CS/MATH 375, Fall 2024 — HOMEWORK # 10
Due : Nov. 8th 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 Ca nvas, both partners should 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.

---

1. Least Squares fitting (cubic!)

   (a) Use the provided matlab function `[x,y] = generate_ls_data(N)` with $N = 100$ to generate 100 data points. Plot the data using `plot(x,y,'o')` (you can add more commands to make the formatting nice, but use the `'o'` syntax so that MATLAB does not connect the data with lines

   (b) Next, we want to find the cubic polynomial given by

   $$p_3(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3,$$

that best fits the data. To do this, we will construct the Vandermonde matrix, A given by

$$A = \left[\vec{x}^3 \mid \vec{x}^2 \mid \vec{x} \mid \vec{x}^0\right],$$

using the data from part (a). Now, finding the cubic polynomial is equivalent to "solving" the problem

$$A \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \vec{y}.$$

Construct the Vandermonde matrix and compute the polynomial coefficients c_0, c_1, c_2, c_3 with a least squares approximation by solving the **normal equations**. It might be helpful to reuse your code from HW 8 to generate the Vandermonde matrix.

Define a fine grid of points using `xfine = linspace(0,1,1000);`. Evaluate the polynomial at these points and plot the results, along with the data from part (a) (you can use MATLAB's `polyval` command if you ordered your c's correctly).

2. Least Squares fitting (cubic again!)

We're going to redo the last problem, but now we'll use the SVD instead of the normal equations.

(a) Use the provided matlab function `[x,y] = generate_ls_data(N)` with $N = 100$ to generate 100 data points. Plot the data using `plot(x,y,'o')`.

(b) Next, we want to find the cubic polynomial given by

$$p_3(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3,$$

that best fits the data. To do this, we will construct the Vandermonde matrix, A given by

$$A = \left[\vec{x}^3 \mid \vec{x}^2 \mid \vec{x} \mid \vec{x}^0\right],$$

using the data from part (a). Now, finding the cubic polynomial is equivalent to "solving" the problem

$$A \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \vec{y}.$$

Construct the Vandermonde matrix and compute the polynomial coefficients c_0, c_1, c_2, c_3 with a least squares approximation by using the **SVD** of A (you can

use Matlab's built-in `svd` command). Do you get the same answer as in the previous homework? Define a fine grid of points using `xfine = linspace(0,1,1000);`. Evaluate the polynomial at these points and plot the results, along with the data from part (a).

(c) Redo the previous two parts but with $n = 500$ data points.

3. QR factorization

Consider the matrix

$$A = \begin{bmatrix} -4 & -4 \\ -2 & 7 \\ 4 & -5 \end{bmatrix}.$$

(a) Use the Gram-Schmidt procedure (as I did in lecture) on the columns of A to produce the QR factorization of A (technically what I showed in class is the "skinny" QR factorization). Do this "by hand".

(b) Use Matlab's built-in `qr` command to compute the QR factorization of A. How does this differ from your answer in part a? (*Hint:* the command `rats` can be useful when comparing Matlab's decimal representation of a matrix to the one you computed by hand.)

(c) With the right hand side $\vec{b} = [3, 9, 0]^T$, use your "skinny" factorization from part a to find the least squares solution.

(d) With the same right hand side, use the full factorization that Matlab gave you in part b to find the least squares solution. Is this the same as your answer to part c? Calculate the 2-norm of the residual *without actually calculating the residual*.