1. Теоретичні відомості

Інтерполяцією називається знаходження проміжних значень величини за наявним дискретним набором відомих значень.

Інтерполяційна формула Гауса – формула, яка використовує в якості вузлів інтерполяції найближчі до точки x інтерполяції вузли.

Перша інтерполяційна формула Гауса використовується для інтерполяції вперед і має вигляд:

$$P(x) = y_0 + q\Delta y_0 + \frac{q^{[2]}}{2!}\Delta^2 y_{-1} + \frac{(q+1)^{[3]}}{3!}\Delta^3 y_{-1} + \frac{(q+1)^{[4]}}{4!}\Delta^4 y_{-2} + \ldots$$

$$\ldots + \frac{(q+n-1)^{[2n-1]}}{(2n-1)!}\Delta^{2n-1} y_{-(n-1)} + \frac{(q+n-1)^{[2n]}}{(2n)!}\Delta^{2n} y_{-n}$$

де $x = x_0 + qh$ $\quad q^{[m]} = q(q-1)\ldots[q-(m-1)]$

Перша інтерполяційна формула Гауса містить центральні різниці

Δy_0, $\Delta^2 y_{-1}$, $\Delta^3 y_{-1}$, $\Delta^4 y_{-2}$, $\Delta^5 y_{-2}$, $\Delta^6 y_{-3}$, …

Друга інтерполяційна формула Гауса (для інтерполяції назад) записується в такому вигляді:

$$P(x) = y_0 + q\Delta y_{-1} + \frac{(q+1)^{[2]}}{2!}\Delta^2 y_{-1} + \frac{(q+1)^{[3]}}{3!}\Delta^3 y_{-2} + \frac{(q+2)^{[4]}}{4!}\Delta^4 y_{-2} + \ldots$$

$$\ldots + \frac{(q+n-1)^{[2n-1]}}{(2n-1)!}\Delta^{2n-1} y_{-n} + \frac{(q+n)^{[2n]}}{(2n)!}\Delta^{2n} y_{-n}$$

де $x = x_0 + qh$

Формули *Гауса* застосовуються для інтерполяції в середині таблиці поблизу $x_\circ$. При цьому перша формула Гауса використовується при $x > x_\circ$, а друга − при $x < x_\circ$.

Інтерполяційна формула Стірлінга представляє собою середнє арифметичне першої та другої формул Гауса і має вигляд:

$$P_n(x) = y_0 + q\frac{\Delta y_{-1} + \Delta y_0}{2!} + \frac{q^2}{2}\Delta^2 y_{-1} + \frac{q(q^2-1^2)}{3!}\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} +$$

$$+ \frac{q^2(q^2-1^2)}{4!}\Delta^4 y_{-2} + \ldots + \frac{q(q^2-1^2)(q^2-2^2)\ldots(q^2-(n+1)^2)}{(2n)!}\cdot$$

$$\frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-(n-1)}}{2} + \frac{q^2(q^2-1^2)(q^2-2^2)\ldots(q^2-(n+1)^2)}{(2n)!}\Delta^{2n} y_{-n}$$

де $q = \frac{x - x_0}{h}$.

Ця формула використовується при інтерполяції функцій при значеннях q, близьких до 0. На практиці її застосовують при $|q| \le 0.25$.

Інтерполяційна формула Бесселя має фигляд:

$$P_n(x)=\frac{y_0+y_{-1}}{2}+(q-1/2)\Delta y_0+\frac{q(q-1)}{2!}\frac{\Delta^2 y_{-1}+\Delta^2 y_0}{2}+$$

$$+\frac{(q-1/2)q(q-1)}{3!}\Delta^3 y_{-1}+\frac{q(q-1)(q+1)(q-2)}{4!}\cdot\frac{\Delta^4 y_{-2}+\Delta^4 y_{-1}}{2}+\ldots$$

$$\ldots+\frac{q(q-1)(q+1)(q-2)(q+2)\ldots(q-n)(q+n-1)}{(2n)!}\cdot\frac{\Delta^{2n} y_{-n}+\Delta^{2n} y_{-(n-1)}}{2}+$$

$$+\frac{(q-1/2)q(q-1)(q+1)(q-2)(q+2)\ldots(q-n)(q+n-1)}{(2n+1)!}\Delta^{2n+1} y_{-n}$$

Формула Бесселя використовується для інтерполяції при значеннях q, близьких до 0,5. На практиці $0.25 < |q| < 0.75$.

