

Numerical Methods

Lab 2 [Polynomial Interpolation]

All questions are from C02

1 Introduction

Part 1: Representing a Polynomial

Polynomials are a function of the following format:

$$p(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n,$$

where $[a_0, a_1, \dots, a_n]$ are called coefficients and n (called the degree or order) is a non-negative integer.

This can also be written as:

$$y = f(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_nx^n$$

Example,

$$y = 1 + 2x^2 + 5x^4$$

is a polynomial of order 4 ($= n$) with $n + 1$ coefficients $a_0 = 1, a_1 = 0, a_2 = 2, a_3 = 0, a_4 = 5$

1.1 Before performing the tasks

1. Open the Colab file shared in BUX.
2. Create a copy of that shared file.
3. Rename the Colab filename using the format
Name-ID-Lab Section

1.2 Task 1 - 1 Marks

You need to set up a Polynomial using the given list of coefficients [1, 0, 2, 0, 5] and use that to approximate the value of Pn(6).

Hint: Set up a Loop to traverse through.

Also, using the same polynomial, determine the approximate values for [1.0, 2.0, 3.0, 4.0, 5.0, 6.0]

1.3 Task 2 - 1.5 Marks

Now, in the Colab file, you will see a Polynomial Class. The constructor is defined for your purpose.

You need to write the code for Constructor to make the object callable. You will have to remove the "raise NotImplementedError()" in the call. You can cross-check whether you have done it successfully by creating an instance of the class.

1.4 Task 3 - 1.5 Marks

Part 2: Polynomial Interpolation (Vandermonde Matrix Method) If we have $n+1$ node, that is, $\{(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ that satisfies a polynomial of order n , it can be written as:

$$a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0$$

$$a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1$$

$$a_0 + a_1x_2 + a_2x_2^2 + \dots + a_nx_2^n = y_2$$

...

$$a_0 + a_1x_{n-1} + a_2x_{n-1}^2 + \dots + a_nx_{n-1}^n = y_{n-1}$$

Here, $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ is called the fitted polynomial of the given data points (nodes). Using this polynomial to find the y_k corresponding to an x_k with the range of the given nodes is called polynomial interpolation.

In matrix form, the equations can be written as

$$Xa = y,$$

From this, we can solve for a using

$$a = X^{-1}y.$$

So, using the Colab file shared, here you need to implement a function that takes a discrete x and y array and returns a Polynomial object (the one we just implemented). This polynomial object can be used to calculate y for any other value of x (not in that list) within the range.

1.5 Daily Evaluation - 4 marks

Students have learned to represent polynomial interpolation using various approaches, including lists, classes, and matrix methods. They are now required to apply this understanding through a set of implementation exercises, which will be provided separately.