Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Университет ИТМО»

Факультет информационных технологий и программирования

Лабораторная работа № 2

Выполнили студенты группы M3435:

Дорохов Ярослав Юрьевич

Бизяев Илья Владимирович

Пономарев Павел Николаевич

Санкт-Петербург

2021

Задание 1.

В первом задании нами были реализованы следующие методы:

1. Метод сопряженных направлений
2. Метод сопряженных градиентов
3. Метод Ньютона

Метод сопряженных направлений

Данный метод предназначен для нахождения точки минимум квадратичной задачи $f(x) = \frac{1}{2}(Qx, x) + bx$, где Q — положительно определенная матрица.

Функция `conjugate_direction_method` принимает следующие параметры: матрица `Q`, вектор `b`, стартовая точка `start`, точность вычисления `eps`. Возвращает найденную точку минимума и количество итераций, которое методу потребовалось. Стоит заметить, что количество итераций не превосходит размерность пространства.

Метод производит следующие вычисления:

```
w_1 = - Q @ start - b
u_1 = w_1
h_1 = np.dot(u_1, u_1) / np.dot(Q @ u_1, u_1)

for i = 2…n:
    w_i = Q @ x_{i-1} - b
    u_i = w_i - (np.dot(Q @ u_{i-1}, w_i) / np.dot(Q @ u_{i-1}, u_{i-1})) * u_{i-1}
    h_i = np.dot(w_i, u_i) / np.dot(Q @ u_i, u_i)
    x_i = x_{i-1}+ h_i * u_i
```

Метод сопряженных градиентов

Данный метод предназначен для нахождения точки минимума некоторой функции (необязательно квадратичной).

Функция `conjugate_gradient_method` принимает следующие параметры: `f` — функция, для которой необходимо найти точку минимума; `f_grad` — градиент функции `f`, точка старта `start`, точность вычисления `eps`, максимальное количество итераций `max_iters`. Возвращает найденную точку минимума и количество итераций, которое методу потребовалось

Метод производит вычисления похожие на те, что производит метод сопряженных направлений. Значение γ_kвычисляется в соответствии с методом Полака-Рибьера, значение параметра спуска h_kвычисляется с помощью метода одномерной оптимизации (метод дихотомии).

Метод Ньютона

Данный метод предназначен для нахождения минимума некоторой функции.

Функция newton_method принимает следующие параметры: `f` — функция, для которой необходимо найти точку минимума; `f_grad` — градиент функции `f`, `f_hess` — матрица Гессе функции `f`, точка старта `start`, точность вычисления `eps`, максимальное количество итераций `max_iters`. Возвращает найденную точку минимума и количество итераций, которое методу потребовалось

Данный метод аппроксимирует функцию $f(x)$ в окрестности текущей точки x_k квадратичной функцией $\varphi(x)$ и находит точку минимума $\varphi(x)$ в этой окрестности x_{min}. По найденной точке минимума в этой окрестности строится следующая точка $x_{k+1} = x_k + \alpha * x_{min}$. Параметр α находится с помощью метода одномерной оптимизации. Точка минимума квадратичной функции $\varphi(x)$ в окрестности находится с помощью метода сопряженных направлений.

Реализацию описанных методов можно посмотреть [здесь](#).
Тесты реализованных методов доступны [здесь](#).

Задание 2.

В соответствии с условием задания, разработанные алгоритмы были исследованы на ниже указанных функциях из различных исходных точек.

Квадратичная функция
$f(x, y) = 100(y - x)^2 + (1 - x)^2$

method	start	result x	result f(x)	iterations
conjugate_dirs	[0, 0]	[1. 1.]	1.779867417404908e-29	2
conjugate_grad	[0, 0]	[0.99999973 1.]	7.503793090826976e-12	3
newton	[0, 0]	[1. 1.]	1.4711538250881558e-17	2
conjugate_dirs	[1, 1]	[1. 1.]	0.0	0
conjugate_grad	[1, 1]	[1. 1.]	0.0	0
newton	[1, 1]	[1. 1.]	0.0	1
conjugate_dirs	[-1, -1]	[1. 1.]	4.930380657631324e-30	2
conjugate_grad	[-1, -1]	[0.99999946 1.00000001]	3.0015172574012365e-11	3
newton	[-1, -1]	[1. 1.]	1.5676499572125325e-17	2
conjugate_dirs	[-1, -2]	[1. 1.]	1.5954711808094964e-28	2
conjugate_grad	[-1, -2]	[0.99999941 1.0000006]	1.423423282723641e-10	3
newton	[-1, -2]	[1.00000001 1.00000001]	1.5299812245384085e-15	2
conjugate_dirs	[1.95, -2]	[1. 1.]	5.566399762465765e-29	2
conjugate_grad	[1.95, -2]	[1. 1.]	5.75060636438309e-18	4
newton	[1.95, -2]	[1. 1.00000001]	1.6155476458810427e-14	2
conjugate_dirs	[-3, 4]	[1. 1.]	1.7261262682367265e-28	2
conjugate_grad	[-3, 4]	[1. 1.]	9.27198991036261e-20	4
newton	[-3, 4]	[1. 1.]	2.7092053560873096e-15	2
conjugate_dirs	[10, 10]	[1. 1.]	5.174927538249837e-28	2
conjugate_grad	[10, 10]	[0.99999997 0.99999997]	9.49081977191811e-16	4
newton	[10, 10]	[0.99999997 0.99999997]	8.381897366377828e-16	2

