CS/MATH 375, Fall 2024 — HOMEWORK # 6
Due : Oct. 5th at 10:00pm on Canvas

Instructions

– **Report:** In general, your report needs to read coherently. That is, start off by answering question 1. Fully answer the question, and provide all the information needed to understand your answer. If Matlab code or output is part of the question, include that code or output (e.g., screenshot) alongside your narrative answer. If discussion is required for a question, include that. Overall, your report is your narrative explanation of what was done, your answers to the specific questions, and how you arrived at your answers. *Your report should include your Matlab scripts, code output, and any figures.*

– **What to hand in:** Submission must be one **single PDF** document, containing your entire report, submitted on Canvas.

– **Partners:** You are allowed to (even encouraged) to **work in pairs**. If you work with a partner, only one member of the group should need to submit a report. On Canvas, both partners sho uld join a group (numbered 1 through 15). Then either member can upload the report for the entire group. Groups of more than 2 students are not allowed.

– **Typesetting:** If you write your answers by hand, then make sure that your handwriting is readable. Otherwise, I cannot grade it.

– **Plots:** All plots/figures in the report must be generated in Matlab or Python and not hand drawn (unless otherwise specified in the homework question).

In general, make sure to (1) title figures, (2) label both axes, (3) make the curves nice and thick to be easily readable, and (4) include a legend for the plotted data sets. The font-size of all text in your figures must be large and easily readable.

Reading: Read Chapters 2.1 through 2.4 in the book.

Problems:

1. *Programming*

   Write the following Matlab functions. Each of them should use the simplest possible algorithms with nested "for" loops. None of them should involve pivoting.

   - `function [L, U] = naiveLU(A)`. This function should return the *unit* lower triangular matrix L and the upper triangular matrix U such that $LU = A$, obtained via naive forward elimination.

- `function y = naiveLTriSol(L,b)`. This function should solve the *unit* lower-triangular system $L\vec{y} = \vec{b}$ using forward substitution. Your code should explicitly assume that L has ones on the diagonal.
- `function x = naiveUTriSol(U,y)`. This function should solve the upper-triangular system $U\vec{x} = \vec{y}$ using backward substitution.

2. Prove or provide a counter-example for each of the following statements. You can use Matlab to find counter-examples. Assume A is a matrix and c is a scalar.

   (a) $\|cA\| = |c|\,\|A\|$

   (b) $\kappa(cA) = \kappa(A)$ for any nonzero constant c

   (c) $\kappa(A)$ is the same for every matrix norm

3. For each of the following systems: verify the solution by hand, report the infinity norm condition number of A (`cond(A,inf)`), solve the system numerically using your functions from problem 1, solve the system numerically using Matlab's built in `lu` routine (with pivoting), and compare norm of the error for your code and Matlab's code (use the infinity norm and report just absolute error).

   (a)
$$A = \begin{bmatrix} 2 & -2 & -1 \\ 4 & 1 & -2 \\ -2 & 1 & -1 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}.$$

   (b)
$$A = \begin{bmatrix} 10^{-16} & 1 & -1 \\ 2 & 0 & 1 \\ 1 & 2 & -3 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} -3 \times 10^{-16} \\ -4 \\ -5 \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} -3 \\ 2 \\ 2 \end{bmatrix}.$$

4. Consider the matrix
$$A = \begin{bmatrix} 1 & 1 \\ b & 1 \end{bmatrix},$$
   where $b \neq 1$.

   (a) Find $\|A\|_1$ in terms of b.

   (b) Find $\|A^{-1}\|_1$ in terms of b. Remember that
$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

   (c) Using the previous two parts, find $\kappa(A)$ in terms of b

   (d) Explain geometrically why the condition number grows as $b \to 1$

(e) If you solve the linear system $Ax = b$ using Gaussian elimination without pivoting, for what b should you expect to have 10 digits of accuracy, when the computations are carried out using double precision? Recall that for double precision we have $\epsilon_m \approx 2.2 \times 10^{-16}$

5. Let

$$A = \begin{bmatrix} 1 & 1+\delta \\ 1-\delta & 1 \end{bmatrix},$$

where δ is some small number.

(a) Find the LU factorization of A without pivoting

(b) Using double precision floating point arithmetic, for what value of δ will solving via LU factorization (without pivoting) fail?

(c) Find the determinant of A

(d) Using double precision floating point arithmetic, for what values of δ will $\det(A)$ equal to 0?

6. The **Hilbert Matrix** of size k, $A = \text{HILB}(k)$, is defined by

$$a_{ij} = 1/(i + j - 1).$$

For example,

$$A = \text{HILB}(3) = \begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix}.$$

Consider the linear system $A\mathbf{x} = \mathbf{b}$, where A is a Hilbert Matrix and the vector $\mathbf{b}$ is given by $b_i = \sum_{j=1}^{n} a_{ij}$ (i.e. $\mathbf{b}$ is the row-sum of A).

(a) Use the Matlab command `hilb` to create Hilbert Matrices for $8 \leq k \leq 12$. Create a table of the condition numbers of these matrices using the matlab command `cond`.

(b) Without using Matlab, what is the true solution, $\mathbf{x}_{\text{true}}$, to the linear systems? Justify. *Hint:* the answer is nearly identical for any k.

(c) Without solving the system, how many decimal digits of accuracy do you expect for $8 \leq k \leq 12$ if you solve the system using Gaussian elimination without pivoting? Explain why.

(d) Now solve the linear systems for $8 \leq k \leq 12$. Use the Matlab *backslash* operator for the linear solve. For each k, write down your Matlab solution. Make sure you set the `format` to `long` so that Matlab displays 16 digits of the solution.

(e) How many digits of accuracy do you see in the solution components for different values of n. Double precision has about 15 or 16 digits of accuracy. Does this agree with your answer in (c)?

(f) Tabulate the relative error in the solution $\|x_t - x\|_2/\|x_t\|_2$ and the relative value of the residual $\|b - Ax\|_2/\|b\|_2$, where x_t is the true solution and x is the solution returned by the Matlab *backslash* operator. Are both relative errors and relative residuals small as n increases? Explain your results.