Пермский национальный исследовательский политехнический университет

Кафедра информационные технологии и автоматизированные системы

Лабораторная работа №2

Вычисление функций с использованием их разложения в степенной ряд

Вариант: 9

|  |  |
| --- | --- |
|  | Выполнил:  студент группы: ИВТ-24-2б  Ваулин Артём Сергеевич  Проверил:  доцент кафедры ИТАС  О.А. Полякова |

Пермь 2024

# Постановка задачи

Для х изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x), используя ее разложение в степенной ряд в двух случаях:

а) для заданного n;

б) для заданной точности ε (ε=0.0001).

Для сравнения найти точное значение функции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | функция | диапазон  изменения аргумента | n | сумма |
| 9 |  |  | 3 |  |

# Математическая модель

1. **Для фиксированного числа членов ряда** n**:**

Формула разложения в степенной ряд:

где n - заданное число членов ряда.

1. **Для заданной точности** **:**

Здесь расчет происходит до тех пор, пока абсолютное значение каждого следующего члена ряда не станет меньше заданной точности .

Формула разложения в степенной ряд:

с условием, что добавление каждого следующего члена

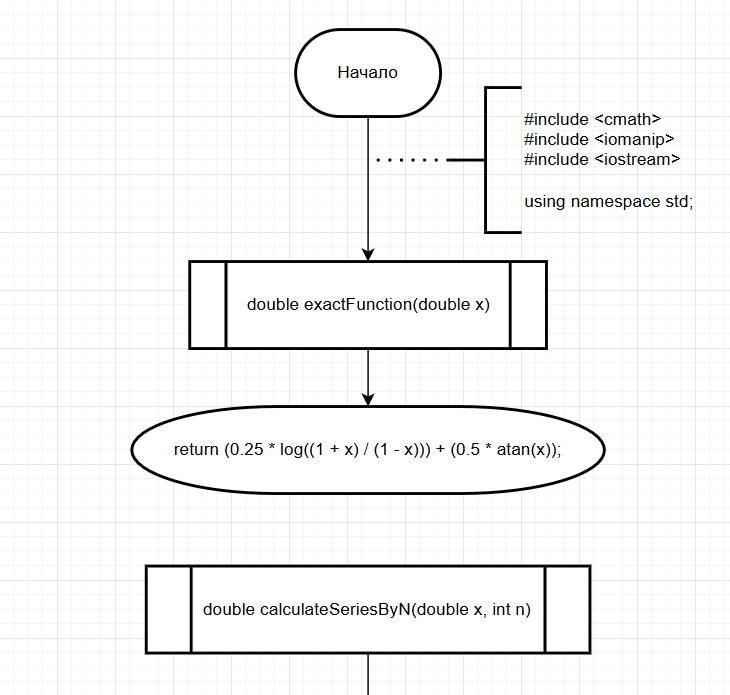
Таким образом, для каждого значения xx в диапазоне [0.1, 0.8] выполняются два расчета:

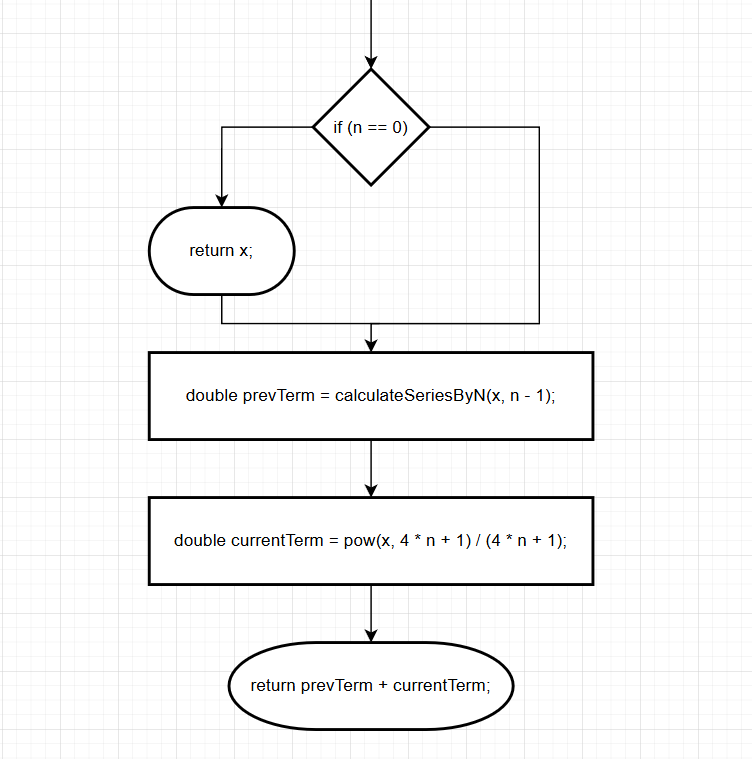
1. **Ряд с фиксированным числом членов** n:.
2. **Ряд до достижения заданной точности** .

