Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерных технологий

Лабораторная работа №3

по дисциплине «Методы оптимизации»

Вариант 13

Выполнила:

Павличенко Софья Алексеевна, Р3215

Преподаватель:

Селина Елена Георгиевна

Санкт-Петербург 2025г.

Оглавление

[Задание 3](#_Toc193370590)

[Решение 4](#_Toc193370591)

[Исходный код программы 5](#_Toc193370592)

[Результаты работы программы 7](#_Toc193370593)

[Вывод 7](#_Toc193370594)

# Задание

Решить задачу методом квадратичной аппроксимации. По 5 шагов метода выполнить вручную + написать программу по методу на одном из языков программирования.

# Решение

**Шаг 1.**

**Шаг 2.**

**Шаг 3.**

**Шаг 4.**

**Шаг 5.**

…

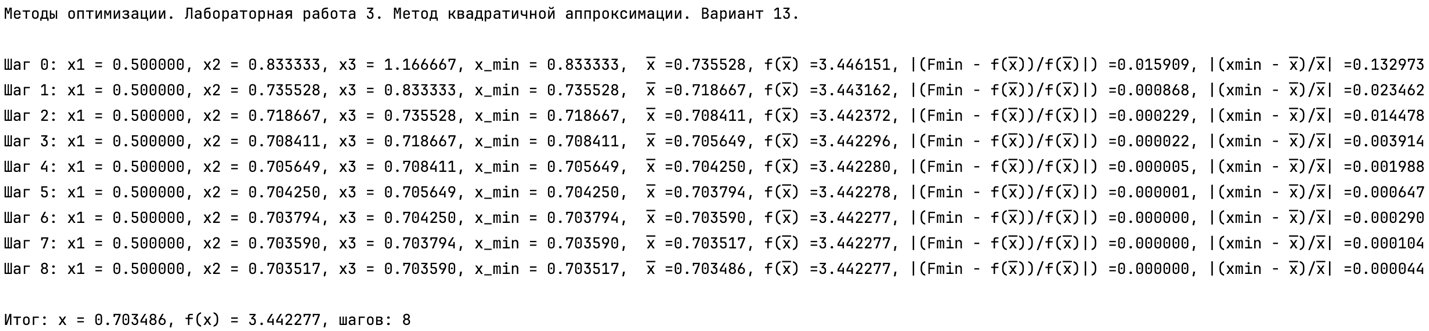
## Исходный код программы

import math  
  
MAX\_ITERATIONS = 500  
  
  
def find\_xs(x1):  
 x2 = x1 + dx  
 x3 = x1 - dx  
 if f(x1) > f(x2):  
 x3 = x1 + 2\*dx  
 return x1, x2, x3  
  
  
def quadratic\_approximation\_method(x1, x2, x3, eps, n=0):  
 if n > MAX\_ITERATIONS:  
 raise ValueError("Метод не сошёлся за максимальное число итераций.")  
  
 f1 = f(x1)  
 f2 = f(x2)  
 f3 = f(x3)  
  
 Fmin = min(f1, f2, f3)  
 xmin = x1 if Fmin == f1 else x2 if Fmin == f2 else x3  
 denominator = (x2-x3) \* f1 + (x3-x1) \* f2 + (x1-x2) \* f3  
 if denominator == 0:  
 print(f"Шаг {n}: x1 = {x1:.6f}, x2 = {x2:.6f}, x3 = {x3:.6f}, знаменатель в формуле для x̅ равен 0 => x1 = xmin ")  
 return quadratic\_approximation\_method(\*find\_xs(xmin), eps, n + 1)  
  
 x = 1/2 \* ((x2\*\*2-x3\*\*2) \* f1 + (x3\*\*2-x1\*\*2) \* f2 + (x1\*\*2-x2\*\*2) \* f3) / denominator  
 fx = f(x)  
 print(f"Шаг {n}: x1 = {x1:.6f}, x2 = {x2:.6f}, x3 = {x3:.6f}, x\_min = {xmin:.6f}, ̅x ={x:.6f}, f(̅x) ={fx:.6f}, "  
 f"|(Fmin - f(̅x))/f(̅x)|) ={abs((Fmin-fx)/fx):.6f}, |(xmin - ̅x)/̅x| ={abs((xmin-x)/x):.6f}")  
 if abs((Fmin-fx)/fx) < eps and abs((xmin-x)/x) < eps:  
 return x, fx, n  
 else:  
 if min(x1, x3) <= x <= max(x1,x3):  
 x2 = min(xmin, x)

if x1 > x3:

x1 = x3  
 x3 = max(xmin, x)  
 return quadratic\_approximation\_method(x1, x2, x3, eps, n + 1)  
 else:  
 return quadratic\_approximation\_method(\*find\_xs(x), eps, n + 1)  
  
  
  
print('Методы оптимизации. Лабораторная работа 3. Метод квадратичной аппроксимации. Вариант 13.')  
  
*# --- Исходные данные ---*a = 0.5  
b = 1.5  
eps = 0.0001  
f = lambda x: 1/x + math.e\*\*x  
  
  
*# --- Вычисление минимума ---*dx = (b - a) / 3  
x, y, n = quadratic\_approximation\_method(\*find\_xs(a), eps)  
  
print(f"\nИтог: x = {x:.6f}, f(x) = {y:.6f}, шагов: {n}")

## Результаты работы программы



# Вывод

В ходе выполнения лабораторной работы я научилась находить минимум функции методом квадратичной аппроксимации, реализовав метод на языке программирования Python. В результате работы были найден минимум уравнения на отрезке с определенной точностью.