Lab Manual: Numerical Analysis



**Lab Manual for Numerical Analysis**

Lab No. 6

INTERPOLATION - NEWTON’S DIVIDED DIFFERENCE

FORMULA

1

Software Engineering Department Bahria University (Karachi Campus)

Lab Manual: Numerical Analysis



**BAHRIA UNIVERSITY KARACHI CAMPUS**

**Department of Software Engineering**

# NUMERICAL ANALYSIS LAB EXPERIMENT # 6

Interpolation - Newton’s Divided Difference Formula

## OBJECTIVE:

This lab aims to introduce students to **interpolation principles**, with a focus on Newton's Divided Difference Formula, in order to equip them with practical skills for estimating intermediate values within scientific and engineering contexts.

# Introduction

**Interpolation**, as discussed in previous lab a fundamental concept in mathematics and data analysis, holds a pivotal role in estimating values between established data points. It is essential for making informed decisions and predictions, and we've previously explored its core principles in our academic pursuits.

In our prior chapter, we have extensively delved into two key interpolation formulas that are integral to our understanding: Newton's Backward Formula and Newton's Forward Formula. These formulas prove to be invaluable tools for approximating intermediate values within datasets. Newton's Backward Formula is particularly advantageous when we encounter situations marked by decreasing intervals, while Newton's Forward Formula shines when we work with increasing intervals. Collectively, these formulas provide us with a versatile and robust toolkit for performing interpolation with precision and efficiency. Through these formulas, we gain the ability to make data-driven decisions and predictions, enabling us to navigate the intricate terrain of data analysis and mathematical modeling with confidence and accuracy.

In this lab, we will further enhance our understanding of interpolation by exploring two additional techniques: Newton’s Divided Difference Formula.

# Implementation of Newton’s Divided Difference Formula

This section will provide a detailed discussion and implementation of **Newton's Divided Difference Formula**, offering a comprehensive understanding of these interpolation techniques and their practical applications.

2

Software Engineering Department Bahria University (Karachi Campus)

Lab Manual: Numerical Analysis



## Newton’s Divided Difference Formula

**Newton's divided difference interpolation** formula, is a method for approximating a **polynomial** that passes through a set of given data points, particularly when the intervals between the data points are **not uniform**. This interpolation technique is named after Sir Isaac Newton and is widely used in numerical analysis and mathematics.

The formula allows for the estimation of function values at intermediate points between the data points. It is constructed using the divided difference coefficients, which are calculated recursively based on the function values at the data points. The polynomial created with these coefficients can be used for interpolation to predict function values at specific locations within the range of the data.

The formula for Newton's divided difference Interpolation is:

𝒇(𝒙) = 𝒇(𝒙𝟎) + (𝒙 − 𝒙𝟎)𝒇[𝒙𝟎, 𝒙𝟏] + (𝒙 − 𝒙𝟎)(𝒙 − 𝒙𝟏)𝒇[𝒙𝟎, 𝒙𝟏, 𝒙𝟐] + ⋯

+ (𝒙 − 𝒙𝟎)(𝒙 − 𝒙𝟏) … . (𝒙 − 𝒙𝒌−𝟏)𝒇[𝒙𝟎, 𝒙𝟏, 𝒙𝟐, … 𝒙𝒌]

Where;

- ***f[xo , x1]*** *=* 𝒇(𝒙𝟏)−𝒇(𝒙𝟎)

𝒙𝟏−𝒙𝟎

- ***f[xo , x1 , x2]*** *=* 𝒇(𝒙𝟏 , 𝒙𝟐)−𝒇(𝒙𝟎 , 𝒙𝟏)

𝒙𝟐−𝒙𝟎

and so on.

## Implementation in Python

# Function to calculate the product term def calculate\_product\_term(i, value, x):

product = 1

for j in range(i):

product = product \* (value - x[j]) return product

# Function for calculating the divided difference table def calculate\_divided\_difference\_table(x, y, n):

for i in range(1, n):

for j in range(n - i):

y[j][i] = ((y[j][i-1] - y[j+1][i-1])/(x[j] - x[i+j]))

return y

# Function for applying Newton's divided difference formula def apply\_newton\_formula(value, x, y, n):

result = y[0][0]

for i in range(1, n): result=result+(calculate\_product\_term(i,value, x)\*y[0][i])

return result

3

Software Engineering Department Bahria University (Karachi Campus)

Lab Manual: Numerical Analysis



# Function for displaying the divided difference table def display\_divided\_difference\_table(y, n):

for i in range(n):

for j in range(n - i): print(round(y[i][j], 4), "\t", end=" ")

print("")

# User input: Number of inputs

n = int(input("Enter the number of inputs: ")) x = []

y = [[0 for i in range(n)] for j in range(n)]

# Input data points for i in range(n):

x\_val = float(input(f"Enter x[{i}]: "))

y\_val = float(input(f"Enter y[{i}]: ")) x.append(x\_val)

y[i][0] = y\_val

# Calculate the divided difference table

y = calculate\_divided\_difference\_table(x, y, n)

# Display the divided difference table print("\nDivided Difference Table:") display\_divided\_difference\_table(y, n)

# User input: Value to interpolate

value = float(input("\nEnter the value to interpolate: "))

# Calculate and display the interpolated value interpolated\_value = apply\_newton\_formula(value, x, y, n) print("\nInterpolated value at", value, "is approximately", round(interpolated\_value, 4))

4

Software Engineering Department Bahria University (Karachi Campus)

# Lab Tasks

1. Write a python program implementing the newton’s difference formula that considers the following data points, and
   1. Find the value of y at x = 301
   2. Find the value of y at any given user input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 300 | 304 | 305 | 307 |
| y | 2.4771 | 2.4829 | 2.4843 | 2.4871 |

1. Write a Python program that implements Newton's difference formula, reads the CSV file 'salary\_data,' and considers 'years of experience' as x and 'salary' as y. Then, upon user input for a specific number of years of experience, it provides an output of the salary they would receive.

***(Note: the file has been uploaded in miscellaneous section)***