AMS213A Project 2

Sutirtha Sengupta (Student id#:1566652)

February 17, 2017

Problem 1

For a given square matrix A, the Choldec and Cholsol subroutines (in the module LinAl.f90) solves the normal equations given by

$$\tilde{A}x = \tilde{b} \tag{1}$$

where $\tilde{A} = A^T A$ and $\tilde{b} = A^T b$ for an unknown vector x.
The Fortran program myprog.f90 solves the least square problem using Cholesky decomposition (performed by the routine Choldec in LinAl.f90 module) of the corresponding Vandermond matrix A given by

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & & & & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^{n-1} \end{pmatrix} \tag{2}$$

by casting it into the standard linear form:

$$Ax = b \tag{3}$$

where b is a column vector containing the $\{y\}$-values corresponding to the $\{x\}$ values read from the atkinson.dat file (given as an argument to the program). The vector b is replaced by the solution for x on executing the program.
Figure 1 shows the fit to the data using with a 3^{rd} order polynomial given by

$$f(x) = 0.574662268 + 4.72584343x - 11.1281977x^2 + 7.66867352x^3 \tag{4}$$

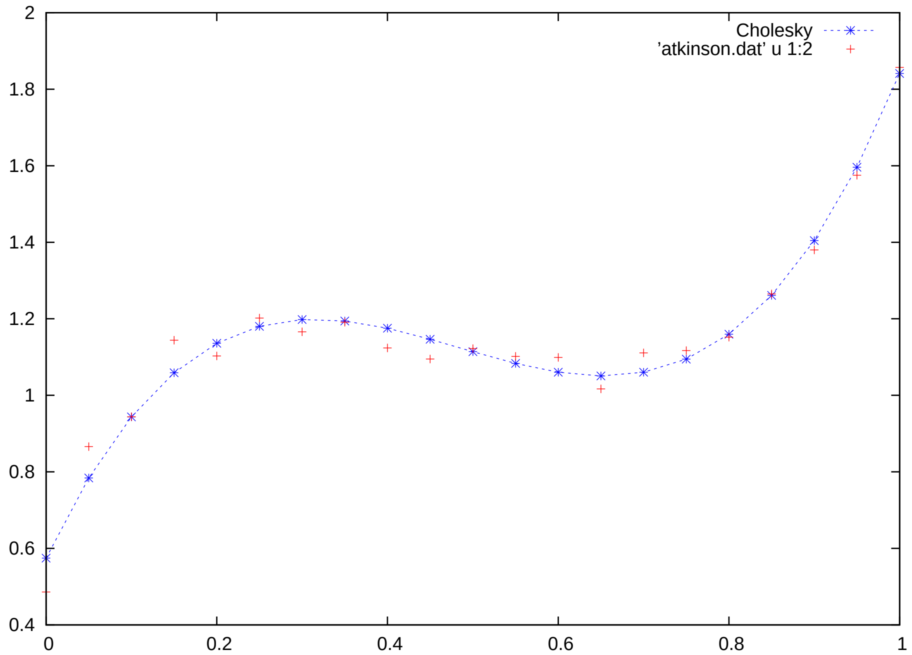

Figure 1: least square fit with 3^{rd}order polynomial (Eq. 4) to the data in atkinson.dat using Cholesky decomposition

with a rms error given by

$$E = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2} = 0.0420603156 \tag{5}$$

Figure 2 hows the fit to the data using with a 5^{th} order polynomial given by

$$f(x) = 0.511144936+7.14999056x-28.0456734x^2+51.0233231x^3-46.5010490x^4+17.7211895x^5 \tag{6}$$

with a rms error given by

$$E = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2} = 0.0306545403 \tag{7}$$

The algorithm breaks down for order of polynomial exceeding 5, since the algorithm give negative values for the diagonal entries (so the matrix is no longer positive-definite, and hence the decomposition doesn't work)

