HW3 Report

Jensen Davies

February 2021

Notes to Ian:

My theory problems are submitted separately as a handwritten PDF. My proof for problem 10) is not rigorous, and I didn't know how to prove a direction of 4a). Dongwook said he would update problem 10) and also email us a way to do the 4a) backwards implication after office hours on 2/18, but neither of these things happened, sadly.

I added the dimensions of the atkinson data to the top of the atkinson.dat file. This way, I could use the readMat subroutine from the previous LinAl.f90 file. If you wish to change the degree of the interpolant between 3, 5, or any other number that works, simply change the value of p on lines 32 and 83 of my submitted Driver_LinAl.f90.

Problem 1: Cholesky solution of the least squares problem

First, we explain in reasonable detail how the Cholesky decomposition and backsubstituion work. The Cholesky decomposition of matrix of symmetric positive definite matrix A is given by

$$A = U^T U = LL^T,$$

where U is upper-triangular and L is lower triangular such that $u_{jj} = l_{jj}$. The construction of the matrix L is done recursively, since we can write out the component form of $A = LL^T$ as

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik} l_{jk}.$$

By examining this summation, one finds that the diagonal entries of L are defined as

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk} l_{jk}},$$

and the entries below the columns as

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}.$$

Once we have decomposed A into the product LL^T by using the above expressions for the entries of L, we can rewrite the linear system $Ax = b$ as

$$LL^T x = b.$$

Hence, we solve this system just like we solve $LUx = b$. First, we let $L^T x = y$, and perform forward-substitution on $Ly = b$. This results in unknown vector y, allowing us to then utilize back-substitution to solve $L^T x = y$, ultimately finding our desired vector x.

The algorithm to perform the Cholesky decomposition presented in LinAl.f90 (submitted alongside this PDF) follows the pseudocode as presented in Chapter 2 lecture notes. Alongside LinAl.f90 is a driver program, Driver_LinAl.f90. For the Cholesky decomposition portion of the driver, the following list has been completed and will be summarized further below (if needed):

- Created the matrices associated with the normal equations, i.e. $A^T A$ and $A^T b$.

In this case, the matrix A is a Vandermonde matrix.

- We printed the solution vector x.
- Computed the fitted curve and prints the data to a file.
- Computed the 2-norm error between the fitted curve and the data
- Provided figures of the 3rd and 5th degree polynomial interpolants.

The atkinson.dat file is read and stored into a 21×2 matrix. The first column of this data is used to construct our Vandermonde matrix A, while the second column serves as our vector b in the normal equation $A^T Ax = A^T b$. We then construct the Gram matrix, $A^T A$, and also the vector $A^T b$. We do this for single precision, as requested. Using the Cholesky decomposition and backsubstitution described above, we solve the normal equation for x, giving us (with truncated values from single precision results)

$$x_{(3)} = \begin{bmatrix} 0.5746 \\ 4.7258 \\ -11.1282 \\ 7.6686 \end{bmatrix}.$$

This solution vector $x_{(3)}$ contains the coefficients on our 3rd degree polynomial interpolant for the pairs of data (x_i, y_i), $i = 1, \dots 21$ provided from atkinson.dat. When we set the degree of our polynomial interpolant to be of degree

5, we have coefficients (with truncated values from single precision results)

$$x_{(5)} = \begin{bmatrix} 0.5111 \\ 7.1499 \\ -28.0456 \\ 51.0233 \\ -46.5010 \\ 17.7211 \end{bmatrix}.$$

The 3rd and 5th fitted curve values, which are contained in the vector $q_{(3)} = Ax_{(3)}$, and $q_{(5)} = Ax_{(5)}$, are

