Lab Session 11

MA-581 : Numerical Computations Lab R. Alam Date: October 25, 2021

The purpose of this lab tutorial is to solve the Least-Squares Problem (in short, LSP) $Ax \simeq b$. Here $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$, and usually m is much bigger than n.

Origin: Suppose that we have a data set (t_i, b_i), for $i = 1 : m$, that have been obtained from some experiment. These data are governed by some unknown laws. So, the task is to come up with a model that best fits these data. A model is generated by a few functions, called model functions, $\phi_1, \ldots, \phi_n$. Therefore once a model is chosen, the task is to find a function p from the span of the model functions that best fits the data.

Suppose that the model functions $\phi_1, \ldots, \phi_n$ are given. For $p \in \mathrm{span}(\phi_1, \ldots, \phi_n)$, we have $p = x_1\phi_1 + \cdots + x_n\phi_n$ for some $x_j \in \mathbb{C}$. Now, forcing p to pass through the data (t_i, b_i) for $i = 1 : m$, we have $p(t_i) = b_i + r_i$, where r_i is the error. We want to choose that p for which the sum of the squares of the errors r_i is the smallest, that is, $\sum_{i=1}^m |r_i|^2$ is minimized.

Now $p(t_i) = b_i + r_i$ gives $x_1\phi_1(t_i) + \cdots + x_n\phi_n(t_i) = b_i + r_i$. Thus in matrix notation,

$$\begin{bmatrix} \phi_1(t_1) & \cdots & \phi_n(t_1) \\ \phi_1(t_2) & \cdots & \phi_n(t_2) \\ \vdots & \cdots & \vdots \\ \phi_1(t_m) & \cdots & \phi_n(t_m) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}.$$

This is of the form $Ax = b + r$ and we have to choose $x \in \mathbb{C}^n$ for which $\|r\|_2 = \|Ax - b\|_2$ is minimized. We write this as LSP $Ax = b$.

1. Over 400 years ago Galileo, attempting to find a mathematical description of falling bodies, studied the paths of projectiles. One experiment consisted of rolling a ball down a grooved ramp inclined at a fixed angle to the horizontal, starting the ball at a fixed height h above a table of height 0.778 meters. When the ball left the end of the ramp, it rolled for a short distance along the table and then descended to the floor. Galileo altered the release height h and measured the horizontal distance d that the ball traveled before landing on the floor. The table below shows data from Galileos notes (with measurements converted from puntos to meters)

   Data from Galileos inclined plane experiment.

h	0.282	0.564	0.752	0.948
d	0.752	1.102	1.248	1.410

   Determine a a least squares linear fit $y = c_1 + c_2x$ with the following MATLAB code:

```
%%% Set up data
h = [0.282; 0.564; 0.752; 0.940]; d = [0.752; 1.102; 1.248; 1.410];

%%% Form the 4x2 matrix A and solve for the coefficient vector c.
A = [ones(size(h)), h]; c = A\d;   % solution of the LSP Ac =d
cc = flipud(c);    % order the coefficients in descending
                   % order for polyval
%%% Plot the data points
plot(h,d,'b '), title('Least Squares Linear Fit'), hold
xlabel('release height'), ylabel('horizontal distance')

%%% Plot the line of best fit
hmin = min(h); hmax = max(h); h1 = [hmin:(hmax-hmin)/100:hmax];
plot(h1, polyval(cc, h1), 'r'), axis tight
```

This code produces the line y = 0.4982 + 0.9926x. This line is plotted along with the data points. A measure of the amount by which the line fails to hit the data points is the residual norm $r := \|Ac - d\|_2$. Determine r.

2. Consider the following least squares approach for ranking sports teams. Suppose we have four college football teams, called simply T1, T2, T3, and T4. These four teams play each other with the following outcomes:

   - T1 beats T2 by 4 points: 21 to 17.
   - T3 beats T1 by 9 points: 27 to 18.
   - T1 beats T4 by 6 points: 16 to 10.
   - T3 beats T4 by 3 points: 10 to 7.
   - T2 beats T4 by 7 points: 17 to 10.

   To determine ranking points r_1, r_2, r_3, r_4 for each team, we do a least squares fit to the overdetermined linear system:

   $$r_1 - r_2 = 4, r_3 - r_1 = 9, r_1 - r_4 = 6, r_3 - r_4 = 3, r_2 - r_4 = 7.$$

   This system does not have a unique least squares solution, however, since if $[r_1, r_2, r_3, r_4]^\top$ is one solution and we add to it any constant vector then we obtain another vector for which the residual is exactly the same.

   To make the solution unique, we can fix the total number of ranking points, say, at 20. To do this, we add the equation $r_1 + r_2 + r_3 + r_4 = 20$ to those listed above.

