Assignment 1

Heine Olsson Aabø

September 2018

1 Introduction and Theory

Using the normal equations $A^T A\hat{x} = A^T\hat{b}$ is a common method to solve the least square problem, where we want to find the $\hat{x}$ that minimize $||A\hat{x}-\hat{b}||_2^2$. One trivial way to solve this system is to invert $A^T A$, however this can lead to numerical problems when it is singular of close to singular. Instead we can examine two other methods; using Cholesky decomposition and QR factorization.

1.1 Cholesky decompostition

The Cholesky decomposition is on the form $A = LDL^T$, where L is lower triangular and D is diagonal. For a real positive-definite matrix A, there is a decomposition $A = RR^T$, where $R = LD^{\frac{1}{2}}$. For the normal equations we set $B = A^T A = RR^T$. If we set $R\hat{w} = \hat{y}$, where $\hat{y} = A^T\hat{b}$, and $R^T\hat{x} = \hat{w}$ this can be solved using forward and backward substitution since R is upper triangular.

1.2 QR factorization

QR factorization gives $A = QR$, where Q is orthogonal, R is upper triangular and A is a nxm matrix ($n > m$). It turns out the least square problem can be solved for $R_1\hat{x} = \hat{c_1}$, where R_1 is a the mxm elements of $R = [R_1, 0]^T$ and $\hat{c_1}$ is the mxm elements of $Q^T\hat{b} = [\hat{c}_1, \hat{c}_2]^T$. Which can be solved using back substitution.

2 Implementation

I wrote my own code using Python for back substitution, forward substitution and the cholesky decomposition, where also the $D^{\frac{1}{2}}$ was computed. The QR factorization was computed using a premade function `numpy.linalg.qr` with an optional "reduced mode" that only computes the mxm elements of Q and R.

The normal equations was solved using both methods, where

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} \end{bmatrix} \tag{1}$$

Where m is the degree of the polynomial fit. For this assignment the normal equations was solved for $m = 3$ and $m = 8$. The data to fit is given by Eq. (2) and (3) with noise r.

$$y = xcos(r + \frac{x^3}{2}) + xsin(\frac{x^3}{2}) \tag{2}$$

$$y = 4x^5 - 5x^4 - 20x^3 + 10x^2 + 40x + 10 + r \tag{3}$$

3 Results

All plots are shown below; titled QR and Cholesky to reference which method, and Eq. (2) or (3) to reference the data set fitted.

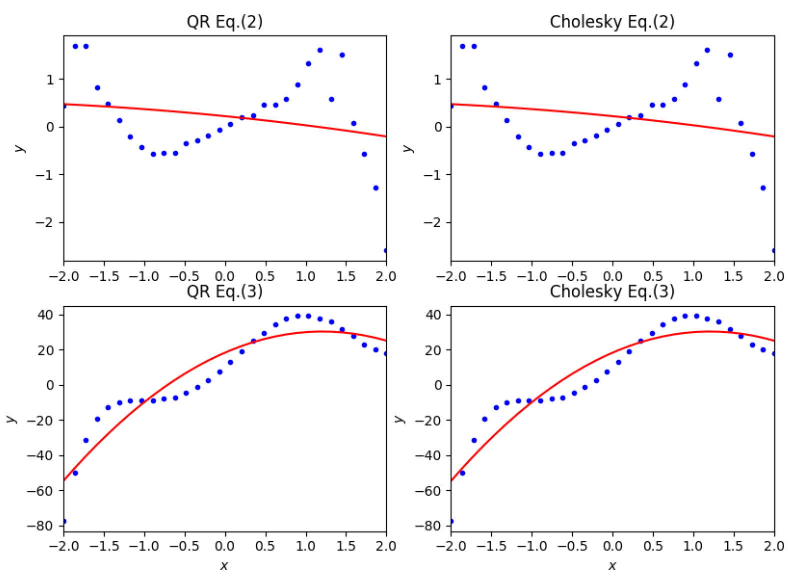
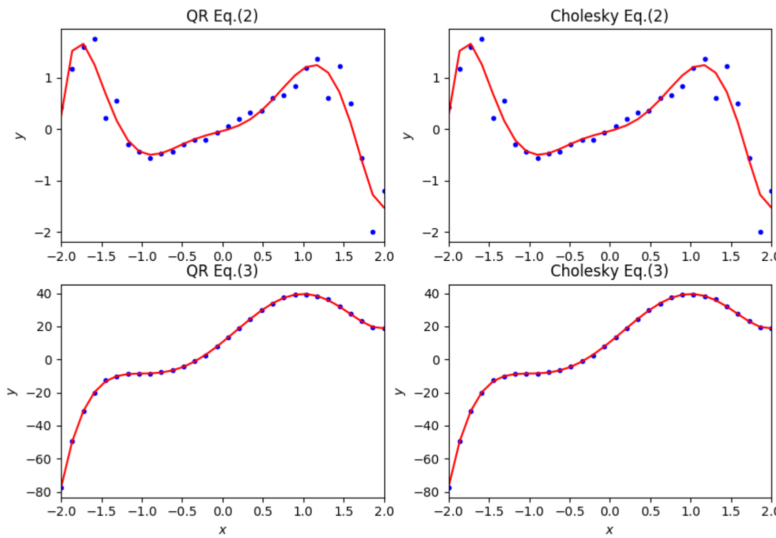


Figure 1: Polynomial fit of degree $m = 3$



Polynomial fit of degree $m = 8$

Both methods got the same mean square error (forward error) as each other for $m = 3$ and for $m = 8$ with the difference being in the order of 10^{-17}. This was found using Eq. (4)

$$MSE(\hat{y}, \hat{\tilde{y}}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \tilde{y}_i)^2 \tag{4}$$

I did not find it necessary to report the forward errors here, the only thing of importance is that they are equal.

4 Conclusion

The numerical efficiency is examined by calculating the total number of flops for each method. For the QR method:
- QR factorization with Gram-Schmidt gives ~2mn^2 flops.
- $Q^T \hat{b} = \hat{c}$ gives $2mn$ flops.
- $R_1 \hat{x} = \hat{c}_1$ gives n^2 flops.

Which gives a total of $2mn^2 + n^2 + 2mn$ flops for the QR method. For the cholesky method:
- $B = A^T A$ gives ~mn^2 flops.
- Cholesky decomposition gives $n^3/3$ flops.
- $\hat{y} = A^T \hat{b}$ gives $2mn$ flops.
- $R\hat{w} = \hat{y}$ gives n^2 flops.

- $R^T\hat{x} = \hat{w}$ gives n^2 flops.

Which gives a total of $\frac{n^3}{3}+mn^2+2n^2+2mn$ flops for the Cholesky method. This indicates that the Cholesky method is more efficient for large matrices. However it contains the operation to find the matrix A^TA which has a condition number $k(A^TA) = [k(A)]^2$. The QR method does not involve this operation, and in addition for an orhogonal matrix the condition number is equal to 1, so that multplications with other vectors or matrices does not increase the condition number. In other words with QR factorization the squaring of $k(A)$ is avoided which is very good for the numerical stability.

This gives an indication that the QR method is more stable, while the Cholesky method is more efficient.