Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод Дэвидона-Флетчера-Пауэлла |
| 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  
Дэвидона-Флетчера-Пауэлла. 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**

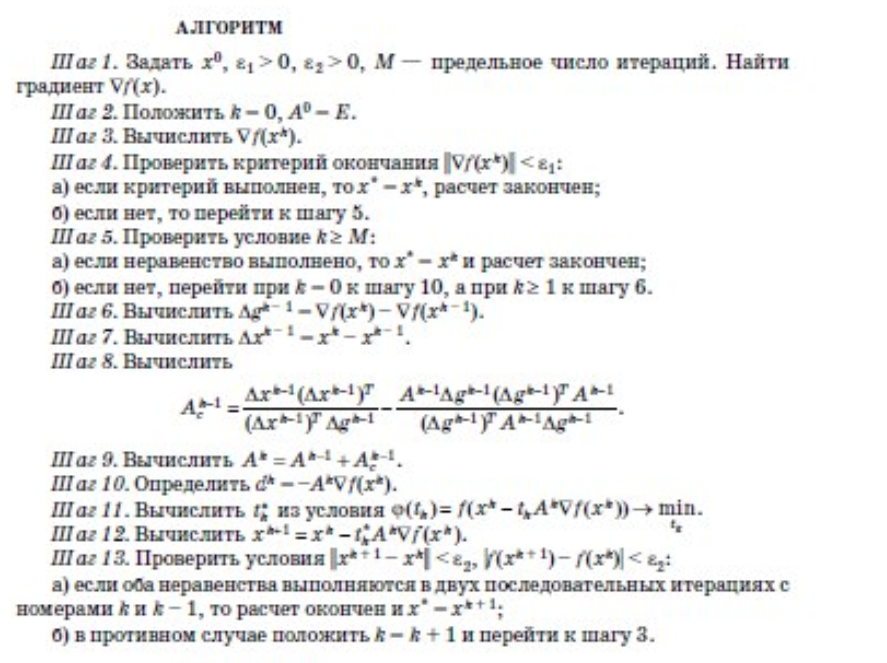


Рисунок 1 – Описание метода

**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 davidon\_fletcher\_powell(f, df1, df2, x0, eps1, eps2, M):  
 global cache\_0  
 global cache\_1  
 global cache\_2  
 df = lambda x: (df1(x), df2(x))  
 k = 0  
 x = [x0]  
 d = [None]  
 check3 = False  
 A = [np.eye(2)]  
 Ac = [None]  
 dg = [None]  
 dx = [None]  
 while True:

Окончание листинга 1

dfk = df(x[k])  
 if norm(dfk) < eps1 or k >= M:  
 result = x[k]  
 break  
 if k != 0:  
 if k - 1 + 1 > len(dg):  
 dg.append(None)  
 dg[k-1] = (dfk[0] - df(x[k-1])[0], dfk[1] - df(x[k-1])[1])  
 if k - 1 + 1 > len(dx):  
 dx.append(None)  
 dx[k-1] = (x[k][0] - x[k - 1][0], x[k][1] - x[k - 1][1])  
 if k - 1 + 1 > len(Ac):  
 Ac.append(None)  
 first = np.matmul(np.array(dx[k-1]), np.transpose(dx[k-1])) / np.mat  
 mul(np.transpose(dx[k-1]), np.array(dg[k-1]))  
 second = (np.matmul(np.matmul(A[k-1], np.array(dg[k-1])), np.transpo  
 se(dg[k-1]))\*A[k-1]) / (np.matmul(np.matmul(np.transpose(dg[k-1]), A  
 [k-1]), np.array(dg[k-1])))  
 Ac[k - 1] = first - second  
 if k + 1 > len(A):  
 A.append(None)  
 A[k] = A[k-1] + Ac[k-1]  
 if k + 1 > len(d):  
 d.append(None)  
 d[k] = - np.matmul(A[k], np.array(dfk))  
 tk\_func = lambda tk: f(tuple(np.array(x[k]) - tk \* np.matmul(d[k], np.ar  
 ray(dfk))))  
 tk = fibonacci\_method(tk\_func)  
 x.append(None)  
 x[k + 1] = np.array(x[k]) - tk \* np.matmul(d[k], np.array(dfk))  
 check1 = norm(tuple(x[k+1] - x[k])) < eps2  
 check2 = abs(f(tuple(x[k+1])) - f(tuple(x[k]))) < eps2  
 if check1 and check2:  
 if check3:  
 return x[k + 1]  
 check3 = True  
 else:  
 check3 = False  
 k = k + 1  
 return result

