $C_1 = \alpha t$, where t is in seconds. Then, to solve for α we have

$$
\begin{aligned}
\alpha =& \frac{C_1}{t} \\
=& \frac{25,000,000}{0.005} \\
=& 5,000,000,000.
\end{aligned}
$$

Then, we denote C_2 as the Gaussian Elimination cost. To find the cost in time, we consider

$$
\begin{aligned}
t &= \frac{C_2}{\alpha} \\
&= \frac{\frac{2n^3}{3}}{\alpha} \\
&= \frac{\frac{2(5000)^3}{3}}{5000000000} \\
&= \frac{250}{15} \\
&= \frac{50}{3}
\end{aligned}
$$

This gives us a elimination time cost of 17 seconds.

(c) On the same computer, assume back substitution for a 4000×4000 upper-triangular system takes 0.002 seconds. Now estimate:

- The time required for forward elimination, and
- The time for back substitution

when solving a general system of size 9000×9000. Use the same operation count assumptions and round your answer to the nearest second.

Solution. We use the same notation as before. First, we find α, which gives us

$$
\begin{aligned}
\alpha &= \frac{C_1}{t} \\
&= \frac{4000^2}{0.002} \\
&= 8,000,000,000.
\end{aligned}
$$

Next, we are interested in solving for the forward elimination t time in seconds for a 9000×9000 matrix. That is,

$$\begin{aligned} t &= \frac{C_2}{\alpha} \\ &= \frac{\frac{2n^3}{3}}{\alpha} \\ &= \frac{\frac{2(9000)^3}{3}}{\alpha} \\ &= \frac{\frac{2(9000)^3}{3}}{8,000,000,000} \\ &= 60.75. \end{aligned}$$

That is, the cost for the forward elimination is 61 seconds. Next, we want to consider the cost for back substitution. Then,

$$\begin{aligned} t &= \frac{n^2}{\alpha} \\ &= \frac{9000^2}{8,000,000,000} \\ &= 0.01. \end{aligned}$$

That is, the backward substitution is 0.01 seconds.

(d) Suppose you want to solve 1000 different systems, all with the same coefficient matrix A (of size 9000×9000) but different right-hand sides $\vec{b}$. Compare the total time required using

- Full Gaussian elimination on each system, versus
- LU decomposition once, followed by back substitution for each right-hand side.

Clearly justify which method is more efficient and by how much.

Solution. Note that the LU factorization costs about n^2 whereas the Gaussian elimination costs about $\frac{2n^3}{3}$. If we are solving 1000 different systems using a 9000×9000 matrix, we can use the estimates above to compare the costs. For the LU factorization, we have

$$LU = 1000(0.01) = 10 \text{ seconds}$$

For the Gaussian Elimination, we have

$$\text{Elimination} = 1000(61) = 61,000 \text{ seconds.}$$

Then, the LU method takes $61+10=71$. Notice that the LU method is more efficient, since it lowers costs by $60,929$ seconds.

Problem 2. Solving Hilbert systems and analyzing conditioning. Let $H \in \mathbb{R}^{n\times n}$ be the Hilbert matrix with $H_{ij} = \dfrac{1}{i+j-1}$ and let $\vec{b} = 1$.

(a) Hand Calculation for $n = 3$.

(1) Gaussian elimination: Solve $H\vec{x} = \vec{b}$ by elimination. Show the row operations (or the equivalent elimination steps) and the back-substitution, and report the final $\vec{x}_{GE}$.

Solution. When we compute H, we get

$$H = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}.$$

We will use Gaussian Elimination to find x_1, x_2, and x_3:

$$\left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & 1 \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & 1 \end{array}\right) \begin{array}{c} (-\frac{1}{2}R_1 + R_2 \mapsto R_2) \\ \to \end{array} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & 1 \end{array}\right)$$

$$\begin{array}{c} (-\frac{1}{3}R_1 + R_3 \mapsto R_3) \\ \to \end{array} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{2} \\ 0 & \frac{1}{12} & \frac{4}{45} & \frac{2}{3} \end{array}\right)$$

$$\begin{array}{c} (12R_2 \mapsto R_2) \\ \to \end{array} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ 0 & 1 & 1 & 6 \\ 0 & \frac{1}{12} & \frac{4}{45} & \frac{2}{3} \end{array}\right)$$

$$\begin{array}{c} (-\frac{1}{12}R_2 + R_3 \mapsto R_3) \\ \to \end{array} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ 0 & 1 & 1 & 6 \\ 0 & 0 & \frac{1}{180} & \frac{1}{6} \end{array}\right)$$

$$\begin{array}{c} (180R_3 \mapsto R_3) \\ \to \end{array} \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{4} & 1 \\ 0 & 1 & 1 & 6 \\ 0 & 0 & 1 & 30 \end{array}\right)$$

