Problem Set 4
Math 146 Spring 2023
Due Thursday, March 9, 11:59 PM

Note: Number 4(a-d) is a "pen and paper" problem for which you should show work. Numbers 1 - 3 & 4e are Matlab problems that require code submissions. All answers should be submitted in a write-up.

1. (a) (20 points) Create a function to solve the system $AX = B$, for A, X, and B, $n \times n$ matrices. The function should take as input matrices A and B and give the output X. Validate your code. Explain how you validated and provide any relevant output. Note: This only requires a slight modification to the code you created in Problem Set 3, Number 3b.

   See directsolve_matrix_pivoting.m for the function and PS4N1.m for the script. I validated my code by creating a random 100×100 matrix A and a random 100×100 matrix X_{exact}. I then multiplied AX_{exact} to form the matrix B. I then used my function to solve $AX = B$ for X. By forming a random X_{exact} instead of a random B, I now have the ability to form the error matrix $|X - X_{exact}|$. I therefore formed this error matrix and then found the maximum absolute value component and divided by the maximum absolute value component of X_{exact}. Note: Finding the absolute value component of the error matrix is NOT using the max norm of a matrix (which is the "max row sum"). That norm is based on seeing a matrix as an operator, which is not what we are doing for matrix X. For an example problem, I found this relative error to be about $4.3(10)^{-12}$ and the condition number of A to be $6.1(10)^3$, which would indicate I would lose about 4 digits of accuracy, which matches the relative error.

   (b) (20 points) Create a function to compute the inverse of a square $n \times n$ matrix A. The function should take as input the matrix A and give the output A^{-1}. Validate your code. Explain how you validated and provide any relevant output.

   See my_inverse.m for the function and PS4N1.m for the script. To validate my code, I used the the same randomly generated matrix A from part (a) and then calculated A^{-1} using my function. Then I found $E = |A^{-1}A - I|$, where I is the identity matrix, and found the maximum absolute value component of this and got $2.4(10)^{-14}$.

2. Included in the PS 4 Folder is a function that uses the Thomas Algorithm to solve the tridiagonal system of equations given below. Notice that the algorithm has been altered from

the pseudocode provided in class in order to eliminate one of the for loops.

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \\ \vdots \\ \\ d_n \end{bmatrix} \tag{1}$$

(a) (20 points) The function takes as input $n \times 1$ vectors (a), $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$. Modify the algorithm to make a new function, Thomas_2, which takes as input the tridiagonal matrix A and vector $\mathbf{d}$ instead. Include in your function errors if A is not a square tridiagonal matrix. (Hint: Use the command `isbanded`.) Validate your code by creating a sparse 1000×1000 tridiagonal matrix made up of random numbers from 0 to 50 and a similarly formed vector for $\mathbf{d}$. Compare your solution to the solution found by using Matlab backslash, and report the relative difference between the two solutions, with respect to the max norm.

See Thomas_2.m for the function and PS4N2.m for the script. The relative difference between my solution and the Matlab solution, with respect to the max norm is $2.1(10)^{-15}$.

(b) (30 BONUS points) Modify the algorithm to solve problems of the form

$$\begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \\ \vdots \\ \\ d_n \end{bmatrix}, \tag{2}$$

where all terms not on the tridiagonal or in the upper right corner are zeros. Show any work you did or explain how you made your modification. (You may want to explore the original algorithm a little first.) Validate your code by solving the system

$$\begin{bmatrix} 1 & -2 & 0 & 0 & 0 & -1 \\ 2 & -2 & 3 & 0 & 0 & 0 \\ 0 & -2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & -2 & -4 & -1 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x1 \\ \\ \vdots \\ \\ x6 \end{bmatrix} = \begin{bmatrix} -9 \\ 7 \\ 9 \\ 22 \\ -34 \\ 17 \end{bmatrix}. \tag{3}$$

The exact solution is $\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$. Report your relative error, with respect to the 1-norm, 2-norm, and max norm.

