Lab Session 6

MA423 : Matrix Computations July-November 2021 S. Bora

---

1. Given a vector $x \in \mathbb{R}^n$, write a MATLAB function program $[\mathtt{u}, \gamma, \tau] = \mathtt{reflect(x)}$ to compute $u \in \mathbb{R}^n$, and $\gamma \in \mathbb{R}$ so that $Qx = [-\tau, 0, \ldots, 0]^T$ where $\tau = \pm\|x\|_2$. Ensure that you choose the sign of τ as explained in Lecture 25 so as to avoid catastrophic cancellation.

2. Write another function program $\mathtt{B} = \mathtt{applreflect(u, gamma, A)}$ to *efficiently* perform the multiplication QA (as explained in Lecture 25) where $Q = I - \gamma uu^T$.

3. Use the programs written above to write another function program $[\mathtt{Q}, \mathtt{R}] = \mathtt{reflectqr(A)}$ that computes the condensed QR decomposition of $A \in \mathbb{R}^{n\times m}, n \geq m$, via reflectors.

   The program should have all the features explained in Lectures 25 and 26. In particular, the zeros created at each step are to be overwritten by the vectors u (apart from the leading 1 entry) required to construct the reflector used at that stage and the values of γ corresponding to each reflector are to be stored as a separate vector. Further, the Q should be assembled column-by-column as explained in Lecture 26.

4. **Test your output** $[\mathtt{Q}, \mathtt{R}] = \mathtt{reflectqr(A)}$ **for various different randomly generated matrices** A **by running** $[\mathtt{Qhat}, \mathtt{Rhat}] = \mathtt{qr(A, 0)}$ **and checking if** $\mathtt{norm(Q * R - A)}, \mathtt{norm(Q' * Q - eye(m))}$, $\mathtt{norm(tril(R, -1))}$, $\mathtt{norm(R - Rhat)}$, **and** $\mathtt{norm(Q - Qhat)}$ **are all** $\approx \mathtt{u}$ **.**
   (If you have chosen the sign of τ correctly in the `reflect.m` program, then $\mathtt{norm(R-Rhat)}$ and $\mathtt{norm(Q - Qhat)}$ will be $\approx u$.)

5. The purpose of this exercise is to solve the Least-Squares Problem (in short, LSP) $Ax = b$ by different methods and compare the solutions. Here $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$, and usually n is much bigger than m.

   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 $p \in \operatorname{span}(\phi_1, \ldots, \phi_n)$, we have $p = x_1\phi_1 + \cdots + x_n\phi_n$ for some $x_j \in \mathbb{R}$. 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 $x \in \mathbb{R}^n$ for which the 2-norm of the residual vector $\|r\|_2$, is minimized. We write this as LSP $Ax = b$.

Your task is to find the polynomial of degree 17 that best fits the function $f(t) = \sin(\frac{\pi}{5}t) + \frac{t}{5}$ for $t_1 = -5, t_2 = -4.5, \ldots, t_{23} = 6$. Set up the LSP $Ax = b$ and determine the polynomial p whose coefficients are determined by x 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 setting up the normal equation $A^T Ax = A^T b$ and solving them for x. You will need to use the Cholesky method for this. Call this polynomial p_2.

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

Set the formatting to `format long e` and compute the condition number of the coefficient matrix associated with each of the systems that you are solving. Which one is the most ill conditioned?

Compute the $\|r\|_2$ which gives the value of $\sqrt{\Sigma_{i=1}^{23} |p_j(t_i) - f(t_i)|^2}\, j = 1, 2, 3$ for each of these methods (again in `format long e.`). This is a measure of the goodness of the fit in each case. Which of the methods provide the best fit?

Make a report of your experiments.