Polynomial fitting using QR and Cholesky factorization

Live Ljungqvist Storborg
(Dated: September 24, 2025)

This report compares QR factorization and Cholesky decomposition for polynomial least squares fitting on trigonometric and polynomial datasets with noise. Both methods produce equivalent results for low degrees, though QR factorization offers superior numerical stability for higher-degree polynomials.

I. INTRODUCTION

In this assignment, we approximate the two datasets

$$y = x\left(\cos\frac{x^3}{2} + \sin\frac{x^3}{2}\right) \tag{1}$$

and

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

with polynomial curves using the least squares method. We seek to fit a polynomial of degree smaller than m represented as:

$$p(x) = \sum_{j=1}^{m} c_j x^{j-1}$$

For polynomial fitting, our design matrix is given by the Vandermonde matrix:

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} \end{bmatrix} \tag{3}$$

where $x_1, x_2, \dots, x_n$ are the data points and n is the number of observations. This leads to the linear system $X\boldsymbol{\theta} = \mathbf{y}$, where $\boldsymbol{\theta}$ contains the polynomial coefficients and $\mathbf{y}$ is the target data.

We will implement and compare two methods for finding the optimal parameters: QR factorization and Cholesky factorization via normal equations.

II. METHOD & THEORY

When considering an overdetermined linear system $X\boldsymbol{\theta} = \mathbf{y}$ where $X \in \mathbb{R}^{n\times m}$ with $n > m$, we cannot solve it exactly. Instead, we find the vector $\boldsymbol{\theta} \in \mathbb{R}^m$ that minimizes the sum of error squared:

$$\|X\boldsymbol{\theta} - \mathbf{y}\|_2^2 \tag{4}$$

This optimization problem has a unique global minimum because the cost function 4 is a convex function in $\boldsymbol{\theta}$. First, we expanding the cost function

$$\begin{aligned} F(x) = \|X\boldsymbol{\theta} - \mathbf{y}\|_2^2 &= (X\boldsymbol{\theta} - \mathbf{y})^T(X\boldsymbol{\theta} - \mathbf{y}) \\ &= \boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\mathbf{y} + \mathbf{y}^T\mathbf{y} \end{aligned}$$

This is followed by taking the derivative of the cost function with respect to $\boldsymbol{\theta}$

$$\nabla_\theta F = 2X^T X\boldsymbol{\theta} - 2X^T\mathbf{y} \tag{5}$$

At last, to attain the minimum, we set the gradient 5 to zero

$$\nabla_\theta F = 0 \quad \Rightarrow \quad X^T X\boldsymbol{\theta} = X^T\mathbf{y}$$

There are two main approaches to solve this system: QR factorization and Cholesky factorization via normal equations.

A. QR Factorization

Any $n \times m$ matrix X with $n > m$ can be factorized as

$$X = QR$$

where $Q \in \mathbb{R}^{n\times n}$ is a real and orthogonal matrix with $Q^TQ = I$, and $R \in \mathbb{R}^{n\times m}$ is an upper triangular matrix.

By substituting $X = QR$ into the linear system, we get:

$$\begin{aligned} QR\boldsymbol{\theta} &= \mathbf{y} \\ Q^TQR\boldsymbol{\theta} &= Q^T\mathbf{y} \\ R\boldsymbol{\theta} &= Q^T\mathbf{y} \end{aligned}$$

By performing back substitution on $R\boldsymbol{\theta} = \tilde{\mathbf{y}}$ where $\tilde{\mathbf{y}} = Q^T\mathbf{y}$ represents the vector $\mathbf{y}$ rotated to align with the column space of X, we get the optimal parameters $\boldsymbol{\theta}$.

B. Cholesky Factorization

Cholesky factorization solves the linear system

$$X\boldsymbol{\theta} = \mathbf{y} \quad \rightarrow \quad X^T X\boldsymbol{\theta} = X^T\mathbf{y}$$

directly by forming the normal equations. The matrix $B = X^TX$ is symmetric and positive definite.

Since B is symmetric positive definite, it can be decomposed using Cholesky factorization:

$$B = RR^T$$

where R is an upper triangular matrix with positive diagonal elements.