See Thomas_3.m for the function and PS4N2.m for the script. See Problem Set 4 2c solutions.pdf for the detailed work to find the new algorithm. The relative errors with respect to the 1, 2, and max norms are $4.4(10^{-16})$, $4.2(10^{-16})$, and $3(10^{-16})$, respectively. Since the condition numbers of A with respect to the various norms are between 10 and 100, I would expect to lose 1-2 digits of accuracy. I have only lost about 1 digit.

3. (a) (32 points) Create a function to implement the Cholesky factorization pseudocode from class. Your function should take as input an $n \times n$ positive definite symmetric matrix A and give output $R^{\intercal}$. Validate your code by factoring the symmetric positive definite matrix $A = \begin{bmatrix} 10 & 5 & 2 \\ 5 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix}$. Explain how you then verify that your solution is correct.

   See choleskyfact.m for the function and PS4N3.m for the script. I validated my code by finding the maximum component of $E = |R^{\intercal}R - A|$, and I got $1.8(10^{-15})$. I also checked that $R^{\intercal}$ is lower triangular.

   (b) (18 points) Create a function that takes as input an $n \times n$ positive definite symmetric matrix A and a vector $\mathbf{b}$ and uses your Cholesky factorization from part (a) to solve the linear system $A\mathbf{x} = \mathbf{b}$. Validate your code. Explain how you validated the code and provide any relevant output.

   See linsolve_symmposdef.m for the function and PS4N3.m for the script. I validated my code by finding a randomly formed vector, $\mathbf{x}_{exact}$ and then finding $\mathbf{b} = A\mathbf{x}_{exact}$. Then, I used my function to solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$. Then I found the relative error of my solution, with respect to the max norm. I got a relative error of $7.8(10^{-15})$, so I have about 14 digits of accuracy. The condition number of A with respect to the max norm is 266, which tells me I would lose about 2 digits of accuracy, which matches what I saw.

4. You would like to fit the following data with a second order polynomial (a parabola).

x	y
-2	9
-1	5
0	3
1	4
2	8
3	12

   (a) (12 points) Let the polynomial be $p(x) = c_0 + c_1x + c_2x^2$. Write by hand the 6 equations given to us by this data. Also give the system of equations in matrix form.

The 6 equations are

$c_0 - 2c_1 + 4c_2 = 9$

$c_0 - c_1 + c_2 = 5$

$c_0 = 3$

$c_0 + c_1 + c_2 = 4$

$c_0 + 2c_1 + 4c_2 = 8$

$c_0 + 3c_1 + 9c_2 = 12$

In matrix form, this is

$$\begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 9 \\ 5 \\ 3 \\ 4 \\ 8 \\ 12 \end{bmatrix}.$$

(b) (17 points) Find the normal equations to this system.

First, we find $A^\intercal A$ and $A^\intercal \mathbf{b}$.

$$A^\intercal A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 & 3 \\ 4 & 1 & 0 & 1 & 4 & 9 \end{bmatrix} \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}$$

$$= \begin{bmatrix} 1+1+1+1+1+1 & -2-1+1+2+3 & 4+1+1+4+9 \\ -2-1+1+2+3 & 4+1+1+4+9 & -8-1+1+8+27 \\ 4+1+1+4+9 & -8-1+1+8+27 & 16+1+1+16+81 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 3 & 19 \\ 3 & 19 & 27 \\ 19 & 27 & 115 \end{bmatrix}$$

Then $A^\intercal \mathbf{b} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 & 3 \\ 4 & 1 & 0 & 1 & 4 & 9 \end{bmatrix} \begin{bmatrix} 9 \\ 5 \\ 3 \\ 4 \\ 8 \\ 12 \end{bmatrix} = \begin{bmatrix} 9+5+3+4+8+12 \\ -18-5+4+16+36 \\ 36+5+4+32+108 \end{bmatrix} = \begin{bmatrix} 41 \\ 33 \\ 185 \end{bmatrix}$

The normal equations are therefore given by

$$\begin{bmatrix} 6 & 3 & 19 \\ 3 & 19 & 27 \\ 19 & 27 & 115 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 41 \\ 33 \\ 185 \end{bmatrix}.$$

(c) (22 points) Solve the normal equations by hand. What polynomial gives the Least Squares best fit polynomial for this data?

