Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод покоординатного спуска |
| 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.............................................................................................................. 3

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

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

5 Анализ результатов.......................................................................................... 4

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

**1 Задание**

Разработать программу, реализующую метод покоординатного спуска.  
Найти безусловный экстремум функции, выбранной в соответствии с заданием,  
с использованием разработанной программы.

**2 Вариант**

f(x) = x1^2 + x2 \* x1 + 7 \* x2^2

**3 Описание метода**

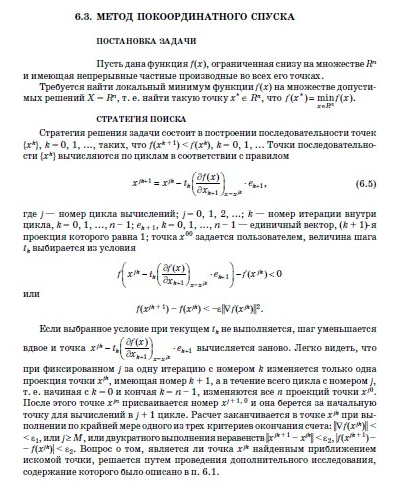


Рисунок 1 – Описание метода

**4 Реализация метода**

Ниже в листинге 1 представлен текст python программы, реализующей  
метод для поиска минимума функции.

Листинг 1 – Текст python программы

def norm(vector):  
 return (vector[0]\*\*2 + vector[1]\*\*2) \*\* 0.5  
def coordinat\_descent(f, df1, df2, x0, eps1, eps2, M, t = 0.1):  
 n = len(x0)  
 if M % n != 0:

Окончание листинга 1

print("Incorrect M")  
 return 0  
 df = lambda x: (df1(x), df2(x))  
 j = 0  
 k = 0  
 x = []  
 x.append([])  
 x[0].append(x0)  
 check3 = False  
 while True:  
 #3  
 if j >= M:  
 return x[j - 1][k]  
 k = 0  
 while True:  
 #5  
 if k == n:  
 x.append([])  
 x[j+1].append(None)  
 x[j + 1][0] = x[j][n]  
 j = j + 1  
 break  
 dfjk = df(x[j][k])  
 if norm(dfjk) < eps1:  
 return x[j][k]  
 tk = t  
 x[j].append(None)  
 if k + 1 == 1:  
 dfk = df1  
 else:  
 dfk = df2  
 if k+1 == 1:e = (1, 0)  
 else:e = (0, 1)  
 while True:  
 x[j][k+1] = (x[j][k][0] - tk \* dfk(x[j][k]) \* e[0], x[j][k][1] -  
 tk \* dfk(x[j][k]) \* e[1])  
 if f(x[j][k+1]) - f(x[j][k]) >= 0:  
 tk = tk / 2  
 else: break  
 check1 = norm((x[j][k+1][0] - x[j][k][0], x[j][k+1][1] - x[j][k][1])  
 ) < eps2  
 check2 = abs(f(x[j][k+1]) - f(x[j][k])) < eps2  
 if check1 and check2:  
 if check3:  
 return x[j][k+1]  
 check3 = True  
 else:  
 check3 = False  
 k = k + 1

**5 Анализ результатов**

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

Найдём частные производные 1-го порядка:

z'x = 2x + y

z'y = x + 14y

M0: x = 0; y = 0

Найдём частные производные 2-го порядка в точке M0 и проверим  
достаточное условие экстремума:

A = z''xx(M0) = 2

B = z''xy(M0) = 1

C = z''yy(M0) = 14

AC - B^2 > 0

2\*14 - 1^2 > 0

При этом A>0, следовательно точка (0;0) - минимум функции.

