Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод Пауэлла (квадратичная интерполяция) |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | В. В. Тынченко |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/1Б, 032156940 |  |  |  | Н.А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Вариант задания................................................................................................ 3

3 Описание метода............................................................................................... 3

4 Реализация метода............................................................................................ 3

5 Анализ результатов.......................................................................................... 5

6 Вывод................................................................................................................. 9

**1 Задание**

Разработать программу, реализующую метод Пауэлла (метод  
квадратичной интерполяции). Найти безусловный экстремум функции,  
выбранной в соответствии с заданием, с использованием разработанной  
программы.

**2 Вариант задания**

f(x) = (6x + 3)^2 − 2x − 1 → min Интервал неопределённости [-6,6].

**3 Описание метода**

Задается начальная точка и с помощью пробного шага находятся три  
опорные точки таким образом, чтобы они располагались как можно ближе к  
искомой точке минимума. В полученных точках вычисляются значения  
функции. Затем строится интерполяционный полином второй степени,  
проходящий через имеющиеся три точки. В качестве приближения точки  
минимума берется точка минимума полинома. Процесс поиска заканчивается,  
когда полученная точка отличается от наилучшей из трех опорных точек не  
более чем на заданную величину

Алгоритм

Шаг 1. Задать начальную точку x , величину шага delta > 0, и epsilon1 и  
epsilon2– малые положительные числа, характеризующие точность.

Шаг 2. Вычислить x2 = x1 + delta

Шаг 3. Вычислить f(x1) и f(x2).

Шаг 4. а) если f(x1) > f(x2), x3 = x1 + 2delta; б) иначе x3 = x1 - delta.

Шаг 5. Вычислить f(x3).

Шаг 6. Найти Fmin =min{f1, f2, f3}, xmin = xi : f(xi) = Fmin.

Шаг 7. Вычислить точку минимума интерполяционного полинома,  
построенного по трем точкам: x' = 1/2 \* ((x2^2 - x3^2)f1 + (x3^2 - x1^2) + (x1^2 -  
x2^2)f3) / ((x2 - x3)f1 + (x3 - x1) + (x1 - x2)f3), и величину функции f(x'). Если  
знаменатель в формуле для x' на некоторой итерации обращается в нуль, то  
результатом интерполяции является прямая. В этом случае рекомендуется  
обозначить x1 = xmin и перейти к шагу 2.

Шаг 8. Проверить выполнение условий окончания: |(Fmin - f(x')) / f(x')| <  
epsilon1, |(xmin - x') / x'| < epsilon2. Тогда: а) если оба условия выполнены,  
процедуру закончить и положить x\* = x'; б) если хотя бы одно из условий не  
выполнено и x' принадлежит [x1, x3] , выбрать наилучшую точку ( xmin или x' )  
и две точки по обе стороны от нее. Обозначить эти точки в естественном  
порядке и перейти к шагу 6; в) если хотя бы одно из условий не выполнено и x'  
не принадлежит [x1, x3] , то положить x1 = x' и перейти к шагу 2.

**4 Реализация метода**

Ниже в листинге 1 представлен текст python программы, реализующей

метод для поиска минимума функции.