Результаты находятся [здесь](#).
Все реализованные методы из всех указанных начальных точек сошлись к точке минимума.
Количество итераций, необходимое методу сопряженных направлений не превосходит размерность (2), метод Ньютона делает не более 2 итераций, метод сопряженных градиентов — не более 4.

Функция Розенброка

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

method	start	result x	result f(x)	iterations
conjugate_grad	[0, 0]	[1. 1.]	2.2570704989884645e-16	22
newton	[0, 0]	[1. 1.]	1.1168484060573177e-19	9
conjugate_grad	[10, 10]	[1. 1.]	3.192558626170139e-16	35
newton	[10, 10]	[1. 0.99999999]	3.401405804802837e-17	28
conjugate_grad	[-1, -2]	[1. 1.]	7.118801872585616e-18	59
newton	[-1, -2]	[0.99999993 0.99999987]	9.229875648479202e-15	14
conjugate_grad	[-3, 3]	[1. 1.]	4.8604382054398646e-18	69
newton	[-3, 3]	[1. 1.]	6.051659698394087e-19	21
conjugate_grad	[-12, -11]	[0.99996447 0.99992296]	4.839444520972246e-09	100
newton	[-12, -11]	[1. 1.]	1.339796776666393e-22	35
conjugate_grad	[2, 2]	[0.99998871 0.99997523]	6.092634845015793e-10	22
newton	[2, 2]	[0.99999998 0.99999996]	7.004067723838306e-16	10

Результаты находятся здесь.

Все реализованные методы из всех указанных начальных точек сошлись к точке минимума.

Количество итераций, необходимое методу Ньютона не превышает 34, метод сопряженных градиентов максимальное сделал 100 итераций.

Тестовая функция

$$f(x, y) = 2\,exp(-(\frac{x-1}{2})^2 - (\frac{y-1}{1})^2) + 3\,exp(-(\frac{x-2}{3})^2 - (\frac{y-3}{2})^2$$

method	start	result x	result f(x)	iterations
conjugate_grad	[3, 3]	[1.96715656 2.88589729]	-3.035063505405846	5
newton	[3, 3]	[1.96526154 2.88609745]	-3.0350623224080544	1000
conjugate_grad	[0, 0]	[1.26271467 1.33402388]	-3.1693170298311917	8
newton	[0, 0]	[-6.26285044e-10 -6.53393312e-10]	-0.7757493397713863	1
conjugate_grad	[-1, -1]	[148.72523612 364.23906005]	-0.0	3
newton	[-1, -1]	[-1. -1.]	-0.03368973496216185	1
conjugate_grad	[1.5, 1.35]	[1.26290768 1.33444681]	-3.1693172016855455	7
newton	[1.5, 1.35]	[1.26207346 1.33408026]	-3.1693166793136527	1000
conjugate_grad	[3.5, 3.5]	[1.2631213 1.33475042]	-3.169317068409744	7
newton	[3.5, 3.5]	[1.96659054 2.88593254]	-3.035063414243407	1000
conjugate_grad	[1.2, 1.2]	[1.26386823 1.33446731]	-3.1693168403810175	5
newton	[1.2, 1.2]	[1.26289202 1.33439783]	-3.1693172046324105	1000
conjugate_grad	[1.5, 1.5]	[1.26346812 1.3342623]	-3.16931706874317	5
newton	[1.5, 1.5]	[1.26237998 1.33412277]	-3.1693169454245482	1000

Результаты находятся здесь.

Методы в основном сходились к локальным максимумам функции:

$(x_1, y_1) = (1.2627, 1.334)$, $f(x_1, y_1) = 3.1693...$

$(x_2, y_2) = (1.9671, 2.8858)$, $f(x_2, y_2) = 3.03506…$

Оба метода разошлись при стартовой точке $(-1, -1)$, а метод Ньютона в дополнение разошелся при выбранной стартовой точке $(0, 0)$.