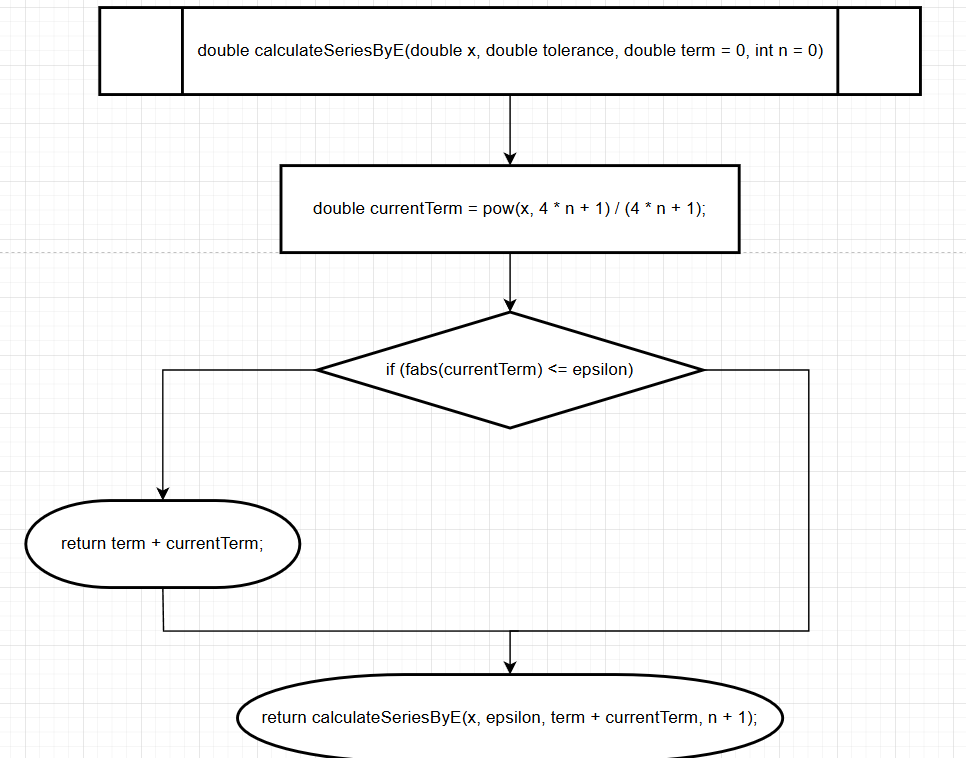
В точке x, точное значение функции вычисляется как:

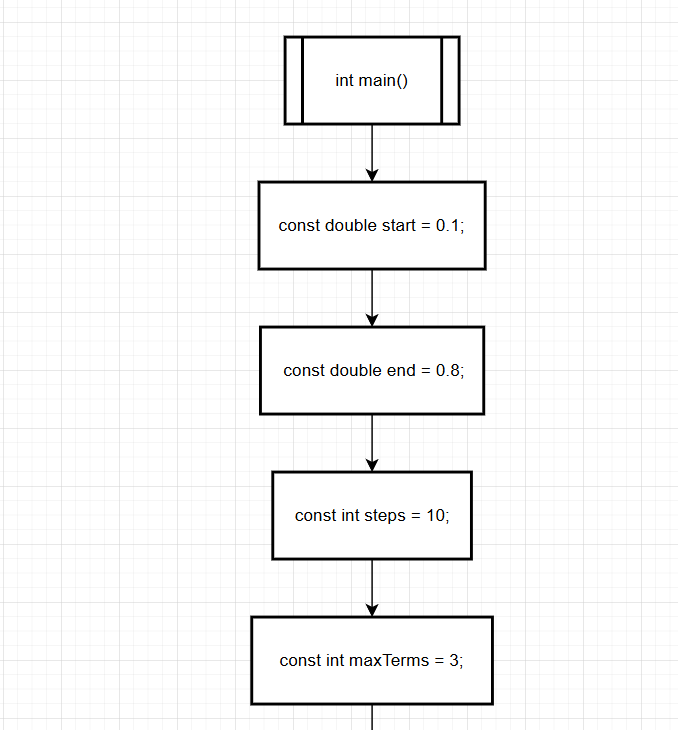
Это позволяет сравнить приближенное значение, полученное разложением в степенной ряд, с точным значением функции.

