Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Национальный исследовательский университет «МЭИ»

ФГБОУ ВО «НИУ «МЭИ»

Кафедра инновационных технологий наукоемких отраслей

Отчет по практике

Выполнил:

студент группы ТФэ-01-20

Бонадыков Никита Михайлович

Содержание

[Введение 3](#_Toc138932148)

[1. Одномерная оптимизация: 4](#_Toc138932149)

[1.1. Методы нулевого порядка: 4](#_Toc138932150)

[1.1.1. Оптимальный пассивный поиск 4](#_Toc138932151)

[1.1.2. Деление отрезка пополам 7](#_Toc138932152)

[1.1.3. Метод Фибоначчи 9](#_Toc138932153)

[1.1.4. Метод золотого сечения 11](#_Toc138932154)

[1.2. Методы первого порядка 14](#_Toc138932155)

[1.2.1. Метод средней точки 14](#_Toc138932156)

[1.2.2. Метод Ньютона 15](#_Toc138932157)

[1.2.3. Метод секущей 18](#_Toc138932158)

[2. Многомерная оптимизация 20](#_Toc138932159)

[2.1. Методы нулевого порядка 20](#_Toc138932160)

[2.1.1. Покоординатный спуск 20](#_Toc138932161)

[2.1.2. Метод Нелдера - Мида 20](#_Toc138932162)

[2.2. Методы первого порядка 20](#_Toc138932163)

[2.2.1. Метод градиентного спуска 20](#_Toc138932164)

# Введение

Одно из ключевых направлений в проектировании изделий и технологических процессов заключается в оптимизации (минимизации или максимизации) некоторой характеристики

Отсюда возникает задача как одномерной минимизации, так и многомерной.

Различают различные методы оптимизации, ниже приведен список самых популярных:

1. Одномерная оптимизация:
   1. Методы нулевого порядка:
      1. Оптимальный пассивный поиск
      2. Деление отрезка пополам
      3. Метод Фибоначчи
      4. Метод золотого сечения
   2. Методы первого порядка:
      1. Метод средней точки
      2. Метод Ньютона
      3. Метод секущей
2. Многомерная оптимизация:
   1. Методы нулевого порядка:
      1. Покоординатный спуск
      2. Метод Нелдера-Мида
   2. Методы первого порядка
      1. Метод градиентного спуска

Рассмотри же каждый выше написанный метод более подробно, с помощью языка программирования Python.

# Одномерная оптимизация:

## Методы нулевого порядка:

### Оптимальный пассивный поиск

Оптимальный пассивный поиск – наверное самый простой и понятный способ нахождения экстремума функции.

Метод состоит в том, чтоб получить значения функции в каждой исследуемой точке и конце сравнить все полученные значения для выбора максимума или минимума целевой функции.

Для примера возьмем следующею функцию:

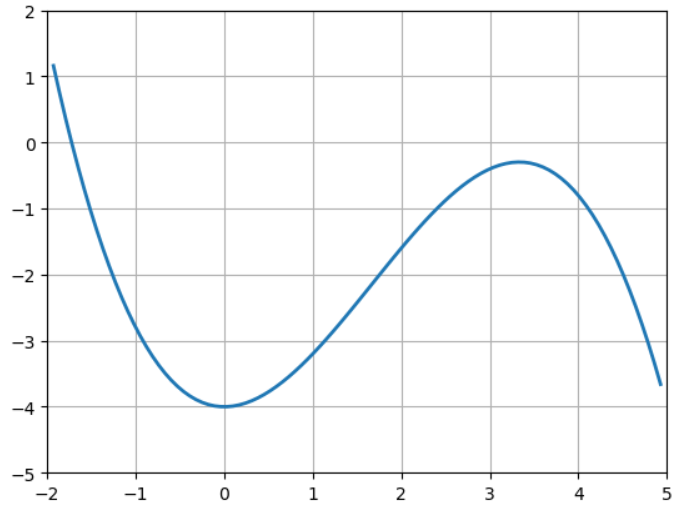
График которой в интервале от [-2, 5] имеет вид:

Рисунок 1. График целевой функции

Как мы видим у данной функции в данном интервале существует две точки экстремума точка минимума с координатами [0, -4] и точка максимума с координатами [3.33, -0.3]. В дальнейших методах будем использовать данную функцию.

