Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Псковский государственный университет»**

Передовая инженерная школа гибридных технологий в станкостроении

Союзного государства

Отделение информационно-коммуникационных технологий

Отчет по лабораторной работе №5

«Поиск корней сложной функции с использованием подпрограмм»

Вариант №22

**Выполнил:** Иванов И.С.

группа 0482-06

**Проверил:** Андреев Д. А.

Псков

2024

1. **Вариант задания**

Вариант №22: найти корень уравнения

методом половинного деления на отрезке . Вычисление интеграла оформить в виде подпрограммы-функции с аргументами α и ε (точность вычисления интеграла). Интеграл вычисляется методом Симпсона. Начальная точность вычисления интеграла , затем увеличивается в 2 раза на каждом шаге по мере уменьшения отрезка, на котором находится корень. Точность вычисления корня .

1. **Используемые вычислительные формулы**

;

, .

Начальное число отрезков , после вычисления каждого промежуточного значения интеграла оно увеличивается в 2 раза.

Формула Симпсона: , где , .

На первом шаге , , затем , , где .

, .

Промежуточные и итоговое значения корня .

1. **Блок-схема алгоритма**

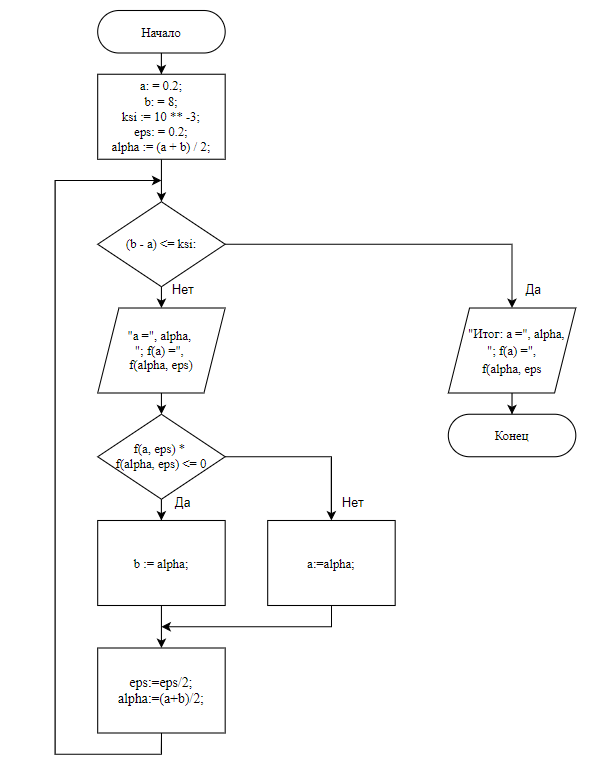
****

Рис. 1. Блок-схема алгоритма основной программы

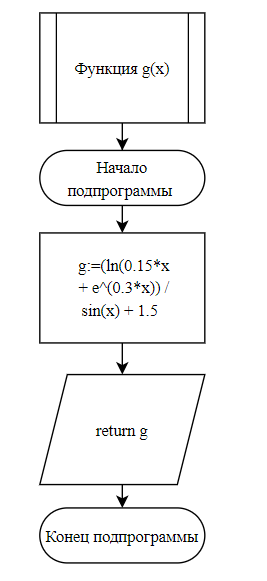


Рис. 2. Блок-схема алгоритма подпрограммы-функции g(x)

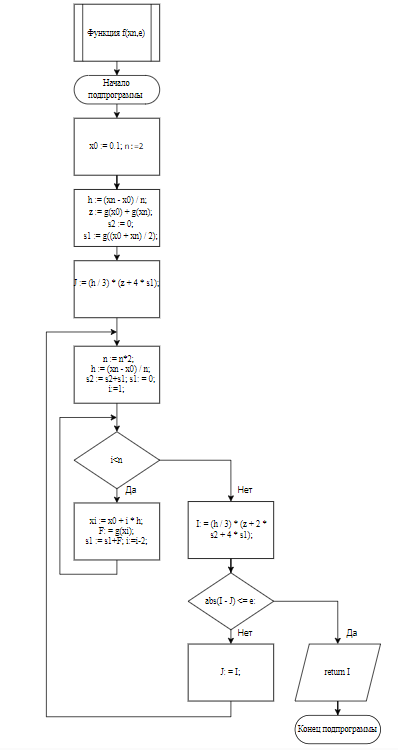


Рис. 3. Блок-схема алгоритма подпрограммы-функции f(xn, e)

