# Table of Contents



# Assignemnt

Newton's Interpolation

```
% Name        : Mohamed Mafaz
% Roll Number : AM25M009
% Department  : Applied Mechanics

clc
clear
close all
```

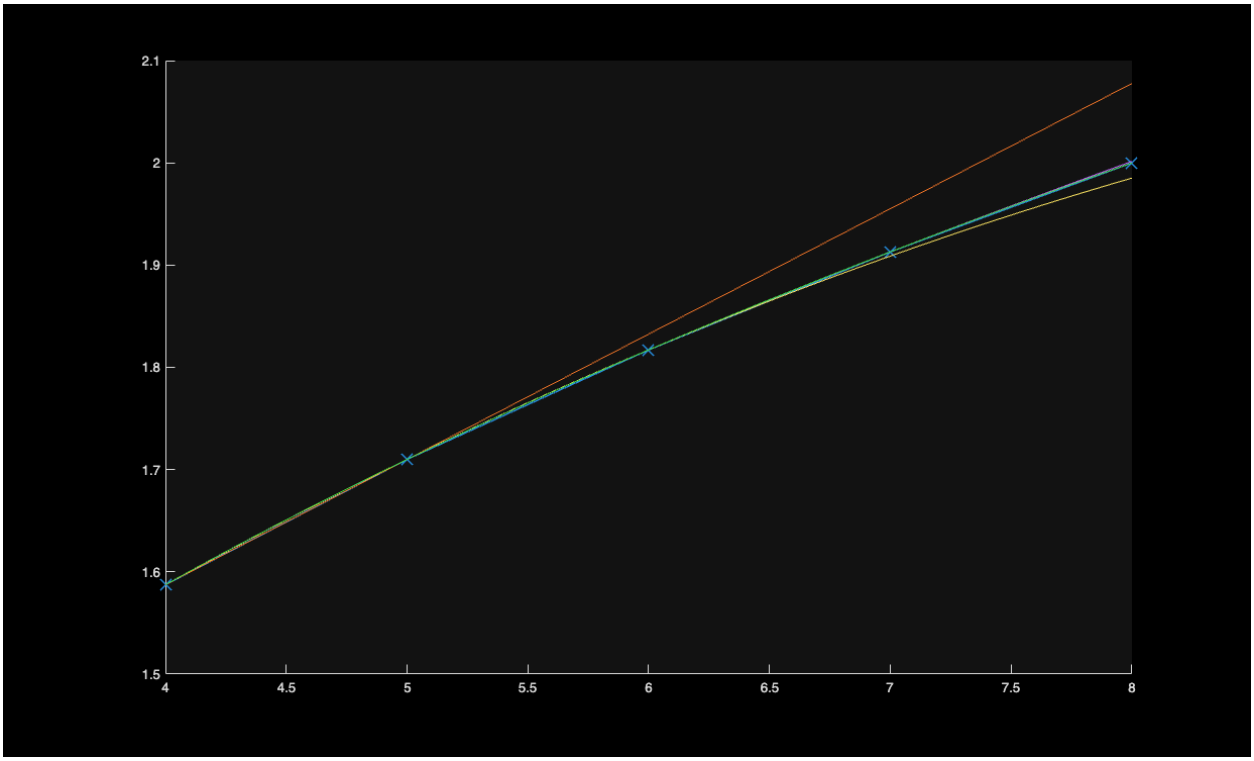
# Part 1 (Preprocessing)

```
x = [4.0, 5.0, 6.0, 7.0, 8.0];
y = [1.58740105, 1.709976, 1.81712059, 1.912931, 2.0];

% x = [0.1, 1.5, 2.95, 3.0]
% y = [1.2, -0.25, 1.71, 5.5]

n = length(x);
xx = linspace(min(x), max(x), 500);

% Compute divided difference table manually
dd = zeros(n, n);
dd(:,1) = y';
```

# Part 2 (Processing / Using the Interpolation Algorithm)

```
for j = 2:n
    for i = 1:n-j+1
        dd(i,j) = (dd(i+1,j-1) - dd(i,j-1)) / (x(i+j-1) - x(i));
    end
end

figure;
hold on;
plot(x, y, 'LineWidth', 1, 'DisplayName', 'Actual Data points', Marker='x',
MarkerSize=12);
```

