Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

УНИВЕРСИТЕТ ИТМО УЧЕБНЫЙ ЦЕНТР ОБЩЕЙ ФИЗИКИ ФТФ

Группа: М32101

Студент: Косовец Роман / Дмитрий Надеждин

Преподаватель: Дорофеева Юлия Александровна

Рабочий протокол и отчет по лабораторной работе № 3

«Методы спуска»

1. Цель работы:

Реализовать перечисленные ниже методы спусков:

- Градиентный спуск с постоянным шагом
- Градиентный спуск с дроблением шага, используя условие Армихо
- Метод наискорейшего спуска
- Метод сопряженных градиентов

2. Задачи, решаемые при выполнении работы:

1. Проанализировать траектории предложенных алгоритмов на примере квадратичных функций.
2. Сравнить эффективность методов с точки зрения количества вычислений.
3. Исследовать работу методов в зависимости от выбора начальной точки

3. Условие задачи:

Прикладная математика (весна'23)

Лабораторная работа # 3

Методы спуска

Предполагаемый язык выполнения лабораторных работ Python 3. Лабораторные работы выполняются студентами индивидуально или в группах по 2-3 человека (по желанию). По результатам выполнения лабораторной работы необходимо подготовить отчет. Отчет должен содержать описание реализованных вами алгоритмов, ссылку на реализацию, необходимые тесты и таблицы.

Постановка задачи

1. Реализуйте градиентный спуск с постоянным шагом.
2. Реализуйте алгоритм спуска с дроблением шага, используя условие Армихо.
3. Реализуйте метод наискорейшего спуска (для этого выберите произвольный метод одномерной оптимизации).
4. Реализуйте метод сопряженных градиентов.
5. Проанализируйте траектории предложенных алгоритмов на примере квадратичных функций. Для этого придумайте две-три квадратичные функции от двух переменных, на которых работа методов будет отличаться.
6. Для каждой функции:
   (a) исследуйте сходимость градиентного спуска с постоянным шагом, сравните полученные результаты для выбранных функций;
   (b) сравните эффективность методов с точки зрения количества вычислений минимизируемой функции и ее градиентов;
   (c) исследуйте работу методов в зависимости от выбора начальной точки;
   (d) в каждом случае нарисуйте графики с линиями уровня и траекториями методов;
7. Реализуйте генератор случайных квадратичных функций n переменных с числом обусловленности k.
8. Исследуйте зависимость числа итераций $T(n, k)$, необходимых градиентному спуску для сходимости в зависимости от размерности пространства $2 \leqslant n \leqslant 10^3$ и числа обусловленности оптимизируемой функции $1 \leqslant k \leqslant 10^3$.
9. По возможности для получения более корректных результатов проведите множественный эксперимент и усредните полученные значения относительно числа итераций.

Важно: для избежания потерь точности, используйте аналитически вычисленный градиент, а не его аппроксимации при помощи конечных разностей.

4. Решение задачи:

Для дальнейшего решения задачи и описания методов спуска возьмем две квадратичные функции (функцию Розенброка и Химмельблау):

- $(x^2 + y - 11)^2 + (x + y^2 - 7)^2$ (Himmelblau)
- $x^2 + y^2 - xy$ (func1)

Для функции func1 были выбраны следующие входные параметры для дальнейшего исследования:

- Initial points = [5; 8]
- Learning rate = 0.01
- Eps = 0.00001
- Search range = [-10; 10]

Для функции Химмельблау были выбраны следующие входные параметры для дальнейшего исследования:

- Initial points = [-2; 2]
- Learning rate = 0.001
- Eps = 0.00001
- Search range = [-4; 4]

Код всех методов (Python)
https://github.com/RomanKosovets/PriMat/tree/main/Lab_3

5. Градиентный спуск с постоянным шагом:

Градиентный спуск с постоянным шагом — это итерационный численный метод оптимизации, используемый для нахождения минимума функции. Он основан на использовании градиента функции для определения направления наискорейшего убывания функции и обновления текущей точки.