У випадку, коли $q = 0.5$, формула Бесселя може бути записана у вигляді:

$$P_n(x)=\frac{y_0+y_{-1}}{2}+\frac{1}{8}\cdot\frac{\Delta^2 y_{-1}+\Delta^2 y_0}{2}+\frac{3}{128}\cdot\frac{\Delta^4 y_{-2}+\Delta^4 y_{-1}}{2}+\ldots$$

$$\ldots+(-1)^n\frac{(1\cdot 3\cdot 5\ldots(2n-1))^2}{2^{2n}(2n)!}\cdot\frac{\Delta^{2n} y_{-n}+\Delta^{2n} y_{-(n-1)}}{2}.$$

Ця формула називається *формулою інтерполяції на середину*.

Інтерполяційні формули Ньютона зручно використовувати, якщо точка інтерполяції знаходиться поблизу початку (перша формула Ньютона) або кінця таблиці (друга формула Ньютона).

Перша формула Ньютона має вигляд:

$$P_n(x)=y_0+q\Delta y_0+\frac{q(q-1)}{2!}\Delta^2 y_0+\ldots+\frac{q(q-1)\ldots(q-n+1)}{n!}\Delta^n y_0,$$

де $q=\dfrac{x-x_0}{h},\ y_i=f_i$

Друга інтерполяційна формула Ньютона має вигляд:

$$P_n(x)=y_n+q\Delta y_{n-1}+\frac{q(q+1)}{2!}\Delta^2 y_{n-2}+\ldots+\frac{q(q+1)\ldots(q+n-1)}{n!}\Delta^n y_0,$$

де $q=\dfrac{x-x_n}{h}$

Інтерполяційна формула Лагранжа має вигляд:

$$f(x) \approx L(x) = f(x_0) \cdot \frac{(x - x_1)(x - x_2) \cdot \ldots \cdot (x - x_m)}{(x_0 - x_1)(x_0 - x_2) \cdot \ldots \cdot (x_0 - x_m)} +$$
$$+ f(x_1) \cdot \frac{(x - x_0)(x - x_2)(x - x_3) \cdot \ldots \cdot (x - x_m)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3) \cdot \ldots \cdot (x_1 - x_m)} + \ldots +$$
$$+ f(x_m) \cdot \frac{(x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{m-1})}{(x_m - x_0)(x_m - x_1) \cdot \ldots \cdot (x_m - x_{m-1})} =$$
$$= \sum_{i=0}^{m} f(x_i) \cdot \frac{(x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{i-1})(x - x_{i+1}) \cdot \ldots \cdot (x - x_m)}{(x_i - x_0)(x_i - x_1) \cdot \ldots \cdot (x_i - x_{i-1})(x_i - x_{i+1}) \cdot \ldots \cdot (x_i - x_m)}.$$

Ітераційні формули засновані на послідовному використанні простих інтерполяційних схем декілька разів. Так, сутність **формули Ейткіна** у повторному використанні лінійної інтерполяції:

$$y_{i1}(x) = \frac{1}{x_i - x_0}\,[y_0(x_i - x) - y_i(x_0 - x)];$$
$$y_{i2}(x) = \frac{1}{x_i - x_1}\,[y_{11}(x_i - x) - y_{i1}(x_1 - x)];$$
$$y_{i3}(x) = \frac{1}{x_i - x_2}\,[y_{22}(x_i - x) - y_{i2}(x_2 - x)];$$
$(\cdots)$