Листинг 1 – Текст python программы

import math  
def quad\_inter\_method(func, x1, delta, eps1=1e-3, eps2=1e-3):  
 work = True  
 check = True  
 calc\_num = 0  
 while work:  
 if check:  
 x2 = x1 + delta  
 f1 = func(x1)  
 f2 = func(x2)  
 calc\_num+=2  
 if f1 > f2:  
 x3 = x1 + 2 \* delta  
 else:  
 x3 = x1 - delta  
 f3 = func(x3)  
 calc\_num += 1  
 else:  
 f1 = func(x1)  
 f2 = func(x2)  
 f3 = func(x3)  
 calc\_num += 3  
 F\_min = min([f1, f2, f3])  
 if F\_min == f1: x\_min = x1  
 elif F\_min == f2: x\_min = x2  
 else: x\_min = x3  
 numerator = (x2\*\*2 - x3\*\*2) \* f1 + (x3\*\*2 - x1\*\*2) \* f2 + (x1\*\*2 - x2\*\*2  
 ) \* f3  
 denominator = (x2 - x3) \* f1 + (x3 - x1) \* f2 + (x1 - x2) \* f3  
 if denominator == 0:  
 x1 = x\_min  
 check = True  
 continue  
 x\_poli = 0.5 \* (numerator / denominator)  
 if x\_poli == 0:x\_poli += eps2 / 2  
 f\_poli = func(x\_poli)  
 calc\_num += 1  
 if abs((F\_min - f\_poli) / f\_poli) < eps1 and abs((x\_min - x\_poli) / x\_po  
 li) < eps2:  
 work = False  
 else:  
 interval = sorted([x1, x3])  
 if x\_poli >= interval[0] and x\_poli <= interval[1]:  
 if min([F\_min, f\_poli]) == F\_min: x\_choosed = x\_min  
 else:  
 x\_choosed = x\_poli  
 dots = sorted(list(set([x1, x2, x3, x\_choosed, x\_poli, x\_min])))  
 x\_choosed\_i = dots.index(x\_choosed)  
 if x\_choosed\_i == 0: x1, x2, x3 = [dots[0], dots[1], dots[2]]  
 elif x\_choosed\_i == len(dots) - 1: x1, x2, x3 = [dots[len(dots)   
 - 3], dots[len(dots) - 2], dots[len(dots) - 1]]  
 else:x1, x2, x3 = [dots[x\_choosed\_i - 1], dots[x\_choosed\_i], dot  
 s[x\_choosed\_i + 1]]  
 check = False  
 else:  
 x1 = x\_poli

Окончание листинга 1

check = True  
 return x\_poli, calc\_num  
def func(x):  
 # -0.472  
 return (6 \* x + 3)\*\*2 - 2\*x - 1  
x0 = -6  
delta\_x = 0.0001  
eps1 = 0.0001  
eps2 = 0.0001  
minimum, calc\_num = quad\_inter\_method(func, x0, delta\_x, eps1, eps2)  
print("Параметры метода: x0 -", x0, ", delta\_x -", delta\_x, ", epsilon\_1 -", eps  
1, ", epsilon\_1 -", eps2)  
print("Минимум функции(x):", minimum)  
print("Количество вычислений функции:", calc\_num)

**5 Анализ результатов**

Для начала найдём реальный минимум функции, для этого произведём  
следующие действия:

Находим первую производную функции: y′=72·x+34 Приравниваем ее к  
нулю: 72·x+34=0 x1=-0.472 Вычисляем значения функции на концах интервала  
f(-0.472)=-0.027 f(-6)=1100 f(6)=1508 В результате реальный минимум функции  
на интервале [-6; 6] находится в точке -0.472.

Теперь получим значения работы метода Пауэлла. Также построим  
графики зависимости количества вычислений целевой функции и отклонения  
от реального минимума в зависимости от изменения параметров метода.  
Результаты представлены на рисунках ниже.

