**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Прикладная математика

*Лабораторная работа №2*

Выполнили студенты:

Чайков Артемий M32031

Демин Вадим M32021

Гусев Дмитрий M32021

Санкт-Петербург

2021

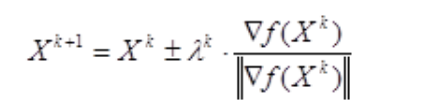
План написания отчёта:

1. Задание 1
   1. Описать каждый метод, что используется в алгоритме + прошарить за код
   2. Для каждого алгоритма по одному примеру выполнения
2. Задание 2
   1. сделать замеры кода при разных методах
   2. вставить всякие таблицы и графики
3. Задание 3
   1. Тесты на 3 различных по выполнению функциях
   2. различные тесты на
      1. числа обусловленности функции
      2. выбора начальной точки
      3. стратегии выбора шага (кажется это пункт 2, поэтому скип)
   3. на этих тестах построить таблицы и графики и тп.
4. Задание 4
   1. пока не понимаю, но кажется это чисто тестирование градиентного спуска на разные значения числа обусловленности и размерности пространства n
   2. Замеры делаются через рандом, он уже написан, надо понять как пользоваться
   3. в итоге надо измерить число итераций для разных переменных

1. **метод Наискорейшего спуска**

Метод наискорейшего спуска - это итерационный численный метод решения оптимизационных задач, который позволяет определить экстремум целевой функции. Является одним из модификаций градиентного спуска.

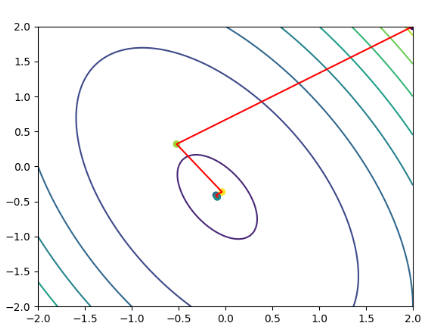
Метод наискорейшего спуска является дальнейшим развитием метода градиентного спуска. В общем случае процесс нахождения экстремума функции является итерационной процедурой, которая записывается следующим образом:



Направление движения выбирается в сторону антиградиента.



определяется как



Answer is: [-0.57693671 -0.46147779] steps are: 7

1. **метод Градиентного спуска**

Основная идея метода заключается в том, чтобы осуществлять оптимизацию в направлении наискорейшего спуска, а это направление задаётся антиградиентом.

Общая формула выглядит как у метода наискорейшего спуска, т.к. наискорейший спуск является одним из разновидностей градиентного спуска.



В данной формуле изменяемым параметром является шаг λ^k, он может:

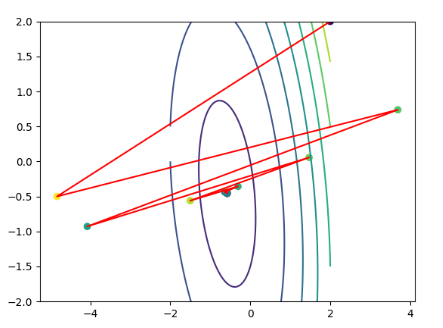
1. быть постоянным
2. в процессе спуска шаг делится на какое-то число
3. наискорейший спуск

В данном алгоритме использовался второй пункт, и итоговая формула, выглядит как 

Где i - количество итераций в цикле. Таким образом шаг будет уменьшаться не все время, когда мы достигаем значения 1/25, мы остаемся на нем. Это сделано для того, чтобы бесконечное уменьшение не привело к бесконечной работе алгоритма.