2. Завдання

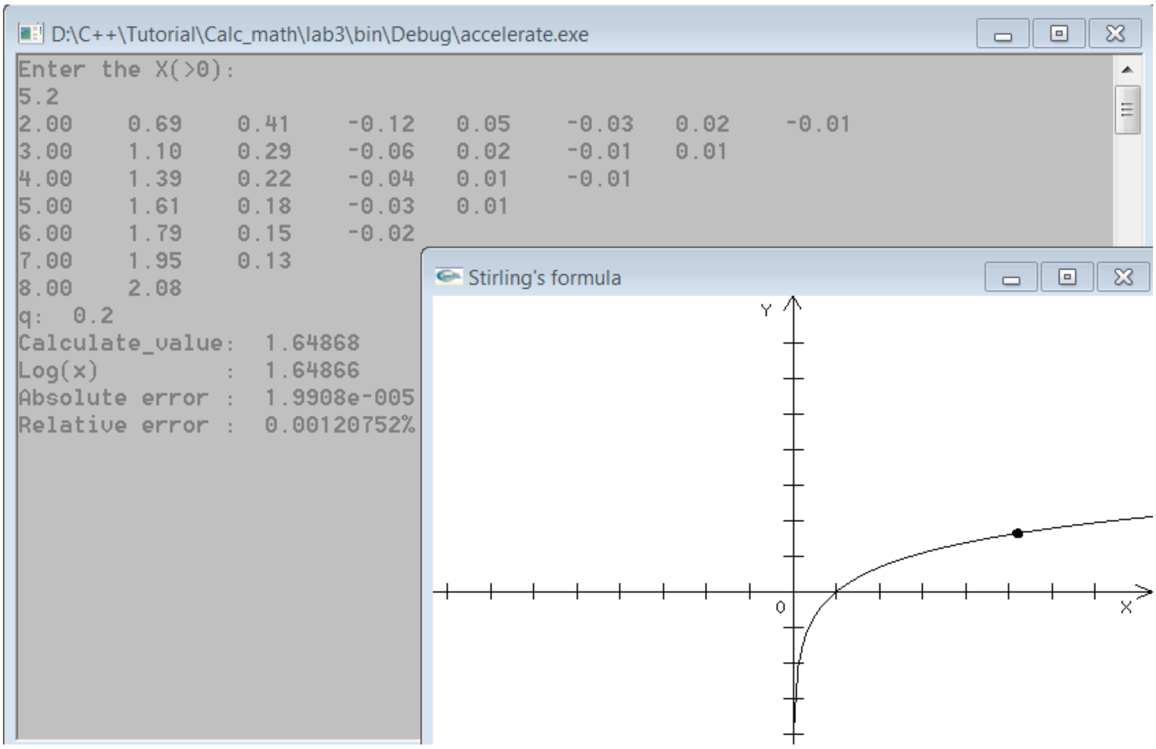
Створити програму, що буде розраховувати значення функції в точці, за методом інтерполяції Стірлінга.

3. Текст програми

```
#include <GL/glut.h>
#include <cmath>
#include <iostream>
#include <stdio.h>
#include <cstdlib>
#include "calc.h"
#include "const.h"
#include "draw.h"
using namespace std;

void Initialize (){
   glClearColor(1, 1, 1, 1.0);
   glOrtho(-width/2, width/2, -height/2, height/2, -1, 1);
}

void display(){
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(0, 0, 0);
   draw_net();      // Cartesian
   glColor3f(0, 0, 0);
   glBegin(GL_LINE_STRIP); // Graph
   for(float x = -width; x < width; x += 0.1)
      glVertex2f(x*SCL, log(x)*SCL); //library function
   glEnd();
   glColor3f(0, 0, 0); // Our Point
   drawCircle(X*SCL, Y*SCL, 4, 20);
   glutSwapBuffers();
}

int main(int argc, char **argv){
   system("color 78"); //good print
   cout << "Enter the X(>0):" << endl;
   cin >> X; //5.2
   Y = Stirling(X);
   //Initialization
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
   glutInitWindowSize(width, height);
   glutInitWindowPosition(x_pos, y_pos);
   glutCreateWindow("Stirling's formula");
   //Registration
   glutDisplayFunc(display);
   Initialize();
   glutMainLoop();
   return 0;
```

```
}



float Stirling(float x){
    int N = 7; //amount of values
    float **Y;
    float Y_val[N];
    float F = 2; // first value in our table;
    float h = 1; //step

    //enter the table
    for (int i = 0; i < N; i++){
        Y_val[i] = log(F + i * h); // Y = table values
    }

    Y = getDiffTable(Y_val, N);
    showDiffTable(F, h, Y, N);

    int mid = ceil((float)(N / 2)); //central index
    float x0 = F + h*mid;
            float P = Y[mid][0]; //Zero part from table
            float temp1, temp2, temp3;
            for (int n = 1; n <= mid; n++){
                        temp1 = Q(x, x0, h, n) / fact(2 * n - 1); //q(q^2 - i^2)/2n!/ (2n-1)!
                        temp2 = (Y[-n + mid][2 * n - 1] + Y[-(n - 1) + mid][2 * n - 1]) / 2; //average diff
                        temp3 = (Q(x, x0, h, n)*q(x, x0, h)) / fact(2 * n); // q^2(q^2 - i^2)/2n!
                        P += (temp1)*(temp2)+(temp3)*(Y[-n + mid][2 * n]); //add parts
                        //cout << (temp1)*(temp2) << "\n";
                        //cout << (temp3)*(Y[-n + mid][2 * n]) << "\n";
            }
            float ln_val = log(x);
            cout << "q:  " << q(x, x0, h) << "\n";
            cout << "Calculate_value:  " << P << "\n";
            cout << "Log(x)         :  " << ln_val << "\n";
            cout << "Absolute error :  " << abs(P - ln_val) << "\n";

    cout << "Relative error :  " << ((ln_val == 0) ? 0 : (abs(P - ln_val) / ln_val * 100)) << "%\n";

    float result = P;
    return result;
}
```