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

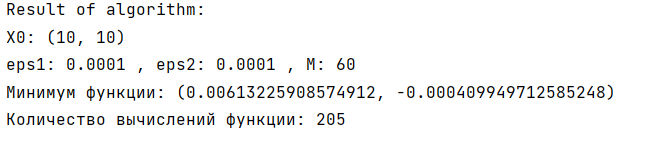


Рисунок 2 – Результат работы метода

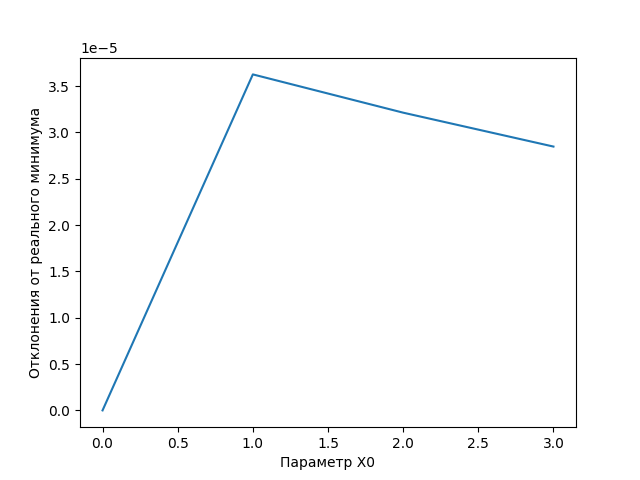


Рисунок 3 – График отклонения от реального минимума для X0

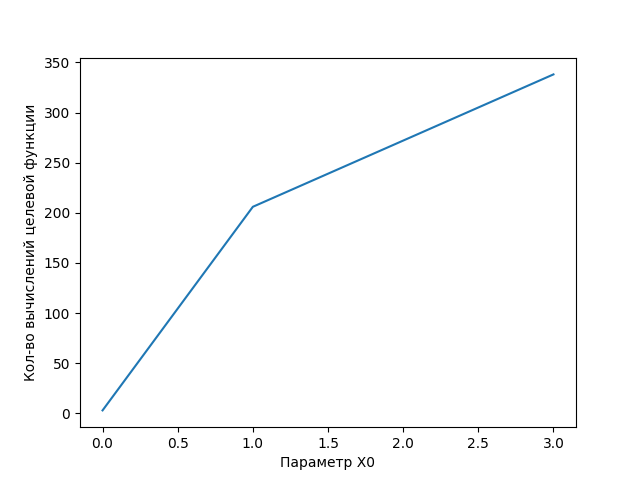


Рисунок 4 – График количества вычислений целевой функции для X0

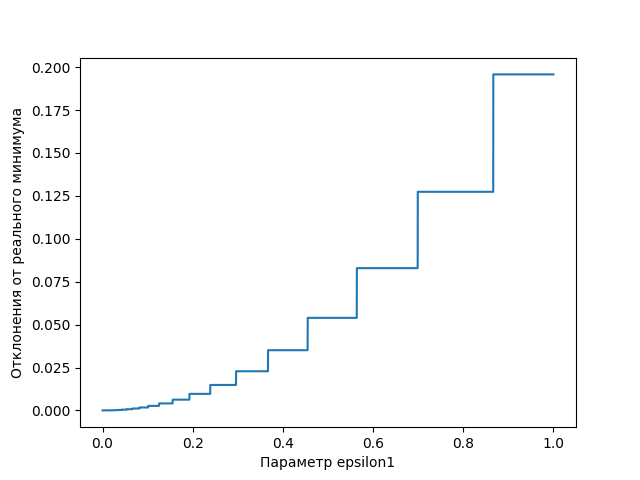


Рисунок 5 – График отклонения от реального минимума для epsilon1

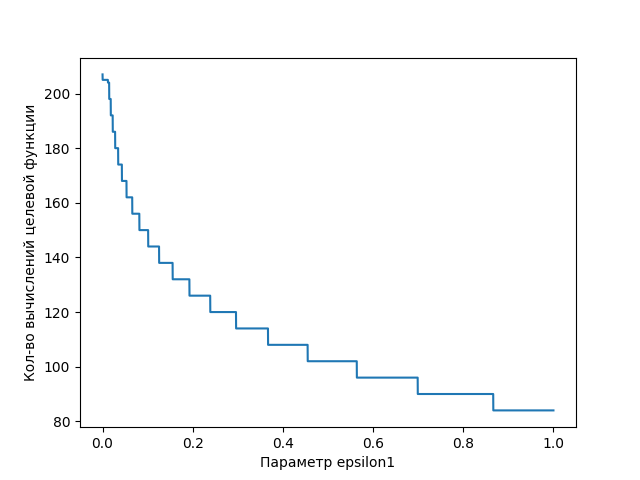


Рисунок 6 – График количества вычислений целевой функции для epsilon1

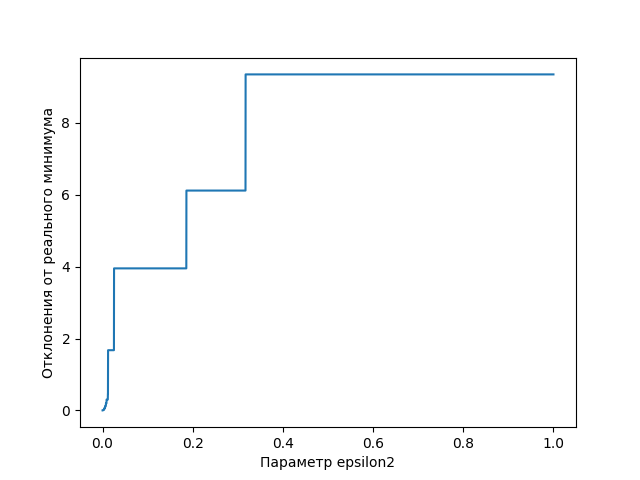


Рисунок 7 – График отклонения от реального минимума для epsilon2

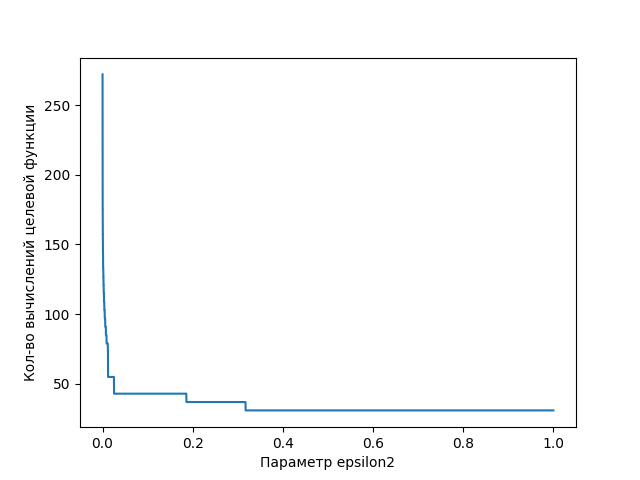