Преимущества градиентного спуска с постоянным шагом:

- Простата реализации.
- Эффективность на выпуклых функциях с гладкой поверхностью градиентный спуск с постоянным шагом может достичь глобального минимума.
- Низкие вычислительные требования: метод не требует хранение матриц или обратных матриц, что позволяет снизить вычислительную сложность.

Недостатки градиентного спуска с постоянным шагом:

- Зависимость от learning rate: выбор неправильного значения может замедлить сходимость или привести к расхождению алгоритма
- Попадание в локальные минимумы. Этот метод не гарантирует нахождение глобального минимума на невыпуклых функциях
- Чувствительность в инициализации. Начальная точка существенно влияет на результат градиентного спуска, особенно при наличии нескольких локальных минимумов

Результаты работы метода на двух выбранных функциях:

На функции func1 алгоритм сошелся и закончил свое выполнение успешно, при этом было затрачено 566 итерации для точного нахождения точки минимума:

- Кол-во итераций = 566
- Точка минимума = [0.02200; 0.02200]

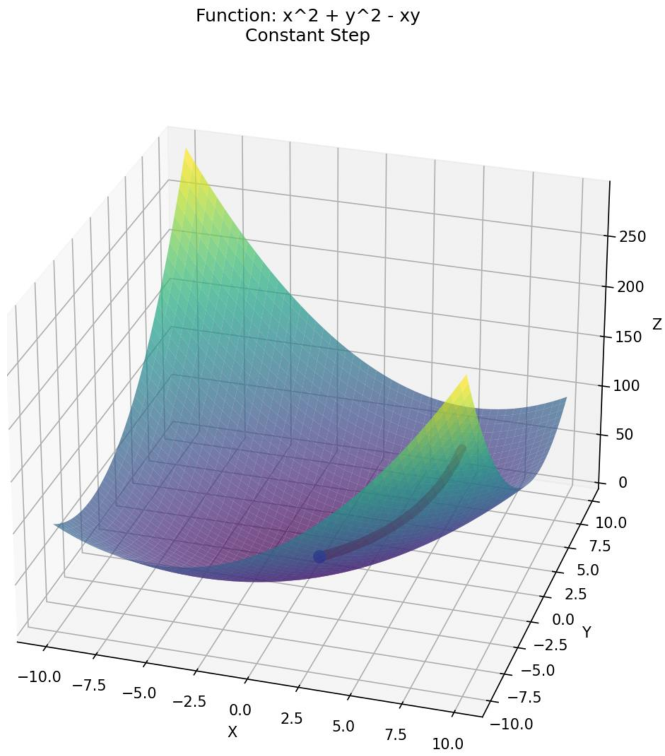

График функции func1

На функции Химмельблау метод также сошелся и закончил свое выполнение успешно, при этом было затрачено всего 106 итераций для точного нахождения точки минимума:

- Кол-во итераций = 106
- Точка минимума = [-2.80380; 3.130913]

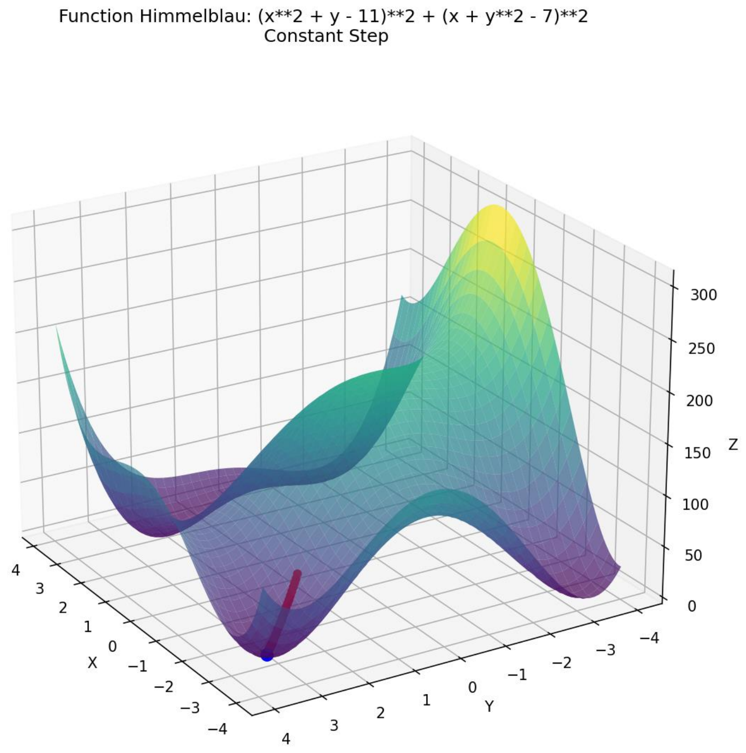

График функции Химмельблау

6. Градиентный спуск с дроблением шага:

Градиентный спуск с дроблением шага по условию Армихо – этот алгоритм является одним из методов оптимизации, который позволяет эффективно выбирать размер шага на каждой итерации градиентного спуска. Этот метод помогает улучшить сходимость и скорость сходимости алгоритма. Условие Армихо гарантирует, что шаг является достаточно малым, чтобы градиентный спуск сходился, и что уменьшение функции достаточно значительно.

Преимущества градиентного спуска с дроблением шага:

- Адаптивность изменения шага. Этот метод позволяет выбирать оптимальный размер шага на каждой итерации, исходя из локальной информации о функции. Это помогает справиться с проблемами, такими как большие значения градиента в функции, где фиксированный размер шага может быть недостаточно эффективным.
- Улучшение сходимости. Данный алгоритм может сходиться быстрее, чем градиентный спуск с фиксированным шагом. Алгоритм быстрее приближается к оптимальному решению благодаря адаптивному шагу

Недостатки градиентного спуска с дроблением шага:

- Вычислительная сложность. Включения дополнительного определения оптимального шага может привести к дополнительным вычислительным затратам.
- В редких случаях алгоритм может выбрать слишком маленький размер шага, что может замедлить сходимость алгоритма.
- Чувствительность к условию Армихо. Выбор параметром для условия Армихо может оказаться нетривиальной задачей. Например, выбор слишком большого значения alpha может привести к слишком большим шагам, которые могут вызвать расходимость.

В целом, градиентный спуск с дроблением шага и условием Армихо является мощным инструментом для оптимизации функции. Он обеспечивает адаптивность размера шага и улучшает сходимость алгоритма.

Результаты работы метода на двух выбранных функциях:

На выбранных функциях (func1 и Химмельблау), с данными параметрами изменилось количество итераций.

На функции func1 алгоритм сошелся и закончил свое выполнение успешно, при тех же выходных данных, что и при градиентном спуске с постоянным шагом:

- Кол-во итераций = 66
- Точка минимума = [0.0221; 0.03210]

