+2 for presentation

CS375 HW8

Ryan Scherbarth

October 2024

1. Can you find the degree d polynomial that passes through $(-1,3),(1,1),(2,3),(3,7)$ for $d = 2,3,6$? If you can find such a polynomial, write it down. If no such polynomial exists, explain why. (Hint): The Newton polynomial basis makes this easy

- (a) $d = 2$

  Given $n+1$ distinct points, we know that we can find a minimum degree unique polynomial $p(x)$ of degree at most n, therefore the lowest value d that will be able to satisfy this requirement is $d = 3$, so we can immediately conclude it is not possible to pass through each of these points with $d = 2$, or any $d < 3$ for that matter.

- (b) $d = 3$

  Part (a) confirms there exists some unique polynomial of degree 3. Following the Newton Polynomial formula;

x	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+2}]$	$f[x_k, x_{k+2}, x_{k+3}]$
-1	3	$\frac{3-1}{-1-1} = -1$	$\frac{-1-2}{-1-2} = 1$	$\frac{1-1}{-1-3} = 0$
1	1	$\frac{1-3}{1-2} = 2$	$\frac{2-4}{1-3} = 1$	
2	3	$\frac{3-7}{2-3} = 4$		
3	7			

-10

$$\begin{aligned} p(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_2)(x - x_1)(x - x_0) \\ &= 3 - 4(x+1) - 1(x+1)(x-1) + 0 \\ &= -x^2 - 4x \end{aligned}$$

what degree is this? Also, this polynomial is not correct

- (c) $d = 6$

  It would also not be possible to replicate this graph given a 6th degree polynomial. By the same logic in part (a), we would need 7 data points in order to fit a 6th degree polynomial.

2. Consider the following points of data(n=3); $(-1,1),(1,1),(2,1),(3,9)$

   (a) Find the monomial polynomial interpolating these data points by constructing the Vandermonde matrix and inverting it (you can use Matlab to do the inversion if you wish)

   We can start by constructing the Vandermonde matrix V;

$$V = \begin{bmatrix} (-1)^3 & (-1)^2 & (-1)^1 & 1 \\ 1^3 & 1^2 & 1^1 & 1 \\ 2^3 & 2^2 & 2^1 & 1 \\ 3^3 & 3^2 & 3^1 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{bmatrix}$$

   We also can create a matrix with the y values $y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 9 \end{bmatrix}$

   I plugged the values in to Matlab and evaluated $inv(V) * y$ where we find the final result $\begin{bmatrix} 1 \\ -2 \\ -1 \\ 3 \end{bmatrix}$

   (b) Find the Lagrange polynomial interpolating these data points

$$L(x) = \sum_{i=0}^{n} y_i * \Pi_{j \neq i} \frac{x - x_j}{x_i - x_j}$$
$$\ell_0 = \frac{(x-1)(x-2)(x-3)}{(-2)(-3)(-4)} = \frac{(x-1)(x-2)(x-3)}{24}$$
$$\ell_1 = \frac{(x+1)(x-2)(x-3)}{(2)(-1)(-2)} = \frac{(x+1)(x-2)(x-3)}{4}$$
$$\ell_2 = \frac{(x+1)(x-2)(x-3)}{(3)(1)(-1)} = \frac{(x+1)(x-2)(x-3)}{-3}$$
$$\ell_3 = \frac{(x+1)(x-1)(x-2)}{(4)(2)(1)} = \frac{(x+1)(x-1)(x-2)}{8}$$
$$L(x) = \frac{(x-1)(x-2)(x-3)}{24} + \frac{(x+1)(x-2)(x-3)}{4} + \frac{(x+1)(x-2)(x-3)}{-3} + 9(\frac{(x+1)(x-1)(x-2)}{8})$$

   (c) Find the Newton polynomial interpolating these data points

x	$f[x_k]$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
-1	1			
1	1	$\frac{1-1}{1-(-1)} = 0$		
2	1	$\frac{1-1}{2-1} = 0$	$\frac{0-0}{2-(-1)} = 0$	
3	9	$\frac{9-1}{3-2} = 8$	$\frac{8-0}{3-1} = 4$	$\frac{4-0}{3-(-1)} = 1$