Выход из алгоритма будет при достижении данного условия:





7x^2 + 2\*y^2 + 2\*x\*y + 9\*x + 3\*y

Answer is: [-0.57696913 -0.46130203], steps are: 16

1. **метод сопряженных градиентов**

использование направление анти градиента как направление убывания функции не всегда удачное, из-за этого можно увидеть зигзаго образный график решения. Поэтому будем менять направление раз в N шагов

1. Задаются значения точности, x0 и частоту обновлений.
2. Направление принимается за антиградиент.
3. найдём Альфа k, такое что функция f(xk+Alpha k \* pk) принимает минимальное значение
4. Находим x(k+1) = xk + Alpha k \* pk
5. Проверяем условие выхода ||grad(xk+1)||<eps
6. Если пришло время обновить направление, то вычисляем новое направление p(k+1)



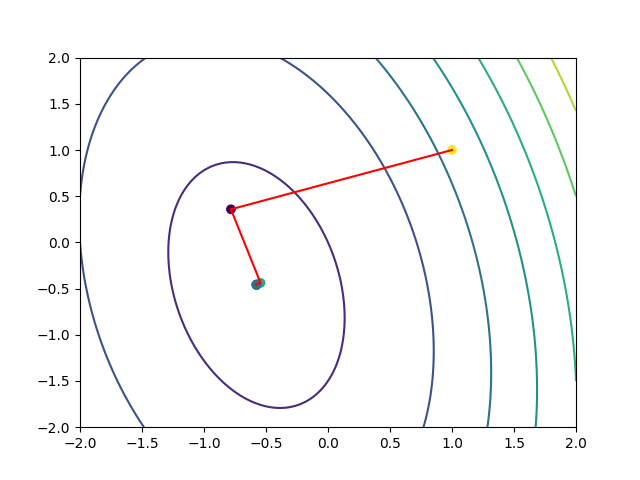
1. повторяем пока не выйдем по условию выхода

Примеры:

Для изменения градиента каждый 2ой шаг:

7x^2 + 2\*y^2 + 2\*x\*y + 9\*x + 3\*y

Answer is: [-0.57688336 -0.4615097 ] steps are: 6



1. **метод сопряженных направлений (метод Пауэлла)**

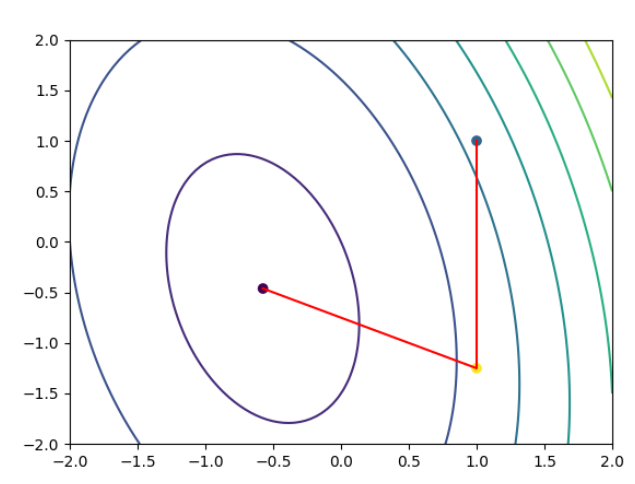
Для N-мерного пространства выбирается изначальная точка x0 изначального приближения. Метод минимизирует функцию путем двунаправленного поиска по очереди по каждому вектору поиска(за вектора поиска как правило принимается N векторов сонаправленных с осями пространства). При этом каждый следующий поиск производится из точки минимума, полученной на предыдущем шаге. После этого получается точка x. Далее происходит замена векторов по принципу s[i] = s[i+1], а вектор смещения относительно изначальной точки становится новым вектором поиска и добавляется в конец списка поиска. Данный алгоритм повторяется до тех пор, пока шаг между изначальной точкой x0 и точкой x не станет меньше эпсилон.

