Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод Розенброка |
| 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................................................................................................................. 10

**1 Задание**

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

**2 Вариант**

f(x) = (x1 - 1)^2 + (x2 + x1)^2

**3 Описание метода**

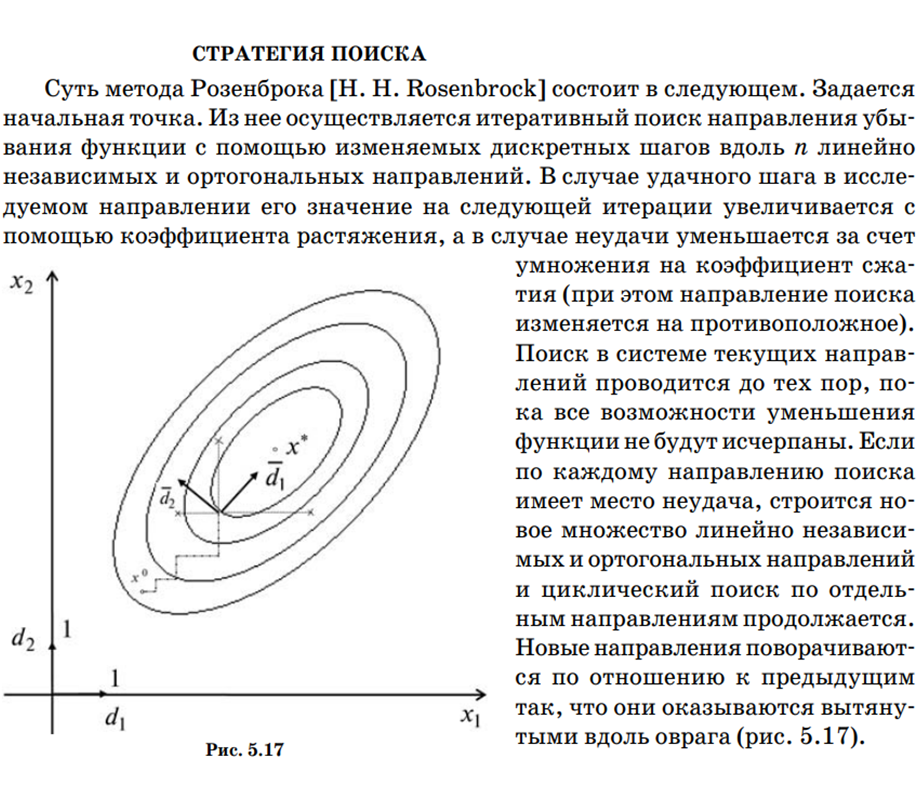


Рисунок 1 – Описание метода

**4 Реализация метода**

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

Листинг 1 – Текст python программы

def rosenbrok(func, x0, a, b, e, N, deltas\_0):  
 n = len(x0)  
 deltas = {i + 1: deltas\_0[i] for i in range(n)}  
 d = {}  
 for i in range(1, n + 1):  
 lst = [0] \* n  
 lst[i - 1] = 1  
 d[i] = np.array(lst)  
 x = {0: x0}  
 y = {}  
 y[1] = x[0]  
 k = 0  
 i = 1  
 l = 0  
 end = False

Окончание листинга 1

while not end:  
 if func(y[i] + deltas[i] \* d[i]) < func(y[i]): # удачный шаг  
 l = 0  
 y[i + 1] = y[i] + deltas[i] \* d[i]  
 deltas[i] = a \* deltas[i]  
 else:  
 l += 1  
 y[i + 1] = y[i]  
 deltas[i] = b \* deltas[i]  
 if i < n:  
 i += 1  
 continue  
 else:  
 if func(y[n + 1]) < func(y[1]):  
 y[1] = y[n + 1]  
 i = 1  
 continue  
 else:  
 if not (func(y[n + 1]) < func(x[k])):  
 if l <= N:  
 if all([abs(t) <= e for t in deltas]):  
 result = x[k]  
 break  
 else:  
 y[1] = y[n + 1]  
 continue  
 x[k + 1] = y[n + 1]  
 if (  
 (x[k + 1][0] - x[k][0]) \*\* 2 + (x[k + 1][1] - x[k][1]) \*\* 2  
 ) \*\* 0.5 <= e:  
 result = x[k + 1]  
 break  
 else:  
 lambdas = {}  
 lambdas[1] = x[k + 1][0] - x[k][0]  
 lambdas[2] = x[k + 1][1] - x[k][1]  
 alphas = {}  
 for i in range(1, n + 1):  
 if lambdas[i] == 0:  
 alphas[i] = d[i]  
 else:  
 alphas[i] = sum([lambdas[j] \* d[j] for j in range(i,  
 n + 1)])  
 bs = {}  
 d = {}  
 for i in range(1, n + 1):  
 if i == 1:  
 bs[i] = alphas[i]  
 else:  
 bs[i] = alphas[i] - (alphas[1] \* d[1])[1] \* d[1]  
 d[i] = bs[i] / ((bs[i][0] \*\* 2 + bs[i][1] \*\* 2) \*\* 0.5)  
 i = 1  
 y[1] = x[k + 1]  
 k += 1  
 deltas = {i + 1: deltas\_0[i] for i in range(n)}  
 return result