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

(d) Show that your answers in a, b, and c are equivalent. We can create a table to illustrate each of our functions evaluate to the given y values for the x inputs we are given:

x	y	$p(x)$	$L(x)$	$N(x)$
-1	1	1	1	1
1	1	1	1	1
2	1	1	1	1
3	9	9	9	9

3. Consider the data points $(1, 0), (2, \ln(2)), (4, \ln(4))$, which came from sampling the function $f(x) = \ln(x)$.

(a) Using Lagrange or Newton form, find the quadratic polynomial interpolation of this data. We will use Newton's form;

x	$f(x)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$
1	0		
2	$\ln(2)$	$\frac{\ln(2)-0}{2-1} = \ln(2)$	
4	$\ln(4)$	$\frac{\ln(4)-\ln(2)}{4-2} = \frac{\ln(2)}{2}$	$\frac{\frac{\ln(2)}{2}-\ln(2)}{4-1} = -\frac{\ln(2)}{6}$

$$N(x) = \ln(2)(x-1) - \frac{\ln(2)}{6}(x-1)(x-2)$$

(b) Use the error formula from lecture to find an upper bound on error if you evaluate your polynomial at $c = 3$, i.e. $|f(3) - p(3)|$.

We can say $|f(x) - p_n(x)| \leq M\frac{h^{n+1}}{4(n+1)}$ to set an upper bound on the error. Given;
$f'(x) = \frac{1}{x}$
$f''(x) = -\frac{1}{x^2}$
$f^{(3)}(x) = \frac{2}{x^3}$

$$|f^{(3)}(1)| = 2$$ /3

Therefore $M = 2$ in this case. We then find h as the max distance between the set of points, in this case when $A = 1$, $B = 4$ we have $h = 3$.

$$|f(3) - p_2(3)| \leq 2(\frac{3^3}{4(3)})$$
$$\leq 4.5$$

So we have an upper bound on our error of 4.5.

-4

(c) Evaluate your polynomial at 3, calculate $|p(3) - \ln(3)|$, and compare with your answer from part b.

$$\begin{aligned} N(x) &= \ln(2)(x-1) - \frac{\ln(2)}{6}(x-1)(x-2) \\ &= \ln(2)(3-1) - \frac{\ln(2)}{6}(3-1)(3-2) \\ &= \frac{6\ln(2)}{3} - \frac{\ln(2)}{3} \\ &= \frac{5\ln(2)}{3} \end{aligned}$$

If we let $ln(3) \approx 1.1$,

$$\begin{aligned} |p(3) - \ln(3)| &= |1.15 - 1.1| \\ &= 0.05 \end{aligned}$$

So the error is significantly smaller than what we found in the previous part. This is still plausible though since we only found an upper bound in part b and this solution is within that bound.

4. In this problem you will write (and use) your own code to compute the monomial based interpolating polynomial for arbitrary data: $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ for $(x_j, y_j) = (x_j, f(x_j)), j = 0, \ldots, n$..

(a) Write a matlab function function a=interp-monomials(x,y) that returns the vector of coefficients $a = [a_0, \ldots, a_n$ given input vectors

$$\begin{aligned} x &= [x_0, x_1, \ldots, x_n] \\ y &= [y_0, y_1, \ldots, y_n] \end{aligned}$$

```
function a = interp_monomials(x, y)
    n = length(x);

    V = zeros(n);
    for i = 1:n
        V(:, i) = x.^(i - 1); % Each column is x^(i-1)
    end

    a = V \ y(:); % Solve V * a = y for the coefficients 'a'

    disp('Polynomial coefficients:');
    disp(a');
end
```

(b) Apply your function to find the polynomial that interpolates $f(x) = \frac{1}{1+25x^2}$

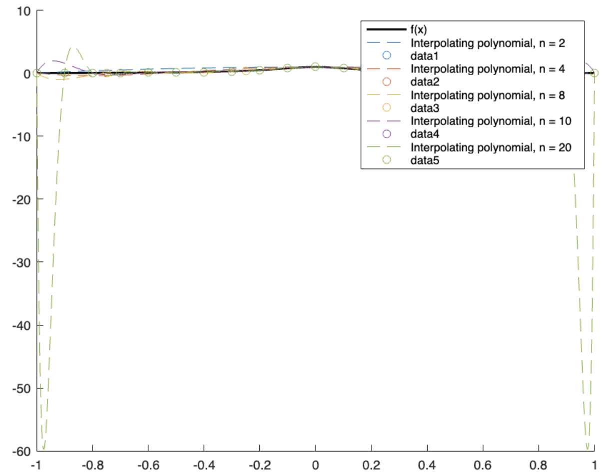
For the points $x_i = -1 + ih, h = \frac{2}{n}, i = 0, \ldots, n$ Plot the data points $(x_i, f(x_i))$ as circles (use plot(x,f,'o') and in the same plot, plot the polynomial and $f(x)$ on a finer grad (still on x $\in$ [-1,1]) by using $n_{fine} = n * 100$ total points for for plotting. Show what happens when you increase N. Try $n = 2, 4, \ldots, 20$. Comment to evaluate the polynomial at some given values of x you can use polyval (note that polyval requires your coefficients to be in a specific order. Do help polyval).

