21-671: Computational Linear Algebra
Homework Assignment Due 11:59pm on 09/30/24 Solutions

Instructions: Write or type solutions to the following problems and create a PDF file of your work. Then upload that PDF to Gradescope before the due date and time.

1. (20 pts.) (Norms and Calculus) Consider the problem of computing the 2-norm of the 2×2 matrix $A = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$. Recall that
$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2.$$
Now every unit vector $x \in \mathbb{R}^2$ is of the form $x = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$ for some $\theta \in [0, 2\pi]$. Thus we have
$$Ax = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \begin{bmatrix} \cos\theta - 2\sin\theta \\ -3\cos\theta + 4\sin\theta \end{bmatrix},$$
so we have
$$\|Ax\|_2^2 = (\cos\theta - 2\sin\theta)^2 + (4\sin\theta - 3\cos\theta)^2 \equiv f(\theta).$$
Now use calculus to find $\max \|Ax\|_2^2$ and thus $\|A\|_2$. Also give a unit vector that maximizes $\|Ax\|_2$. (You may wish to use a plot of $f(\theta)$ to see where f is maximized as finding the inverse of a periodic function can be tricky.)

2. (40 pts.) (Programming) **Include the code and any results from the code to solve this problem in your single PDF file. Your code should be commented.** In this problem we are going to investigate how long the built-in linear system solvers in MATLAB or Python require to solve linear systems of increasing size. We will visualize the results by generating a plot of the run time vs. the matrix size.

   Write Python or MATLAB code to perform the following tasks:

   (a) For each n in the table below, Construct a random matrix A (with entries between 0 and 1) of size $n \times n$. Use the `rand()` function in MATLAB or the `numpy.random.rand()` function in Python to do this.

   (b) Construct a random true solution vector x (with entries between 0 and 1) of size $n \times 1$.

   (c) Find the correct right hand side vector b by computing $b = Ax$.

   (d) Use the built-in linear solver (\ in MATLAB or `numpy.linalg.solve` in Python) to find the approximate solutions $\hat{x}$. Calculate the time each solve takes using the built-in timing functions (`tic` and `toc` in MATLAB or `time.time()` in Python). For example, in MATLAB, a code snippet that will do this is:
   `tic;xhat = A\b;toc`.

   (e) Now that you have $\hat{x}$, calculate the 2-norm of x, the 2-norm of the relative error, and the relative 2-norm of the residual $b - A\hat{x}$ using the built-in norm function. Enter these values in the table below.

   (f) Once you have completed the above steps for all n in the table below, construct a log-log scale plot with n on the horizontal axis and the Solve Time on the vertical axis. Label your axes and give your plot a title. Can you draw any conclusions from the curve you plotted? Include your plot in your homework submission.

n	Solve Time	$\|\hat{x}\|_2$	Relative Error $\frac{\|x-\hat{x}\|_2}{\|x\|_2}$	Residual Norm $\frac{\|b-A\hat{x}\|_2}{\|b\|_2}$
100				
200				
500				
800				
1000				
2000				
5000				
8000				
10000				
15000				
20000				