Computer Aided Numerical Methods – I

Report on Assignment – 3

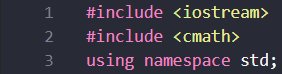
Problem Statement:

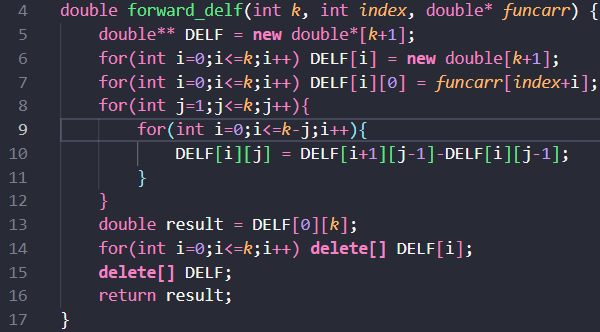
The following table gives equi-spaced data of . Compute error at , where .

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Explanation of Code:

0 – Preprocessing

This program has been coded in C++ language. Hence, the main library ‘iostream’ has been included to allow use of the basic functions, control-flow statements, etc. of C++. Here the library ‘cmath’ has also been included to allow the use of in-built math functions in C++. ‘Using namespace std;’ prevents having to prefix input and output functions with “std::’.

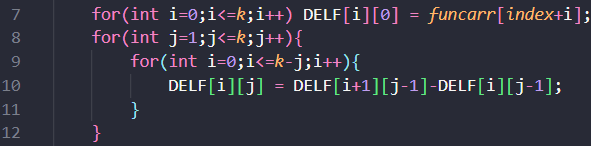
1 – Forward Difference Function

This function takes the order of difference, , array consisting of values which we use to find order difference, , and the index value corresponding to the element for which we find the order difference, as inputs and returns order difference at index , i.e. .

Functionality:



1. Creation of Forward Difference Table: Initializes and allocates memory for the Forward Difference Table array of size , .



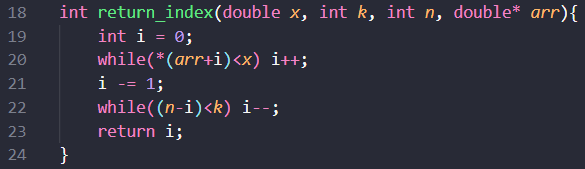
1. Building the Forward Difference Table: We initially set the first column of our table to the elements of array starting from initial index to index . We then fill the rest of the table using the recursive formula: and in the Forward Difference Table , this formula becomes: (Here, we let for the sake of ease of computation) Thus, we get the final Table as:

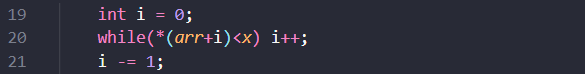
A computer screen shot of a math equation

AI-generated content may be incorrect.

1. Returning Value of required Forward Difference and Freeing Memory: Clearly from the table, we can get by returning . Then we use ‘delete’ to free up the memory of table as we have no need for it anymore.

2 – Returning Index Function

This function takes an input , length of required subarray of interpolation points , an array , and length of , , and return the starting index to the subarray of length which we use for our interpolation, .

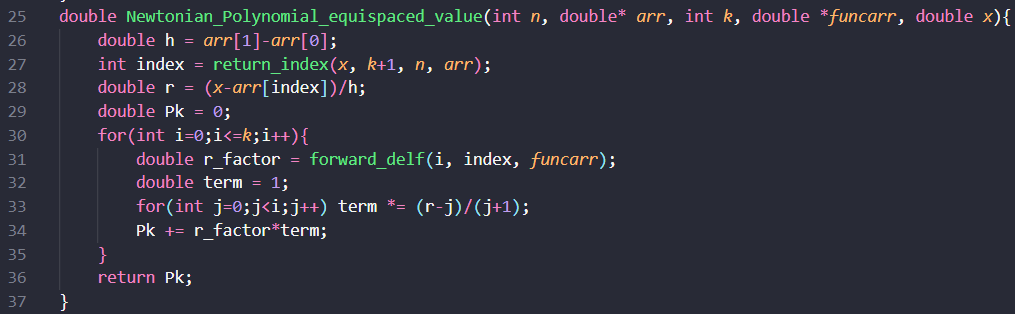
Functionality:

1. Initialization and Forward Traversal: First, we set index and then move forward in the array until we find the index such that , and then move back one step to get the closest left-hand point.



1. Backward Traversal: We check if the remaining points from that index can form a subarray of length . If not, we move further left until we can find a valid starting point for the subarray and return the final .

3 – Newtonian Polynomial Function (for equi-spaced values)

This function takes total number of points, , the array of length containing input values , , order of the Newtonian polynomial, array consisting of output values , and the value for which we compute the interpolated output value using the equi-spaced data as inputs and returns the value of interpolated Newtonian polynomial of order at .

The general formula for Newtonian polynomial of order is given by:

Where is corresponding output to and .

( is the step size, i.e., )

Functionality:



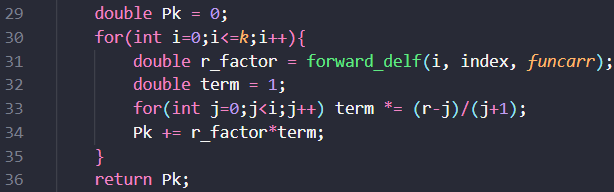
1. Step-Size Calculation: Since the data is equi-spaced, step size is computed.