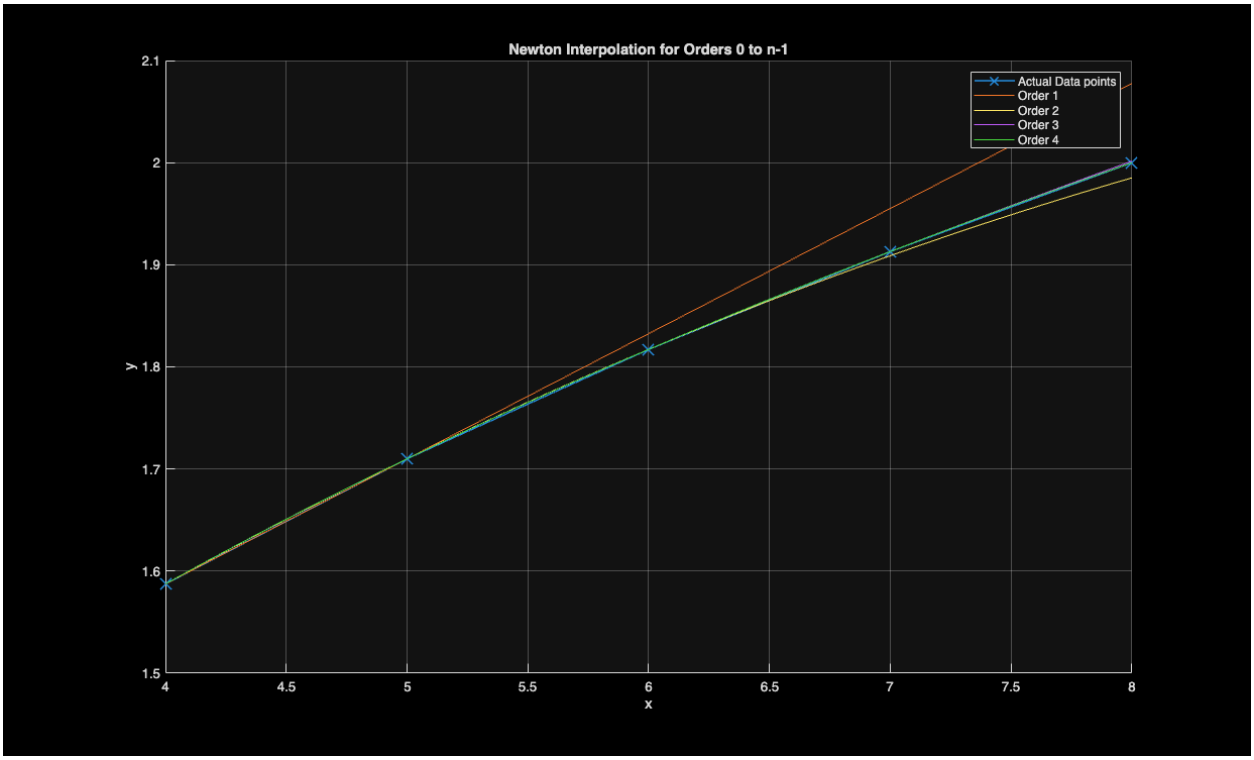
```
for order = 1:n-1
                                % Allocate storage for the interpolated
values
    yy = zeros(size(xx));
                                % Evaluate the Newton polynomial of given
order at each point in xx
    for k = 1:length(xx)
                                % Start with the first divided difference
(constant term)
        val = dd(1,1);
                                % This will accumulate (x - x0)(x - x1)...
progressively
        product_term = 1;
        for j = 1:order
                                % Build the product (x - x0)(x - x1)...(x -
x_{j-1})
            product_term = product_term * (xx(k) - x(j));
            val = val + dd(1,j+1) * product_term;
        end
        yy(k) = val;
    end
    plot(xx, yy, 'DisplayName', ['Order ', num2str(order)]);
    hold on
end
```




# Part 3 (Post Processing / Plotting)

Plotting actual Data plot(x, y, 'LineWidth', 1, 'DisplayName', 'Actual Data points', Marker='x', MarkerSize=12);

```
legend show;
title('Newton Interpolation for Orders 0 to n-1');
xlabel('x');
ylabel('y');
grid on;
hold off;
```




*Published with MATLAB® R2025a*