**5 Анализ результатов**

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

Найдём частные производные 1-го порядка:

z'x = 4x + 2y - 2

z'y = 2x + 2y

Приравняем их к нулю и решим систему:

M0: x = 1; y = -1

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

A = z''xx(M0) = 4

B = z''xy(M0) = 2

C = z''yy(M0) = 2

AC - B^2 > 0

4\*2 - 2^2 > 0

4>0

При этом A>0, следовательно точка (1;-1) - минимум функции.

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

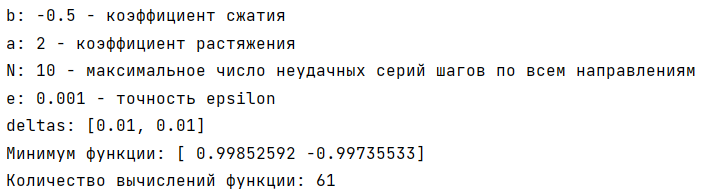


Рисунок 2 – Результат работы метода

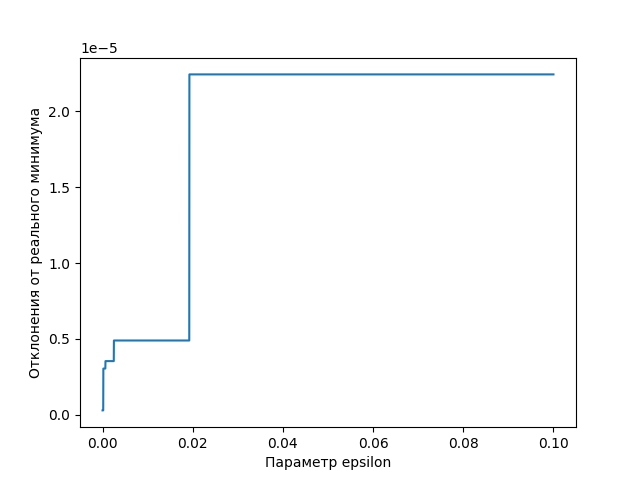


Рисунок 3 – График отклонения от реального минимума для epsilon

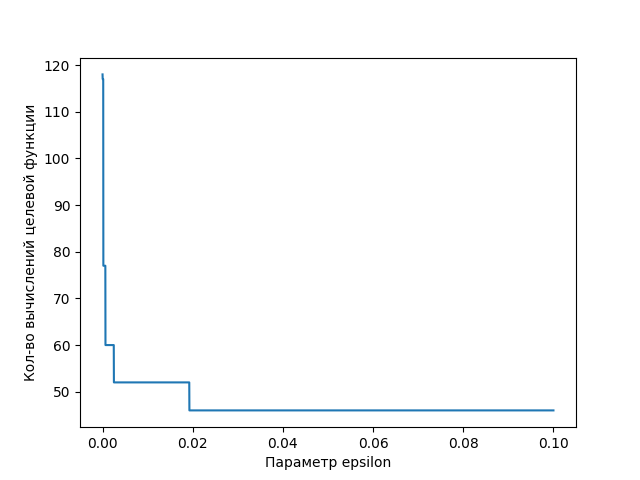


Рисунок 4 – График количества вычислений целевой функции для epsilon

