Assignment: Aerodynamic Efficiency Curve

Modelling in Aerospace Engineering

Assignment II

23^{rd} **of April, 2025**

Author:

Andrés Velázquez Vela

This report aims to explain the methods used to solve the systems and to ensure a proper understanding of the results.

Task 1: Least squares polynomial fitting

The Least Squares (LS) method fits a polynomial by minimizing the sum of squared errors between the data and the polynomial. For a 3rd-degree polynomial $E = c_0 + c_1 C_L + c_2 C_L^2 + c_3 C_L^3$, the coefficients $\mathbf{x} = [c_0, c_1, c_2, c_3]^T$ are found solving the normal equations $A^T A\mathbf{x} = A^T\mathbf{b}$, where A known as the Vandermonde matrix, is constructed from the C_L values, and $\mathbf{b}$ is the vector of efficiencies E.

$$A = \begin{bmatrix} 1 & Cl_1 & Cl_1^2 & Cl_1^3 \\ 1 & Cl_2 & Cl_2^2 & Cl_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & Cl_n & Cl_n^2 & Cl_n^3 \end{bmatrix}$$

In MATLAB, this system is easyly solved using the backslash operator (\).

```
    M = A' * A;          % 4 x 4
    b = A' * E;          % 4 x 1
    % Solve
    x = M \ b;     %output coeficients
```

Task 2: Error minimization polynomial fitting

The Gradient Descent (GD) method minimizes the error function:

$$\varepsilon(\mathbf{x}) = \sum_{i=1}^{81} \left(c_0 + c_1 C_{L_i} + c_2 C_{L_i}^2 + c_3 C_{L_i}^3 - E_i\right)^2$$

by iteratively updating the coefficients in the direction opposite to the gradient $\nabla\varepsilon(\mathbf{x}) = 2A^T(A\mathbf{x} - \mathbf{b})$. The linear shearch adjusts the step size α, starting from $\alpha_{\max} = 1$, with parameters $\beta = 0.5$, for the reduction factor, and $\sigma = 10^{-4}$, chosen to be the most optimal. The iteration stops when $\|\nabla\varepsilon(\mathbf{x})\| < 10^{-1}$ or after 10^6 iterations, using an initial guess $\mathbf{x}_0 = [1, 10, -10, 1]^T$.

Output and Results

Polynomial Fits

The fitted polynomials from both methods were plotted separately against the dataset. The first plot shows the LS fit overlaid on the experimental data, while the second plot displays the GD fit overlaid on the same dataset. Both fits closely follow the trend of the data, indicating successful polynomial approximations.

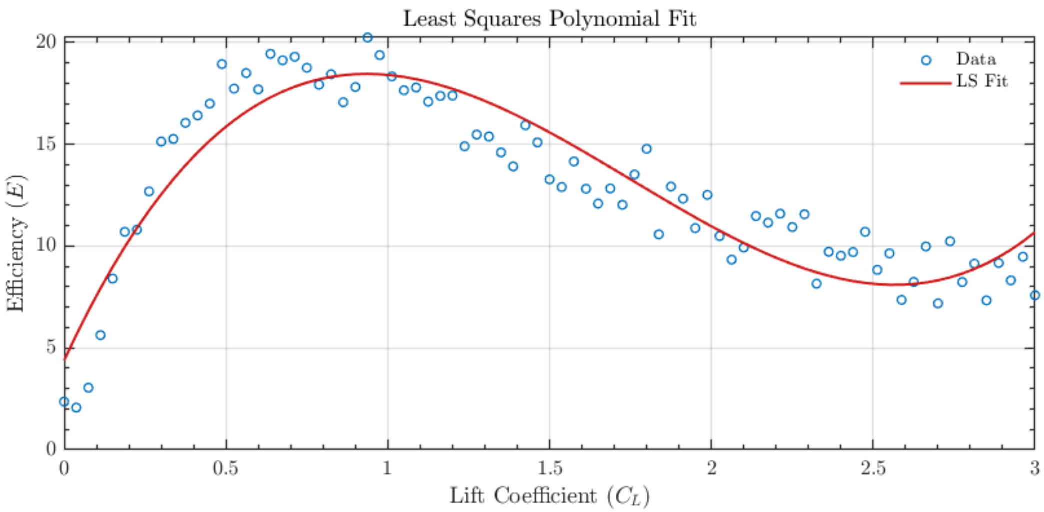

Figure 1: Least Squares.