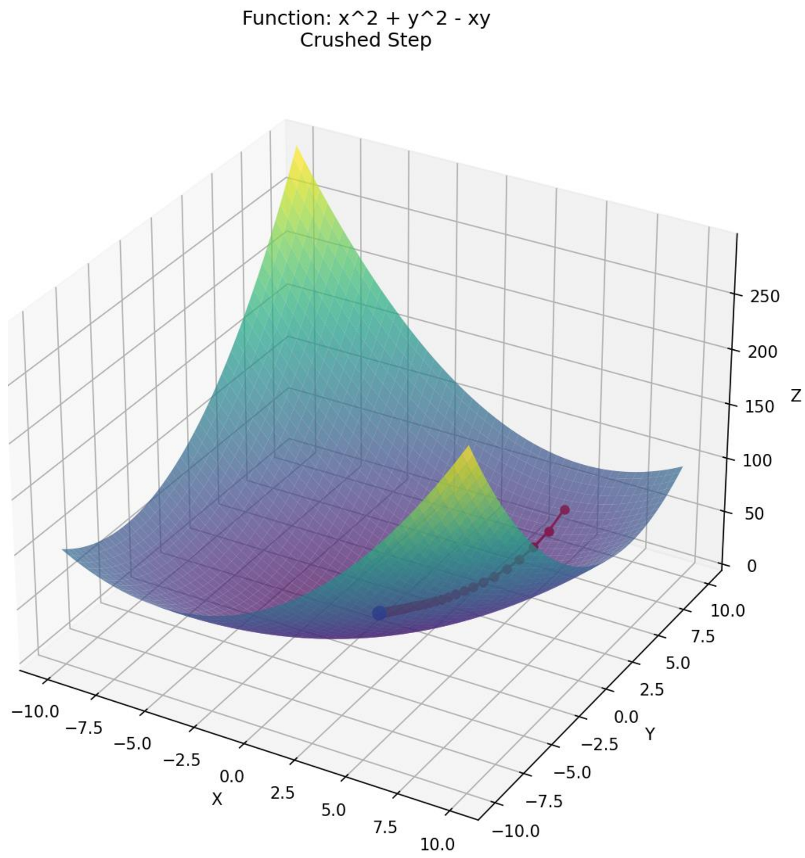

График функции func1

На функции Химмельблау метод также сошелся при тех же параметрах, количество итераций стало меньше:

- Кол-во итераций = 86
- Точка минимума = [-2.80380; 3.130913]

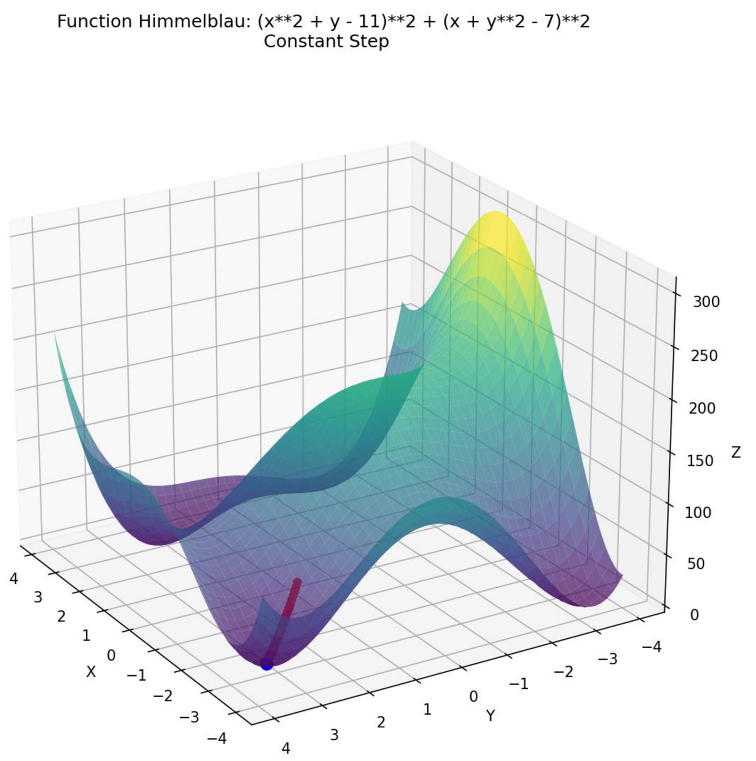

График функции Химмельблау

7. Метод наискорейшего спуска:

Метод наискорейшего спуска – этот метод является одним из вариантов градиентного спуска, который основывается на выборе наиболее крутого направления спуска в каждой точке. Вместо постоянного шага, как в обычном градиентном спуске, метод наискорейшего спуска адаптивно выбирает оптимальный шаг вдоль выбранного направления.

Преимущества метода наискорейшего спуска:

- Простота реализации.
- Глобальная сходимость. Данный метод гарантирует глобальную сходимость к локальному минимуму, если функция является выпуклой и дифференцируемой.
- Независимость от начальной точки. В отличие от алгоритма с постоянным шагом, метод наискорейшего спуска не требует тщательного выбора начального шага, так как шаг определяется адаптивно на каждой итерации.

Недостатки метода наискорейшего спуска:

- Вычислительная сложность. На каждой итерации метода требуется вычислять градиент функции, что может быть затратным для больших и сложных функций.

- Чувствительность к выбору шага. Выбор оптимального шага может быть сложной задачей. Слишком маленький шаг может привести к медленной сходимости, а слишком большой к расходимости.

В целом, метод наискорейшего спуска является простым и глобально сходящимся методом оптимизации. Однако есть вероятность медленной сходимости, а также вычислительная сложность может ограничивать его применение в некоторых случаях.