На языке программирования функция по нахождению точки минимума будет выглядеть следящим образом:

Рисунок 2.Блок схема оптимального пассивного поиска

Код и вывод результата Вы можете наблюдать в Приложении №1.

Результатом выполнения функции является построения графика функции с точкой минимума, а также вывод самого значения *x* при котором функция принимает минимальное значение.

Минусом такого метода является погрешность так как что бы найти значение с точностью 10-1 необходимо исследовать 9 точек. А для значения с точностью 10-2 необходимо исследовать уже 99 точек.

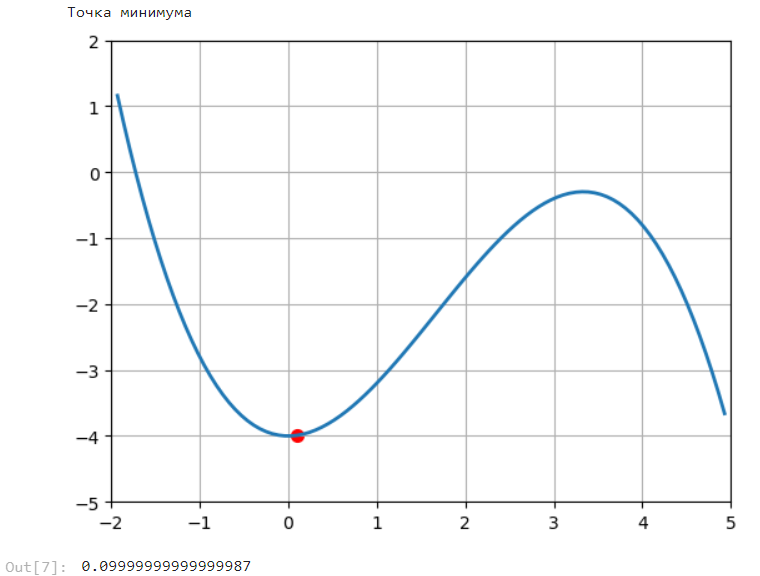
 Вышеприведённый график иллюстрирует решение данной задачи с точностью 10-1.

Рисунок 3. Точка минимума целевой функции

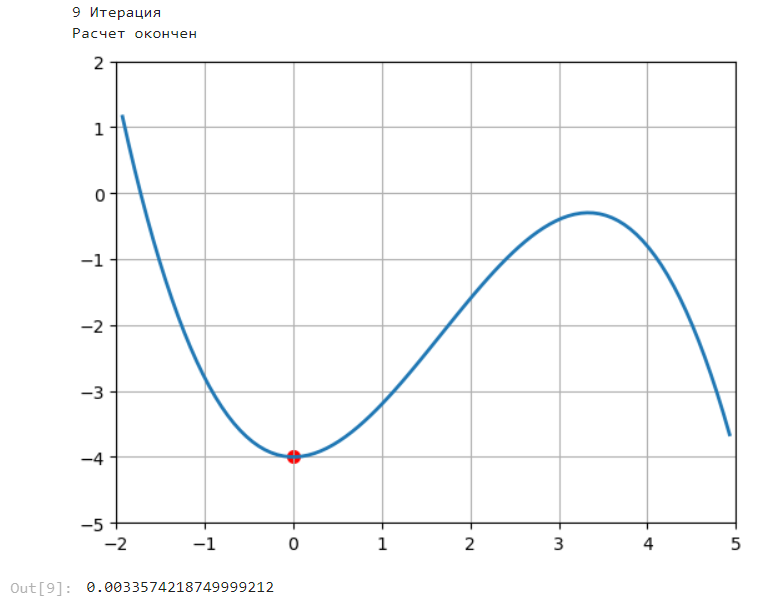
### Деление отрезка пополам

Деление отрезка пополам – тоже достаточно простой метод нахождения экстремума функции. В нем используется принцип последовательного сокращения отрезка локализации.

