

**Discovering Knowledge**

**Numerical Analysis**

**Report**

**Lagrange's Interpolation for Data Reconstruction**

**Group Members**

|  |  |
| --- | --- |
| **Enrollment #** | **Name** |
| 02-131202-054 | Muhammad Abdullah |
| 02-131202-036 | Jawad Ahmed |
| 02-131202-027 | Tahir Zaman |

**Submitted to**

Engr. Rahemeen

**Department of Software Engineering**

Bahria University Karachi Campus

**Introduction:**

In various scientific and engineering fields, data is often collected with missing values or irregular intervals. The ability to accurately reconstruct missing data is crucial for making informed decisions and predictions. Lagrange's Interpolation, a numerical method, presents an effective solution to this problem by constructing polynomials that pass through given data points. This report aims to provide a comprehensive understanding of the principles and practical applications of Lagrange's Interpolation.

**Problem statement:**

In various fields like signal processing, data analysis, and computer graphics, there's often a need to estimate values between given data points or reconstruct missing data. Missing data can occur due to mistakes in data entry or data collection processes. This may include typos, misinterpretations, or oversights during data entry. Problems with measurement instruments or data collection tools can contribute to missing data. Malfunctioning equipment or incomplete recording of information can result in missing values. Errors in data processing, transformation, or integration can result in missing values. This may occur during the cleaning or merging of datasets.

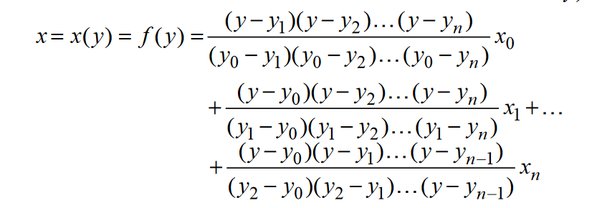
**Lagrange's Interpolation Overview:**

Lagrange's Interpolation is a numerical technique used to construct a polynomial that passes through a given set of data points. Its simplicity and versatility make it a popular choice for data reconstruction tasks. The method involves the use of Lagrange polynomials, which enable the creation of a continuous function representing the underlying trends in the data.

**Mathematical Basis - Lagrange Polynomials:**

Lagrange polynomials are the foundation of Lagrange's Interpolation. These polynomials are designed to interpolate a function or set of data points. The construction involves a weighted sum of terms, where each term corresponds to a data point. This weighted sum ensures that the resulting polynomial passes through all given data points.

**Interpolation Formula:**



**Advantages of Lagrange’s Interpolation:**

1. **Simplicity and Ease of Implementation:** Lagrange's interpolation is a simple polynomial interpolation method that can be easily implemented without the need for sophisticated algorithms or extensive computational resources.
2. **Precision and Accuracy:** If the underlying relationship between data points is well-understood and can be accurately represented by a polynomial, Lagrange's interpolation may provide precise and accurate results.
3. **Interpretability:** Lagrange's interpolation provides a clear mathematical formula (polynomial) that represents the interpolated function. This can be advantageous in situations where interpretability and understanding of the underlying model are crucial.
4. **Computational Efficiency:** Lagrange's interpolation can be computationally more efficient for small datasets, as it directly calculates the coefficients of the interpolating polynomial.

**Implementation:**

* **Find the missing values using Lagrange’s Interpolation.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Y** | **32.0** | **?** | **35.4** | **36.8** | **35.1** | **?** | **43.4** | **49.8** |

**Code:**

import numpy as np

# Define a class to represent data points with x and y values

class DataPoint:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

# Function for Lagrange Interpolation

def lagrange\_interpolation(data\_points, x\_value):

result = 0.0

# Iterate through each data point

for i in range(len(data\_points)):

term = data\_points[i].y

# Calculate the Lagrange term for the current data point

for j in range(len(data\_points)):

if j != i:

term \*= (x\_value - data\_points[j].x) / (data\_points[i].x - data\_points[j].x)

# Add the term to the result

result += term

return result

if \_\_name\_\_ == "\_\_main\_\_":

# Provide x and y values as arrays

x\_values = [1, 3, 4, 5, 7, 8]

y\_values = [32.0, 35.4, 36.8, 35.1, 43.4, 49.8]

# Create DataPoint instances based on the arrays

data\_points = [DataPoint(x, y) for x, y in zip(x\_values, y\_values)]

# Input the x value for interpolation

x\_value = float(input("Enter the x value for interpolation: "))

# Calculate and print the interpolated value

interpolated\_value = np.round(lagrange\_interpolation(data\_points, x\_value), decimals=1)

print(f"Interpolated value at {x\_value} is: {interpolated\_value}")

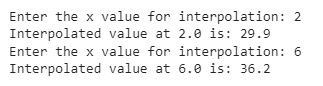
x\_value = float(input("Enter the x value for interpolation: "))

# Calculate and print the interpolated value

interpolated\_value = np.round(lagrange\_interpolation(data\_points, x\_value), decimals=1)

print(f"Interpolated value at {x\_value} is: {interpolated\_value}")

**Output:**



**Applications of Lagrange’s Interpolation:**

1. Numerical Analysis: Lagrange interpolation is often used in numerical analysis for approximating values between known data points. It helps in constructing a polynomial that passes through given points, enabling the estimation of intermediate values.
2. Computer Graphics: Lagrange interpolation is used in computer graphics for tasks such as curve fitting and surface interpolation. It helps generate smooth curves or surfaces based on a set of control points.
3. Signal Processing: In signal processing, Lagrange interpolation is employed for tasks like signal reconstruction. It is used to estimate missing or irregularly sampled data points in a signal.
4. Physics and Engineering: Lagrange interpolation is applied in physics and engineering for data analysis and modeling. It helps create polynomial functions that represent experimental or observational data.
5. Control Systems: Control systems often involve working with discrete data points. Lagrange interpolation can be used to interpolate values in control systems, aiding in the design and analysis of controllers.
6. Finance: Lagrange interpolation may be used in finance for tasks such as pricing options or interpolating yield curves. It can help estimate values between observed market data points.
7. Geographical Information Systems (GIS): GIS applications use Lagrange interpolation for spatial data analysis. It can be applied to interpolate elevation or other geographic attributes between known data points.
8. Image Processing: Lagrange interpolation is utilized in image processing for tasks like image resizing and reconstruction. It helps fill in the gaps when scaling images to different dimensions.
9. Circuit Design: In electrical engineering and circuit design, Lagrange interpolation can be applied to approximate values in circuits or to design filters.
10. Economics: Lagrange interpolation may be used in economic modeling to estimate values between observed economic data points, facilitating analysis and prediction.