   Note that this equation will be satisfied exactly since it will not affect how well the other equalities can be approximated. Determine the values r_1, r_2, r_3, r_4 that most closely satisfy these equations, and based on your results, rank the four teams.

3. Find the polynomial of degree 10 that best fits the function $f(t) = 1/(1 + 25t^2)$ at $30, 50, 100$ equally-spaced points t between -1 and 1. Set up the matrix A and right-hand side vector b, and determine the polynomial coefficients in two different ways:

   (a) By using the MATLAB command `x = A \ b` (which uses a QR decomposition).

   (b) By solving the normal equations $A^\top Ax = A^\top b$. This can be done in MATLAB by typing

   ```
   x = (A' * A) \ (A' * b).
   ```

   Plot the data points, three polynomials and the function $f(t)$ in a single plot. Compute the residual norm in each case and comment on the results. Do you observe Runge's phenomenon?

   [Note: You can compute the condition number of A or of $A^\top A$ using the MATLAB function `cond`.]

4. Determine the polynomial of degree 19 that best fits the function $f(t) = \sin(\frac{\pi}{5}t) + \frac{t}{5}$ for $t_1 = -5, t_2 = -4.5, \ldots, t_{23} = 6$. Setup the LSP $Ax = b$ and determine the polynomial p in three different ways:

   (a) By using the matlab command
   `>> A \ b`
   This uses QR factorization to solve the LSP $Ax = b$. Call this polynomial p_1.

   (b) By solving the normal equation $A^* Ax = A^* b$. Use `x = (A'*A)\(A'*b).` Call this polynomial p_2.

   (c) By solving the system $\begin{bmatrix} I_m & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} -r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$. Call this polynomial p_3.

Compute the condition number (use the matlab command `cond(A)`) of the coefficient matrix associated with each of the systems that you are solving. If $\mathrm{cond}(A) = 10^t$ then we can expect the solution to be have $16 - t$ correct digits and hence lose t digits of accuracy.

Print the result to 16 digits (use `format long e`). Which one is the most ill conditioned?

The norm of the residual $\|r\|_2 = \sqrt{\Sigma_{i=1}^{23} |p_j(t_i) - f(t_i)|^2}\, j = 1, 2, 3$ for each of these methods gives an idea of the goodness of the fit in each case. Compute these norms (in `format long e`). You may use the `polyval` command to evaluate the polynomials p_1, p_2 and p_3 at the points $t_i, i = 1 : 23$. However you must flip the vectors p_1, p_2 and p_3 upside down by using the `flipud` command before this. Type `help polyval` and `help flipud` for details.

Finally, plot the polynomials p_1, p_2, p_3 and the function f on $[-5, 6]$. Use different colors to distinguish these plots. Do you observe any difference? If yes, which polynomial is a better approximation of f?

5. **Analysis of "Filip" data set from NIST:** The filip data set consists of several dozen observations of a variable y at different x. Your task is to model y by a polynomial $p(x)$ of degree 10. You will find the filip dataset at the following URL:

   `http://www.itl.nist.gov/div898/strd/lls/data/Filip.shtml`

   This dataset is controversial because the NIST certified polynomial cannot be reproduced by many algorithms. You task is to report what MATLAB does with it.

   (a) Your first task is to download the data from the above website. Next, extract the value of x and y and load the data into MATLAB. Plot y versus x (plot it with '.') and then invoke Basic Fitting tool available under the Tools menu on the figure window. Select the 10th degree polynomial fit. (Ignore the warning that MATLAB may give.) From the Tools menu compute the coefficients of the polynomial fit. How do the coefficients compare with the certified values on NIST web page? How does the plotted fit compare with the graphic on the NIST Web page? The basic fit tools also displays the norm of the residuals $\|r\|$. Compare this with the NIST quantity "Residual Standard Deviation", which is $\frac{\|r\|}{\sqrt{n-p}}$. Here p is the degree of the polynomial and n is the number of data.

   (b) Next, examine the dataset by using the following methods to compute the polynomial fit. Explain all the warning messages you received during these computations.

   * MATLAB Backslash command.
   * Pseudoinverse (`pinv`).
   * Normal equation (theoretically the LSP is of full rank).
   * Certified coefficients: Obtain the coefficients from the NIST Web page.

   Prepare a table giving coefficients of the polynomial fit and the norm of the residuals obtained by each method. Plot the polynomial fits. Use dots, '.' at the data values and plot the curves $y = p(x)$ by evaluating $p(x)$ at a few hundred points over the range of the x's. Some plots may not be visibly distinct. Which methods produce which plots? Generate similar plots as given in the NIST web page.

*** End ***