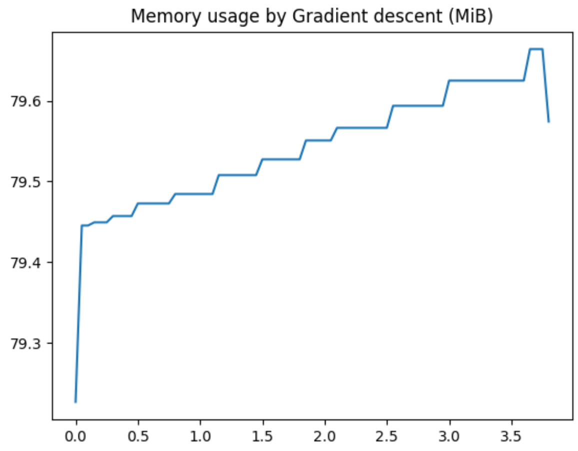
Количество итераций, необходимое в данном случае методу Ньютона сильно превышает количество итераций, необходимое методу сопряженных градиентов.

Задание 3.

$f(x, y, z) = 100(y - x)^2 + (1 - x)^2 + (z - y)^2 + 5(x + y)^2$

Градиентный спуск

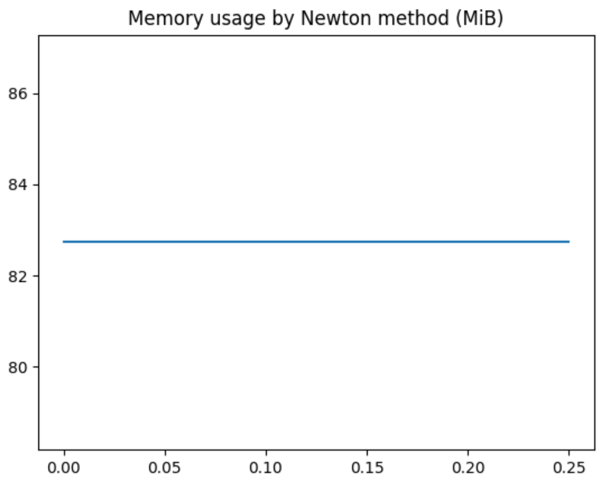
Завершился за 2731 итераций
Максимальное потребление памяти: 79.66 Мб



	Число вызовов
f	114744
f_grad	117476

Метод Ньютона

Завершился за 2 итерации
Максимальное потребление памяти: 82.73 Мб

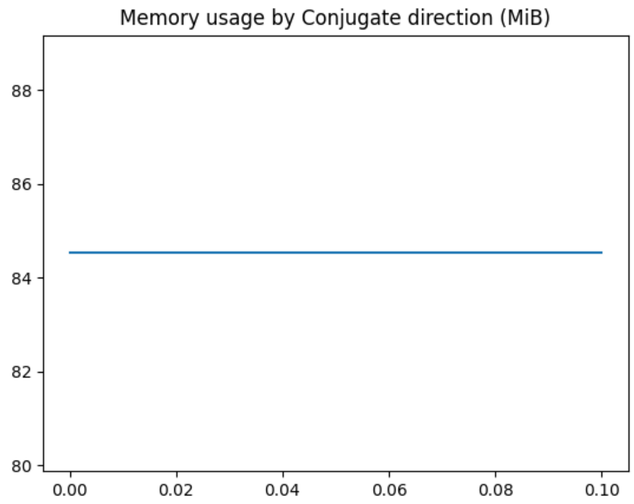


	Число вызовов
f	198

f_grad	3

Метод сопряжённых направлений

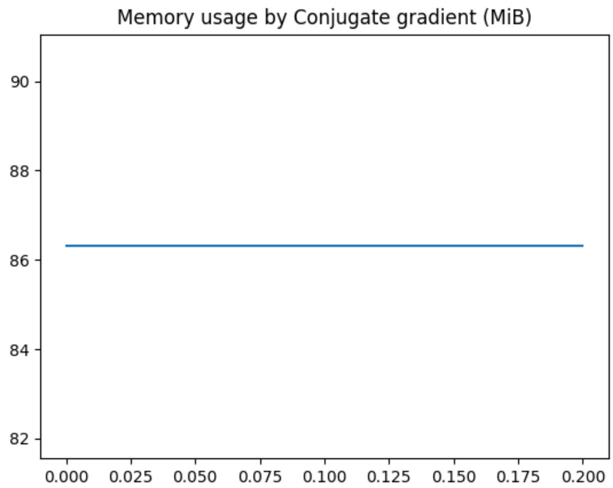
Завершился за 4 итерации
Максимальное потребление памяти: 84.53 Мб