4. Приклад виконання програми



Власний розрахунок

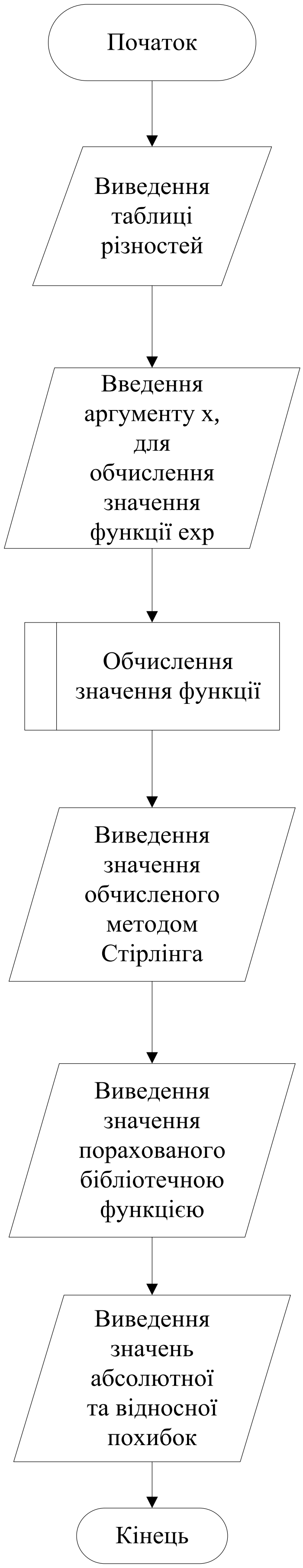
			Δy_i					
Індекс	x_i	y_i	1	2	3	4	5	6
0	2	0.69	0.41	-0.12	0.05	-0.03	0.02	-0.01
1	3	1.10	0.29	-0.06	0.02	-0.01	0.01	
2	4	1.39	0.22	-0.04	0.01	-0.01		
3	5	1.61	0.18	-0.03	0.01			
4	6	1.79	0.15	-0.02				
5	7	1.95	0.13					
6	8	2.08						

x= 5.2;
h = 1;
$q= (x-x_0)/h = (5.2-5)/1 = 0.2;$

$P(x) = 1.61 + 0.2*(0.22+0.18)/2 + (0.2)^2/2! * (-0.04) + 0.2(0.2^2 -1)/3! *(0.02+0.01)/2 * (-0.01) + 0.2(0.2^2 -1)* (0.2^2 -1)/5! *(0.02+0.01)/2 +0.2^2 (0.2^2 -1)* (0.2^2 -1)/6! * (-0.01) = 1.61 + 0.0405 * (-0.0008) - 0,000581 * 0.0000177 - 0.000077 * 0.00000263 = 1.64868$

Реальне значення: 1.64866
Абсолютна похибка: 0.00002
Відносна похибка: 0.0012%

5. Блок – схема програми



Висновок

В результаті виконання лабораторної роботи було розроблено програму для інтерполяції функції за інтерполяційною формулою Стірлінга. Значення x вводиться користувачем з клавіатури, таблиця значень функції розраховується за допомогою бібліотечної функції (вважаємо точність даної функції абсолютною для даної обмеженої числової сітки).

Значення функції натурального логарифму розраховується за допомогою додавання членів поліному. Програма розраховує значення абсолютної та відносної похибок. Отримані значення не перевищують $3*10^{-5}$ та 0,0002% відповідно у діапазоні застосування формули Стірлінга ($| q | \leq 0.25$). В залежності від вибору точок точність розрахунку змінюється (чим більша відстань від точок, по яким рахується поліном – тим більша похибка). Для доцільної інтерполяції в інших діапазонах застосовуються інші чисельні методи з розкладанням многочлена та деякі ітераційні методи, які повторно використовують більш прості інтерполяційні методи, наприклад, лінійну інтерполяцію.