ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»

Дисциплина:
«Прикладная математика»

Отчет по лабораторной работе №3
Методы спуска

Студенты:
Никитин Александр, M32041
Игнатьев Андрей, M32041
Курепин Даниил, M32041

Преподаватель:
Гомозова Валерия Эдуардовна

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 $2 \leq n \leq 10^3$и числа обусловленности оптимизируемой функции $1 \leq k \leq 10^3$
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.

Теория

Градиентный спуск с постоянным шагом

Задача решается посредством построения последовательности точек такой, что с каждым разом $f(x_{k+1}) < f(x_k)$. Точки вычисляются по принципу $x_{k+1} = x_k + \lambda_k \nabla f(x_k), \lambda_k \geq 0$, при этом начальная точка выбирается произвольно. Шаг не меняется, пока соблюдается условие $f(x_{k+1}) < f(x_k)$, в противном случае длина шага корректируется пока условие снова не выполнится. Алгоритм завершается, когда выполняется условие $||\nabla f(x_k)|| < \varepsilon$

Ограничения на функцию для корректной работы алгоритма:

1. *Функция выпуклая дифференцируемая*

При этом, даже при выполнении этих условий метод может не сойтись

Градиентный спуск с дроблением шага по условию Армихо

Метод состоит в обновлении текущей точки в направлении, противоположном градиенту
Перед стартом алгоритма:

1. Берётся произвольная x_0
2. Вычисляется $g = \nabla f(x_0)$
3. Задаётся начальный шаг α, коэффициент дробления γ (обычно меньше 1)

Далее повторяются следующие шаги до выполнения критерия остановки (к примеру, размер шага) или достижения максимального числа итераций:

1. Вычисляется новая точка $x_{k+1} = x_k - \alpha \nabla f(x_k)$ и значение функции в ней
2. Проверяется условие Армихо: $f_{k+1} \leq f_k - \gamma \alpha {g_k}^2$, где g_k – норма градиента в x_k
   При невыполнении условия шаг уменьшается (обычно, $\alpha = \alpha\gamma$), шаги 1 и 2 повторяются

Алгоритм эффективно приближается к минимуму, избегая осцилляций и перепрыгиваний минимума.

Метод наискорейшего спуска

Метод наискорейшего спуска является одним из наиболее распространенных методов оптимизации, используемых для решения задач безусловной оптимизации.

Основная идея метода заключается в том, чтобы на каждой итерации двигаться в сторону наиболее быстрого убывания функции. Для этого вычисляется градиент функции в текущей точке и движение происходит в направлении, противоположном градиенту, с определенным шагом. Таким образом, метод наискорейшего спуска позволяет искать локальный минимум функции.

$$L = \{x = x_k - \lambda f'(x_k);\ \lambda \leq 0\}:\ \lambda_k = arg_{\lambda \in [0,\infty]} min f(x_k - \lambda f'(x_k))$$

Алгоритм метода наискорейшего спуска следующий:

1. Задать начальную точку x0.

2. Найти градиент функции f(x) в точке x0.

3. Вычислить направление движения как противоположное градиенту: d = -grad f(x0).

4. Определить длину шага alpha, на которую нужно двигаться в направлении d. Для этого можно использовать методы одномерной оптимизации, такие как поиск наискорейшего спуска (англ. line search) или фиксированный шаг (В нашей реализации мы использовали метод одномерной оптимизации Брента)

5. Выполнить шаг в направлении d: x1 = x0 + alpha * d.

6. Повторять шаги 2-5 до тех пор, пока не будет достигнута заданная точность или не будет достигнуто максимальное число итераций.

Однако, метод наискорейшего спуска может столкнуться с рядом проблем, таких как медленная сходимость в области сильного выпуклости, неэффективность в области сильного вытянутого минимума и т.д.

Метод сопряженных градиентов

Метод каждый цикл вычисляет следующее направление сопряжённости, при этом учитывая предыдущие направления и градиенты.

Рассмотрим алгоритм на квадратичной функции:
$f(x) = 0.5x^TAx - b^Tx + c$, где

x – Вектор переменных
A – Симметричная положительно-определённая матрица
b – Вектор
$c = const$

Работа алгоритма начинается из выборочного x_0 с начальным направлением d_0.
Далее на каждой итерации метода:

1. Вычисляем градиент функции $g = Ax_k - b$
   Если k=0, то $d = -g$
2. Вычисляем шаг $\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k}$
3. Вычисляем следующую точку $x_{k+1} = x_k + a_k d_k$, вычисляем градиент g_{k+1} в ней
4. Вычисляем коэффициент направления сопряжённости $\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$,
   Обновляем направление сопряжённости: $d_{k+1} = -g_{k+1} + \beta_k d_k$

Данные шаги повторяются до достижения критерия остановки – количества итераций или достаточно малой нормы градиента.

