1 Experiment No. 8

2 Experiment Title

Finding intermediate points using Lagrange Interpolating Polynomial.

3 Objective

The objectives of this lab are:

- To know how to fit an n^{th} order Lagrange interpolating polynomial through (n+1) data points..
- To find intermediate values in between tabulated data points using Lagrange interpolating polynomial through MATLAB programming.

4 Lagrange Interpolation

Lagrange interpolation is a polynomial interpolation technique used to estimate intermediate data points within the range of a discrete set of known data points. The theory behind the Lagrange interpolation method is based on constructing a polynomial that passes through a given set of points (x_i, y_i). This is especially useful in numerical analysis for interpolation.

4.1 Definition

Given $n+1$ data points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, the Lagrange interpolating polynomial $P(x)$ is expressed as:

$$P(x) = \sum_{i=0}^{n} y_i L_i(x) \tag{1}$$

where $L_i(x)$ are the Lagrange basis polynomials defined as:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{2}$$

4.2 Example

Suppose three data points are given: $(x_0, y_0) = (1, 2)$, $(x_1, y_1) = (3, 4)$, and $(x_2, y_2) = (4, 3)$.

1. Compute the basis polynomials $L_0(x)$, $L_1(x)$, and $L_2(x)$:

$$\begin{aligned}
L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{(x-3)(x-4)}{6} \\
L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x-1)(x-4)}{(3-1)(3-4)} = -\frac{(x-1)(x-4)}{2} \\
L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{(x-1)(x-3)}{3}
\end{aligned}$$

2. Form $P(x)$:

$$P(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \tag{3}$$

3. Simplify $P(x)$ to get the final polynomial if required.

4.3 Algorithm

Input: Data points $x = [x_0, x_1, ..., x_n]$ and $y = [y_0, y_1, ..., y_n]$
Output: Simplified Lagrange interpolating polynomial $P(X)$ and its value at a given X_value

1. Initialize polynomial $P \leftarrow 0$
2. Define symbolic variable X
3. Let $n \leftarrow$ length(x)
4. **For** $i = 1$ to n **do**:
   (a) Initialize Lagrange basis polynomial $L_i \leftarrow 1$
   (b) **For** $j = 1$ to n **do**:
      - **If** $i \neq j$ **then**
        Update $L_i \leftarrow L_i \times \frac{X - x[j]}{x[i] - x[j]}$
   (c) Update polynomial $P \leftarrow P + y[i] \times L_i$
5. Simplify the polynomial: $P \leftarrow$ simplify(P)
6. Display `"Lagrange Interpolating Polynomial:"`, P
7. **Input:** X_value (specific value to evaluate P)
8. Compute $P\_value \leftarrow$ double(substitute P with $X = X\_value$)

4.4 Finding intermediate points using Lagrange Interpolating Polynomial

4.4.1 MATLAB Code:

```
x = [1, 4, 6];
y = [0, log(4), log(6)];
v = 2;
n = 3;
L = zeros(1, n);
for i = 1:n
Li = 1;
for j = 1:n
if i ~= j
Li = Li * ((v - x(j)) / (x(i) - x(j)));
end
end
L(i) = Li;
end
f = y(1) * L(1) + y(2) * L(2) + y(3) * L(3)
```

Listing 1: Lagrange Interpolating Polynomial in MATLAB.

4.5 Result Shown in Command Window

```
f = 0.5658
```

Listing 2: Command Window for Finding intermediate points using Lagrange Interpolating Polynomial

5 Discussion

In this experiment, we successfully applied the Lagrange Interpolating Polynomial to estimate intermediate values between known data points. The Lagrange method constructs a polynomial that passes through all given data points, and it is particularly useful when only a few data points are available and an estimation is needed at a point not explicitly listed.
We selected three known points and used MATLAB to implement the Lagrange interpolation formula. The program correctly calculated the Lagrange basis polynomials and combined them to evaluate the function value at a specific point, $x = 2$. The result obtained was approximately $f = 0.5658$ for $\ln(2)$ and expected natural for $\ln(2) \approx 0.6931$ which aligns with the expected natural logarithmic value between $\ln(1) = 0$ and $\ln(4) \approx 1.3863$.