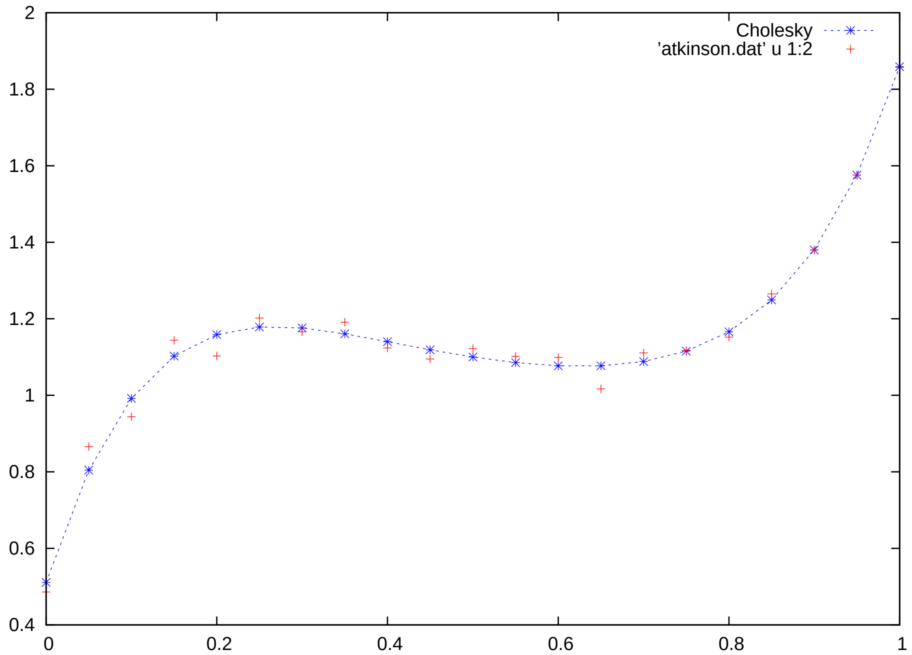

Figure 2: least square fit with 5^{th}order polynomial (Eq. 6) to the data in atkinson.dat using Cholesky decomposition

Problem 2

The QR decomposition turns a matrix A (in this case, non-square) into the form

$$A = QR \tag{8}$$

,where Q is an orthogonal matrix and R is an upper triangular one. The QRdec subroutine (in LinAl.f90 module) constructs the QR decomposition of the $m \times n$ matrix A and it returns the upper triangular matrix R in the upper triangle of A, while Q is returned separately as the product of n Householder matrices defined by

$$H_j = I_m - 2v_j v_j^T \tag{9}$$

with

$$
\begin{aligned}
v_j &= \frac{(0, ..., 0, a_{jj} + s_j, a_{j+1,j}, ..., a_{mj})^T}{||v_j||}, && (10) \\
(||v_j||^2 &= (a_{jj} + s_j)^2 + \sum_{k=j+1}^{m} a_{kj}^2 = 2s_j^2 + 2a_{jj}s_j, && (11) \\
s_j &= sign(a_{jj})(\sum_{k=j}^{m} a_{kj}^2)^{\frac{1}{2}} && (12)
\end{aligned}
$$

So,

$$
Q = H_n H_{n-1} .....H_2 H_1 \tag{13}
$$

To see how this procedure works, we operate H_j on the the j-th column of A denoted by a_j:

$$
H_j a_j = a_j - 2v_j(v_j^T a_j) \tag{14}
$$

$$
v_j^T a_j = \frac{1}{||v_j||}(0, ..., 0, a_{jj} + s_j, a_{j+1,j}, ..., a_{mj}) \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{jj} \\ a_{j+1,j} \\ \vdots \\ a_{m,j} \end{pmatrix} \tag{15}
$$

$$
= \frac{1}{||v_j||}(a_{jj}s_j + \sum_{k=j}^{m} a_{kj}^2) \tag{16}
$$

Using this in Eq.14, we have

$$
H_j a_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{jj} \\ a_{j+1,j} \\ \vdots \\ a_{m,j} \end{pmatrix} - \frac{2s_j(a_{jj} + s_j)}{||v_j||^2} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ a_{jj} + s_j \\ a_{j+1,j} \\ \vdots \\ a_{m,j} \end{pmatrix} = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ -s_j \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{17}
$$

$$
\text{(using Eq. 11)} \tag{18}
$$

Thus successive application of the Householder matrices progressively zero out the sub-diagonal elements in each column of A thereby reducing A into upper triangular form. The Fortran program myprog.f90 solves the least square problem using Householder-based QR decomposition (performed by the routine QRdec in LinAl.f90 module) of the corresponding Vandermond matrix A given by Eq. 2 by casting it into the standard linear form:

$$Ax = b \tag{19}$$

where b is a column vector containing the $\{y\}$-values corresponding to the $\{x\}$ vector read from the atkinson.dat file (given as an argument to the program). The *backsub* routine actually solves the system 19 by using the outputs from QRdec routine. First, the code creates $Q^T b$ in order to solve the system

$$Rx = Q^T b \tag{20}$$

By finding $Q^T b$ and solving the system 20 is is the same like solving 19, since

$$Ax = QRx = b \Rightarrow Rx = Q^{-1} b \Rightarrow Rx = Q^T b \tag{21}$$

using the orthogonality of Q ($Q^T = Q^{-1}$). So, in the backsub routine, $Q^T b$ is returned as an output in initial vector b which now is a $(m \times 1)$ vector. Finally, vector b is overwritten by the solution x and so the new b is returned as the final output of the routine.
Figure 3 shows the fit to the data using with a 3^{rd} order polynomial given by

