**Пермский национальный исследовательский политехнический университет**

**Кафедра информационные технологии и автоматизированные системы**

**Лабораторная работа №3**

**Методы решения нелинейных уравнений**

Выполнил: Пономарев Кирилл Денисович ИВТ-24-2б

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 (=0.0001).

Для сравнения найти точное значение функции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 |  |  | 10 |  |

# 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 x в диапазоне [0.1, 0.8] выполняются два расчета:

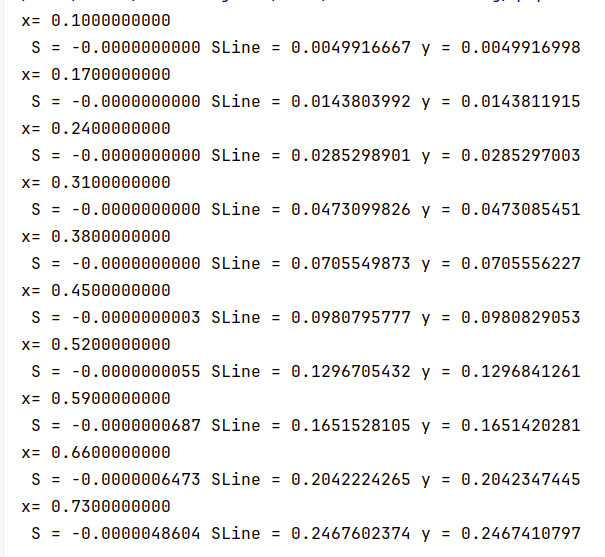
1. **Ряд с фиксированным числом членов** n:.
2. **Ряд до достижения заданной точности** .

В точке x, точное значение функции вычисляется как:

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

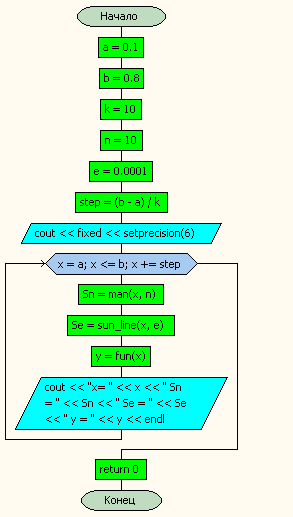
# Код

#include <iostream>  
#include <cmath>  
#include <iomanip>  
  
using namespace std;  
  
double fun(double x)  
{  
 return (x\* atan(x) - log(pow(1+ pow(x, 2), 0.5)));  
}  
  
double man(double x, int n)  
{  
  
 return pow(-1, n + 1) \* pow(x, 2 \* n) / (2 \* n \* (2 \* n - 1));  
}  
  
double sum\_line(double x, double e)  
{  
 double el;  
 double sum =0;  
 int n = 1;  
 do {  
 el = pow(-1, n + 1) \* pow(x, 2 \* n) / (2 \* n \* (2 \* n - 1));  
 sum += el;  
 n++;  
 } while (abs(el) > e);  
 return sum;  
}  
  
int main()  
{  
 double a = 0.1;  
 double b = 0.8;  
 int k = 10;  
 int n = 10;  
 double e = 0.0001;  
  
 cout << fixed << setprecision(10);  
 for (double x = a; x <= b; x += (b - a) / k)  
 {  
 double S = man(x, n);  
 double SLine = sum\_line(x, e);  
 double fx = fun(x);  
  
 cout << "x= " << x <<'\n'<< " Sn = " << S <<'\n'<< " Se = " << SLine <<'\n'<< " y = " << fx << endl;  
 }  
  
 return 0;  
}

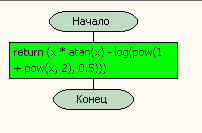


**Блок схема**

int main()



double fun(double x)



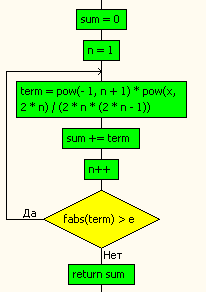
double man(double x,

int n)



double sun\_line(double x,

double e)



Найти 9 число последовательности Фибоначи рекурсивным алгоритмом

Код

#include <iostream>

int fibonacci(int n) {

if (n <= 0) {

return 0;

} else if (n == 1) {

return 1;

} else {

return fibonacci(n - 1) + fibonacci(n - 2);

}

}

int main() {

int n = 9;

int result = fibonacci(n);

std::cout << "9-е число в последовательности Фибоначчи: " << result << std::endl;

return 0;

}



Блок схема