	Число вызовов
f	0
f_grad	0

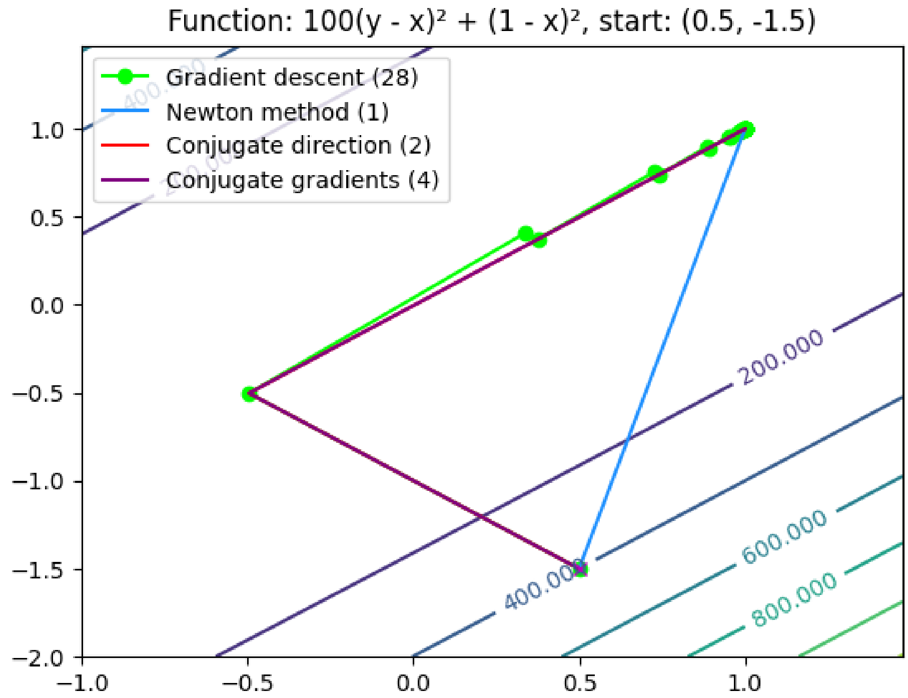
Метод сопряжённых градиентов

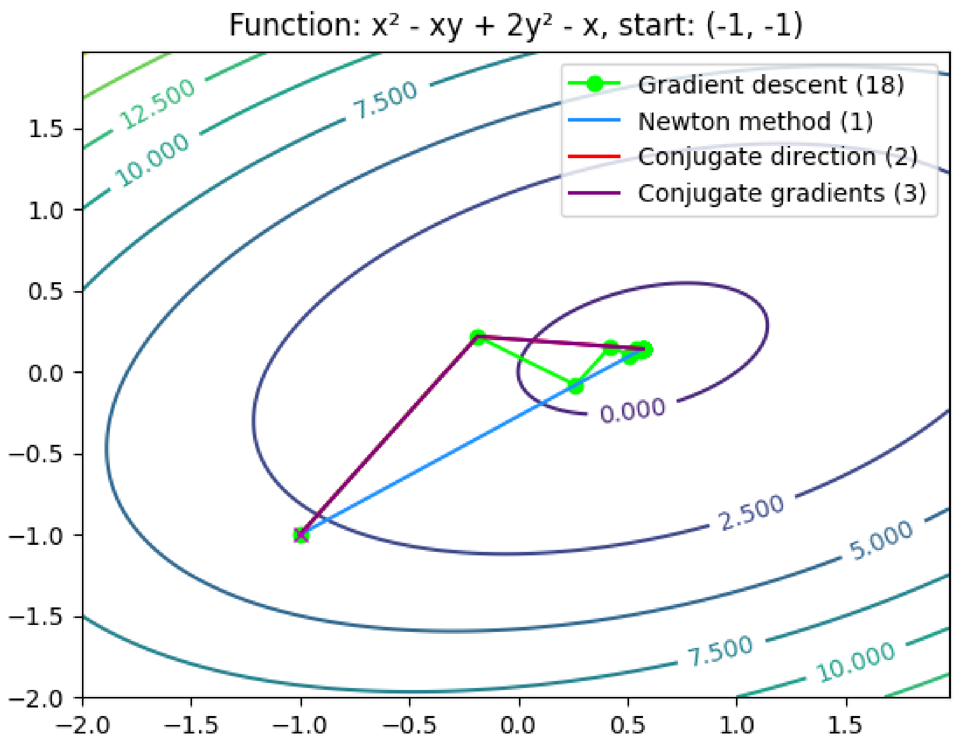
Завершился за 14 итераций
Максимальное потребление памяти: 86.30 Мб



	Число вызовов
f	924
f_grad	15

Задание 4.


Function: 100(y - x)² + (1 - x)², start: (0.5, -1.5)
Gradient descent (28)
Newton method (1)
Conjugate direction (2)
Conjugate gradients (4)
200.000
400.000
600.000
800.000
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.0
−0.5
0.0
0.5
1.0


Function: x² - xy + 2y² - x, start: (-1, -1)
Gradient descent (18)
Newton method (1)
Conjugate direction (2)
Conjugate gradients (3)
12.500
10.000
7.500
0.000
2.500
5.000
7.500
10.000
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5