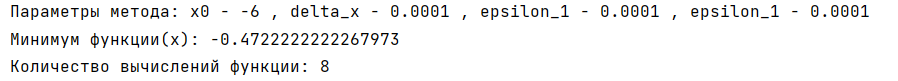


Рисунок 1 – Результат работы метода

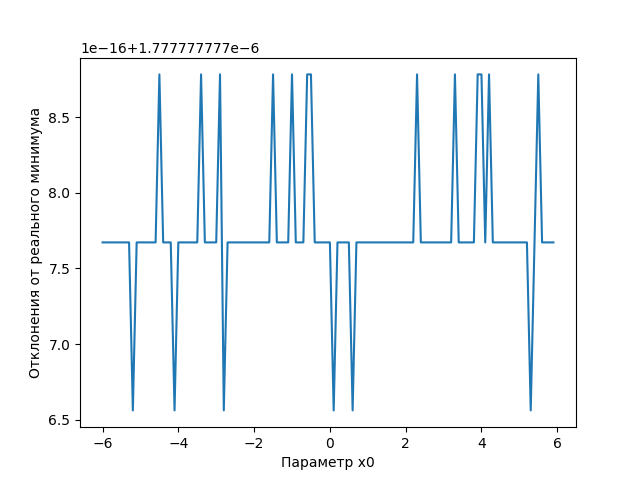


Рисунок 2 – График отклонения от реального минимума для x0

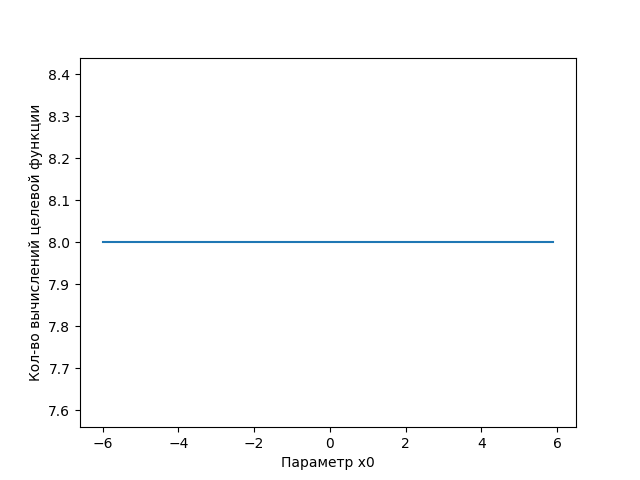


Рисунок 3 – График количества вычислений целевой функции для x0

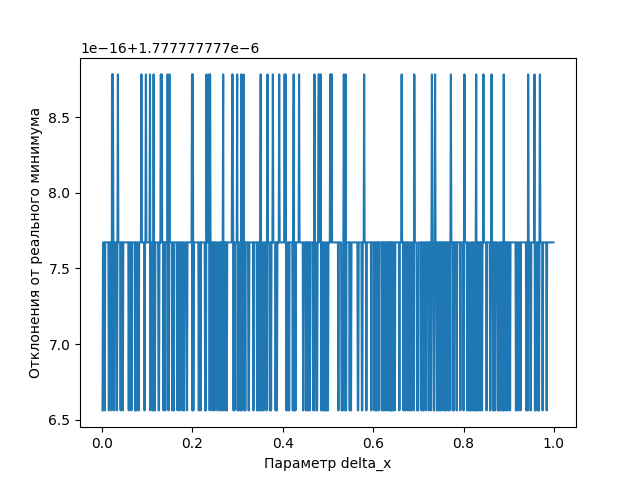


Рисунок 4 – График отклонения от реального минимума для deltaX

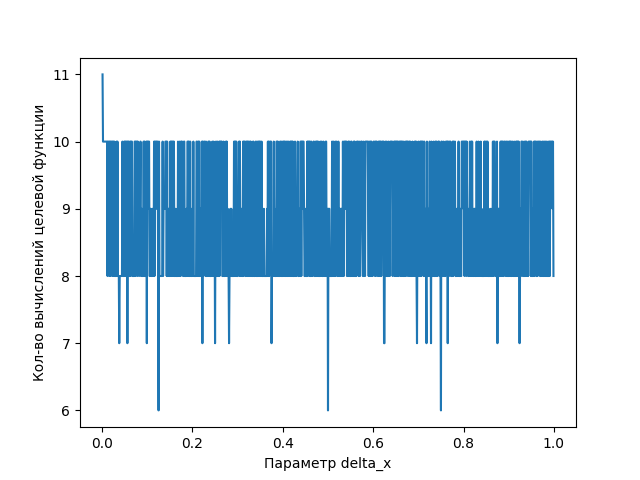


Рисунок 5 – График количества вычислений целевой функции для deltaX

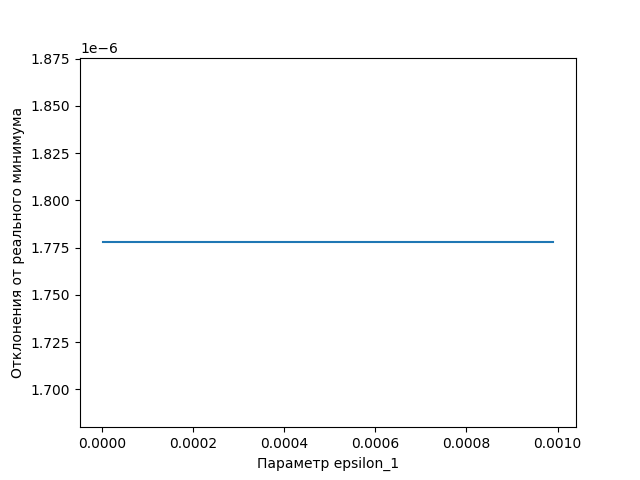


Рисунок 6 – График отклонения от реального минимума для epsilon1

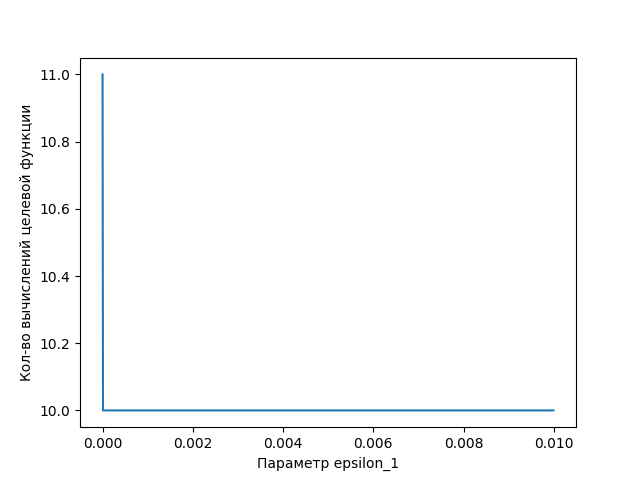