На языке программирования функция по нахождению точки минимума будет выглядеть следящим образом:



Рисунок 4.Блок схема метода деления отрезка пополам

Результатом выполнения функции является построения графика функции с точкой минимума, а также вывод самого значения *x* при котором функция принимает минимальное значение.

Вышеприведённый график иллюстрирует решение данной функции с точностью 10-2. Данное решение получено спустя 9 итераций так же, как и прошлым методом, но в этот раз точность на порядок выше.

### Метод Фибоначчи

Метод Фибоначчи является оптимальным последовательным методом, т. е. методом, обеспечивающим максимальное гарантированное сокращение отрезка локализации при заданном числе N вычислений функции. Этот метод основан на использовании чисел Фибоначчи.

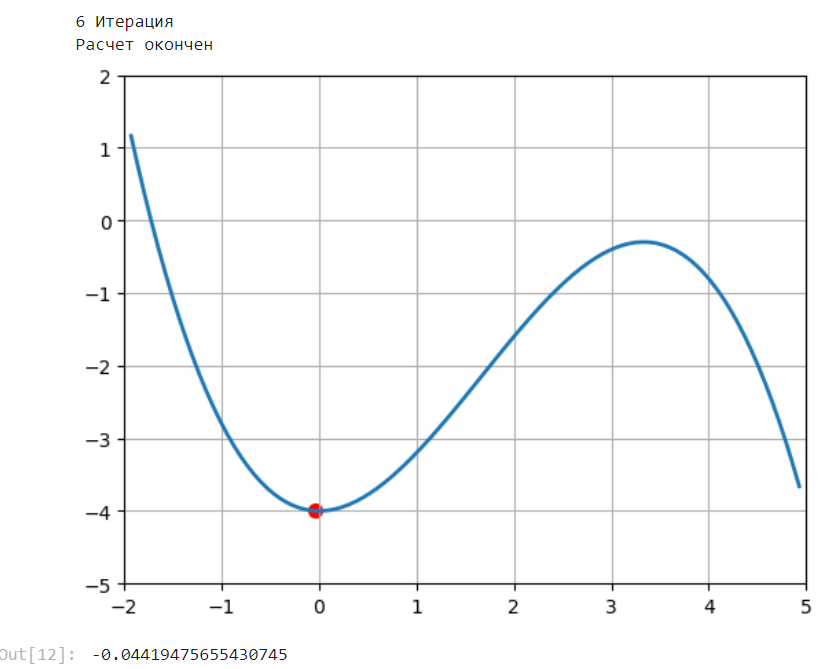
Алгоритм нахождения минимума:Результатам выполнения данного алгоритма будет значение минимума функции и график нахождения данного минимума.

Рисунок 5. Блок схема метода Фибоначчи

Рисунок 6. Точка минимума целевой функции

Как можно заметить, что при точности 10-2 для нахождения минимума потребовалось 6 итераций, что говорит о высокой скорости нахождения минимума.

Хотя метод Фибоначчи и оптимален в указанном выше смысле зачастую он не неудобен для использования.

### Метод золотого сечения

Метод золотого сечения — метод поиска экстремума действительной функции одной переменной на заданном отрезке. В основе метода лежит принцип деления отрезка в пропорциях золотого сечения. Является одним из простейших вычислительных методов решения задач оптимизации. Впервые представлен Джеком Кифером в 1953 году.

Алгоритм:

* + 1. На первой итерации заданный отрезок делится двумя симметричными относительно его центра точками и рассчитываются значения в этих точках.
    2. После чего тот из концов отрезка, к которому среди двух вновь поставленных точек ближе оказалась та, значение в которой максимально (для случая поиска минимума), отбрасывают.
    3. На следующей итерации в силу показанного выше свойства золотого сечения уже надо искать всего одну новую точку.
    4. Процедура продолжается до тех пор, пока не будет достигнута заданная точность.

Результатом выполнения функции является построения графика функции с точкой минимума, а также вывод самого значения *x* при котором функция принимает минимальное значение.

