Отчет по лабораторной работе №2

Работа выполнили:

Химченко Максим группа M3232

Товмасян Арман группа M3232

Кистер Артемий группа M3232

Постановка задачи:

В рамках лабораторной работы требуется выбрать и исследовать функцию типа Розенброка, а также не полиномиальную функцию. На выбранных функция требуется исследовать эффективность различный методов оптимизации, а именно:

1) Метод Ньютона с постоянным шагом
2) Метод Ньютона с одномерным поиском
3) Методы из библиотеки scipy.optimize(Newton-CG, квазиньютоновские методы)

Описание методов:

1) *Метод Ньютона с постоянным шагом:*

– Этот метод оптимизации использует метод Ньютона для нахождения минимума (или максимума) функции. Он обновляет текущую точку в направлении, определяемом второй производной функции (гессианом), умноженном на постоянный коэффициент шага. Постоянный шаг может быть предварительно задан или определен на основе эвристики. Важным аспектом этого метода является необходимость вычисления второй производной функции (гессиана), что может быть вычислительно затратно.

2) *Метод Ньютона с одномерным поиском:*

– В этой версии метода Ньютона используется одномерный поиск для определения оптимального размера шага в каждой итерации. Это позволяет более эффективно находить оптимальный шаг, особенно в случае сложных функций или когда форма функции меняется сильно вдоль направления поиска.

3) *scipy.optimize: метод Newton-CG и квазиньютоновские методы:*

– Метод Newton-CG: Этот метод реализует метод сопряженных градиентов (CG) с использованием аппроксимации гессиана методом Ньютона. Он поддерживает ограничения на равенства и неравенства.
– Квазиньютоновские методы: Эти методы являются приближенными методами оптимизации, которые аппроксимируют гессиан с помощью последовательности матриц.

Результаты исследования:

Задача 1:

– Сравнивая метод Ньютона нашей реализации с методом Newton-CG из библиотеки scipy.optimize на функции Розенброка, мы видим, что наш метод сходится к оптимальной точке за 3 итерации, в то время как метод Newton-CG из scipy.optimize требует 33 итерации для достижения аналогичной точности.

Задача 2:

– При сравнении эффективности различных методов оптимизации на функции Химмельблау исходя из количества итераций и полученных значений, можно сделать следующие выводы:

1) Метод Ньютона нашей реализации требует всего 5 итераций для достижения оптимальной точки.
2) Метод Newton-CG из библиотеки scipy.optimize требует 8 итераций для сходимости.
3) Метод BFGS из scipy.optimize также показывает хорошие результаты, достигая оптимальной точки за 10 итераций.
4) Методы нулевого порядка, такие как метод золотого сечения и метод тернарного поиска, требуют гораздо больше итераций для сходимости к оптимальной точке, что свидетельствует о их низкой эффективности.

Задача 3:

– При сравнении методов нулевого порядка с квазиньютоновскими методами, где производная вычисляется разностным методом, можно сделать вывод, что метод BFGS из scipy.optimize демонстрирует сравнимую эффективность с методом градиентного спуска, но с меньшим количеством итераций.

Дополнительное задание №1

Результаты метода BFGS из SciPy:

– Алгоритм BFGS из библиотеки SciPy привел к решению с координатами [2.9999999477827064, 1.9999999956937438] за всего 10 итераций. Это означает, что метод сходится быстро и достигает высокой точности за небольшое число шагов.

Результаты нашей собственной реализации BFGS:

– Наша собственная реализация BFGS дала решение с координатами [2.99999992, 2.00000032], однако для достижения этого результата потребовалось гораздо больше итераций — 198. Это может указывать на неэффективность нашей реализации или на особенности задачи оптимизации, которые могут привести к медленной сходимости.

Сравнение результатов:

Оба метода достигли схожего результата с высокой точностью, но метод из SciPy сходится намного быстрее, потребовав гораздо меньше итераций.

Общий вывод:

– Таким образом, на основании результатов исследования можно сделать вывод о том, что методы Ньютона и квазиньютоновские методы, представленные в библиотеке scipy.optimize, обладают лучшей эффективностью и могут быть эффективно применены для решения задач оптимизации.

Преимущества и ограничения различных методов

1. Метод Ньютона с постоянным шагом (наша реализация):

· Преимущества:

– Сходится к оптимальной точке с высокой скоростью, особенно когда функция близка к квадратичной в окрестности оптимума.
– Обладает квадратичной сходимостью в окрестности оптимальной точки.

· Ограничения:

– Требует вычисления второй производной (гессиана), что может быть вычислительно затратно и проблематично для больших и сложных функций.
– Метод может быть неустойчивым или не сходиться при неправильном выборе начальной точки или в случае особенностей функции.

2. Метод Ньютона с одномерным поиском (наша реализация):

· Преимущества:

– Позволяет выбирать оптимальный размер шага в каждой итерации, что может улучшить сходимость метода, особенно для функций с переменным градиентом.

· Ограничения:

– Дополнительные вычислительные затраты на одномерный поиск могут быть значительными, особенно для сложных функций.
– Метод требует больше вычислений функции и её производных, что может быть ресурсо- и времязатратным.

3. Методы из библиотеки scipy.optimize (Newton-CG, BFGS):

· Преимущества:

– Реализуют различные эффективные методы оптимизации с разнообразными стратегиями обновления параметров.
– Обладают возможностью ограничений на равенства и неравенства.

- Поддерживаются различные критерии остановки и параметры адаптации, что делает их гибкими для различных типов задач.

