Пермский национальный исследовательский политехнический университет

Кафедра информационные технологии и автоматизированные системы

Лабораторная работа №2

Методы решения нелинейных уравнений

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.

# Код программы

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

double f\_exact(double x)

{

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

}

double power\_series\_n(double x, int n)

{

return x + pow(x, 4 \* n + 1) / (4 \* n + 1);

}

double power\_series\_e(double x, double e)

{

double sum = x;

double term;

int n = 1;

do

{

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

sum += term;

n++;

} while (fabs(term) > e);

return sum;

}

int main()

{

double a = 0.1;

double b = 0.8;

int k = 10;

int n = 3;

double e = 0.0001;

double step = (b - a) / k;

cout << fixed << setprecision(6);

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

{

double Sn = power\_series\_n(x, n);

double Se = power\_series\_e(x, e);

double y = f\_exact(x);

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

}

return 0;

}

# Результат

