Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**Лабораторная работа №2**

**Вычисление функций с использованием их разложения в степенной ряд**

**Вариант: 4**

Выполнил работу

студент группы ИВТ-24-2б

К.Д.Горбушин

Проверил

Доцент кафедры ИТАС

О.А.Полякова

Пермь, 2024

**Содержание**

1. Цель
2. Постановка задачи
3. Математическая модель
4. Блок-схема
5. Код программы
6. Результат вычислений
7. Ссылка на GitHub
8. **Цель:** Практика в организации итерационных и арифметических циклов.
9. **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.



# 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

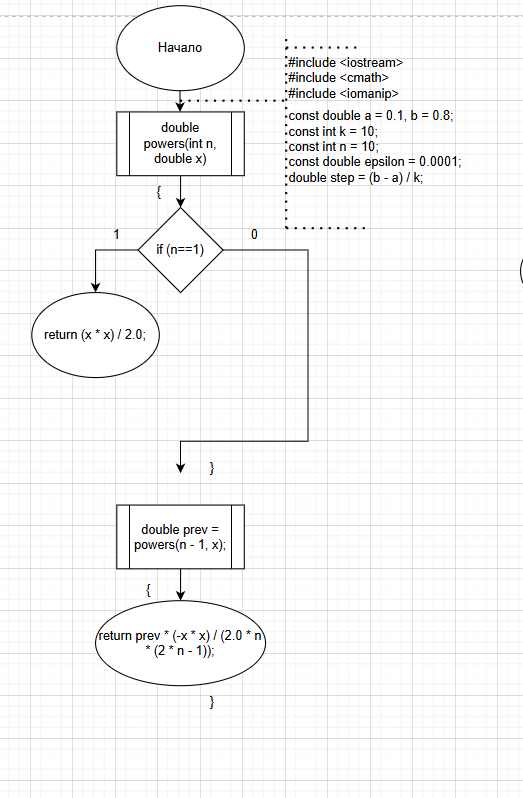
Таким образом, для каждого значения xx в диапазоне [0.1, 0.8] выполняются два расчета:

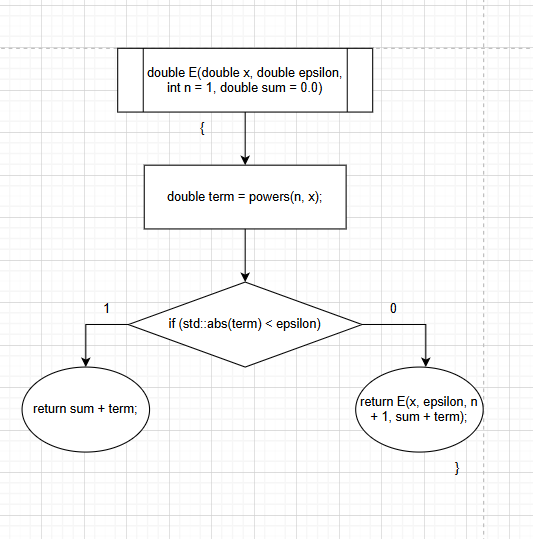
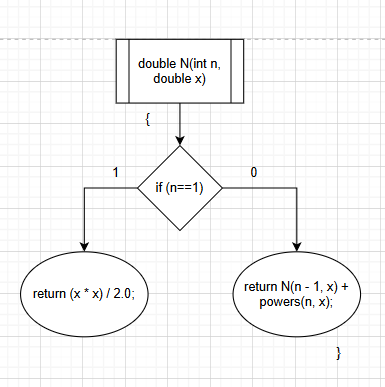
1. **Ряд с фиксированным числом членов** n:.
2. **Ряд до достижения заданной точности** .