Примеры:

7x^2 + 2\*y^2 + 2\*x\*y + 9\*x + 3\*y

Ответ [-0.57607882 -0.46258643]

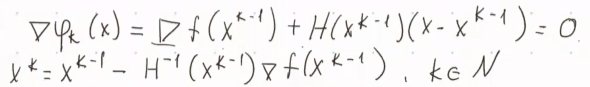
Число шагов: 2

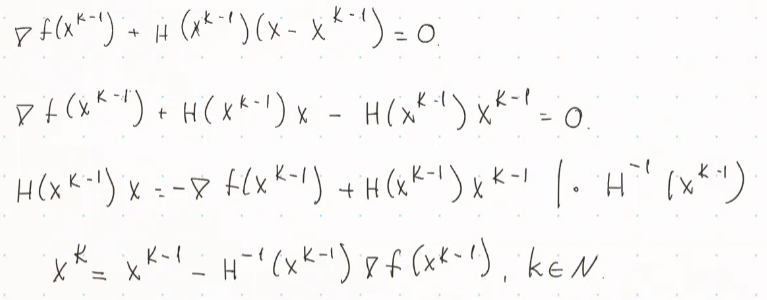


1. **метод Ньютона**

Основная идея метода Ньютона заключается в итеративном использовании квадратичной аппроксимации целевой функции в текущей точке поиска и минимизации этой аппроксимации.

Выбирается начальное положение x0, дальше на каждом шагу будем находить вектор перемещения в точку минимума квадратичной функции.Складываем это значение с x0, пока не выполнится условие выхода ||s|| <= Eps



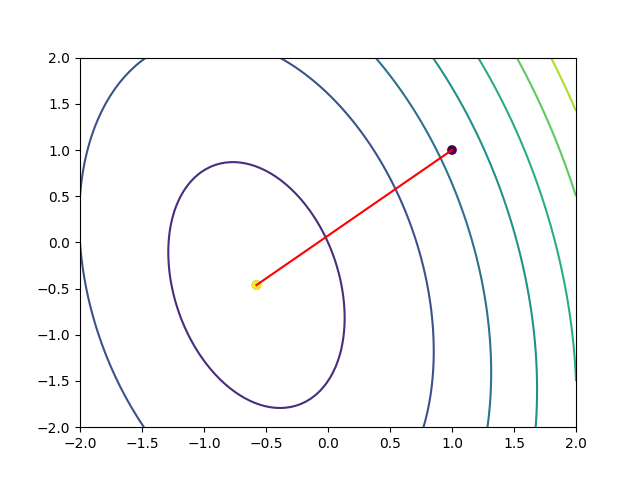


Примеры:

7x^2 + 2\*y^2 + 2\*x\*y + 9\*x + 3\*y

Answer is: [-0.57692308 -0.46153846] steps are: 2

Так как функция квадратичная, мы находим ответ этим методом всего за 2 шага



**Задание 2 + 3**

**см табличку, очень красивая получилась**

**7x^2 + 2\*y^2 + 2\*x\*y + 9\*x + 3\*y**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| метод поиска | Постоянная величина | Метод дробления шага | Метод золотого сечения | Метод Фибоначчи |
| Наискорейший спуск | 31 для 1/20 | 16 | 7 | 8 |
| Сопряженных градиентов | 23 | 13 | 4 | 5 |
| метод Пауэлла | не выполняется | не выполняется | 2 | 2 |
| метод Ньютона | 2 | 2 | 2 | 2 |

**6 \* x[0]\*\*2 - 4 \* x[0]\*x[1] + 3 \* x[1]\*\*2 + 4 \* sqrt(5) \* (x[0] + 2 \* x[1]) + 22**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| метод поиска | Постоянная величина | Метод дробления шага | Метод золотого сечения | Метод Фибоначчи |
| Наискорейший спуск | 34 для 1/20 | 18 | 11 | 12 |
| Сопряженных градиентов | 25 | 13 | 5 | 5 |
| метод Пауэлла | не выполняется | не выполняется | 2 | 2 |
| метод Ньютона | 2 | 2 | 2 | 2 |