**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.

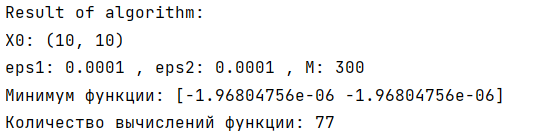


Рисунок 2 – Результат работы метода

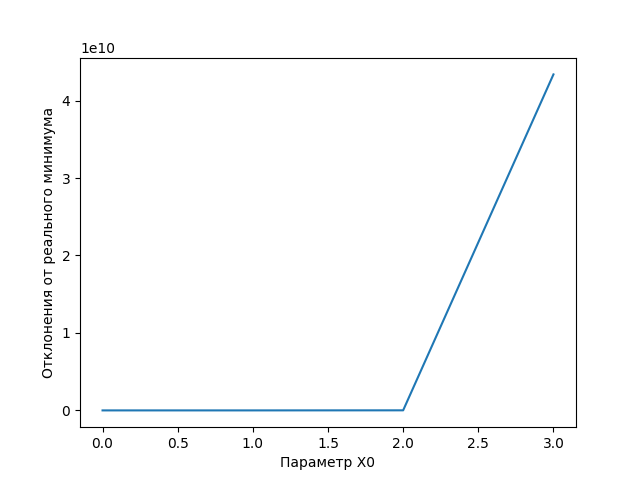


Рисунок 3 – График отклонения от реального минимума для X0

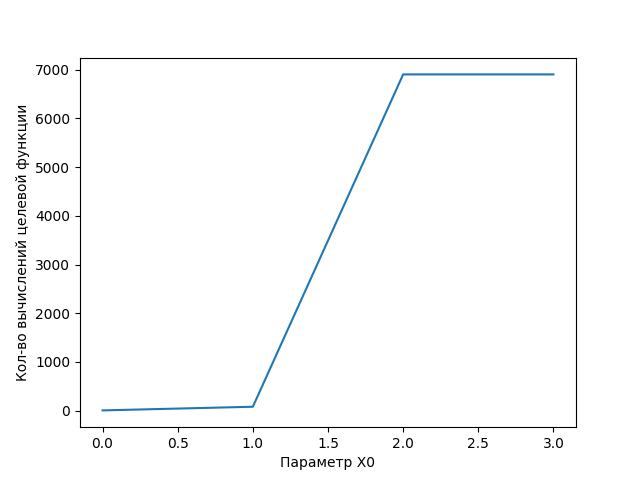


Рисунок 4 – График количества вычислений целевой функции для X0

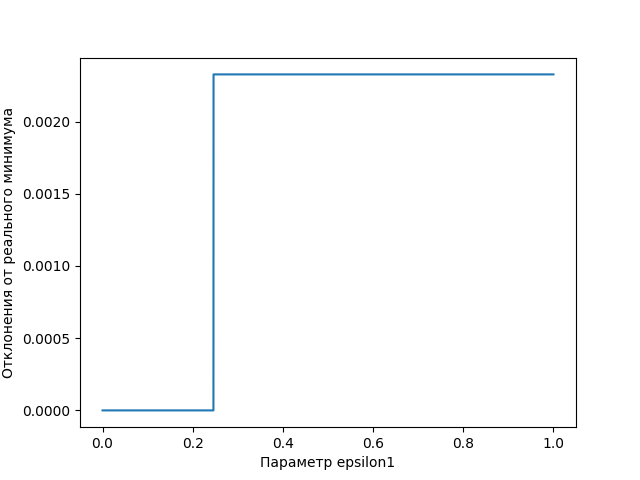


Рисунок 5 – График отклонения от реального минимума для epsilon1

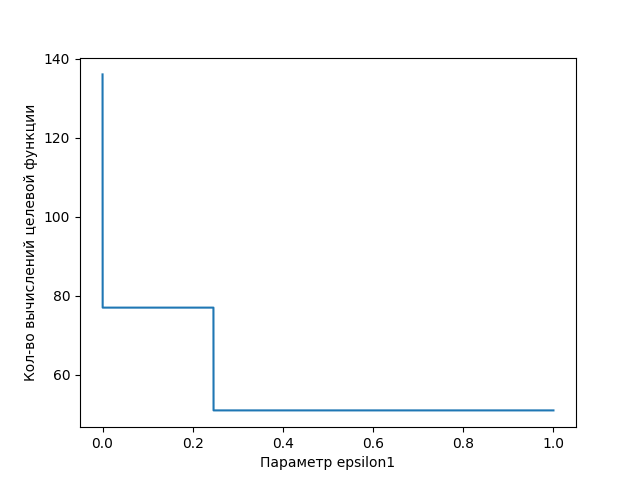


Рисунок 6 – График количества вычислений целевой функции для epsilon1

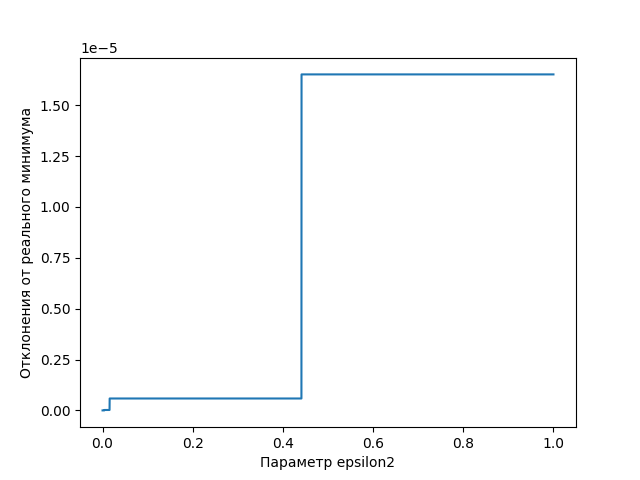