Next, we use backward substitution to obtain the solution. First, note that $x_3 = 30$. Then,

$$\begin{aligned} x_2 &= 6 - x_3 \\ &= 6 - 30 \\ &= -24. \end{aligned}$$

So, $x_2 = -24$. Then, we want to compute x_3, which gives us

$$\begin{aligned} x_1 &= 1 - \frac{x_2}{2} - \frac{x_3}{3} \\ &= 1 - \frac{-24}{2} - \frac{30}{10} \\ &= 3. \end{aligned}$$

So, $\vec{x} = \begin{pmatrix} 3 \\ -24 \\ 30 \end{pmatrix}$.

(2) LU factorization (no pivoting): Compare $H = LU$; write L and U explicitly. Solve $L\vec{c} = \vec{b}$ and $U\vec{x} = \vec{c}$, and verify that $\vec{x}$ matches $\vec{x}_{GE}$ in part (1).

Solution. First, we would like to obtain U. Take the matrix from the previous problem:

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{12} & \frac{1}{12} \\ 0 & \frac{1}{12} & \frac{4}{45} \end{pmatrix} \quad \begin{matrix} (-R_2 + R_3 \mapsto R_3) \\ \rightarrow \end{matrix} \quad \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{12} & \frac{1}{12} \\ 0 & 0 & \frac{1}{180} \end{pmatrix} = U$$

Next, we form L by identifying the transformations to make U and placing them in their respective position:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 1 & 1 \end{pmatrix}.$$

First, we perform forward substitution on $L\vec{c} = \vec{b}$, which gives the following transformation below:

$$L|\vec{b} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ \frac{1}{2} & 1 & 0 & 1 \\ \frac{1}{3} & 1 & 1 & 1 \end{array}\right).$$

Then, using forward substition, where $c_1 = 1$, we can find c_2 and c_3:

$$\begin{aligned} c_2 &= 1 - \frac{c_1}{2} \\ &= 1 - \frac{1}{2} \\ &= \frac{1}{2} \end{aligned}$$

and

$$\begin{aligned} c_3 &= 1 - \frac{c_1}{3} - c_2 \\ &= 1 - \frac{1}{3} - \frac{1}{2} \\ &= 1 - \frac{5}{6} \\ &= \frac{1}{6}. \end{aligned}$$

Next, we solve for $U\vec{x} = \vec{c}$, which is represented by the following transformation:

$$U|\vec{c} = \left(\begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{1}{3} & 1 \\ 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{2} \\ 0 & 0 & \frac{1}{180} & \frac{1}{6} \end{array}\right).$$

Then, we can use backward substitution to solve for $\vec{x}$. Notice that $x_3 = 30$. Notice that the steps for the backward substitution follow similarly to the backward substitution in (1). Then,$\vec{x} = \begin{pmatrix} 3 \\ -24 \\ 30 \end{pmatrix}$.

(3) Conditional numbers: Compute

$$\kappa_\infty(H) = \|H\|_\infty \left\|H^{-1}\right\|_\infty, \quad \kappa_2(H) = \frac{\sigma_{\max}(H)}{\sigma_{\min}(H)}.$$

Give a 1-2 sentence interpretation of what each condition number says about the sensitivity of the solution to perturbations in b (and in H).

Solution. Define

$$\kappa_\infty(H) = \|H\|_\infty \left\|H^{-1}\right\|_\infty$$

To find $\max(H)$, we know that the largest sum of the rows is the first. So, we can say

$$\|H\|_\infty = 1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}.$$

To find $\|H^{-1}\|_\infty$, we compute the inverse of H, which gives us

$$H^{-1} = \begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}.$$

The absolute value of the second row has the largest sum, so

$$\begin{aligned} \left\|H^{-1}\right\| &= |-36| + |192| + |-180| \\ &= 408. \end{aligned}$$

Then, $\kappa_\infty = 408 \cdot \dfrac{11}{6}$. This tells us that small errors in $\vec{b}$ may lead to large errors in $\vec{x}$. Next, we want to compute $\kappa_2(H)$. To do this, we first must find the singular values of $H^T H$. That is, we need to find the eigenvalues and square them. First, we find that

$$H^T H = \begin{pmatrix} \frac{49}{36} & \frac{3}{4} & \frac{21}{40} \\ \frac{3}{4} & \frac{61}{144} & \frac{3}{10} \\ \frac{21}{40} & \frac{3}{10} & \frac{769}{3600} \end{pmatrix}.$$