Рисунок 7. Блок схема метода золотого сечения

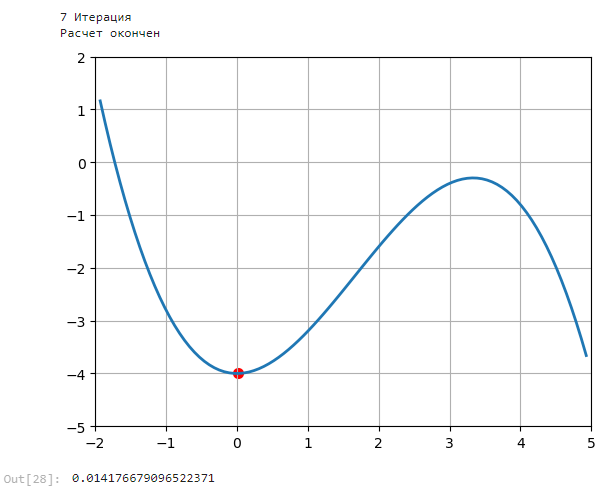


Рисунок 8. Точка минимума целевой функции

Как видно из графика точка минимума находится за 7 итераций с точностью 10-2.

## Методы первого порядка

### Метод средней точки

Алгоритм метода средней точки основан на сокращении длины текущего отрезка неопределенности [a;b], путем отбрасывания той половины отрезка, которая не содержит точки минимума. В основу метода положено основное свойство унимодальной функции, то есть, для того чтобы на отрезке [a;b] существовал минимум, необходимо, чтобы первая производная на нем была неубывающей.

Рисунок 9. Блок схема метода средней точки

Результатом выполнения функции является построения графика функции с точкой экстремума, а также вывод самого значения x при котором функция принимает минимальное или максимальное значение.

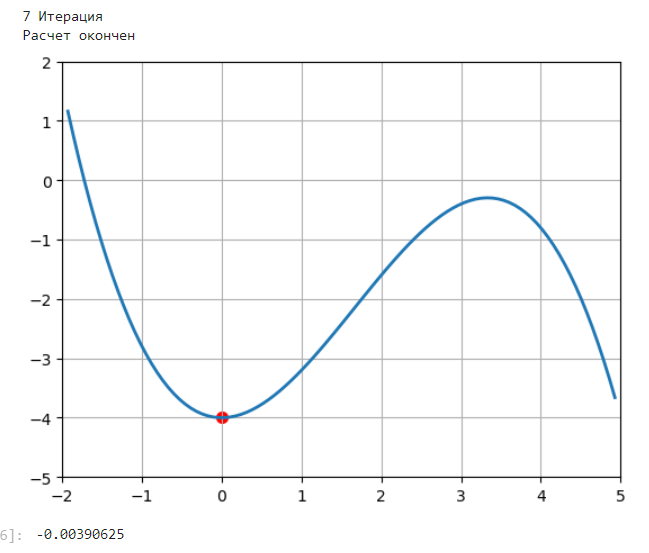


Рисунок 10. Точка минимума целевой функции

### Метод Ньютона

Метод Ньютона, алгоритм Ньютона (также известный как метод касательных) — это итерационный численный метод нахождения корня (нуля) заданной функции. Метод был впервые предложен английским физиком, математиком и астрономом Исааком Ньютоном (1643—1727). Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Метод обладает квадратичной сходимостью. Модификацией метода является метод хорд и касательных. Также метод Ньютона может быть использован для решения задач оптимизации, в которых требуется определить ноль первой производной либо градиента в случае многомерного пространства.

Алгоритм решения задачи с помощью метода ньютона выглядит следующим образом:

Рисунок 11. Блок схема метода Ньютона

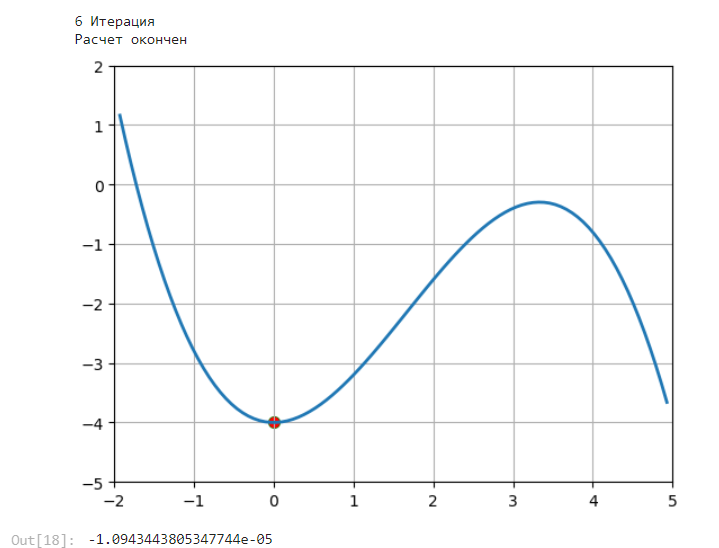
Результатом выполнения функции является построения графика функции с точкой экстремума, а также вывод самого значения x при котором функция принимает минимальное или максимальное значение. 

Рисунок 12. Точка минимума целевой функции

### Метод секущей

В численном анализе метод секущих представляет собой алгоритм поиска корней, который использует последовательность корней из секущих линий для лучшей аппроксимации корня функции f. Метод секущих можно рассматривать как конечно-разностную аппроксимацию метода Ньютона. Однако метод секущих предшествует методу Ньютона более чем на 3000 лет.

Рисунок 13. Блок схема метода секущей

Результатом выполнения функции является построения графика функции с точкой экстремума, а также вывод самого значения x при котором функция принимает минимальное или максимальное значение.

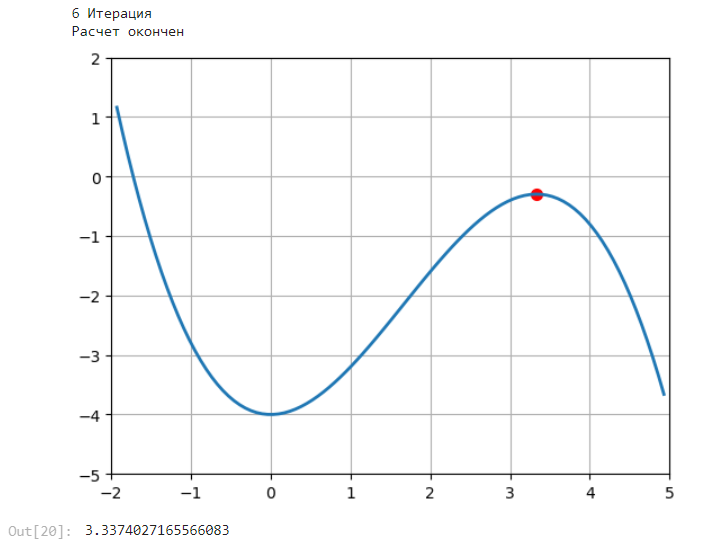
На графики видно, что, выполнив алгоритм можно найти экстремум в нашем случае точку максимума так как производная в нем примерно равна 0.

Рисунок 14. Точка экстремума целевой функции

# Многомерная оптимизация

## Методы нулевого порядка

### Покоординатный спуск

Алгоритм оптимизации, который последовательно минимизирует по координатным направлениям, чтобы найти минимум функции. На каждой итерации алгоритм определяет координату или блок координат с помощью правила выбора координат, затем точно или неточно минимизирует по соответствующей координатной гиперплоскости, фиксируя при этом все остальные координаты или блоки координат.



Рисунок 15. Блок схема метода покординатного спуска

Для проверки данного алгоритма возьмем одну из тестовых функций, к примеру функцию Химмельблау:

График которой имеет вид:

