Numerical Methods

Report

Date of the exercise: **28/03/2019**

Exercise: **Interpolation**

Group: 2, Team:

Subsection (names):

1. Mateusz Nowotnik

2. Dawid Tomala

Solving the interpolation problem of calculating the interpolation function value using the Lagrange and Chebyshev method.

```
#undef __STRICT_ANSI__
#include <cmath>
#include <windows.h>
#include <iostream>
using namespace std;

double f(double x)
{
    return 5/(x*x+2);
}

double F(double xi[], double yi[], double wi[], int n, double x, bool finalWi)
{
    double F = 0;
    for(int i = 0; i <= n; i ++ )
    {
        wi[i] = 1;
        for(int j = 0; j<=n;j++)
        {
            if(j!=i)
                // the auxiliary polynomials
                wi[i] *= (x-xi[j])/(xi[i]-xi[j]);
        }
        F += wi[i]*yi[i];
    }

    if(finalWi == true) {
        for (int i = 0; i < n+1; i++) {
            cout << "wi[" << i << "] : "<< wi[i] << endl;
        }
    }

    return F;
}

void maxError(double xi[], double yi[], int a, int b, int n, double wi[], double x,
bool finalWi)
{
    // find max error in the interval determined by a and b with step equal to 0.01
    double err1;
    double i = a;
    double err2 = f(a) - F(xi, yi, wi, n, x, finalWi);
    while(i <= b) {
        i+=0.01;
        err1 = f(a) - F(xi, yi, wi, n, x, finalWi);
        err2 = max(err2, err1);
    }

    cout << "Max error: " << fabs(err2) << endl;
}

int main()
{
    int n;
    double a, b, x, arg;
    double inter = 0;
    bool finalWi = false;

```

```
    cout << "Enter number of values: ";
    cin >> n;

    cout << "Enter the interval: ";
    cin >> a; cin >> b;

    cout << "Enter the value of x: ";
    cin >> x;

    if ((x < a) || (x > b)) {
        cout << "\nFAIL: x is outside of the interval!\n";
        return false;
    }

    double xiLag[n+1];
    double xiCheb[n+1];
    double yiLag[n+1];
    double yiCheb[n+1];
    double wi[n+1];
    double ri[n+1];

    // step, xi[0] xi[n+1] and their function values for the Lagrange method
    inter = (b-a)/(n);
    xiLag[0] = a;
    xiLag[n+1] = b;
    yiLag[0] = f(xiLag[0]);
    yiLag[n+1] = f(xiLag[n+1]);

    xiCheb[0] = a;
    xiCheb[n+1] = b;
    yiCheb[0] = f(xiCheb[0]);
    yiCheb[n+1] = f(xiCheb[n+1]);
    arg = M_PI/(2*n+2);

    for (int i = 1; i <= n+1; i++) {
        // Lagrange xi, yi
        xiLag[i] = xiLag[i-1] + inter;
        yiLag[i] = f(xiLag[i]);

        // Chebyshev xi, yi
        xiCheb[i] = 0.5*((b-a)*cos(arg*(2*n+1-2*i))+b+a);
        yiCheb[i] = f(xiCheb[i]);
           }

           finalWi = true;

           // Task 1
           cout << "\nLagrange: " << endl;
           cout << "\nInterpolation function: " << F(xiLag, yiLag, wi, n, x, finalWi) << endl;
           cout << "Interpolated function: " << f(x) << endl << endl;

           // Task 2
           cout << "Chebyshev: " << endl;
           cout << "\nInterpolation function: " << F(xiCheb, yiCheb, wi, n, x, finalWi) << endl;
           cout << "Interpolated function: " << f(x) << endl << endl;

           finalWi = false;
           // Task 3
           maxError(xiCheb, yiCheb, a, b, n, wi, x, finalWi);

           system("PAUSE");

           return 0;
       }
```

First part of the program calculates the Lagrange's auxiliary polynomials $\boldsymbol{w_i(x)}$ and the **function values** $\boldsymbol{y_i}$ of the arguments x_i, and from that, the **interpolation function value**; having as the inputs:

- the number of nodes
- interval [a,b]
- the value x

Example:

Giving to the program following inputs:

```
Enter number of values: 9
Enter the interval: -3.2 12.8
Enter the value of x: 4.3
```

The first part of the program yields:

```
Lagrange:
wi[0] : 0.000367136
wi[1] : -0.00433078
wi[2] : 0.0251307
wi[3] : -0.106752
wi[4] : 0.892141
wi[5] : 0.249799
wi[6] : -0.0730408
wi[7] : 0.0200481
wi[8] : -0.00368653
wi[9] : 0.000323944

Interpolation function: 0.261304
Interpolated function: 0.244021
```

In the second task, for the same inputs given by the user, the program calculates roots of the Chebyshev polynomial and Chebyshev (interpolation) nodes, as well as the values of the interpolation function at the points where those nodes exist. The output is:

```
Chebyshev:
wi[0] : 0.0107885
wi[1] : -0.0399309
wi[2] : 0.0863285
wi[3] : -0.182755
wi[4] : 0.851186
wi[5] : 0.366667
wi[6] : -0.140521
wi[7] : 0.0749354
wi[8] : -0.0388598
wi[9] : 0.0121613

Interpolation function: 0.195676
Interpolated function: 0.244021
```

And for the third task, the **maximum error** is being found, subtracting the value of the interpolation function from the value of the interpolated function for the argument $\boldsymbol{x}$ in the interval determined by $\boldsymbol{a}$ and $\boldsymbol{b}$ with the step equal to $\mathbf{0.01}$, which gives the greatest absolute value of that subtraction.

Output:

```
Max error: 0.258869
```