To set up computing the eigenvalues, we get

$$\det(H^T H - \lambda I) = \begin{vmatrix} \frac{49}{36} - \lambda & \frac{3}{4} & \frac{21}{40} \\ \frac{3}{4} & \frac{61}{144} - \lambda & \frac{3}{10} \\ \frac{21}{40} & \frac{3}{10} & \frac{769}{3600} - \lambda \end{vmatrix}$$

When we solve, we get $\lambda \approx 7.22, 0.015, 1.98$. Then, we can find $\sigma_{\max} = \sqrt{7.22} \approx 2.687$ and $\sigma_{\min} = \sqrt{0.015} = 0.122$. Then,

$$\begin{aligned} \kappa_2(H) &= \frac{\sigma_{\max}(H)}{\sigma_{\min}(H)} \\ &\approx \frac{2.687}{0.122} \\ &\approx 22. \end{aligned}$$

Accordingly, this tells that that a 1% error could lead to a 22%, which means that this systems is not very stable.

(b) Timing the solvers: Solve $H\vec{x} = \vec{b}$ for $n \in \{2, 3, 5, 10\}$ using:

- the lecture-note implementation of partial-pivoting LU ($PA = LU$), and
- np.linalg.solve.

Compare their solutions. For each n when using the partial-pivoting LU, report in a table: n, time, and the residual norm $\left\|\vec{b} - H\vec{x}\right\|_2$. You can measure wall-clock time with time.perf_counter().

(c) Conditioning vs. Size: For each n, compute the conditional number with np.linalg.cond:

$$\kappa_2(H) = \text{np.linalg.cond}(H, 2).$$

Also report $\kappa_\infty(H)$. Discuss how condition numbers grow with n. Based on your results, comment on whether the Hilbert matrix is well-conditioned or ill-conditioned as n increases.

Solution. Consider the table in the Google colab on part (c). Notice that the condition numbers grow with n. This tells us that the Hilbert matrix is ill-conditioned as n increases.

Problem 3. LU decomposition with and without pivoting: Let

$$A(\varepsilon) = \begin{pmatrix} \varepsilon & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}, \quad \text{with} \quad \varepsilon = 10^{-12}.$$

(Keep ε symbolic in your hand work; substitute $\varepsilon = 10^{-12}$ for the numeric comparison.)

(a) Hand Calculation

(1) LU without pivoting: Perform $A = LU$. Write L and U explicitly and solve $L\vec{c} = \vec{b}, U\vec{x} = \vec{c}$. Show all steps, including the elimination process to obtain L and U, and the substitution steps to find x.

Solution. We begin by attempting the LU factorization.

$$\begin{pmatrix} \varepsilon & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \overset{(-\frac{1}{\varepsilon}R_1 + R_2 \mapsto R_2)}{\longrightarrow} \begin{pmatrix} \varepsilon & 1 & 1 \\ 0 & 1 - \frac{1}{\varepsilon} & 1 - \frac{1}{\varepsilon} \\ 1 & 1 & 2 \end{pmatrix}$$

$$\overset{(-\frac{1}{\varepsilon}R_1 + R_3 \mapsto R_3)}{\longrightarrow} \begin{pmatrix} \varepsilon & 1 & 1 \\ 0 & 1 - \frac{1}{\varepsilon} & 1 - \frac{1}{\varepsilon} \\ 0 & 1 - \frac{1}{\varepsilon} & 2 - \frac{1}{\varepsilon} \end{pmatrix}$$

$$\overset{(-R_2 + R_3 \mapsto R_3)}{\longrightarrow} \begin{pmatrix} \varepsilon & 1 & 1 \\ 0 & 1 - \frac{1}{\varepsilon} & 1 - \frac{1}{\varepsilon} \\ 0 & 0 & 1 \end{pmatrix} = U$$

That is, we have

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{\varepsilon} & 1 & 0 \\ \frac{1}{\varepsilon} & 1 & 1 \end{pmatrix}.$$

So when we solve $L\vec{c} = \vec{b}$, with $c_1 = 2$, we get

$$\begin{aligned} c_2 &= 3 - \frac{c_1}{\varepsilon} \\ &= 3 - \frac{2}{\varepsilon}. \end{aligned}$$

Then,

$$\begin{aligned} c_3 &= 4 + \frac{c_1}{\varepsilon} - c_2 \\ &= 4 - \frac{2}{\varepsilon} - \left(2 - \frac{2}{\varepsilon}\right) \\ &= 1. \end{aligned}$$