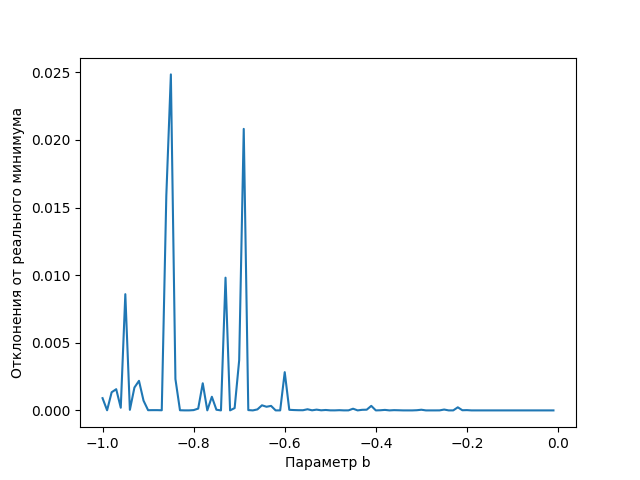


Рисунок 5 – График отклонения от реального минимума для beta

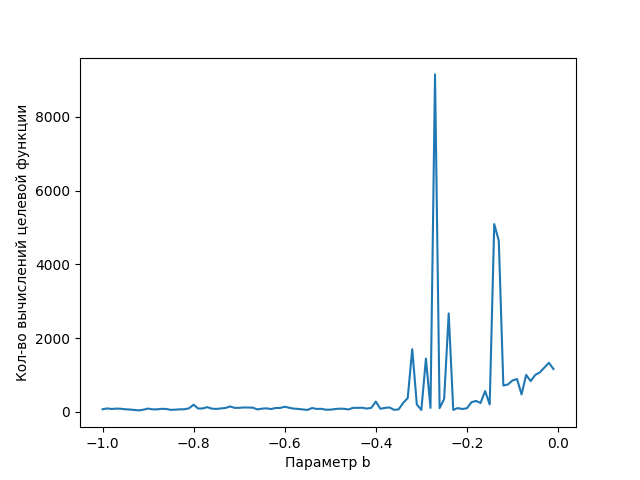


Рисунок 6 – График количества вычислений целевой функции для beta

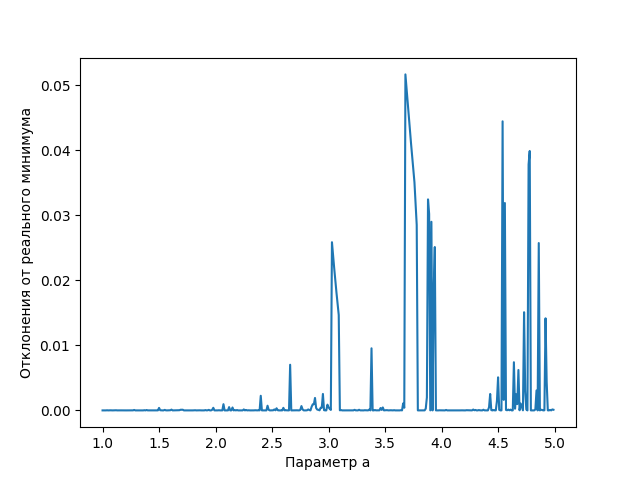


Рисунок 7 – График отклонения от реального минимума для alpha

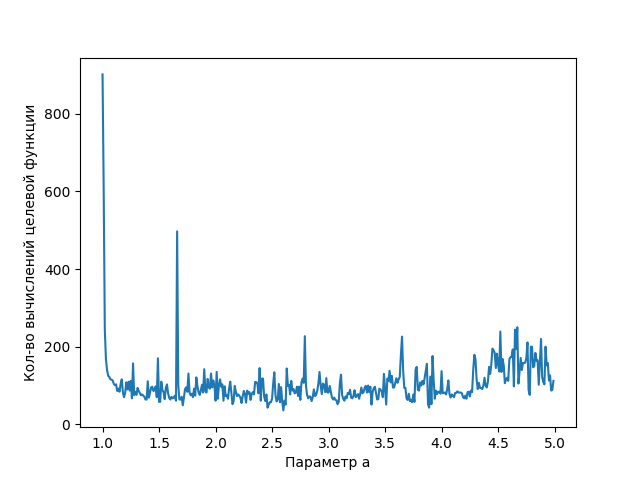


Рисунок 8 – График количества вычислений целевой функции для alpha

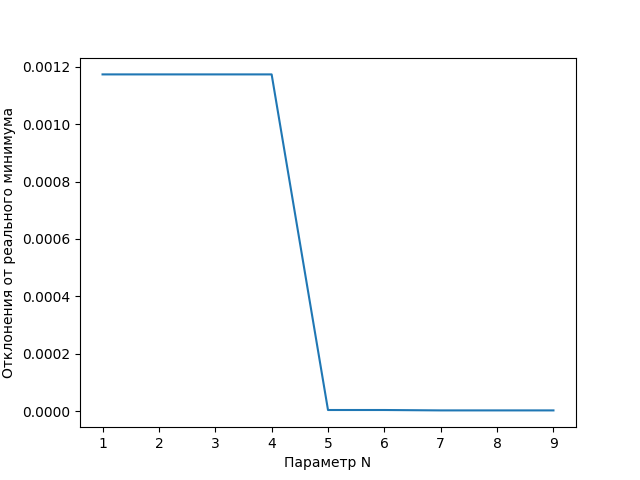


Рисунок 9 – График отклонения от реального минимума для N

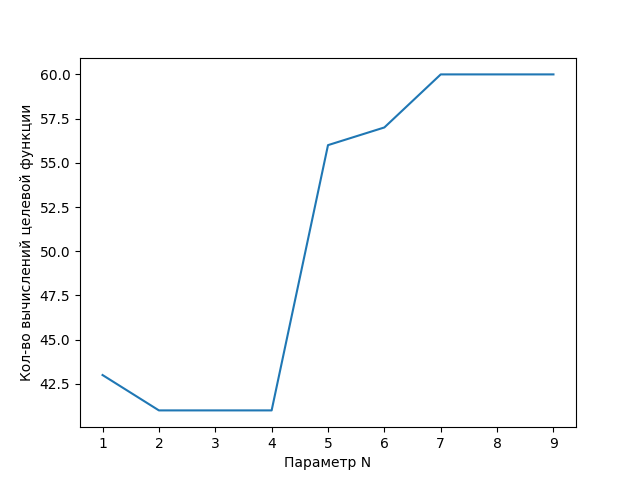


Рисунок 10 – График количества вычислений целевой функции для N

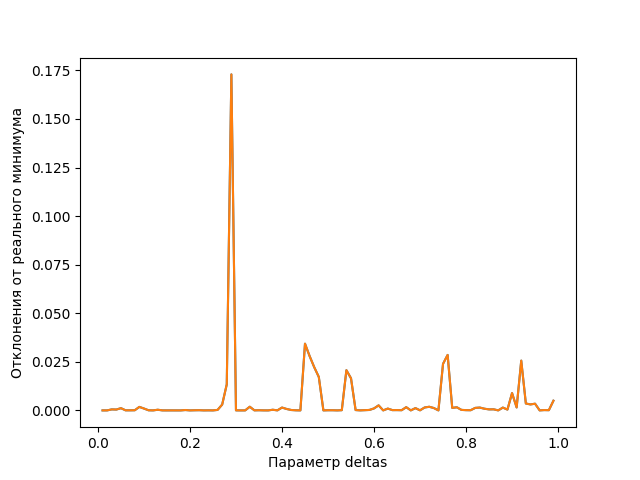


Рисунок 11 – График отклонения от реального минимума для deltas