Рисунок 7 – График количества вычислений целевой функции для epsilon1

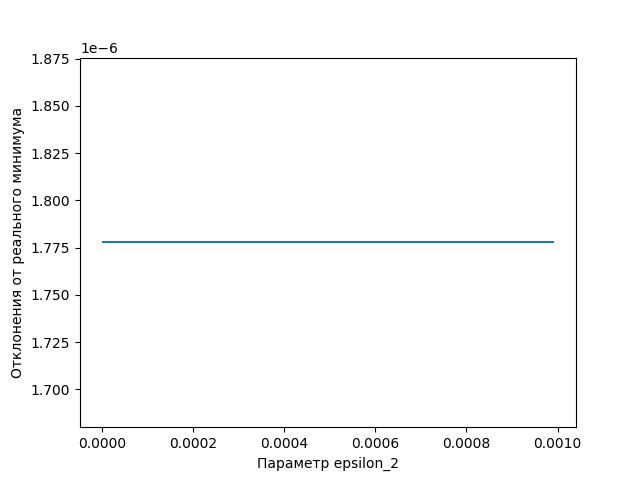


Рисунок 8 – График отклонения от реального минимума для epsilon2

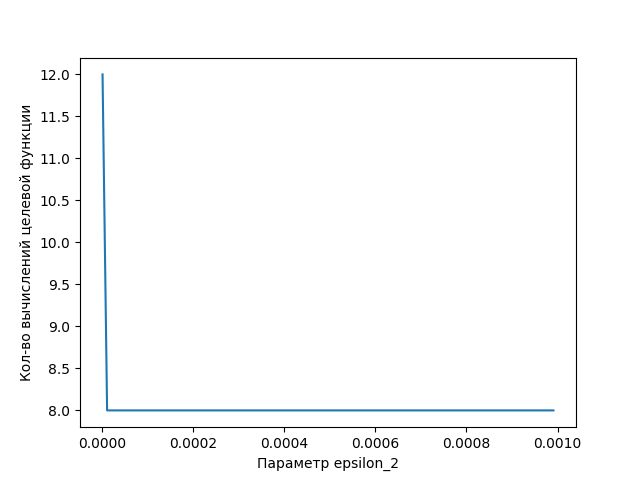


Рисунок 9 – График количества вычислений целевой функции для epsilon2

В результате можно сказать, что найденный минимум функции близок к  
реальному. При этом сложно определить влияние параметров метода на  
результат, для данной функции они практически не значимы.

**6 Вывод**

При выполнении задания был успешно реализован метод Пауэлла,  
результаты работы метода сравнены с реальным и близки к нему, исследована  
зависимость работы метода от значений его параметров.