Он обладает высокой скоростью сходимости для квадратичных функций, однако для не квадратичных функций его использование может быть затруднительно и/или требовать модификаций

Результаты

Из пункта (5):

Нам предлагается взять 2-3 квадратичные функции от двух переменных и проанализировать траектории предложенных алгоритмов. Ниже приведены две функции, их градиентные функции:

Функция 1:

$\boldsymbol{f(x, y) = 5x^2 + 2y^2}$

А также ее градиентная функция:

$\boldsymbol{f(x, y\,) = 10x + 4y}$

Функция 2:

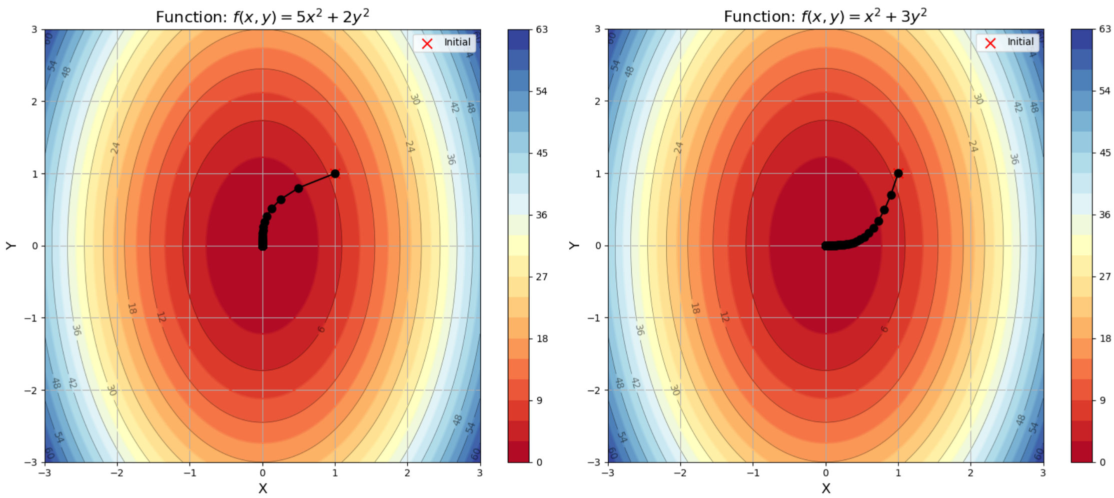
$\boldsymbol{f(x, y) = x^2 + 3y^2}$

А также ее градиентная функция:

$\boldsymbol{f(x, y) = 2x + 6y}$

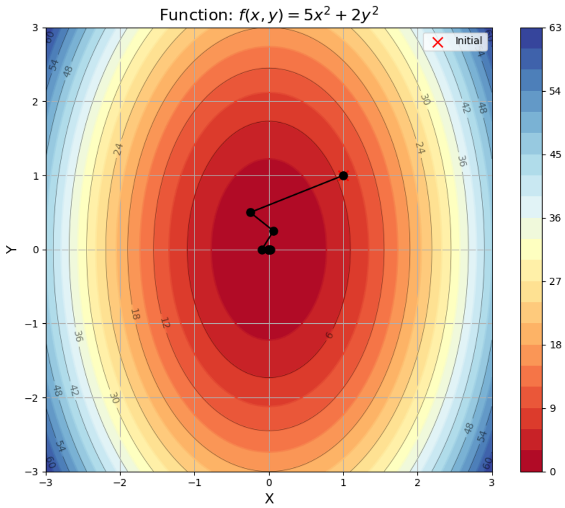
Графики:

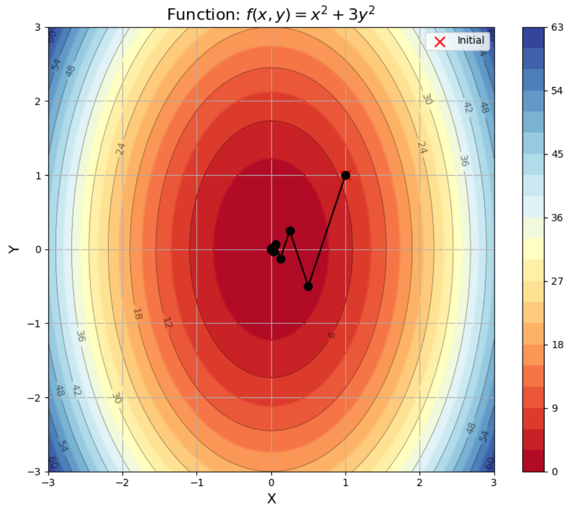
1) **Метод градиентного спуска с постоянном шагом:**



При слишком большом шаге метод расходится.
При слишком малом сходится медленно и точность хуже.
Таким образом, шаг подбирается так, чтобы он был наибольшим из тех, при котором функция в методе может сойтись.

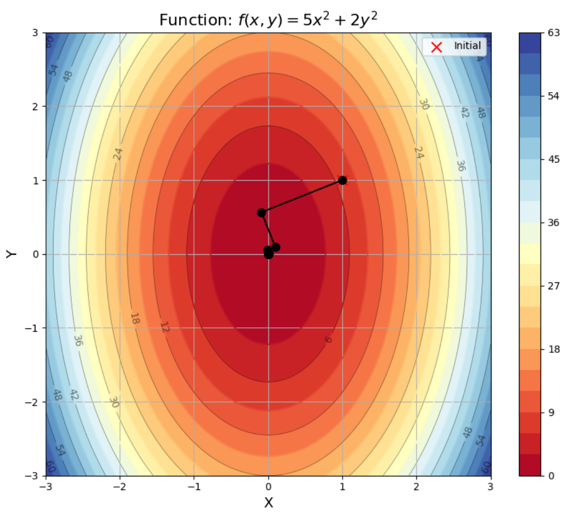
2) Алгоритм спуска с дроблением шага, используя условие Армихо:

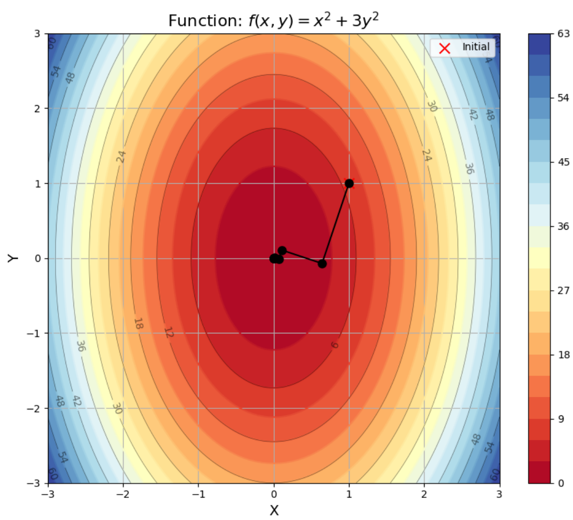




Из графиков видно, что число итераций, необходимых для сходимости, зависит от выбранной начальной точки. Градиентный спуск сначала быстро приближается к минимуму, а затем требуется больше усилий для дальнейшего улучшения. Начальное положение также влияет на количество итераций для достижения минимума. Алгоритм градиентного спуска с условием Армихо позволяет достичь одного и того же оптимального решения независимо от выбора начальной точки, что подтверждает правильность и консистентность алгоритма.

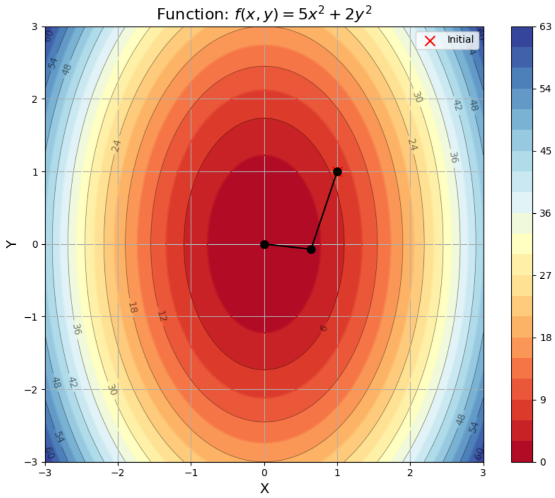
3) Метод наискорейшего спуска с одномерной оптимизацией:

