



Khulna University of Engineering & Technology, Khulna

Department of Biomedical Engineering

SESSIONAL REPORT

Course No: **BME 2152**

Experiment No : **08**

Name of the Experiment: **Function Approximation Using Lagrange Interpolation in MATLAB.**

Remarks:

Name: **Md. Fuad Hasan Hamim**

Roll : **2215029**

Year : **2nd**

Term : **1st**

Date of performance: 5 February 2025

Date of submission : 12 February 2025

Objective:

The main objectives of this sessional are:

1. To understand and implement the Lagrange Interpolation method.
2. To learn how iterative numerical methods work and their application in solving equations.

Introduction:

The Lagrange Interpolation Method is a mathematical technique used to estimate the value of a function when we have only a few known data points. Imagine you have a set of points on a graph, but you want to find the value of the function at a point in between—without knowing the exact formula of the function. This method helps to construct a polynomial that smoothly passes through all the given points and predicts values in between. The core idea behind Lagrange interpolation is to create small polynomials (called Lagrange basis polynomials) for each data point. These polynomials are designed in such a way that they equal 1 at their corresponding data point and 0 at all other given points. By combining these small polynomials with the given function values, we create a single polynomial that represents the function as closely as possible.

This method is widely used in numerical analysis, physics, engineering, and computer graphics where an exact function might not be available, but discrete data points are known. It helps in curve fitting, function approximation, and even in digital signal processing.

Formula:

Mathematically, if we have n data points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ the Lagrange polynomial is given by:

$$P_n(x) = \sum_{i=0}^n f_i l_i(x) = f_0 l_0 + f_1 l_1 + f_2 l_2 + \dots + f_n l_n$$
$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

Algorithm for Lagrange Interpolation Method:

Step 1. Define a set of known data points.

(e.g. x_0, x_1, x_2 recommended to take an array)

Step 2. Define the functional value of the known data points.

(e.g. f_0, f_1, f_2)

Step 3. Take user-defined input for reference value.

Step 4. Find the length of the array (which defined the set of known data points).

Step 5. Create a loop (for loop) for “i” which will run till the end of the array and initiate “ l_i ”.

Step 6. Initiate a nested (for) loop for “j”.

Step 7. Initiate another nested (if) loop to check the condition whether $i \neq j$.

Step 8. Formula implementation in amidst of the nested (if) loop to get values of l_i .

Step 9. Close the nested loops (“if” and “for” loop respectively).

Step 10. Another formula implementation to get values of $P_n(x)$.

Step 11. End the “for” loop and print the result.

Task || Estimate the value of a function by Lagrange interpolation method.

Code:
<pre>clc; clear all; x = [2 3 4]; y = [1.4142 1.7321 2]; a =input("Give input value: "); n = length(x); p=0; for i= 1:1:n l=1; for j = 1:1:n if i ~= j l = l * (a - x(j)) / (x(i) - x(j)); end end p = p + l * y(i); end fprintf("pnx= %g", p);</pre>
Output:
<pre>Give input value: 3.6 pnx= 1.89884>></pre>
Comment:
<p>We perfectly got value for our reference point. It indicates that the process is properly working.</p>

Conclusion:

In this experiment, we successfully implemented the Lagrange Interpolation Method to approximate function values based on given discrete data points. By constructing Lagrange basis polynomials, we were able to estimate unknown function values without requiring an explicit

mathematical equation. The method was effectively implemented in MATLAB, demonstrating its application in numerical analysis, engineering, and scientific computing. Overall, this experiment provided a practical understanding of polynomial interpolation techniques and reinforced the importance of numerical methods in real-world applications, such as data fitting, computer graphics, and signal processing.

Reference:

1. Numerical Method by E Balagurusamy
2. MathWorks - MATLAB Documentation on Interpolation
<https://www.mathworks.com/help/matlab/ref/interp1.html>