We see the Runge phenomena taking place near the edge of the graph on either side, as the derivative approaches 0 our estimates become less and less accurate. We range as far down as -60 on this graph.

```
function run_interpolation()
    f = @(x) 1 ./ (1 + 25 * x.^2);

    x_fine = linspace(-1, 1, 1000);
    y_fine = f(x_fine);

    figure;
    hold on;

    plot(x_fine, y_fine, 'k', 'LineWidth', 1.5, 'DisplayName', 'f(x)');

    colors = lines(10);
    color_index = 1;

    for n = [2, 4, 8, 10, 20]
        h = 2 / n;
        x = -1 + (0:n) * h;  % xi = -1 + ih for i = 0, ..., n
        y = f(x);

        a = interp_monomials(x, y);

        y_poly = polyval(flip(a), x_fine);

        plot(x_fine, y_poly, '--', 'Color', colors(color_index, :), ...
             'DisplayName', ['Interpolating polynomial, n = ', num2str(n)]);

        plot(x, y, 'o', 'Color', colors(color_index, :), 'MarkerSize', 6);

        color_index = color_index + 1;
    end

    legend;
end
```

(c) Keep increasing n until the polynomial does not even interpolate $f(x)$. For what n does this happen? (you can find this by evaluating your polynomial at the original interpolating points x_i and then subtracting them from y_i). What is the condition number fo the matrix V for this n? Can you explain what is happening?

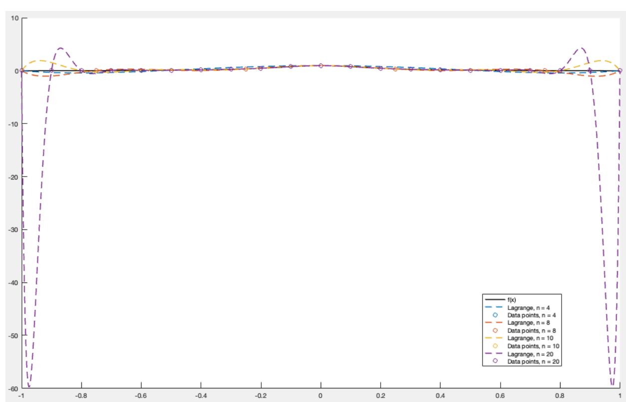
Incrementing n we find the smallest n for which we fail to interpolate the function is $n = 33$. Here, we have a maximum error of 2.11189e-06 and a Condition number 1.62306e+15. The condition



number of V for $n = 34$ becomes 4.98e+15

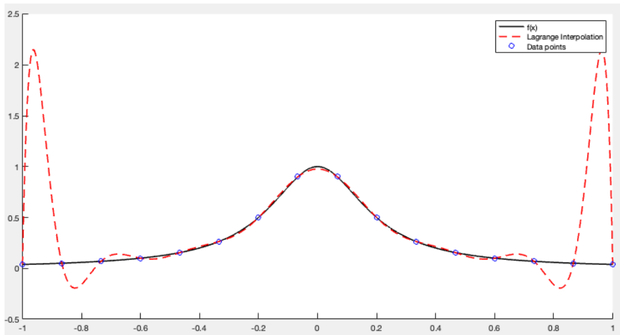
5. Solving the interpolation problem usin the monomial basis (as above) is notoriously ill-conditioned, in fact it is possible to shoe $\text{cond}(V) \approx \pi^{-}1e^{\frac{\pi}{4}}(3.1)^n$ where V is the Vandermonde matrix. Perhaps Lagrage polynomials will help! I have provided a MATLAB function my-lagrange which can be used to evaluate the Lagrange interpolating polynomial at an arbitarry set of points. It's syntax is:
y = my-lagrange(x, pointx, pointy)
where pointx and pointy are arrays of the x and y values of your interpolating data respectively, and x is an array of locations where you would like to evaluate the polynomial.

   (a) Show some plots that the Lagrange polynomial still suffers from Runge phenomenon near the end points -1 and 1.



Plotting n values up to $n = 20$ easily illustrates the Runge phenomenon for this function.

(b) Find the Lagrange polynomial for a value of n which failed to interpolate the data in Problem 3. Does this polynomial correctly interpolate the data now? What changed from Problem 3? We



see we now have a much better interpolation for the function when we use $n = 15$. Note that even when implementing this method but we still run into the issue of Runge phenomenon around the end points. The Lagrange interpolation resolves some of the issues with ill conditioning in monomial-based interpolation though. Implementing Chebyschev nodes would do a better job if we were trying to combat the Runge issue on the end points.