1. **Текст программы на языке высокого уровня**

import math  
def g(x):  
 g = (math.log(0.15 \* x) + math.exp(0.3 \* x)) / (math.sin(x) + 1.5)  
 return g  
def f(xn, e):  
 x0 = 0.1  
 n = 2  
 h = (xn - x0) / n  
 z = g(x0) + g(xn)  
 s2 = 0  
 s1 = g((x0 + xn) / 2)  
 J = (h / 3) \* (z + 4 \* s1)  
 n \*= 2  
 h = (xn - x0) / n  
 s2 += s1  
 s1 = 0  
 for i in range(1, n, 2):  
 xi = x0 + i \* h  
 F = g(xi)  
 s1 += F  
 I = (h / 3) \* (z + 2 \* s2 + 4 \* s1)  
 while not abs(I - J) <= e:  
 J = I  
 n \*= 2  
 h = (xn - x0) / n  
 s2 += s1  
 s1 = 0  
 for i in range(1, n, 2):  
 xi = x0 + i \* h  
 F = g(xi)  
 s1 += F  
 I = (h / 3) \* (z + 2 \* s2 + 4 \* s1)  
 return I  
# Параметры для поиска корня  
a = 0.2  
b = 8.0  
ksi = 10 \*\* -3  
eps = 0.2  
# Начальное значение alpha  
alpha = (a + b) / 2  
while not (b - a) <= ksi:  
 print("a =", alpha, "; f(a) =", f(alpha, eps))  
 if f(a, eps) \* f(alpha, eps) <= 0:  
 b = alpha  
 else:  
 a = alpha  
 alpha = (a + b) / 2  
print("Итог: a =", alpha, "; f(a) =", f(alpha, eps))

Рис. 4. Текст программы

1. **Результат**

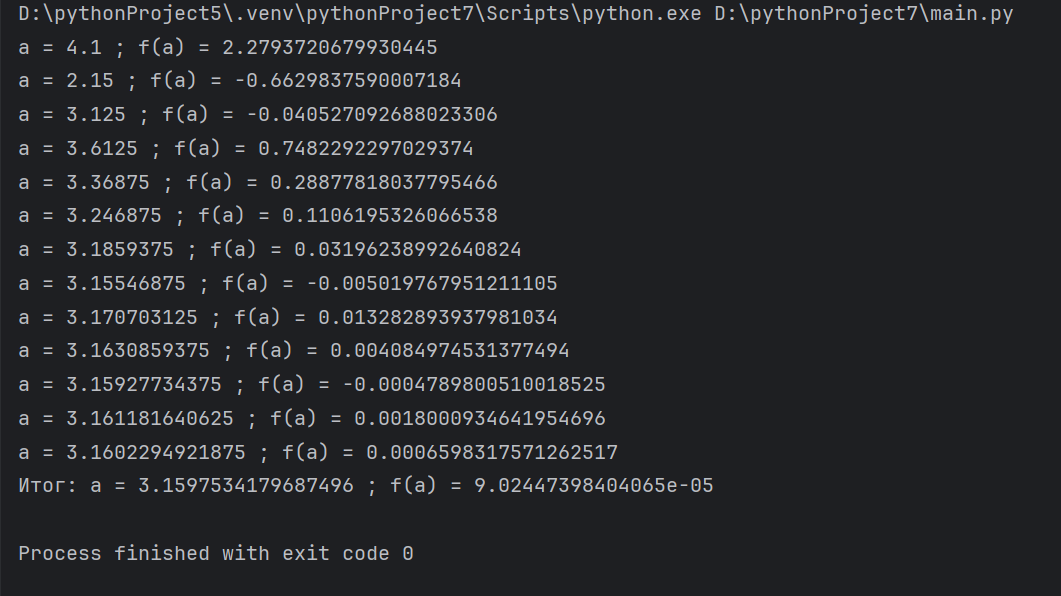


Рис. 5. Результат работы программы

1. **Вывод**

В этой лабораторной работе был найден корень уравнения с использованием подпрограммы-функции. В подпрограмме вычисляется значение определённого интеграла при помощи алгоритма с удвоением отрезков разбиения и метода Симпсона. В основной программе происходит поиск корня при помощи метода половинного деления