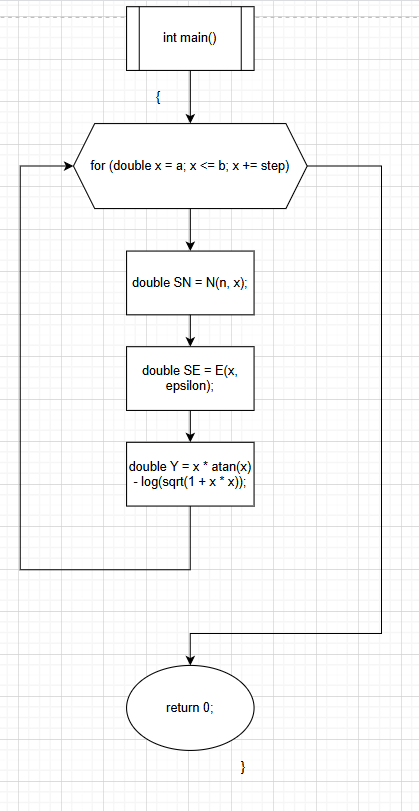
В точке x, точное значение функции вычисляется как: 

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

**4. Блок – схема**

****

****

****

**5. Код программы**

#include <iostream>

#include <cmath>

#include <iomanip>

// Рекурсивная функция для расчета элемента ряда

double powerSeriesElement(int n, double x)

{

if (n == 1)

{

return (x \* x) / 2.0; // Первый элемент

}

double prev = powerSeriesElement(n - 1, x);

return prev \* (-x \* x) / (2.0 \* n \* (2 \* n - 1));

}

// Рекурсивная функция для суммы ряда при заданном n

double N(int n, double x)

{

if (n == 1)

{

return (x \* x) / 2.0; // Первый элемент

}

return N(n - 1, x) + powerSeriesElement(n, x);

}

// Рекурсивная функция для суммы ряда при заданной точности ε

double E(double x, double epsilon, int n = 1, double sum = 0.0)

{

double term = powerSeriesElement(n, x);

if (std::abs(term) < epsilon)

{

return sum + term; // Условие остановки

}

return E(x, epsilon, n + 1, sum + term);

}

// Основная функция

int main()

{

const double a = 0.1, b = 0.8; // Диапазон

const int k = 10; // Количество шагов

const int n = 10; // Заданное количество членов ряда

const double epsilon = 0.0001; // Заданная точность

double step = (b - a) / k; // Шаг изменения x

// Вывод заголовка таблицы

std::cout << std::fixed << std::setprecision(6);

std::cout << "X\t\tSN\t\tSE\t\tY\n";

// Цикл по значениям x

for (double x = a; x <= b; x += step)

{

double SN = N(n, x); // Сумма ряда для заданного n

double SE = E(x, epsilon); // Сумма ряда для заданной точности

double Y = x \* atan(x) - log(sqrt(1 + x \* x)); // Точное значение функции

// Вывод результатов

std::cout << x << "\t" << SN << "\t" << SE << "\t" << Y << "\n";

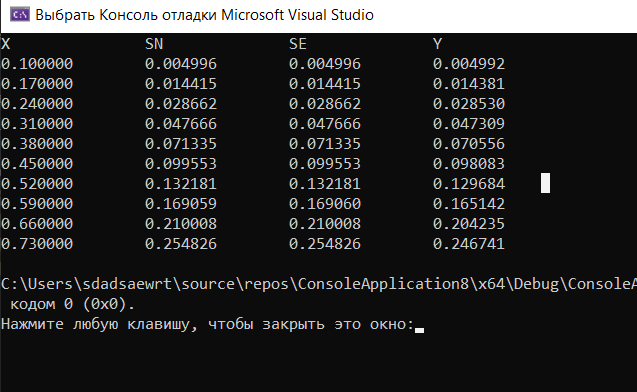
}

return 0;

}

**6. Результат вычислений:**

Заданные значения: n = 10, x = 0.1<=x<=0.8



**7. Ссылка на GitHub**

<https://github.com/hgfgfgffgfg>