1. Index Calculation**:** Calls the function to find the starting index of the subarray of length consisting of values used for interpolation.



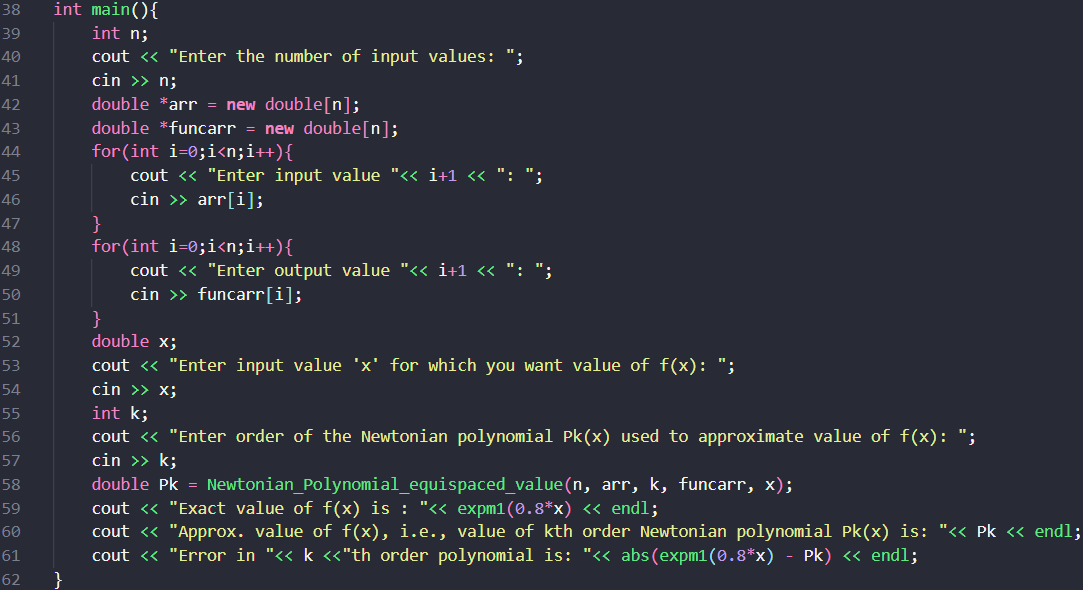
1. **Computing value of r:** , which is used in Newton's forward interpolation formula.



1. Finding the interpolated value of , and returning its value:

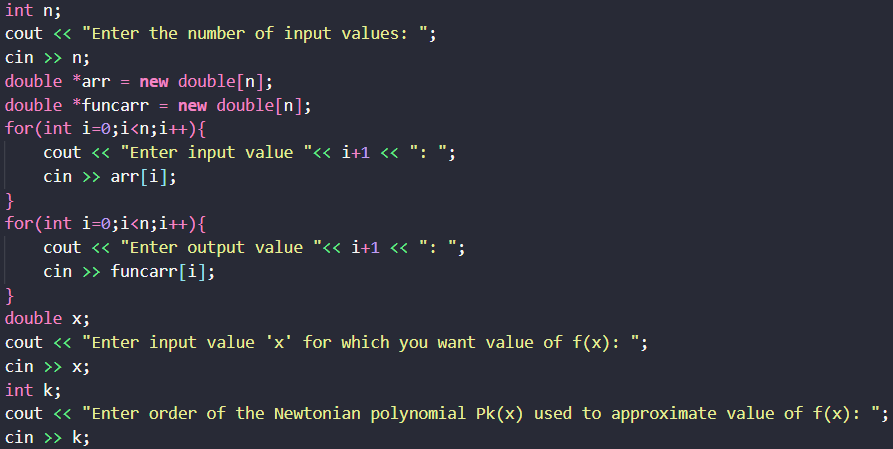
* First, we set .
* A loop runs from to to compute and sum all terms of the Newtonian polynomial:
  + Calls to get .
  + Computes in the inner loop.
  + Multiplies with forward difference and adds to result .
* Finally, the function returns the computed value .

4 – Main Function

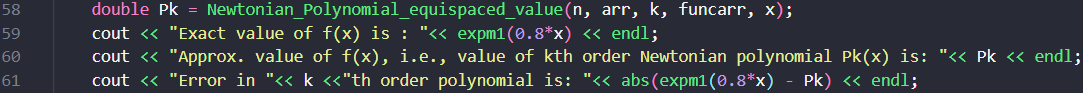


The function serves as the driver of the program and handles all user interaction and function calls.

Functionality:



1. Taking Input: The user is prompted to input the total number of data points , all input values and corresponding output values the value of for which we find the corresponding interpolated output value and order of interpolated Newtonian polynomial .



1. Computing and Printing the Output:

* We find by calling the function and set it to variable .
* The exact value of is calculated using the function
* We then calculate error as absolute difference of exact value and approximate value, i.e.,
* Finally, the exact value of , approximate value of , i.e., and error in approximation are printed.

Solution to Assignment – 3:

* A close up of a sign

  AI-generated content may be incorrect.Error in approximation at and order, :
* ****Error in approximation at and order, :
* A white text on a black background

  AI-generated content may be incorrect.Error in approximation at and order, :