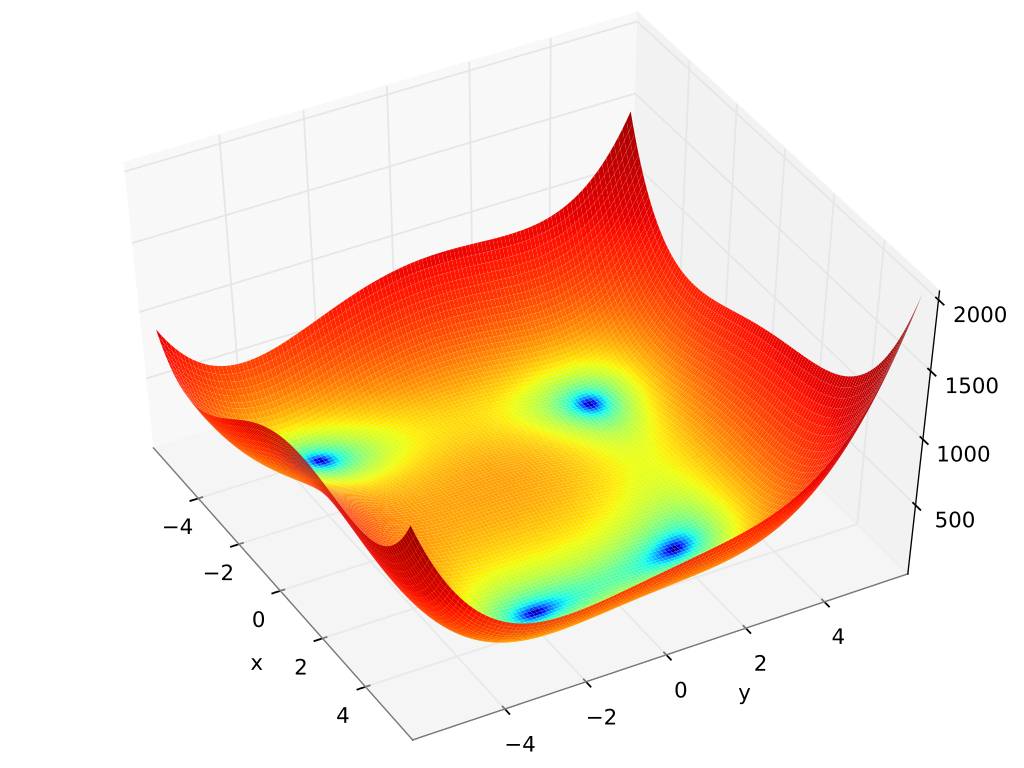


Рисунок 16. График целевой функции в изометрии

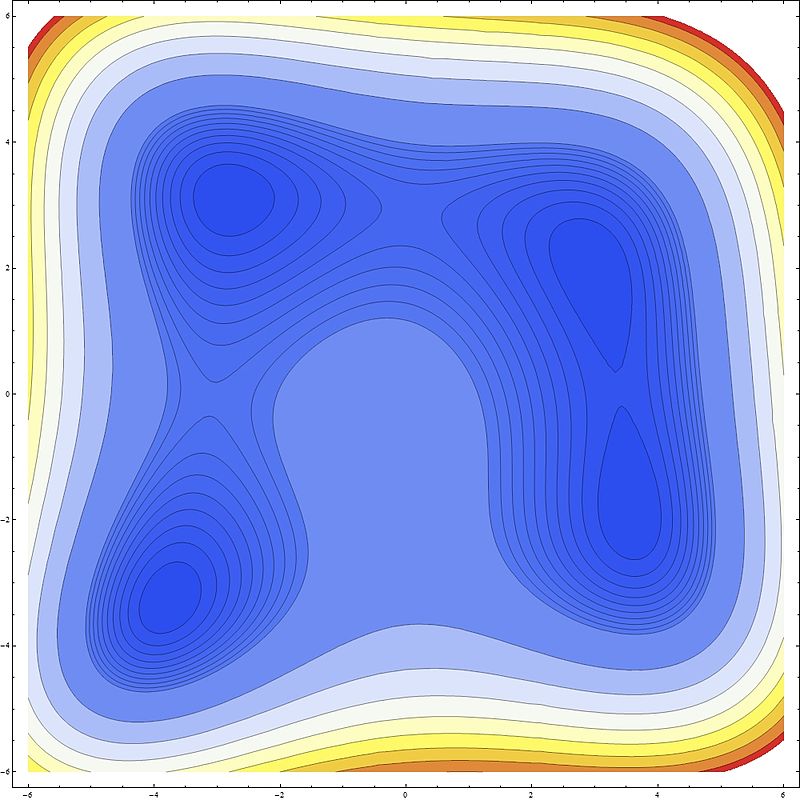


Рисунок 17. График целевой функции в изолиния

С одним локальным максимумом

И четырьмя различными локальными минимумами:

Эту же функцию мы будем использовать далее для исследования многомерных методов.

