

ICSI 401 Final

James Oswald

December 6th, 2020

1. (10 points) Calculate the Lagrange form of the interpolating polynomial for the following points: $(0, 1)$, $(1, 3)$, $(2, 12)$.

I solve by using the formula for the Lagrange form of the interpolating polynomial with three points and plugging in our respective values.

$$P(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}y_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}y_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}y_3$$

$$P(x) = \frac{(x - 1)(x - 12)}{(0 - 1)(0 - 1)}1 + \frac{(x - 0)(x - 2)}{(1 - 0)(1 - 2)}3 + \frac{(x - 0)(x - 1)}{(2 - 0)(2 - 1)}12$$

$$P(x) = \frac{(x - 1)(x - 2)}{(-1)(-2)}1 + \frac{(x)(x - 2)}{(1)(1 - 2)}3 + \frac{(x)(x - 1)}{(2)(1)}12$$

$$P(x) = \frac{x^2 - 3x + 2}{2} + \frac{x^2 - 2x}{-1}3 + \frac{x^2 - x}{2}12$$

$$P(x) = \frac{x^2 - 3x + 2}{2} + \frac{-6x^2 + 12x}{2} + \frac{12x^2 - 12x}{2}$$

$$P(x) = \frac{x^2 - 3x + 2 - 6x^2 + 12x + 12x^2 - 12x}{2}$$

$$P(x) = \frac{7x^2 - 3x + 2}{2}$$

$$P(x) = \frac{7}{2}x^2 - \frac{3}{2}x + 1$$

Thus the Lagrange form of the interpolating polynomial is $\frac{7}{2}x^2 - \frac{3}{2}x + 1$

2. (10 points) Complete the divided differences table for the above points. Your table should be in the following form:

$f[0]$	X	X
$f[1]$	$f[0, 1]$	X
$f[2]$	$f[1, 2]$	$f[0, 1, 2]$

Here, an X indicates that there should NOT be a number in that entry.

In other words, what is expected is that you compute $f[0], f[1], f[0, 1]$, etc. We begin by calculating our first column based off of the points provided.

$$f[0] = 1, f[1] = 3, f[2] = 12$$

Next we use the interpolation formula to calculate terms in the second column:

$$\begin{aligned} f[a, b] &= \frac{f[b] - f[a]}{b - a} \\ f[0, 1] &= \frac{f[1] - f[0]}{1 - 0} = \frac{3 - 1}{1 - 0} = 2 \\ f[1, 2] &= \frac{f[2] - f[1]}{2 - 1} = \frac{12 - 3}{2 - 1} = 9 \end{aligned}$$

Finally we use:

$$\begin{aligned} f[a, b, c] &= \frac{f[b, c] - f[a, b]}{c - a} \\ f[0, 1, 2] &= \frac{f[1, 2] - f[0, 1]}{2 - 0} = \frac{9 - 2}{2} = 3.5 \end{aligned}$$

Thus our final divided differences table is:

1		
3	2	
12	9	3.5

3. (10 points) Consider the following divided differences table for the points $(0, 1), (1, 2), (2, 4)$.

1		
2	1	
4	2	1/2

Use the divided differences table to write down the Newton form of the interpolating polynomial for the given points.

To calculate the Newton form of the interpolating polynomial, we use the formula:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

Thus we can use the diagonal values in the provided by the divided differences table to plug in:

$$P(x) = 1 + 1(x - 0) + \frac{1}{2}(x - 0)(x - 1)$$

$$P(x) = 1 + x + \frac{1}{2}x^2 - \frac{1}{2}x$$

$$P(x) = \frac{1}{2}x^2 + \frac{1}{2}x + 1$$

Thus the Newton form of the interpolating polynomial for these points is $\frac{1}{2}x^2 + \frac{1}{2}x + 1$

4. (10 points) Suppose that we wish to find a piecewise polynomial interpolation for the points $(x_0, y_0) = (0, 1)$, $(x_1, y_1) = (1, 3)$, $(x_2, y_2) = (2, 1)$. In particular, we want two polynomials $P_0(x)$ and $P_1(x)$ satisfying various matching conditions, and we define

$$P(x) = \begin{cases} P_0(x) & x \in [x_0, x_1) \\ P_1(x) & x \in [x_1, x_2] \end{cases}$$

Write down a condition on the values of P_0 and P_1 that is necessary and sufficient to conclude that $P(x)$ is continuous on the interval $[x_0, x_2]$.