I will solve using the Cholesky Factorization (You can use regular Gaussian Elimination if you prefer.) See Problem Set 4 4c solutions for details. The least squares polynomial I obtained is $p(x) = \frac{24}{7} - \frac{3}{7}x + \frac{8}{7}x^2$.

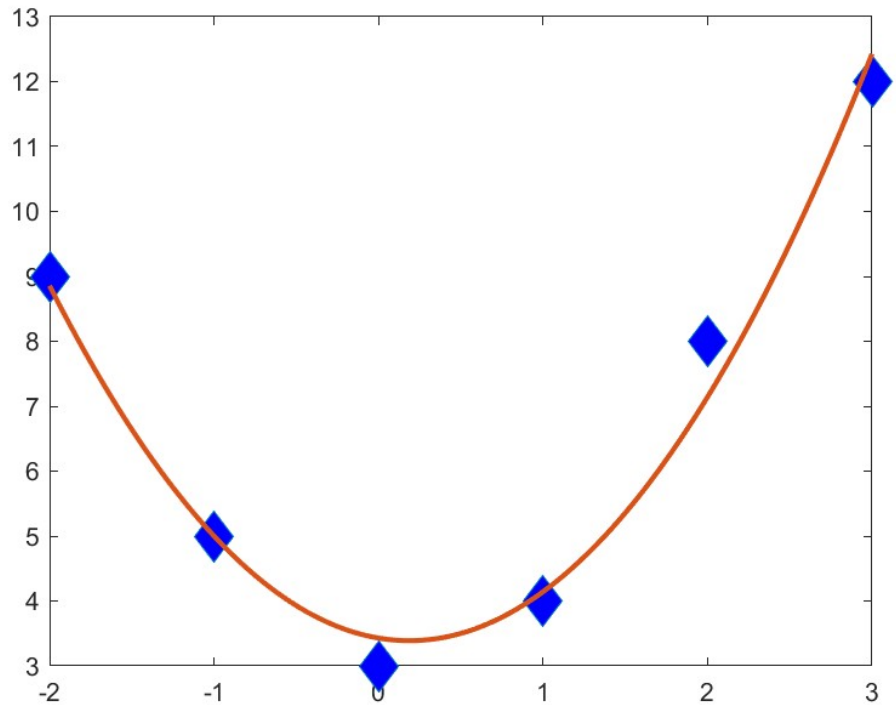
(d) (17 points) Find the residual and its magnitude.

The residual is $\mathbf{r} = \mathbf{y} - A\mathbf{c} = \mathbf{y} - p(\mathbf{x}) = \begin{bmatrix} 9 - p(-2) \\ 5 - p(-1) \\ 3 - p(0) \\ 4 - p(1) \\ 8 - p(2) \\ 12 - p(3) \end{bmatrix} = \begin{bmatrix} 1/7 \\ 0 \\ -3/7 \\ -1/7 \\ 6/7 \\ -3/7 \end{bmatrix}.$

It's magnitude is $||\mathbf{r}||_2 = \sqrt{8/7} \approx 1.069$.

(e) (22 points) Confirm your work in Matlab. Note: You can solve the normal equations using function from Problem (3b) since $A^\intercal A$ is a symmetric positive definite matrix. Plot the data and your Least Squares polynomial.

See PS4N4.m for the script. Everything matched, and below is the plot of the data and the least squares parabola.


13
12
11
10
9
8
7
6
5
4
3
-2
-1
0
1
2
3