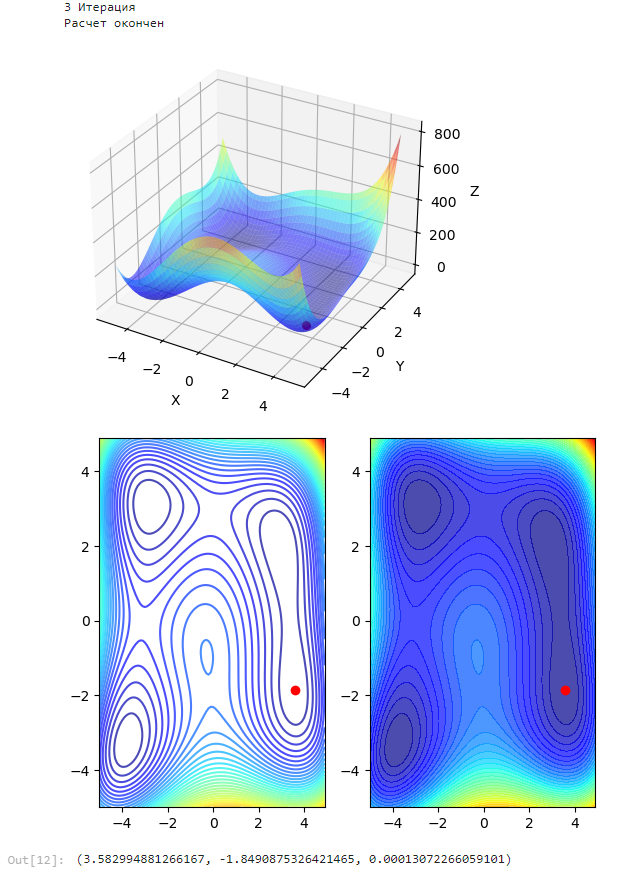
Выполнив расчет экстремума методом покоординатного спуска, получаем в итоге координаты x, y и график с точкой экстремума.

Рисунок 18. Графики нахождения минимума целевой функции

Хоть данный метод и смог определить экстремум за 3 итерации, но в то же момент он три раза искал минимум по каждой координате что в общем замедляло расчет.

### Метод Нелдера – Мида

Метод Нелдера — Мида, также известный как метод деформируемого многогранника и симплекс-метод, — метод безусловной оптимизации функции от нескольких переменных, не использующий производной (точнее — градиентов) функции, а поэтому легко применим к негладким и/или зашумлённым функциям.

Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума.

Метод находит локальный экстремум и может «застрять» в одном из них.

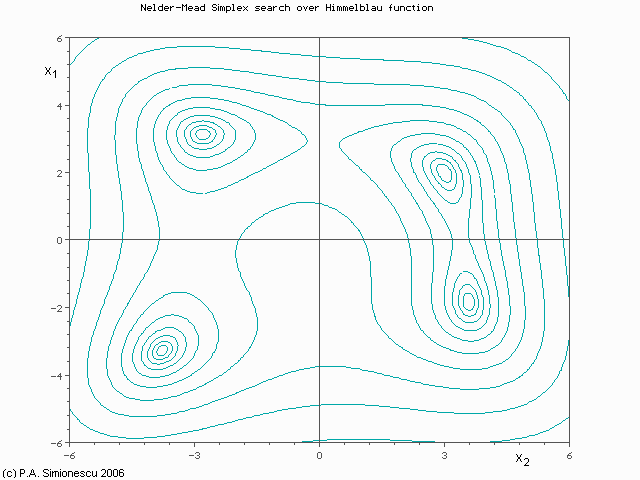


Рисунок 19.Последовательные симплексы в методе Нелдера-Мида для функции Химмельблау

Алгоритм заключается в формировании симплекса (simplex) и последующего его деформирования в направлении минимума, посредством трех операций:

1) Отражение (reflection);

2) Растяжение (expansion);

3) Сжатие (contract);

