CS/MATH 375: Homework 8

Adam Fasulo

October 20, 2025

Problem 1 (By Hand)

We are given four data points: $(-1, 3)$, $(1, 1)$, $(2, 3)$, and $(3, 7)$. The goal is to determine if a polynomial of degree $d = 2, 3$, or 6 can pass through all of them.

A polynomial of degree at most $N - 1 = 3$ can uniquely interpolate $N = 4$ points. However, the question asks whether *any* polynomial of a specific degree d can fit the data, not necessarily the minimal one.

Case $d = 2$

Let $p(x) = a_2x^2 + a_1x + a_0$. Substituting the four data points gives:

$$\begin{aligned} a_2 - a_1 + a_0 &= 3 \\ a_2 + a_1 + a_0 &= 1 \\ 4a_2 + 2a_1 + a_0 &= 3 \\ 9a_2 + 3a_1 + a_0 &= 7 \end{aligned}$$

This system has four equations but only three unknowns, so it's overdetermined. Using the first three equations:

$$2a_1 = -2 \Rightarrow a_1 = -1$$

Substituting $a_1 = -1$ gives:

$$\begin{aligned} a_2 + a_0 &= 2 \\ 4a_2 + a_0 &= 5 \end{aligned}$$

Subtracting these gives $3a_2 = 3 \Rightarrow a_2 = 1$, and hence $a_0 = 1$. Thus, $p(x) = x^2 - x + 1$.

Checking the fourth point:

$$p(3) = 9 - 3 + 1 = 7$$

It works. Therefore, a degree-2 polynomial exists:

$$\boxed{p(x) = x^2 - x + 1}$$

Case $d = 3$

A degree-3 polynomial can also fit the points. We can simply write:

$$p(x) = 0x^3 + x^2 - x + 1$$

which is technically a cubic polynomial, even though the cubic term is zero.

Case $d = 6$

Infinitely many degree-6 polynomials will work. Starting with $p_2(x) = x^2 - x + 1$, we can add any polynomial that vanishes at all four x-values:

$$Z(x) = (x + 1)(x - 1)(x - 2)(x - 3)$$

This $Z(x)$ has degree 4. Multiplying by a degree-2 polynomial such as $Q(x) = x^2$ gives a degree-6 term:

$$P_6(x) = p_2(x) + Q(x)Z(x) = (x^2 - x + 1) + x^2(x + 1)(x - 1)(x - 2)(x - 3)$$

Since $Z(x_j) = 0$ for each interpolation point x_j, $P_6(x_j) = p_2(x_j)$ for all j. Hence, $P_6(x)$ passes through all the points.

Problem 2

(a) Derivation (By Hand)

Given $n + 1$ data points (x_j, y_j), we want coefficients $a_0, a_1, \ldots, a_n$ such that:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

and $p(x_j) = y_j$ for all j.

This gives a system of linear equations:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

This matrix is called the **Vandermonde matrix**, with entries $V_{ij} = x_i^j$. The system can be written compactly as $Va = y$.

(b) MATLAB Implementation

```matlab
function c = interp_monomials(x, y)
    % Compute monomial interpolation coefficients
    x = x(:); y = y(:);
    n = length(x) - 1;
    V = x.^(0:n);
    c = V \ y;
end
```

Listing 1: interp_monomials.m

(c) Runge's Phenomenon

We tested $f(x) = 1/(1 + 25x^2)$ for $n = 2, 4, \ldots, 20$. The plots show the classic **Runge's phenomenon**: as n increases, the polynomial fits the center of the interval $[-1, 1]$ quite well, but oscillates heavily near the endpoints. The interpolation error grows large there because the points are equally spaced.

