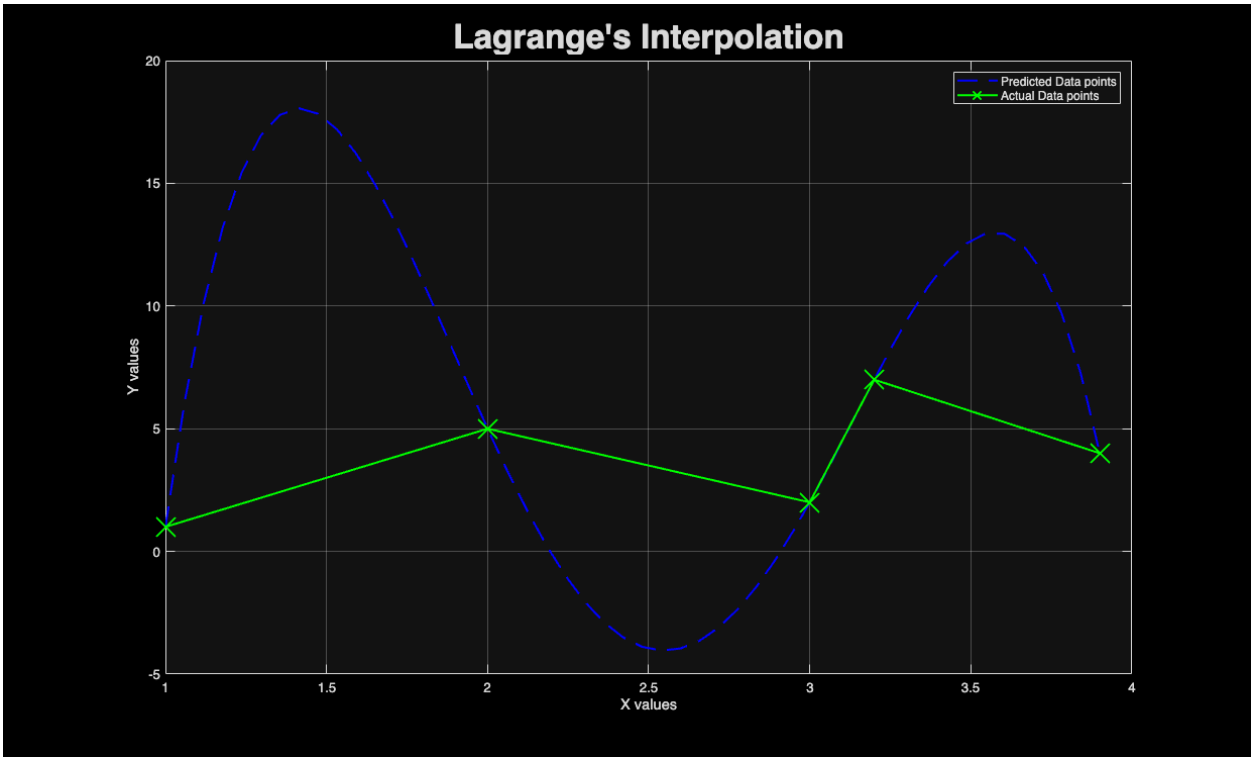
```
clc
clear
```

Lagrange's Interpolation Function

```
function [sum] = LI(xs, ys, number)
    % Straight forward brute force way to find l
    l = [];

    for j = 1:length(xs)
        a = 1;
        c = 1;
        for i = 1:length(xs)
            if i ~= j       % Or else it will always give 0
                a = a * (number - xs(i));  % Calculating Numerator and
Denominator differently
                c = c * (xs(j) - xs(i));
            end
        end
        l(j) = a / c;
    end

    % This calculates l0 x y0 + l1 x y1 + ....
    sum = 0;
    for i = 1: length(l)
        sum = sum + (ys(i) * l(i));
    end
end
```



Using the function

```
x = [1,  2,    3,    3.2,    3.9];
y = [1,  5,    2,    7,      4];


sample_points = 50;

% Predicting
test_xs = linspace(1, x(end), sample_points);
test_ys = [];
for i = 1: sample_points
    test_ys = [test_ys, LI(x, y, test_xs(i)) ];
end

% Plotting predicted Data
plot(test_xs, test_ys, '--b', 'LineWidth', 1.5, 'DisplayName', 'Predicted
Data points');
xlabel('X values');
ylabel('Y values');
title("Lagrange's Interpolation", 'FontSize', 25);

hold on

% Plotting actual Data
plot(x, y, 'g', 'LineWidth', 1.5, 'DisplayName', 'Actual Data points',
Marker='x', MarkerSize=20);
```

```
legend show
grid on;
```

Published with MATLAB® R2025a