To prove that $P(x)$ is continuous on the interval $[x_0, x_2]$ (assuming that it is a given that $P_0(x)$ and $P_1(x)$ are continuous) it is necessary and sufficient to prove that $P_0(x_1) = P_1(x_1)$. If this condition is met then $P(x)$ is guaranteed to be continuous on the interval $[x_0, x_2]$.

5. (10 points) The following pseudocode implements Horner's rule for polynomial evaluation

```
% Implements Horner's method for evaluating a polynomial at a point x.
%
% Inputs:
% * coefficients: a row vector of polynomial coefficients, with the lowest-degree
% coefficient first. For example, [1, 2, 3] corresponds to the polynomial
%  $p(x) = 1 + 2x + 3x^2$ .
% * x: a floating point number.
%
% Output:
% The value of  $p(x)$ .
%
function y = horner(coefficients, x)
    degree = size(coefficients)-1;
    degree = degree(2);
    if (degree == 0)
        y = coefficients(1)
    else
        % degree > 1.
        z = horner(coefficients(2:(degree+1)), x)
        y = coefficients(1) + x * z;
    end
end
```

coefficients	x	y
[1, 1, 3, 1]		
[1, 3, 1]		
[3, 1]	1	3
[1]	N/A	1

Complete either table using the corresponding pseudocode for full credit.

From the table I see that when $z = 1$ then $y = 3$, thus we solve for what x we're calculating with via $y = \text{coef}(1) + x * z$. We plug in and see $3 = 3 + x * 1$, thus it is apparent we are solving for when $x = 0$. With this knowledge we can finish the rest of the table by hand just by going through each iteration.

coefficients	z	y
[1, 1, 3, 1]	1	1
[1, 3, 1]	3	1
[3, 1]	1	3
[1]	N/A	1

6. (10 points) Power method
Consider the following matrix:

$$A = \begin{pmatrix} 2 & 4 \\ 4 & 3 \end{pmatrix}$$

Consider the following Matlab code for computing an eigenvector corresponding to the second largest eigenvalue of A.

```
A = [2, 4; 4, 3];
%
% Code for computing an eigenvector corresponding to the second-largest
% eigenvector of A.
%
x = [1, 0]';
first_eigenvector = [0.661802563235740, 0.749678175815866]';
% Make x orthogonal to the first eigenvector by subtracting off its projection.
x = x - x' * first_eigenvector * first_eigenvector;
% Apply the power method for 1000 iterations.
for j = 1:1000
    % Line B
    x = A*x;
    x = x/norm(x); % Line D
end
% Line C
display(x);
```

Which of the following modifications of the code will ensure convergence to the correct answer when run on a computer?

- A. No modification is needed. It will converge to the correct answer.
- B. At the beginning of the for loop (marked with “Line B”), insert the following code:

```
x = x - x' * first_eigenvector * first_eigenvector;
```

- C. After the for loop (marked “Line C”), insert the following code:

```
x = (A' * A) \ (A' * x);
```

so as to solve the normal equations.

- D. Remove the line marked by “Line D”.

The correct choice is option B. It’s literally in the comment that we need to perform that operation to make x orthogonal to the first eigenvector by subtracting off its projection. Without adding that on line B we will get the wrong value.

7. (10 points) Consider the backward difference approximation for the derivative of a function f :

$$\hat{f}'(x) = \frac{f(x) - f(x-h)}{h}$$

for a small parameter $h > 0$.

Ignoring rounding error, what is the absolute error in approximating $f'(x)$ by $\hat{f}'(x)$ expressed in terms of h ?

- A. $|f'(x) - \hat{f}'(x)| = O(h)$ as $h \rightarrow 0$.
- B. $|f'(x) - \hat{f}'(x)| = O(h^2)$ as $h \rightarrow 0$.
- C. $|f'(x) - \hat{f}'(x)| = O(h^3)$ as $h \rightarrow 0$.
- D. $|f'(x) - \hat{f}'(x)| = O(h^{-1})$ as $h \rightarrow 0$.

Hint: Use the Taylor series of f centered around x .