Рисунок 7 – График отклонения от реального минимума для epsilon2

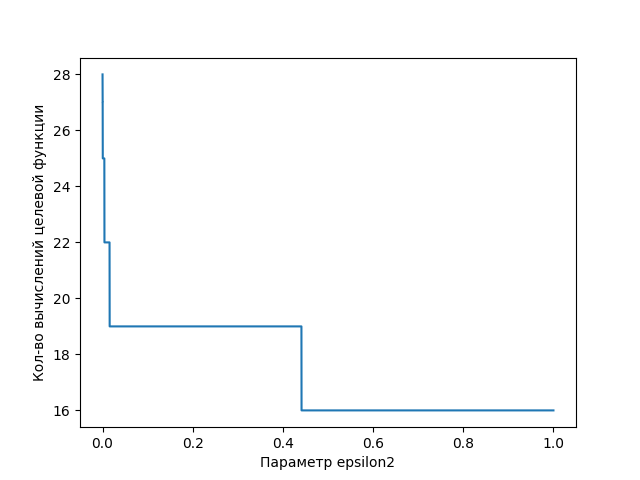


Рисунок 8 – График количества вычислений целевой функции для epsilon2

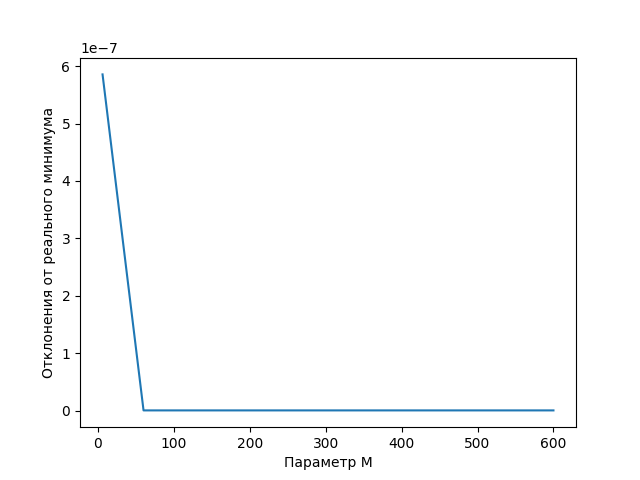


Рисунок 9 – График отклонения от реального минимума для M

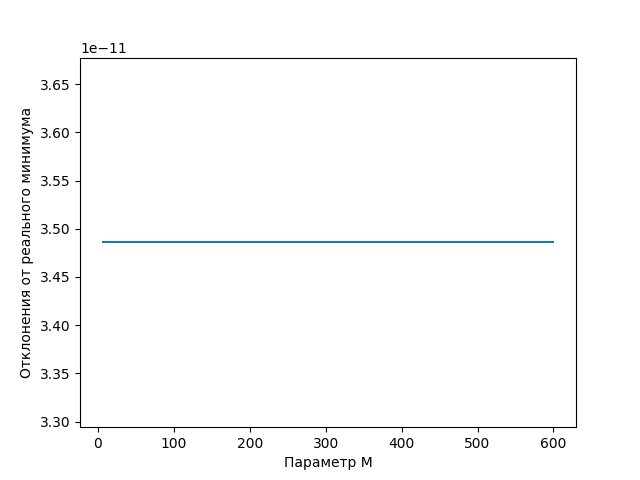


Рисунок 10 – График количества вычислений целевой функции для M

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  
Дэвидона-Флетчера-Пауэлла, результаты работы метода сравнены с реальным  
и близки к нему, исследована зависимость работы метода от значений его  
параметров.