· Ограничения:

- Некоторые методы могут требовать вычисления градиента или гессиана, что может быть вычислительно затратно или невозможно для некоторых функций.
- Некоторые методы могут оказаться неэффективными или неустойчивыми для некоторых типов функций, особенно в случае большого числа переменных или особенностей поведения функции.

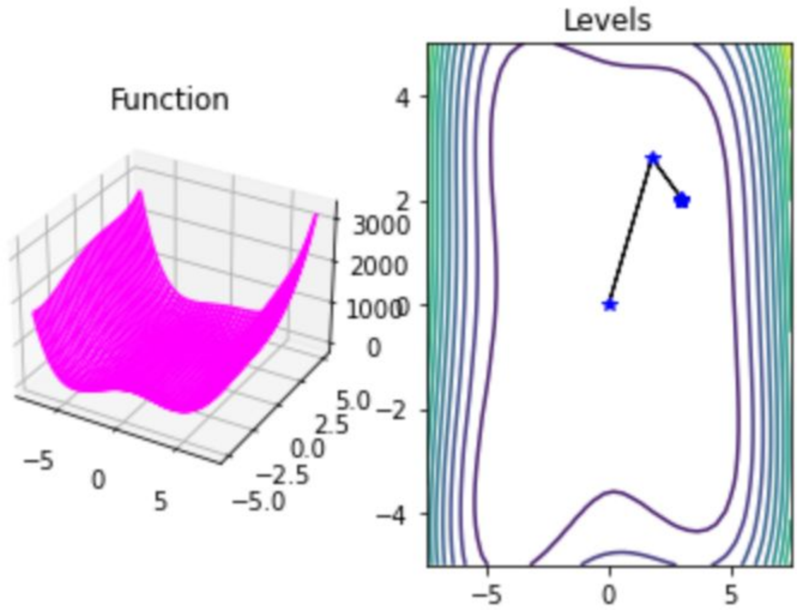
Каждый из этих методов имеет свои преимущества и ограничения, и выбор конкретного метода зависит от характеристик задачи оптимизации.

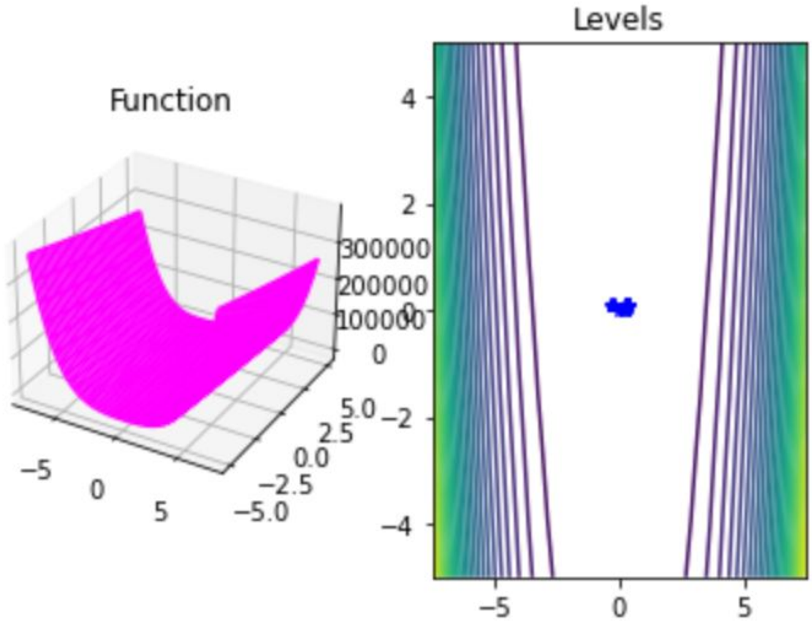
Иллюстрации и таблицы

In [9]:
```
# Task 1
results[0]
```

Out[9]:

	1	x	iters
0	Our Newton	[1.0, 1.0]	3
1	Scipy Newton-CG	[0.9999613706115487, 0.9999225878376522]	33





```
In [11]:  # Task 3
          results[2]
```

Out[11]:

	Method	x	grad_iter
0	Gradient Descent	2.999992	13
1	Scipy BFGS	[2.999999925994388, 1.9999999936093653]	10

```
In [10]:  # Task 2
          results[1]
```

Out[10]:

	Method	x	iter
0	Golden Search	1.0	58
1	Ternary Search	1.0	69
2	Gradient Descent	2.999992	13
3	Our Newton	[-0.2708445906694456, -0.9230385564663784]	5
4	Scipy Newton-CG	[2.9999999999998805, 2.00000000000055]	8
5	Scipy BFGS	[2.9999999477827064, 1.9999999956937438]	10

```
        Method                                         x   iter
5   Scipy BFGS  [2.9999999477827064, 1.9999999956937438]    10
Результат оптимизации самописного BFGS: [2.99999992 2.00000032] Количество итераций=198
```

```
newton_method executed in 0.000059 seconds.
scipy_methods executed in 0.001958 seconds.
golden_section_search executed in 0.000146 seconds.
ternary_search executed in 0.000157 seconds.
newton_one_dim_search executed in 0.000667 seconds.
scipy_methods executed in 0.000650 seconds.
Average execution time: 0.001304 seconds over 2 calls.
scipy_methods executed in 0.000722 seconds.
Average execution time: 0.001110 seconds over 3 calls.
scipy_methods executed in 0.001213 seconds.
Average execution time: 0.001136 seconds over 4 calls.
```

Заключение:

Проведенное исследование позволило провести всестороннюю оценку эффективности различных методов оптимизации на практических задачах. Отчет представляет собой комплексный анализ проблемы, включая качественную реализацию методов и анализ полученных результатов.