I start with expanding the definition $|f'(x) - \hat{f}'(x)|$ and then expanding Taylor series for $f(x+h)$ and $f(x-h)$

$$\begin{aligned} & |f'(x) - \hat{f}'(x)| \\ &= \left| \frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h} \right| \\ &= \left| \frac{f(x+h) + f(x-h) - 2f(x)}{h} \right| \\ &= \left| \frac{f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \dots + f(x) - hf'(x) + \frac{1}{2}h^2f''(x) + \dots - 2f(x)}{h} \right| \\ &= \left| \frac{h^2f''(x) + \dots}{h} \right| \\ &= |hf''(x) + \dots| \\ &= O(h) \text{ as } h \rightarrow 0 \end{aligned}$$

Thus the absolute error in approximating $f'(x)$ by $\hat{f}'(x)$ expressed in terms of h is choice A, $O(h)$ as $h \rightarrow 0$.

8. (10 points) Suppose that you wish to compute a least-squares fit to the following points using a polynomial $p(x)$ of degree ≤ 4 : $(x_0, y_0) = (0, 1)$, $(x_1, y_1) = (1, 3)$, $(x_2, y_2) = (2, 4)$, $(x_3, y_3) = (3, 2)$, $(x_4, y_4) = (4, 5)$. What is the value of the approximation error of the least-squares degree ≤ 4 polynomial for these points (i.e., you need to calculate $\sum_{j=0}^4 (y_j - p(x_j))^2$). **Hint:** If you're doing a lot of calculating for this problem, then you're probably doing something wrong.

I begin by computing the degree 4 Vandermonde matrix of the problem

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}$$

From this obtain the coefficient vector using the pseudoinverse of the Vandermonde matrix and multiplying it with the y values.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}^{\dagger} \begin{bmatrix} 1 \\ 3 \\ 4 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{2}{3} \\ \frac{61}{12} \\ -\frac{17}{6} \\ \frac{5}{12} \end{bmatrix}$$

To solve for $p(x)$ I plug the coefficients into a polynomial and get $p(x) = \frac{5}{12}x^4 - \frac{17}{6}x^3 + \frac{61}{12}x^2 - \frac{2}{3}x + 1$. To calculate the approximation error I plug into $\sum_{j=0}^4 (y_j - p(x_j))^2$, however it should be obviously 0 since this polynomial was designed to perfectly fit these points.

$$\begin{aligned} & \sum_{j=0}^4 (y_j - p(x_j))^2 \\ &= (y_0 - p(x_0))^2 + (y_1 - p(x_1))^2 + (y_2 - p(x_2))^2 + (y_3 - p(x_3))^2 + (y_4 - p(x_4))^2 \\ &= (1 - p(0))^2 + (3 - p(1))^2 + (4 - p(2))^2 + (2 - p(3))^2 + (5 - p(4))^2 \\ &= (1 - 1)^2 + (3 - 3)^2 + (4 - 4)^2 + (2 - 2)^2 + (5 - 5)^2 \\ &= (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 \\ &= 0 \end{aligned}$$

9. (10 points) Suppose that we want to approximate $f'(x)$, with $f(x) = \sin(x)$ and $x = \pi/3$, so that $f'(x) = 0.5$.

We will use the forward difference formula:

$$\hat{f}'(x) = \frac{f(x+h) - f(x)}{h}$$

Below is a table giving the relative error of the approximation for various values of h , when implemented in Matlab.

h	$\frac{ f'(x) - \hat{f}'(x) }{f'(x)}$
10^{-5}	8.660265948035038e-06
10^{-6}	8.660562260676128e-07
10^{-7}	8.601352896597801e-08
10^{-8}	6.077471192966753e-09
10^{-9}	8.274037077704576e-08
10^{-10}	8.274037077704576e-08
10^{-11}	8.274037077704576e-08
10^{-12}	8.890058234078956e-05
10^{-13}	7.992778373593354e-04
10^{-14}	7.992778373593354e-04
10^{-15}	0.110223024625156

Notice, in particular, that the relative error decreases until $h10^{-8}$, and then it begins to increase as we further decrease h . What is the source of this phenomenon?

- A. The Runge phenomenon.
- B. Rounding error in the computation of $f(x+h)$ and $f(x)$.
- C. A bug in Matlab's source code.
- D. f is not differentiable near $x = \pi/3$.

The source of this phenomenon is choice B, Rounding error in the computation of $f(x+h)$ and $f(x)$. This is because at scales this small, even if h can still be reasonably encoded with minimal within a floating point number, The result of operations and functions computing with h may not be.