Рисунок 20. Блок схема метода Нейдера - Мида

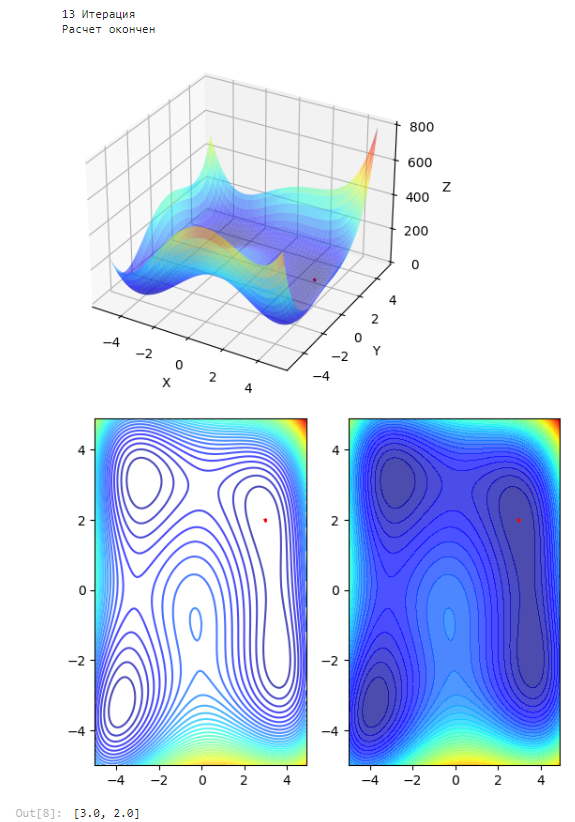
Выполнив расчет экстремума методом Нейдера - Мида, получаем в итоге координаты x, y и график с точкой экстремума.

Рисунок 21. Поиск экстремума целевой функции.

## Методы первого порядка

### Метод градиентного спуска

Градиентный спуск, метод градиентного спуска — численный метод нахождения локального минимума или максимума функции с помощью движения вдоль градиента, один из основных численных методов современной оптимизации.

Активно используется в вычислительной математике не только для непосредственного решения задач оптимизации (минимизации), но и для задач, которые могут быть переписаны на языке оптимизации (решение нелинейных уравнений, поиск равновесий, обратные задачи и т. д.). Метод градиентного спуска можно использовать для задач оптимизации в бесконечномерных пространствах, например, для численного решения задач оптимального управления.

Особенно большой интерес к градиентным методам в последние годы связан с тем, что градиентные спуски и их стохастические / рандомизированные варианты лежат в основе почти всех современных алгоритмов обучения, разрабатываемых в анализе данных.



Рисунок 22. Блок схема градиентного метода

Выполнив расчет экстремума методом градиентного спуска, получаем в итоге координаты x, y и график с точкой экстремума.

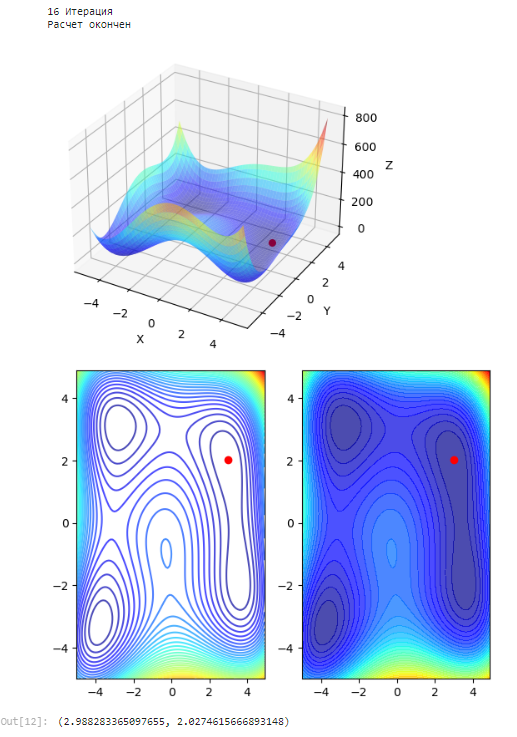


Рисунок 23. График экстремума целевой функции