Numerical Analysis

Assignment #3



Submitted to: Mam Sadaf Khalid

Date: May 31, 2021

Submitted by

Bazila Afridi (008) Saad Iqbal (030) Sarmad Ashfaq Chaudhary (033) Muhammad Hayyan Khan (042) Manal Talat (046)

Section: BSE-6A

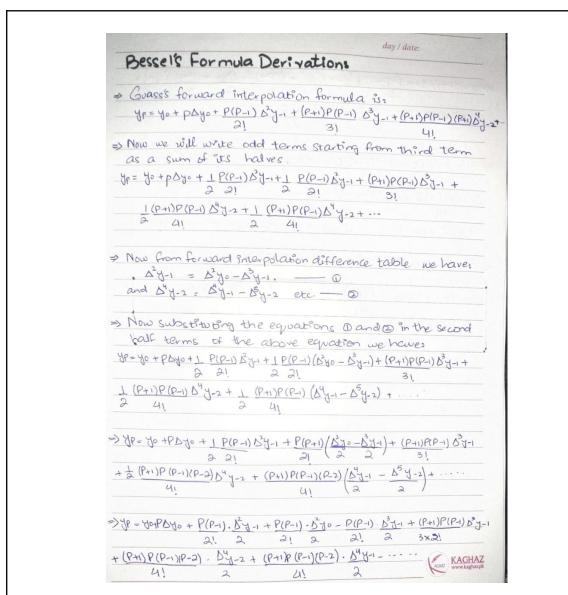
DEPARTMENT OF SOFTWARE ENGINEERING
BAHRIA UNIVERSITY ISLAMABAD

Implement the concept of Numerical Differentiation using programming techniques of C++. Your programming solution should cater the following interpolation formulae:

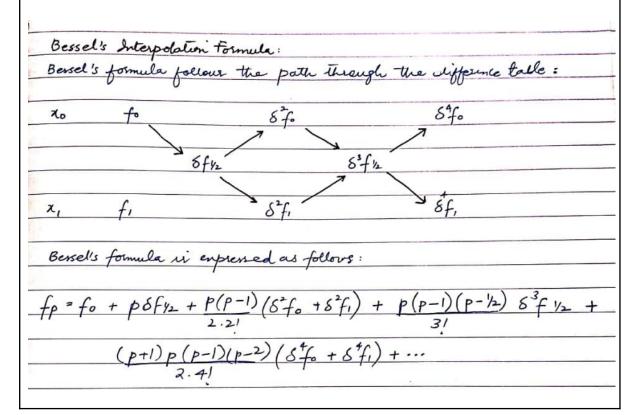
- 1. Stirling's formula.
- 2. Bessel's formula.
- 3. Everett's formula.
- 4. Gauss Forward difference formula.
- 5. Gauss Backward difference formula.

Formula (assigned): Bessel's Interpolation

Formula Derivation:



$$\Rightarrow \forall r = \forall 0 + PD + 0 + P(P-1) / D^{2}_{3-1} + D^{2}_{30}) + \frac{P(P-1)}{2!} / \frac{D^{2}_{3-1} + (P+1)}{3!} D^{2}_{3-1} + \frac{P(P+1)}{3!} D^{2}_{3-1} + \frac{P(P+1)}{3!} D^{2}_{3-1} + \frac{P(P-1)}{3!} D^{2}_{3-1} + \frac{P(P-1)}{3!} D^{2}_{3-1} + \frac{P(P-1)}{3!} D^{2}_{3-1} D^{2}_{3-1} + \frac{P(P-1)}{3!} D^{2}_{3-1} D^{2}_{3-$$



(View on Github)

Code:

bessel.cpp

```
#include <iostream>
#include <time.h>
using namespace std;
class BesselsInterpolation
private:
  double xp, xo, p;
  // dynamic arrays for difference table
  double *x, *fx;
  double *delta, *delta sq, *delta cube, *delta quad;
public:
  BesselsInterpolation(double xp)
       this->xp = xp;
      this->n = 5;
      x = new double[n];
       fx = new double[n];
      generateValues();
  BesselsInterpolation(double xp, int num)
      this->xp = xp;
      this->n = num;
       x = new double[n];
      fx = new double[n];
       getInputValues();
```

```
void generateValues()
    srand(1);
       fx[i] = rand() % 10;
// fill difference table from user input
void getInputValues()
        cout << "Enter value for x[" << i << "]: ";</pre>
       cout << "Enter value for fx[" << i << "]: ";</pre>
       cin >> fx[i];
void buildDifferenceTable()
   delta = new double[n];
   delta_sq = new double[n];
    delta_cube = new double[n];
    delta_quad = new double[n];
           delta[i] = fx[i] - fx[i - 1];
```

```
delta_sq[j] = delta[j] - delta[j - 1];
        delta cube[k] = delta sq[k] - delta sq[k - 1];
        delta_quad[1] = delta_cube[1] - delta_cube[1 - 1];
    cout << "\t\t" << fx[i];</pre>
cout << "\ndelta: ";</pre>
for (int i = 1; i < n; i++)
    cout << "\t\t" << delta[i];</pre>
cout << "\ndelta^2: ";</pre>
    cout << "\t\t" << delta_sq[i];</pre>
cout << "\ndelta^3: ";</pre>
```

```
cout << "\t\t" << delta_cube[i];</pre>
    cout << "\ndelta^4: ";</pre>
        cout << "\t\t" << delta_quad[i];</pre>
double findInterval()
   double h = x[1] - x[0];
   return h;
// calculate p for a specified origin
double calculateP(int xo)
   return ((xp - xo) / findInterval());
void findOrigin()
        p = calculateP(x[i]);
        origin_index = i;
            break;
        else
    cout << "\n\nOrigin (selected): " << xo;</pre>
    cout << "\nOrigin Index: " << origin_index;</pre>
    cout << "\nValue of P at Origin: " << p;</pre>
```

```
int factorial(int n)
       return n * factorial(n - 1);
       return 1;
double calcFirstTerm()
   return fx[origin_index]; // value of f at origin
// calculate bessel interpolation's second term
double calcSecondTerm()
   double sigma1by2 = delta[origin_index];
   return (p * sigma1by2);
// calculate bessel interpolation's third term
double calcThirdTerm()
   double sigmaSq0 = delta_sq[origin_index];
    double sigmaSq1 = delta_sq[origin_index + 1];
   return ((p * (p - 1) * (sigmaSq0 + sigmaSq1)) / (2 * factorial(2)));
double calcFourthTerm()
    double sigmaCube1by2 = delta_cube[origin_index];
    return ((p * (p - 1) * (p - 1 / 2) * sigmaCube1by2) / (factorial(3)));
```

main.cpp

```
{
    BesselsInterpolation BI(xp);
    BI.buildDifferenceTable();
    BI.findOrigin();
    BI.findBesselResult();
}
else
{
    int num;
    cout << "\nNo. of terms in table: ";
    cin >> num;

    BesselsInterpolation BI(xp, num);
    BI.buildDifferenceTable();
    BI.findOrigin();
    BI.findBesselResult();
}
return 0;
}
```

Output:

(User specified values)

```
--- Bessel's Interpolation --- Interpolate for: 27.4
Select an option:
1. Auto-generate values
Specify values
No. of terms in table: 6
Enter value for x[0]: 25
Enter value for fx[0]: 4.000
Enter value for x[1]: 26
Enter value for fx[1]: 3.846
Enter value for x[2]: 27
Enter value for fx[2]: 3.704
Enter value for x[3]: 3.704
Enter value for fx[3]: 3.571
Enter value for x[4]: 29
Enter value for fx[4]: 3.448
Enter value for x[5]: 30
Enter value for fx[5]: 3.333
                                                  26
3.846
                                                                           27
3.704
                                                                                                    3.571
                                                                                                                             3.448
                                                                                                                                                      3.333
delta:
                         -0.154
                                                   -0.142
                                                                            -0.133
                                                                                                     -0.123
                                                                                                                              -0.115
delta^2:
                                     0.012
                                                              0.009
                                                                                       0.01
                                                                                                                0.008
delta^3:
                                      -0.003
                                                                                       -0.002
                                                              0.001
delta^4:
                                     0.004
                                                              -0.003
Origin (selected): 27
Origin Index: 2
Value of P at Origin: 0.4
-> Bessel's Interpolation Result: 3.64498
(base) sarmad@Sarmads—Macbook NA Asgn 3 % □
```

(Auto-generated values)

```
--- Bessel's Interpolation ---
Interpolate for: 27.4
Select an option:
1. Auto-generate values
2. Specify values
                                  2
                                                   3
                                                                     8
                                  9
                                                    3
fx:
delta:
                 2
                                  -6
                                                                     -8
delta^2:
delta^3:
                          -8
                                           11
                                                            -13
                          19
                                           -24
delta^4:
                          -43
Origin (selected): 5
Origin Index: 4
Value of P at Origin: 22.4
-> Bessel's Interpolation Result: -249679
```

REFERENCES

(Book) Chapter 3.4.2 - Bessel's Interpolation, Numerical Analysis with C++