Результаты работы метода на двух выбранных функциях:

Метод наискорейшего спуска при исследовании двух функций действительно оказался намного эффективнее. Для его реализации мы использовали алгоритм золотого сечения, который принимает на вход интервал [a, b], где a и b – начальные значения для определения шага. Затем алгоритм делит интервал на две части для выбора нового шага. Вычисляются значения функции на новых точках и осуществляется сравнение.

На функции func1 алгоритм сошелся и закончил свое выполнение успешно, при этом количество итераций уменьшилось во много раз (86 против 12), а точность в свою очередь немного увеличилась:

- Кол-во итераций = 12
- Точка минимума = [0.0020; 0.0022]

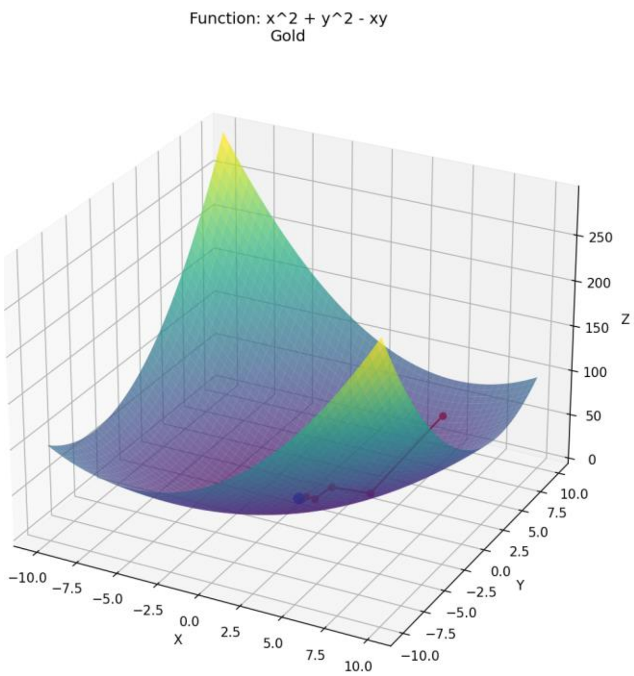

График функции func1

На функции Химмельблау метод также сошелся при тех же параметрах, количество итераций уменьшилось значительно (86 против 5):

- Кол-во итераций = 5
- Точка минимума = [-2.80510; 3.13131]

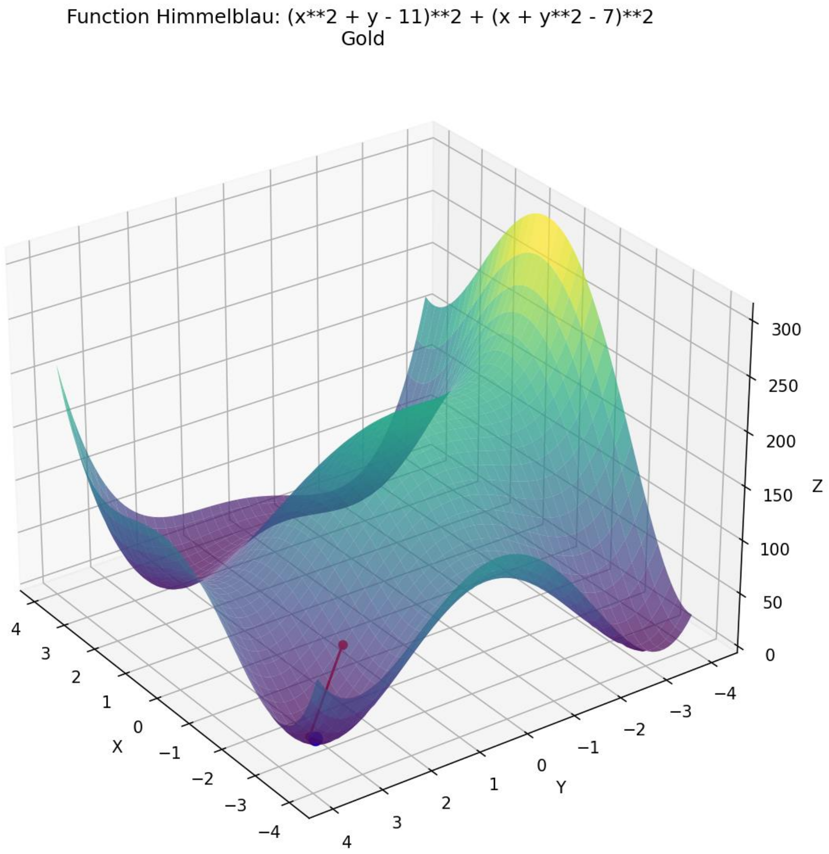

График функции Химмельблау

8. Метод Сопряженных градиентов:

Метод Сопряженных градиентов — это итерационный метод для безусловной оптимизации в многомерном пространстве. Основным достоинством метода является то, что он решает квадратичную задачу оптимизации за конечное число шагов.

Преимущества метода сопряженных градиентов:

- Быстрая сходимость. Метод способен достичь оптимального решения за конечное число итераций, равное числу переменных в задаче.
- Эффективное использование памяти. Метод требует хранения только текущего градиента и нескольких дополнительных векторов. Это делает его практичным для работы с большими системами уравнений.
- Независимость от начальной точки. Метод обладает свойством независимости от начальной точки, то есть он сходится к оптимальному решению, независимо от выбора начального приближения.

Недостатки метода сопряженных градиентов:

- Ограничения на класс функций. Данный метод может быть неэффективным для функций с большой кривизной или для не квадратичных функций, так как может потребоваться большое число итераций для достижения оптимального решения.
- Чувствительность к условиям задачи. Метод сопряженных градиентов чувствителен к нестабильностям и плохо масштабируется для задач с неравномерными условиями численной устойчивости.

В целом, метод сопряженных градиентов является мощным и эффективным методом оптимизации, особенно для задач с квадратичными функциями. Он обладает несколькими преимуществами, такими как быстрая сходимость, эффективное использование памяти и независимость от начальной точки.

Результаты работы метода на двух выбранных функциях:

На выбранных функциях (func1 и Химмельблау), с данными параметрами, функции сошлись намного быстрее. Таким образом, можно сказать, что метод сопряженных градиентов находит минимум функции быстрее всего.

На функции func1 алгоритм сошелся и закончил свое выполнение успешно, при этом количество итераций стало сильно меньше при тех же выходных данных, что и при градиентном спуске с постоянным шагом:

- Кол-во итераций = 2
- Точка минимума = [0.02200; 0.0220]

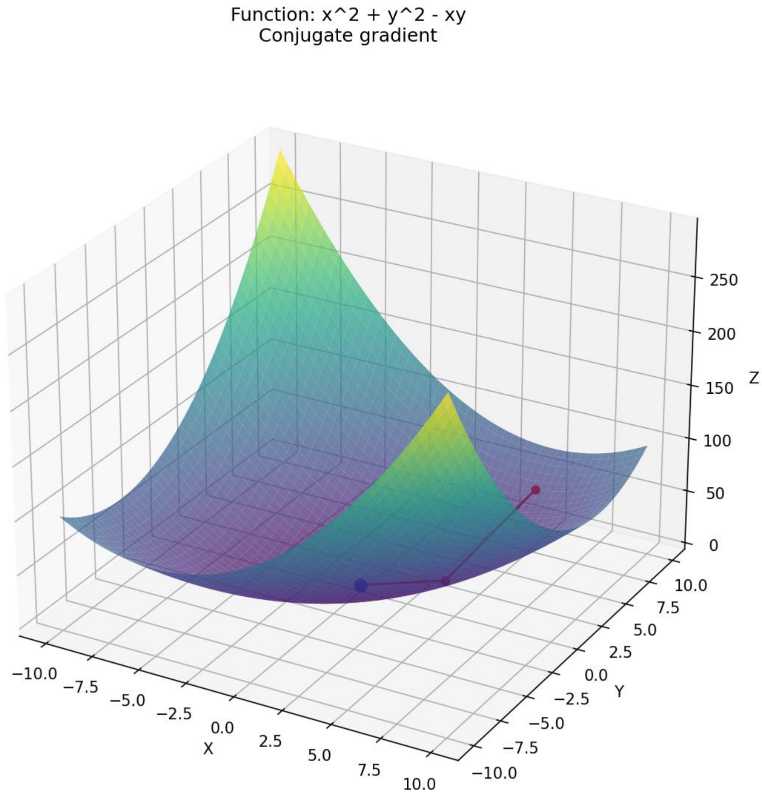

График функции Бута

На функции Химмельблау метод также сошелся при тех же параметрах, но количество итераций стало немного больше, чем при методе наискорейшего градиентного спуска:

- Кол-во итераций = 11
- Точка минимума = [-2.80510; 3.13131]

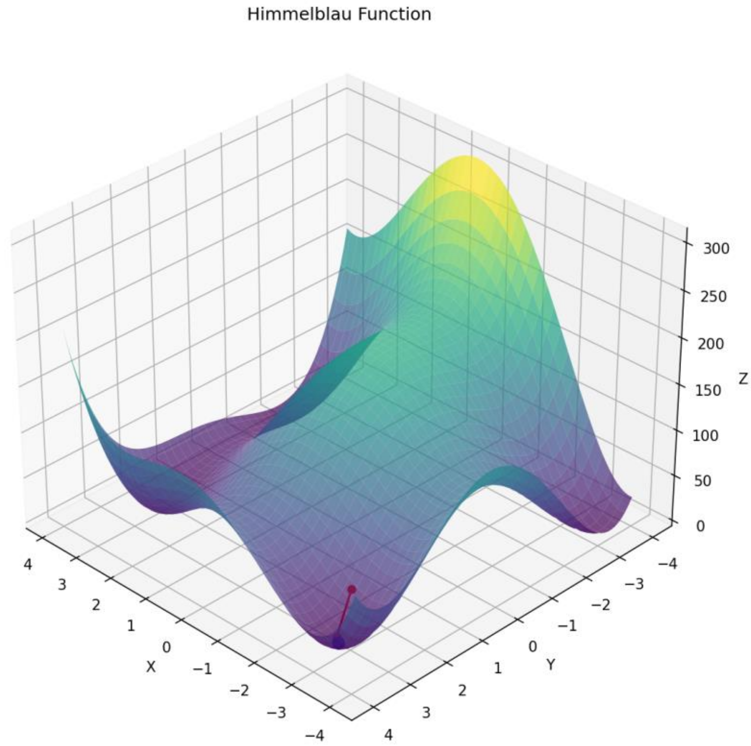

График функции Химмельблау

9. Вывод:

Для начала ответим на вопрос: какова зависимость числа итераций Т (n, k), необходимых градиентному спуску для сходимости в зависимости от размещения пространства $2 \leq n \leq 10^3$ и числа обусловленности оптимизируемой функции $1 \leq k \leq 10^3$.

Оценка зависимости числа итераций Т (n, k) для градиентного спуска в зависимости от размерности пространства n и числа обусловленности k является сложной задачей и зависит от многих факторов, включая функцию, которую мы оптимизируем, выбор шага и точности сходимости.

Однако общая тенденция заключается в том, что с увеличением размерности пространства n и числа обусловленности k, число итераций для достижения сходимости будет возрастать.

Также хочется отметить, что для задач оптимизации в больших размерностях, применение стандартных методов градиентного спуска может быть затруднительным из-за проблемы рассеивания градиента. В таких случаях могут применяться более продвинутые методы оптимизации, такие как стохастический градиентный спуск (SGD).

В итоге, в лабораторной работе были реализовали различные методы градиентного спуска. Конечный выбор метода оптимизации зависит от множества факторов, таких как тип задачи, размерность пространства, особенности функции, наличие ограничений и требования к скорости и точности оптимизации. Каждый метод имеет свои преимущества и недостатки, и выбор наиболее подходящего метода может быть неоднозначным.

Однако можно распределить задачи на несколько пунктов:

- Если функция является гладкой и выпуклой, то метод наискорейшего спуска может быть хорошим выбором из-за его простоты и широкой применимости.
- Если функция является квадратичной или близкой к квадратичной, то метод сопряженных градиентов может обеспечить быструю сходимость.
- Если же требуется обработка больших объемов данных, то метод стохастического градиентного спуска будет предпочтительнее.