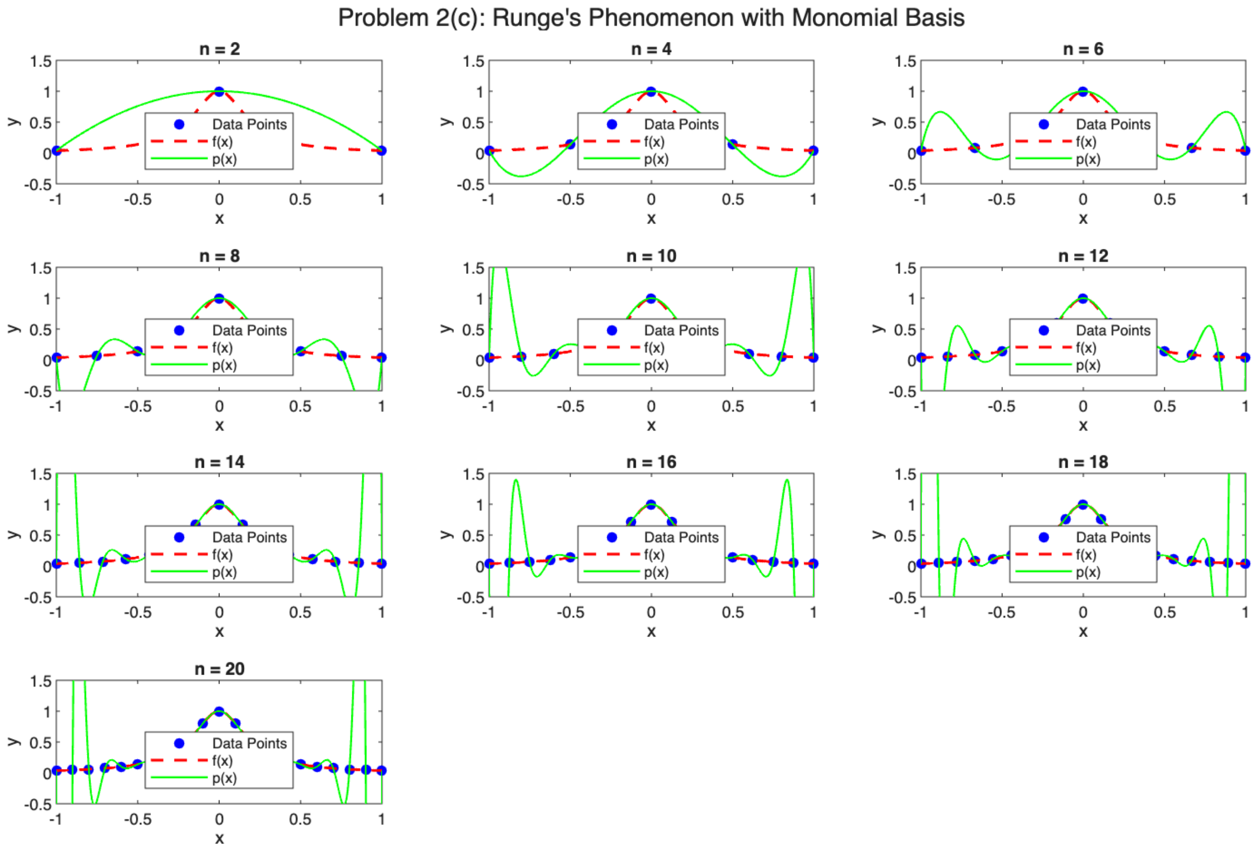

Figure 1: Runge's phenomenon for $n = 2, 4, \ldots, 20$.

(d) Numerical Instability

Running the main script reveals where interpolation starts to fail:

```
Failure to interpolate detected at n = 26
Max interpolation error: 6.09e-08
Condition number of V: 6.53e+11
```

At $n = 26$, the Vandermonde matrix becomes severely ill-conditioned. The condition number $\kappa(V)$ grows exponentially with n, amplifying rounding errors in floating-point arithmetic. Even MATLAB's stable backslash solver can't fully compensate, so the computed coefficients are inaccurate, leading to a noticeable mismatch between $V * c$ and y.