Substituting this into the normal equations:

$$RR^T\boldsymbol{\theta} = X^T\mathbf{y}$$

This system is solved in two steps:

1. **Back substitution**: Solve $R^T\mathbf{z} = X^T\mathbf{y}$ for $\mathbf{z}$ (since R^T is lower triangular)

2. **Forward substitution**: Solve $R\boldsymbol{\theta} = \mathbf{z}$ for $\boldsymbol{\theta}$ (since R is upper triangular)

The Cholesky factorization can alternatively be written as $B = LL^T$ where L is lower triangular, and solved by using forward substitution followed by back substitution. Both formulations are equivalent with $L = R^T$.

C. Condition Numbers and Numerical Stability

The condition number of a matrix A measures how sensitive the linear system is to small perturbations in the input data.

$$K(A) = \|A\|\|A^{-1}\| \tag{6}$$

A large condition number indicates an ill-conditioned problem where small errors in input data can lead to large errors in the solution. This is particularly relevant for polynomial fitting, as Vandermonde (Equation 3) matrices become increasingly ill-conditioned with higher polynomial degrees.

The two approaches has distinctly different conditioning properties. While QR factorization works directly with the matrix X with conditioning number

$$K_{QR}(X) = \|X\|\|X^{-1}\|, \tag{7}$$

the Cholesky method operates on X^TX, with conditioning number

$$K_{Chol}(X^TX) = \|X^TX\|\|(X^TX)^{-1}\| = [K_{QR}(X)]^2 \tag{8}$$

This squared condition number makes the Cholesky approach less stable for ill-conditioned problems.

D. Tools

The code can be found on GitHub by following this link [1]. All code is written in Python.

III. RESULTS & DISCUSSION

Both QR factorization and Cholesky decomposition methods were implemented and used to fit polynomials of degree 2 and 7 to the two datasets described in Equations 1 and 2. The results are presented in Figures 1 through 4.

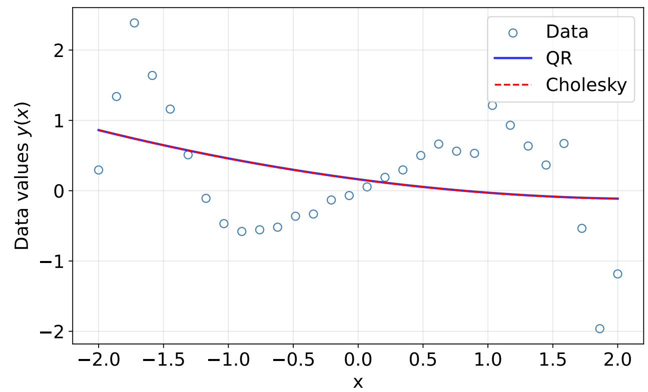

Figure 1. Degree 2 polynomial fitting for 1 using QR factorization (solid blue) and Cholesky decomposition (dashed red) with noise $\epsilon = 1$, fitted to $n = 30$ points over $x \in [-2, 2]$.

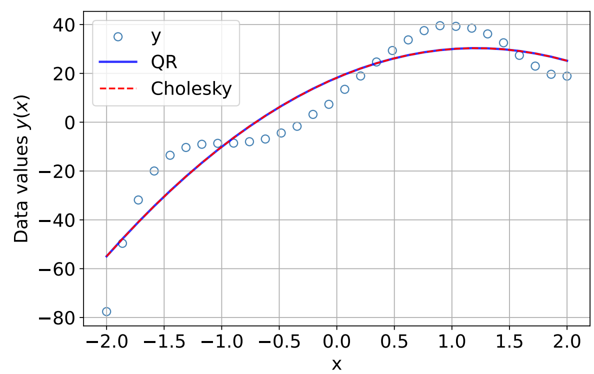

Figure 2. Degree 2 polynomial fitting for 2 using QR factorization (solid blue) and Cholesky decomposition (dashed red) with noise $\epsilon = 1$, fitted to $n = 30$ points over $x \in [-2, 2]$.

The results demonstrate that both QR factorization and Cholesky decomposition produce identical polynomial fits, as expected. However, the choice of polynomial degree significantly affects how well the model is able to capture the underlying pattern.