Next, we solve for $U\vec{x} = \vec{c}$, with $x_3 = 1$, we get

$$\begin{aligned} x_2 &= \frac{3 - \frac{2}{\varepsilon}}{1 - \frac{1}{\varepsilon}} - 1 \\ &= \frac{\frac{3\varepsilon - 2}{\varepsilon}}{\frac{\varepsilon - 1}{\varepsilon}} - 1 \\ &= \frac{3\varepsilon - 2}{\varepsilon - 1} - 1. \end{aligned}$$

Finally,

$$\begin{aligned} x_1 &= 4 - \left(\frac{3\varepsilon - 2}{\varepsilon - 1} - 1\right) - \frac{1}{\varepsilon} \\ &= 4 - \frac{3\varepsilon - 2}{\varepsilon - 1} + \frac{\varepsilon - 1}{\varepsilon} \\ &= -\frac{1}{\varepsilon - 1}. \end{aligned}$$

That is, our final solution is $\vec{x} = \begin{pmatrix} -\frac{1}{\varepsilon - 1} \\ \frac{3\varepsilon - 2}{\varepsilon - 1} - 1 \\ 1 \end{pmatrix}$.

(2) LU with partial pivoting: Compute $PA = LU$ using partial pivoting. Write P, L and U explicitly and solve $L\vec{c} = P\vec{b}, U\vec{x} = \vec{c}$. Show all steps, including pivot selection, row swaps, elimination, and substitution processes.

Solution. We attempt a similar elimination as in the part above, except that we set

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

So, $PA = LU$ gives us

$$\begin{pmatrix} 1 & 1 & 1 \\ \varepsilon & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{matrix} (-\varepsilon R_1 + R_2 \mapsto R_2) \\ \to \end{matrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1-\varepsilon & 1-\varepsilon \\ 1 & 1 & 2 \end{pmatrix}$$

$$\begin{matrix} (-R_1 + R_3 \mapsto R_3) \\ \to \end{matrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1-\varepsilon & 1-\varepsilon \\ 0 & 0 & 1 \end{pmatrix} = U.$$

That is, we find that

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \varepsilon & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Next, we solve $L\vec{c} = P\vec{b}$, where $c_1 = 3$, we get

$$\begin{aligned} c_2 &= 2 - \varepsilon c_2 \\ &= 2 - 3\varepsilon. \end{aligned}$$

Then,

$$\begin{aligned} c_3 &= 4 - 3 \\ &= 1. \end{aligned}$$

Next, we would like to solve $U\vec{x} = \vec{c}$, where $x_3 = 1$. Then,

$$\begin{aligned} x_2 &= \frac{2 - 3\varepsilon}{1 - \varepsilon} - 1 \\ &= \frac{3\varepsilon - 2}{\varepsilon - 1} - 1. \end{aligned}$$

Finally, we get

$$\begin{aligned}
x_1 &= 3 - x_3 - x_2 \\
&= 3 - 1 - \left(\frac{2-3\varepsilon}{1-\varepsilon} - 1\right) \\
&= 3 - \frac{2-3\varepsilon}{1-\varepsilon} \\
&= \frac{1}{1-\varepsilon} \\
&= -\frac{1}{\varepsilon - 1}.
\end{aligned}$$

That is, our final solution is $\vec{x} = \begin{pmatrix} -\frac{1}{\varepsilon-1} \\ \frac{3\varepsilon-2}{\varepsilon-1} - 1 \\ 1 \end{pmatrix}$.

(b) Numerical Comparison Using the code provided in the lecture notes, implement the following for $\varepsilon = 10^{-12}$:

- LU decomposition without pivoting (as in part (a)).
- LU decomposition with partial pivoting (as in part (b)).

Compute the numerical solutions to $\vec{x}$ for both methods. Additionally, compute a reference solution using NumPy's np.linalg.solve(A, b). Present the results in a table comparing:

- The solutions $\vec{x}$ from both methods and the reference solution.
- The absolute error $\|\vec{x}_{\text{method}} - \vec{x}_{\text{ref}}\|_2$ for each method, where $\vec{x}_{\text{ref}}$ is the NumPy solution.

Discuss the differences between the solutions and explain why partial pivoting matters for this system. In your discussion, consider:

- The effect of the small value $\varepsilon = 10^{-12}$ on the matrix's conditioning (you may compute the condition number $\kappa(A)$ using np.linalg.cond(A)).
- The role of partial pivoting in improving numerical stability, especially in the presence of small pivots.
- Any numerical issues (e.g., roundoff errors, small pivots) observed in the computations.

Solution. For the solution, please look at the Google colab.