Рисунок 8 – График количества вычислений целевой функции для epsilon2

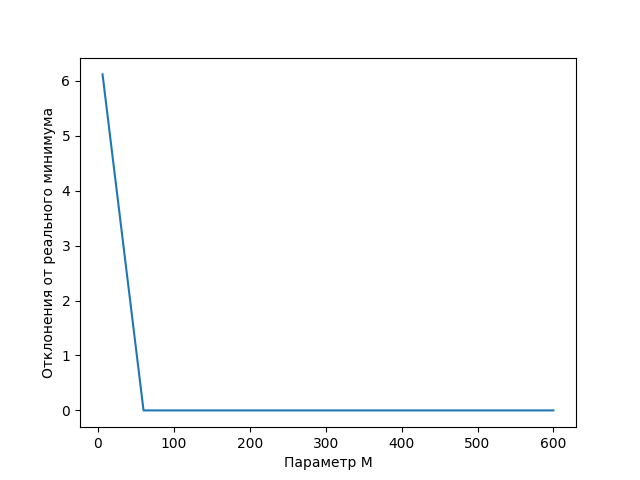


Рисунок 9 – График отклонения от реального минимума для M

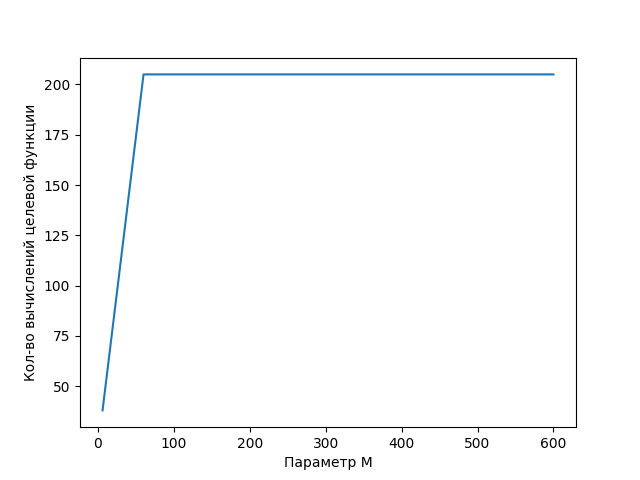


Рисунок 10 – График количества вычислений целевой функции для M

В результате можно сказать, что найденный минимум функции  
соответствует реальному. При этом можно определить следующее влияние  
параметров на результат: чем больше эпсилон 1 и 2 тем больше отклонение, но  
меньше кол-во вычислений. Чем больше x0 отличается от реального минимума  
тем больше вычислений функции. При низких M высокое отклонение и малое  
количество вычислений.

**6 Вывод**

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