A. Method Comparison

All figures demonstrate that the polynomial approximations produced by QR factorization and Cholesky decomposition are identical for both datasets for both polynomial degrees. This confirms that both methods solve the same least squares optimization problem and yield numerically equivalent results. The perfect overlap of the fitted curves validates the correctness of both implementations and demonstrates that the choice between methods can be based on computational considerations,

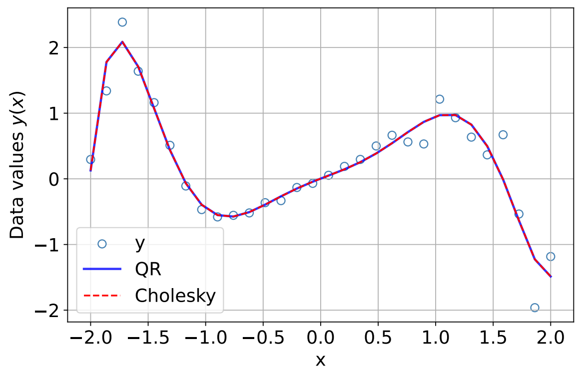

Figure 3. Degree 7 polynomial fitting for 1 using QR factorization (solid blue) and Cholesky decomposition (dashed red) with noise $\epsilon = 1$, fitted to $n = 30$ points over $x \in [-2, 2]$.

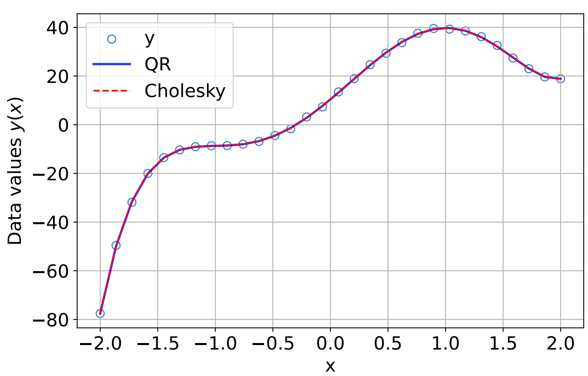

Figure 4. Degree 7 polynomial fitting for 2 using QR factorization (solid blue) and Cholesky decomposition (dashed red) with noise $\epsilon = 1$, fitted to $n = 30$ points over $x \in [-2, 2]$.

such as speed and memory usage, rather than accuracy concerns.

B. Impact of condition number

For $m = 3$ and $m = 8$, both methods compute reasonable results. Polynomial fitting is an ill-conditioned problem since the computation of the Vandermonde matrix (Equation 3) requires high precision for representing x^m when m becomes large. As the input data experiences roundoff errors when the number of features increases, the QR method is superior for stability.

The Cholesky method will produce solutions with increasing error, and the preferred method for complex models is QR factorization.

C. Effect of Polynomial Degree

The figures demonstrate polynomial fitting of degrees 2 and 7, showing that both QR and Cholesky methods achieve better approximation when the model has higher complexity. In Figures 1 and 2, the low-degree polynomial provides poor approximation of the target data due to insufficient model complexity. The fitted line is not very flexible to the data, and the model is not able to capture the underlying pattern.

In contrast, Figures 3 and 4 show significantly improved fits. The higher-degree models possess sufficient complexity to capture the underlying patterns in both datasets from Equations 1 and 2. A higher degree polynomial has more freedom as it has more degrees of freedom to help capture all datapoints from the target data.

D. Dataset-Specific Observations

The polynomial dataset (Equation 2) shows better approximations compared to the trigonometric function (Equation 1). This is expected since polynomial models naturally represent polynomial target functions more accurately than transcendental functions.

For the trigonometric dataset, while the degree 7 polynomial achieves reasonable approximation, neither QR nor Cholesky methods can fully capture all oscillatory details inherent in the trigonometric function, demonstrating the fundamental limitation of polynomial approximation for non-polynomial functions.

IV. CONCLUSION

In this project, I have used both QR factorization and Cholesky decomposition as methods to approximate the two datasets from Equations 1 and 2 with polynomial fitting of degrees 2 and 7. Analyzing the figures, there are no differences in accuracy between the two methods. A model with lower complexity (degree 2) performs poorer than the model with slightly higher complexity (degree 7) for both datasets. The results also show that polynomial fitting of an actual polynomial (Equation 2) provides a better fit than for trigonometric data (Equation 1). In the analysis of the results for $m = 3$ and $m = 8$, both methods are stable, but for larger m, the QR method is preferred for numerical stability.

[1] L. L. Storborg, Assignment 1, https://github.com/livelstorborg/Numerical-Analysis.

[2] A. C. Faul, *A Concise Introduction to Numerical Analysis*, CRC Press, Boca Raton, FL, 2016.