**sin(x)+y^2**

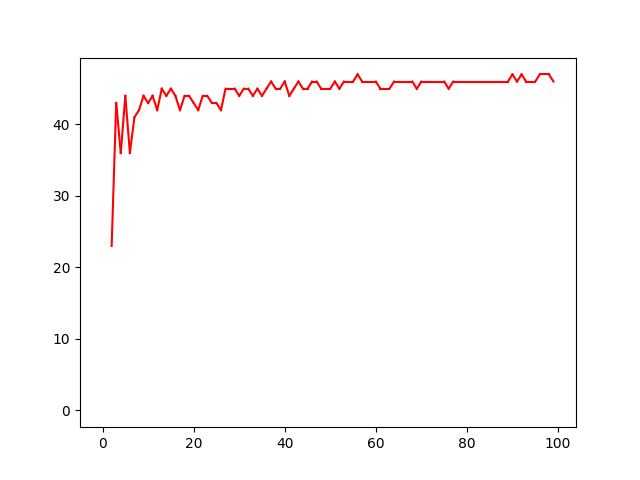
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| метод поиска | Постоянная величина | Метод дробления шага | Метод золотого сечения | Метод Фибоначчи |
| Наискорейший спуск | 123 для 1/20 | 130 | 6 | 6 |
| Сопряженных градиентов | 51 | 31 | 5 | 4 |
| метод Пауэлла | не выполняется | не выполняется | 2 | 2 |
| метод Ньютона | 4 | 4 | 4 | 4 |

\*Для метода Пауэлла не работает постоянная величина и метод дробления шага, так как они строго задают вектора и соответственно путь метода, что, очевидно, неверно, т..к. путь поиска должен изменяться.

**Задание 4**

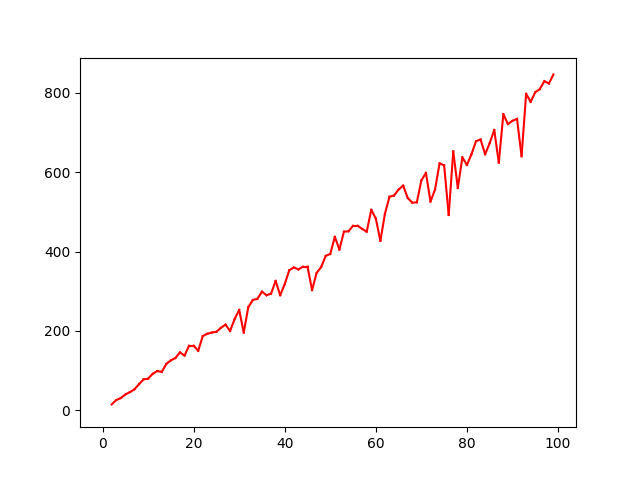
Составляется матрица A с числом обусловленности(число обусловленности можно рассчитать как отношение максимального и минимального собственных чисел матрицы. Следовательно, составим матрицу, где на главной диагонали есть как минимум одно число 1, как минимум одно число равное k, остальные числа на диагонали выбираются рандомно), по которой строится квадратичная форма: f(x)=xT \* A\*x + bT \* x, где b вектор - рандомный массив длины n

Зависимость кол-ва итераций от изменения n - размерности пространства, при k = 5

****

Как видно из графика, число итераций градиентного спуска для функции, полученной из хорошо обусловленной матрицы, не зависит от размерности пространства.

Второй график показывает зависимость количества итераций от изменения числа обусловленности k, при n = 5. График линейный**.**



**Выводы:**

В ходе выполнения данной лабораторной работы мы выяснили, что самыми эффективными являются метод Ньютона и Пауэлла. Оценивая скорость сходимости, используя различные методы для поиска величины шага, мы увидели, что методы Золотого сечения и Фибоначчи показывают наилучший результат, на некоторых функциях метод Золотого сечения вырывается вперед.

Метод Пауэлла находит для квадратичной функции минимум за n шагов, при этом если функция не строго квадратичная, то поиск необходимо провести несколько раз. Также стоит заметить, что для работы метода Пауэлла не надо искать производные и градиенты функции, следовательно его можно использовать на не дифференцируемых функциях.

Метод Ньютона имеет лучшую эффективность, но при запуске функции с тригонометрическим параметром, эффективность ухудшилась, так как функция стала не квадратичной. Также была обнаружена ошибка, при которой метод Ньютона находи минимум по оси Y но при этом являлся максимумом для функции в плоскости OX.

Зависимость количества итераций не зависит от размерности пространства, но линейно зависит от числа обусловленности