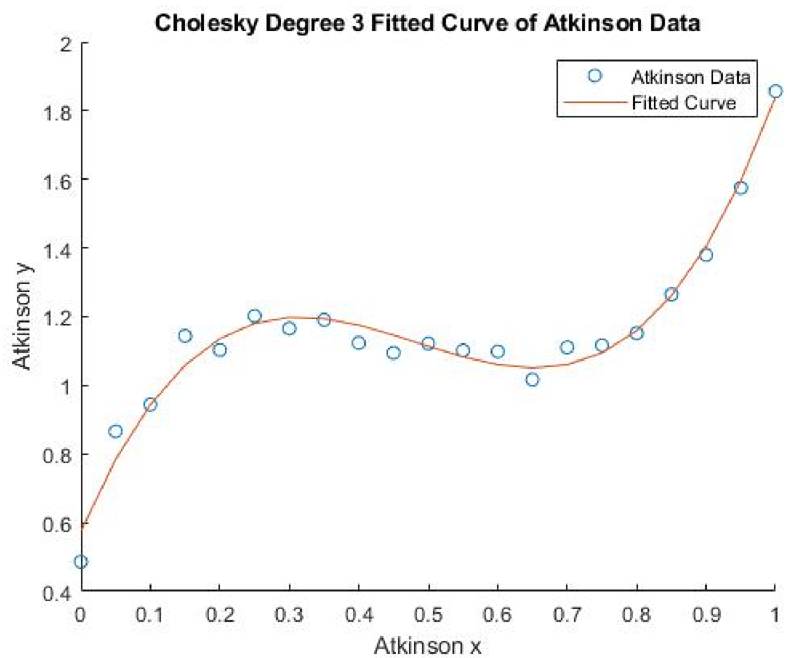
$$q_{(3)} = \begin{bmatrix} 0.5746 \\ 0.7840 \\ 0.9436 \\ 1.0590 \\ 1.1360 \\ 1.1804 \\ 1.1979 \\ 1.1942 \\ 1.1752 \\ 1.1466 \\ 1.1141 \\ 1.0834 \\ 1.0604 \\ 1.0508 \\ 1.0602 \\ 1.0946 \\ 1.1596 \\ 1.2610 \\ 1.4045 \\ 1.5959 \\ 1.8409 \end{bmatrix}, \qquad q_{(5)} = \begin{bmatrix} 0.5096 \\ 0.8049 \\ 0.9932 \\ 1.1035 \\ 1.1593 \\ 1.1787 \\ 1.1756 \\ 1.1603 \\ 1.1397 \\ 1.1185 \\ 1.0998 \\ 1.0856 \\ 1.0775 \\ 1.0775 \\ 1.0887 \\ 1.1158 \\ 1.1659 \\ 1.2491 \\ 1.3794 \\ 1.5750 \\ 1.8594 \end{bmatrix}$$

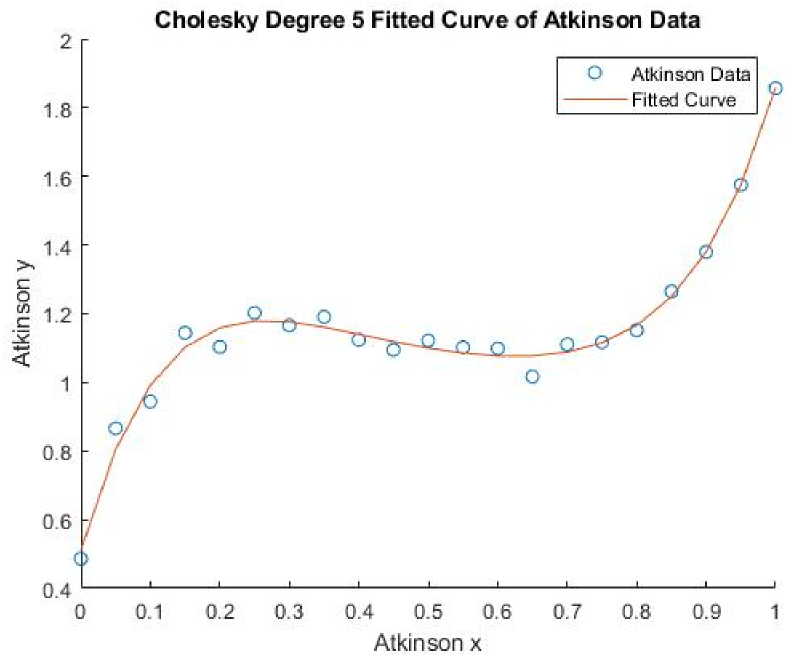
The 2-norm errors between the fitted curves and the data are

$$\|y - q_{(3)}\|_2 = 0.19274458, \text{ and } \quad \|y - q_{(5)}\|_2 = 0.14047675,$$

where y is the second column of the provided atkinson data.

Below are figures labeled for both cases of when the degree of the interpolant is 3, and 5. The 5th degree case looks extremely similar to the 3rd degree, but for $atkinson\_x \in [0.3, 0.6]$ it is easy to tell that this fits the data a tad more snugly. Theoretically, a polynomial of degree 20 is the largest degree polynomial we could use for this set of data, since the Vandermonde matrix A is of dimension would be 21×21. This algorithm fails when the matrix we try to perform the Cholesky decomposition on is either singular, or not positive definite. In the single precision case, it occurs when the degree of the desired polynomial interpolant exceeds 5. This is because of how our atkinson.dat data is constructed–the first three entries (of the second column) are values are less than one, meaning as they are raised to higher and higher powers, they approach zero rapidly. Thus, naturally, the Vandermonde matrix becomes singular if we wish to interpolate with too high of a degree. It becomes singular since the pivots that previously existed at a lower degree were numerically nonzero, but as the degree is raised, they become numerically zero.





.