Problem 3

(i) Lagrange Interpolation and Runge's Phenomenon

Repeating the experiment with Lagrange interpolation (for $n = 20$) produces nearly identical plots: the same oscillations appear near the endpoints. This confirms that **Runge's phenomenon is caused by equally spaced nodes, not by the polynomial basis**. Both the monomial and Lagrange methods compute the same interpolating polynomial.

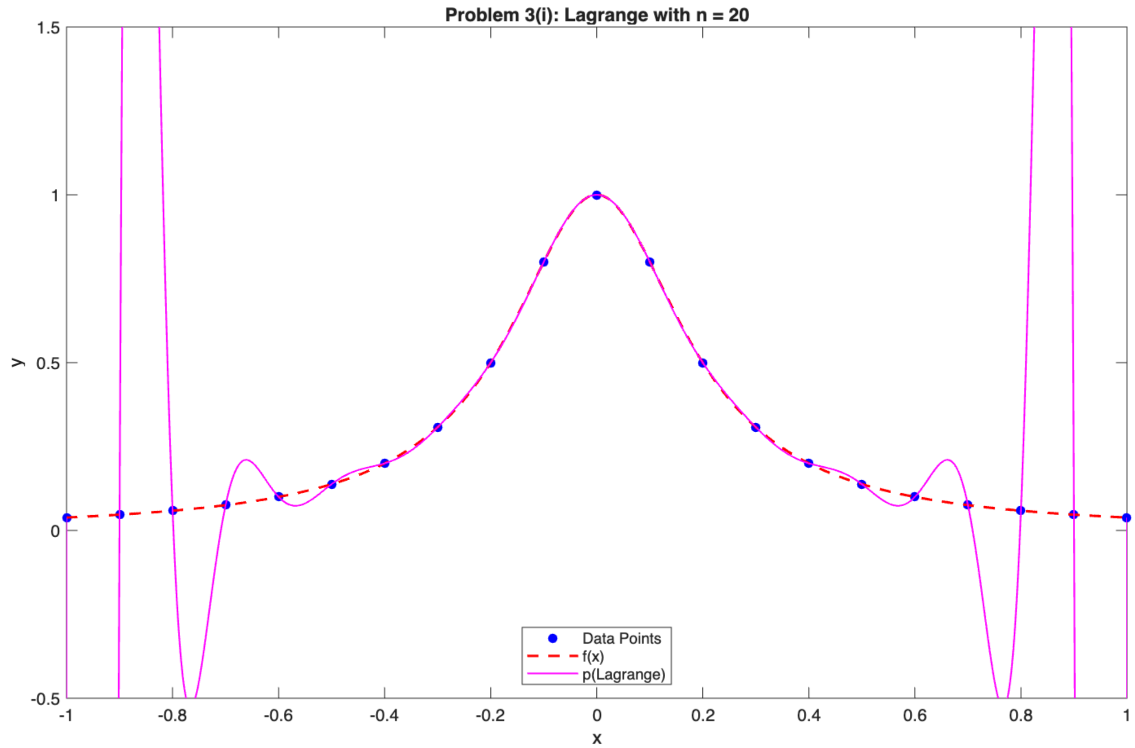

Figure 2: Lagrange interpolation for $n = 20$, still showing Runge's phenomenon.

(ii) Numerical Stability

For $n = 26$, where the monomial approach failed, Lagrange interpolation still produced a perfect fit:

```
Max interpolation error (Lagrange): 0.000000e+00
```

The error is effectively zero (within machine epsilon). This demonstrates that **Lagrange interpolation is numerically stable**, since it evaluates the polynomial directly instead of solving an ill-conditioned system.

Summary

- A quadratic polynomial fits all four given points exactly.
- Runge's phenomenon appears for large n with equally spaced nodes, regardless of the basis used.
- The monomial system fails around $n = 26$ due to the ill-conditioned Vandermonde matrix.
- Lagrange interpolation avoids this issue and remains stable for larger n.