$$f(x) = 0.574658096 + 4.72586584x - 11.1282244x^2 + 7.66868114x^3 \tag{22}$$

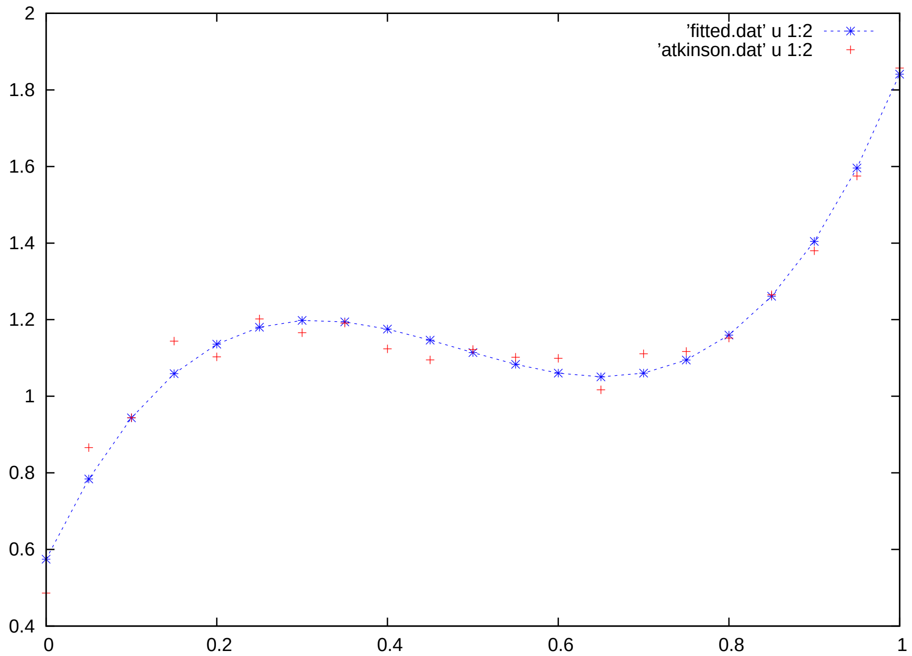

Figure 3: least square fit with 3^{rd}order polynomial (Eq. 22) to the data in atkinson.dat using Householder-based QR decomposition

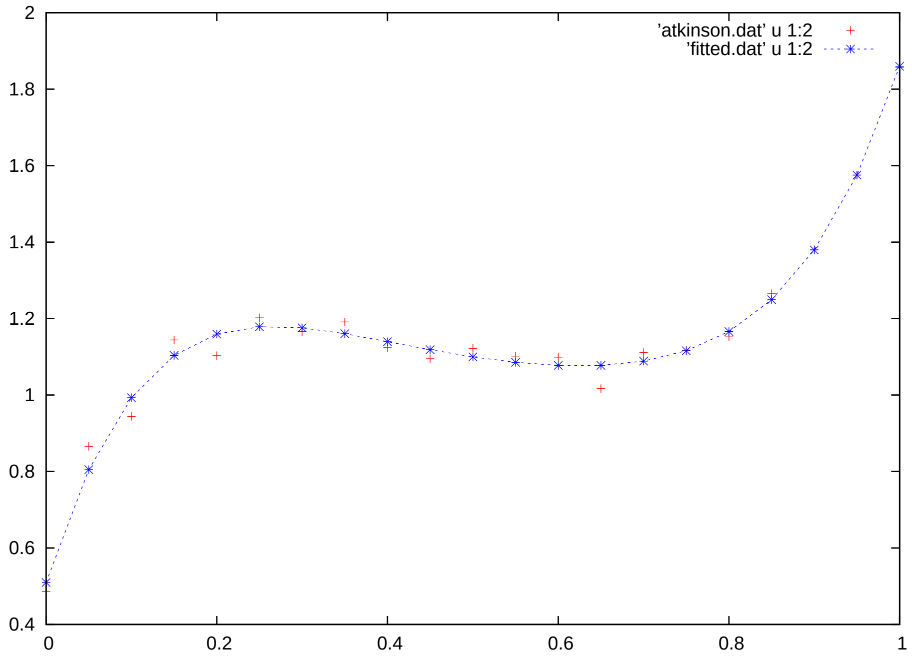

Figure 4: least square fit with 5^{th}order polynomial (Eq. 24) to the data in atkinson.dat using Householder-based QR decomposition

with a rms error given by

$$E = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2} = 0.0420603491 \tag{23}$$

which is greater than the corresponding rms error in the fit obtained using the Cholesky decomposition method in Problem 1.
Figure 4 hows the fit to the data using with a 5^{th} order polynomial given by

$$f(x) = 0.509619713 + 7.20328665x - 28.408123x^2 + 51.9441299x^3 - 47.4873962x^4 + 18.0979404x^5 \tag{24}$$

with a rms error given by

$$E = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2} = 0.0306485258 \tag{25}$$

which is smaller than the corresponding rms error in the fit obtained using the Cholesky decomposition method in Problem 1.
The algorithm gives poor fits (in terms of rms error) for order of polynomial exceeding 13. However, it continues to work (till order of polynomial 20) though the rms error becomes larger with increasing order, thereby suggesting poorer quality of fitting the given data.
The Frobenius norms of $A - QR$ and $Q^TQ - I$ are given by

$$||A - QR||_F = 2.05531614e - 06 \tag{26}$$

$$||Q^TQ - I||_F = 2.03146305e - 06 \tag{27}$$

which are both order of machine accuracy, suggesting that $A \backsimeq QR$ and $Q^TQ \simeq I$ within order of machine accuracy showing that the QR decomposition was indeed successful.