Problem 2: QR solution of the least squares problem

First, we will explain in reasonable detail how the Householder QR decomposition works. Performing the QR factorization of the matrix A via Householder matrices is also known as the orthogonal triangularization of A. This is because A is slowly transformed into the upper triangular matrix R through the repeated multiplication of orthogonal matrices, giving us

$$Q_n Q_{n-1} \dots Q_1 A = R.$$

Since the product of orthogonal matrices is also orthogonal, we can then say $Q_n Q_{n-1} \dots Q_1 = Q^T$, recovering the decomposition

$$A = QR.$$

These orthogonal matrices are Householder matrices, which are described by

$$H_j = I - 2v_j v_j^T, \ \ 1 \leq j \leq n,$$

where

$$v_j = \frac{1}{\|v_j\|}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{jj}+s_j \\ a_{j+1,j} \\ \vdots \\ a_{mj} \end{bmatrix}, \qquad s_j = \text{sign}(a_{jj})\Big(\sum_{k=j}^{m} a_{kj}^2\Big).$$

For the QR decomposition portion of the driver, the following list has been completed and will be summarized further below (if needed):

- Performed the QR decomposition of A via Householder matrices.
- Computed $A - QR$ and printed it to the screen.
- Computed the Frobenius norm of $A - QR$ and printed it to the screen.
- Computed $Q^TQ - I$ and printed it to the screen.
- Computed the Frobenius norm of $Q^TQ - I$and printed it to the screen.
- Solved the least squares equations projected onto the span of A, namely $Ax = P_Ab$ as in the notes.
- Printed the solution vector x.

The matrices $A - QR$ and $Q^TQ - I$ are omitted from the report simply due to size. However, the Frobenius norms of $A - QR$ and $Q^TQ - I$ are

$$\|A - QR\|_F = 0.00000021677, \qquad \|Q^TQ - I\|_F = 0.0000001235,$$

which, in single precision, agrees with what we'd expect analytically.

As requested, we solve the system $Rx = Q^Tb$ by applying the Householder QR decomposition to the Gram matrix A^TA, (where again, A is the Vandermonde matrix) and then applying back-substitution. For a 3rd degree interpolant, our solution x is

$$x_{(3)} = \begin{bmatrix} 0.5746 \\ 4.7258 \\ -11.1282 \\ 7.6686 \end{bmatrix}.$$

For a 5th degree interpolant, our solution x is

$$x_{(5)} = \begin{bmatrix} 0.5096 \\ 7.2033 \\ -28.4084 \\ 51.9449 \\ -47.4882 \\ 18.0982 \end{bmatrix}.$$

Notice that the result for the 5th degree interpolant via QR decomposition starts to vary significantly from the 5th degree interpolant via the Cholesky decomposition. As mentioned above, as the degree of the interpolant increases, the Vandermonde matrix becomes more and more ill-conditioned. Thus, the Gram matrix A^TA is terribly conditioned for higher degrees, since (in the theory section of the homework) we have showed that

$$\mathrm{cond}(A^TA) = (\mathrm{cond}(A))^2.$$

Hence, since the Cholesky decomposition requires a nonsingular/positive definite matrix, it loses numerical accuracy when decomposing a matrix with a large condition number. The Householder QR decomposition, on the other hand, doesn't care whether the matrix it is decomposing is singular or not positive definite. The only reason I can see for this algorithm to fail is when we try to interpolate our set of data with a degree polynomial greater than or equal to the number of points available (or else our Vandermonde matrix isn't defined for the system). In this case, the maximum degree is 20.

The fitted curve values, $q_{(3)} = Ax_{(3)}$ and $q_{(5)} = Ax_{(5)}$, are displayed below.

$$q_{(3)} = \begin{bmatrix} 0.5746 \\ 0.7840 \\ 0.9436 \\ 1.0590 \\ 1.1360 \\ 1.1804 \\ 1.1979 \\ 1.1942 \\ 1.1752 \\ 1.1466 \\ 1.1141 \\ 1.0834 \\ 1.0604 \\ 1.0508 \\ 1.0602 \\ 1.0946 \\ 1.1596 \\ 1.2610 \\ 1.4045 \\ 1.5959 \\ 1.8409 \end{bmatrix}, \qquad {}_{(5)} = \begin{bmatrix} 0.5096 \\ 0.8049 \\ 0.9932 \\ 1.1035 \\ 1.1593 \\ 1.1787 \\ 1.1756 \\ 1.1603 \\ 1.1397 \\ 1.1185 \\ 1.0998 \\ 1.0856 \\ 1.0775 \\ 1.0775 \\ 1.0887 \\ 1.1158 \\ 1.1659 \\ 1.2491 \\ 1.3794 \\ 1.5750 \\ 1.8594 \end{bmatrix}.$$

The 2-norm errors for $q_{(3)}$ and $q_{(5)}$ are

$$\|y - q_{(3)}\|_2 = 0.19274467, \text{ and } \quad \|y - q_{(5)}\|_2 = 0.14044936.$$

Below are two figures, the first being the QR solution for degree 3, and the second being the QR solution for degree 5. The difference between either method in this case is visually indistinguishable, although in both cases the 5th degree interpolant clearly models the atkinson.dat data in a better fashion.