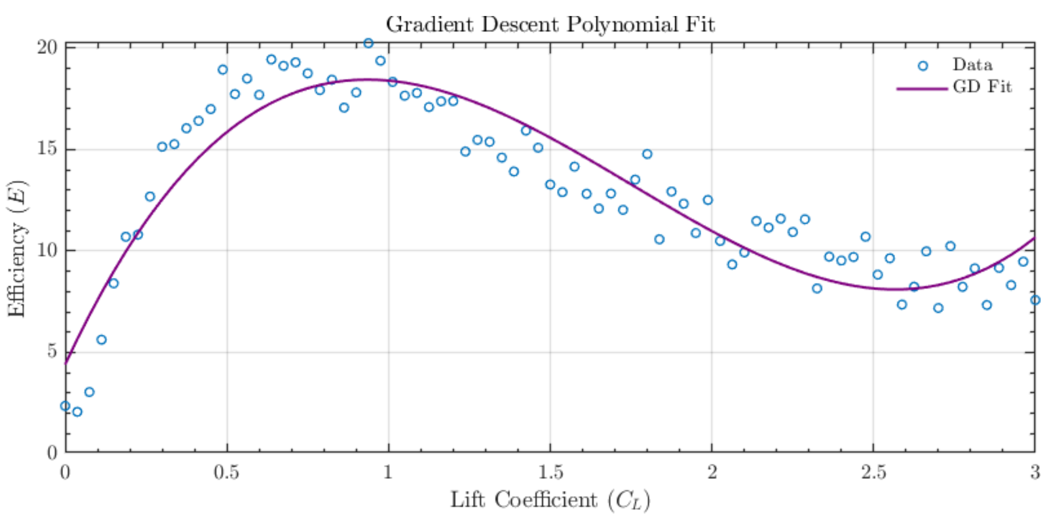

Figure 2: Gradient descent with linear shearch.

As suposed, the difference between both methots is almost negligible, so we can use both for curve fitting.

Coefficients Comparison

The coefficients obtained from both methods are presented in Table

Method	c_0	c_1	c_2	c_3
LS	4.3843	34.2773	-25.0243	4.7662
GD	4.4074	34.1968	-24.9626	4.7532
Difference	0.0231	0.0805	0.0617	0.0129

Table 1: Comparison of polynomial coefficients from LS and GD methods.

The results obtained using the least squares (LS) method should be considered as the reference or ground truth values. This is because the LS approach, through the use of the normal equations, yields the exact polynomial that best fits a given set of data points.

In contrast, the `minError` function employs an iterative gradient descent (GD) method. While this method can approximate the LS solution, its accuracy depends on the tolerance parameter used in the convergence criterion.

For instance, when using a tolerance of $t = 10^{-4}$, the difference between the LS solution and the GD solution is as follows:

$$\text{Difference (LS - GD):}$$
$$\Delta c_0 = -2.581606\times10^{-5}, \quad \Delta c_1 = 8.992568\times10^{-5}, \quad \Delta c_2 = -6.891075\times10^{-5}, \quad \Delta c_3 = 1.445838\times10^{-5}$$

These differences demonstrate that as the tolerance decreases, the gradient descent solution approaches the least squares result.

Execution Times

The execution times for both the Least Squares (LS) method and the Gradient Descent (GD) method were recorded as follows:

- LS method: 0.000429 seconds
- GD method (with tolerance $t = 10^{-1}$): 3.414161 seconds

Interestingly, under this configuration, the GD method appears to take longer than the LS method, despite both being computationally efficient for a small dataset.

However, it is important to note that the runtime of the GD method increases significantly as the tolerance is decreased. For example, reducing the tolerance to $t = 10^{-4}$ results in an execution time of 7.463520 seconds for GD.

Conclusion

This study successfully fitted a 3rd-degree polynomial to the A320 aerodynamic efficiency data using both Least Squares and Gradient Descent methods. The coefficients from both methods were closely aligned, with differences on the order of 10^{-2}, confirming their equivalence in this context. The LS method, leveraging the normal equations, provides a direct and exact solution, while GD offers a flexible, iterative alternative that can be advantageous in terms of computational efficiency for certain problems.