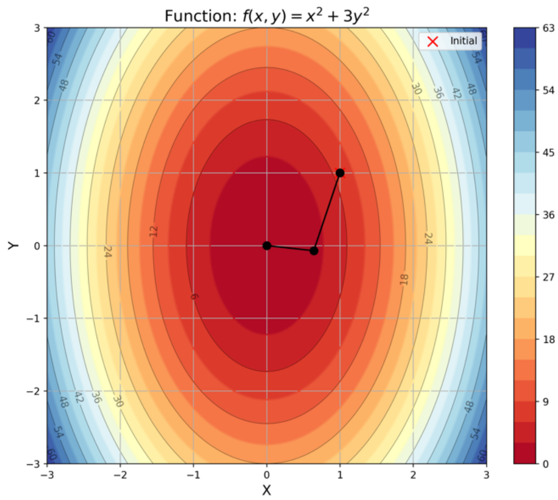




Теоретическая скорость сходимости метода наискорейшего спуска не выше скорости сходимости градиентного метода с оптимальным шагом в общем случае.

4) Метод сопряженных градиентов:



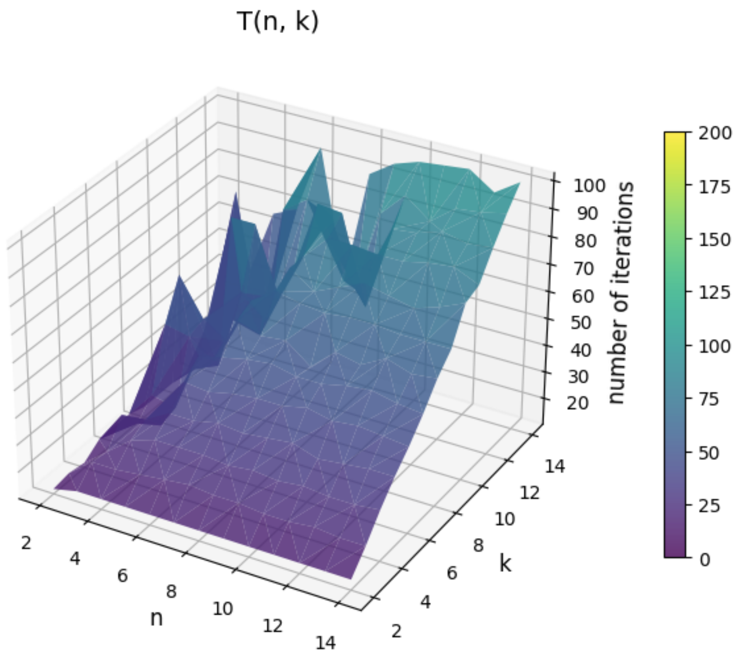


Метод сопряженных градиентов сходится к оптимальному решению функции во всех случаях начальных точек. Он эффективно приближается к минимуму функций с каждой итерацией, уменьшая разницу между текущим приближением и оптимальным решением. Графики демонстрируют уменьшение значений функции с каждой итерацией, следовательно функция приближается к минимуму.

Из пункта (8):

Нам нужно было исследовать зависимость числа итераций $T(n, k)$, необходимых градиентному спуску для сходимости в зависимости от размерности пространства $2 \leq n \leq 10^3$и числа обусловленности оптимизируемой функции $1 \leq k \leq 10^3$. Сделав генератор случайных квадратичных функций n переменных с числом обусловленности k, мы можем теперь, используя наш генератор, а также метод наискорейшего спуска получить трехмерную точечную диаграмму, которая показывает, как количество ограничений и размерность влияют на количество итераций, необходимых для сходимости метода быстрого градиентного спуска. Чем больше размерность и число ограничений, тем больше итераций требуется для сходимости метода.

Полученная диаграмма:



Вывод

В результате лабораторной работы сделаны следующие выводы:

- Метод сопряжённых градиентов – наименьшее количество итераций до сходимости
- Градиентный спуск с постоянным шагом – не учитывает геометрическую структуру решения, оттого достижение оптимального решения требует наибольшего количества итераций для достижения оптимального решения
- Метод наискорейшего спуска – лучше чем метод, использующий условие Армихо

Таким образом, метод сопряжённых градиентов – лучший, особенно когда речь идёт о линейных системах.

При исследовании зависимости $T(n, k)$ мы выяснили, что увеличение размерности пространства и/или числа обусловленности увеличивается количество итераций на пути к минимуму.