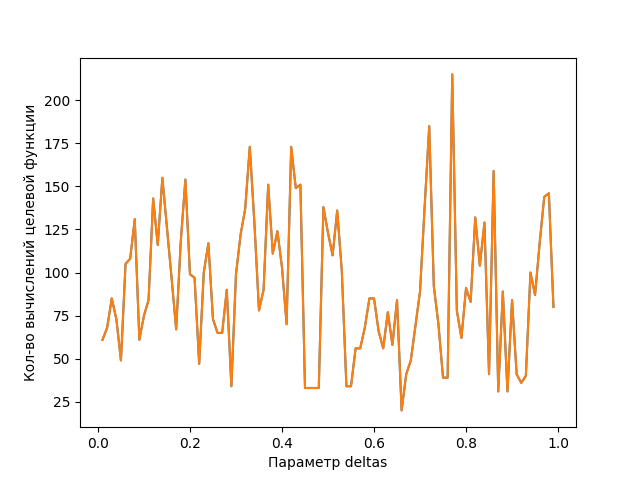


Рисунок 12 – График количества вычислений целевой функции для deltas

В результате можно сказать, что найденный минимум функции  
соответствует реальному. При этом можно определить следующее влияние  
параметров на результат: чем больше эпсилон тем больше отклонение, но  
меньше кол-во вычислений. При меньших beta больше отклонение, при  
больших больше вычислений функции. При меньших alpha больше вычислений  
функции, при больших больше отклонение. При меньших N больше  
отклонение, при больших больше вычислений функции.

**6 Вывод**

При выполнении задания был успешно реализован метод Розенброка,  
результаты работы метода сравнены с реальным и близки к нему, исследована

зависимость работы метода от значений его параметров.