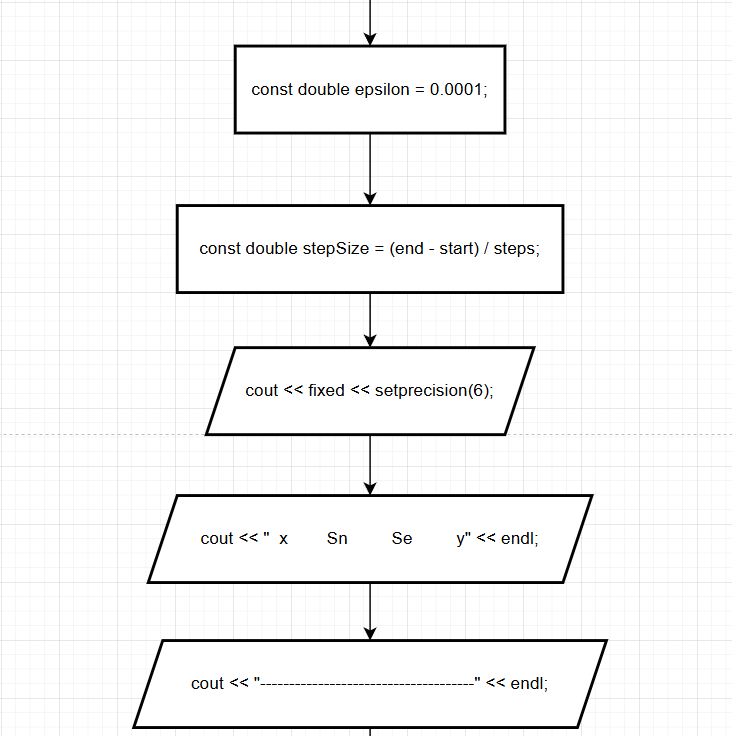
# Блок-схема

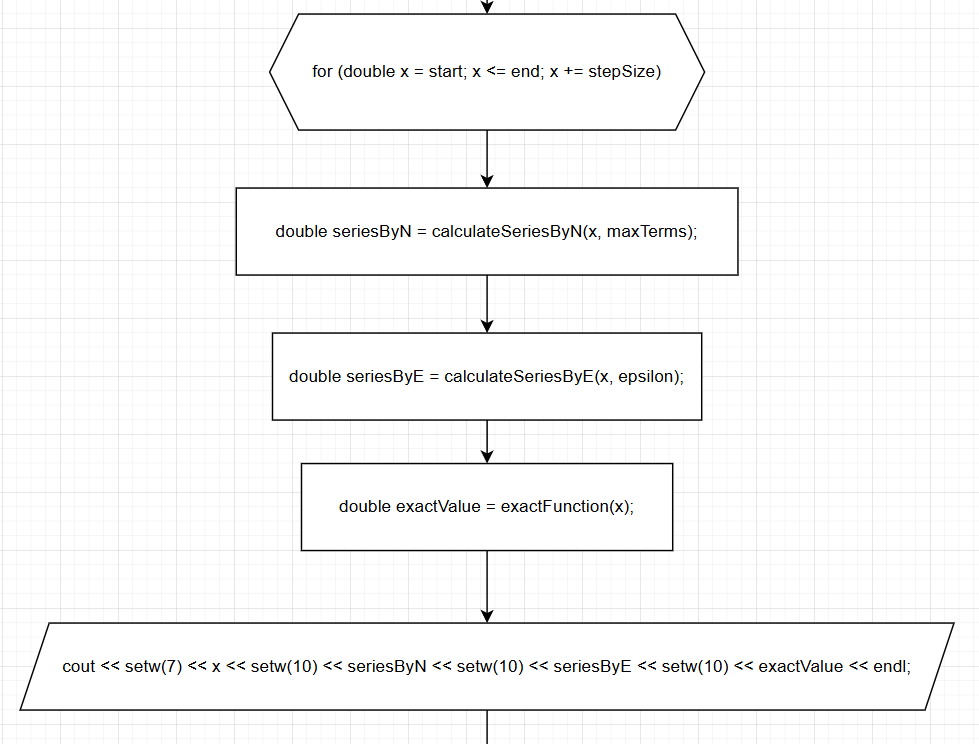


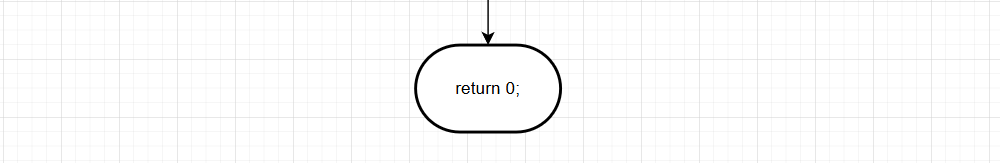












# Код программы

#include <cmath>

#include <iomanip>

#include <iostream>

using namespace std;

// Вычисление точного значения функции

double exactFunction(double x)

{

return (0.25 \* log((1 + x) / (1 - x))) + (0.5 \* atan(x));

}

// Вычисление ряда с заданным числом членов (рекурсивно)

double calculateSeriesByN(double x, int n)

{

if (n == 0)

return x; // Первый член ряда

double prevTerm = calculateSeriesByN(x, n - 1);

double currentTerm = pow(x, 4 \* n + 1) / (4 \* n + 1);

return prevTerm + currentTerm;

}

// Вычисление ряда с заданной точностью (рекурсивно)

double calculateSeriesByE(double x, double epsilon, double term = 0, int n = 0)

{

double currentTerm = pow(x, 4 \* n + 1) / (4 \* n + 1);

if (fabs(currentTerm) <= epsilon)

return term + currentTerm;

return calculateSeriesByE(x, epsilon, term + currentTerm, n + 1);

}

int main()

{

const double start = 0.1;

const double end = 0.8;

const int steps = 10;

const int maxTerms = 3;

const double epsilon = 0.0001;

const double stepSize = (end - start) / steps;

cout << fixed << setprecision(6);

cout << " x Sn Se y" << endl;

cout << "-------------------------------------" << endl;

for (double x = start; x <= end; x += stepSize)

{

double seriesByN = calculateSeriesByN(x, maxTerms);

double seriesByE = calculateSeriesByE(x, epsilon);

double exactValue = exactFunction(x);

cout << setw(7) << x

<< setw(10) << seriesByN

<< setw(10) << seriesByE

<< setw(10) << exactValue << endl;

}

return 0;

}

# Результат

