Лабораторная работа №2

Методы первого и высших порядков

Задачи:

1. Описание следующих методов:
   - метод Ньютона (с постоянным шагом, с одномерным поиском)
   - метод Newton-CG (использование готовой реализации в scipy.optimize)
   - квазиньютоновский метод (использование готовой реализации в scipy.optimize)
2. Реализация вышеперечисленных методов
3. Исследование методов на эффективность на нескольких функциях:
   - сравнение эффективности методов нулевого порядка и градиентного спуска, метода Ньютона, квазиньютоновских методов
   - сравнение эффективности моей реализации метода Ньютона с методом Newton-CG из библиотеки scipy.optimize
   - сравнение эффективности методов нулевого порядка с квазиньютоновскими методами, если в последних производная вычисляется разностным методом

Часть 1. Описания используемых методов

Методы одномерного поиска

Метод золотого сечения, алгоритм:

1. Поиск значения функции f в точках $x_1 = b - \frac{b-a}{\varphi}, x_2 = a + \frac{b-a}{\varphi}$, где a, b $(a < b)$ – концы отрезка, на котором производится поиск минимума, $\varphi = \frac{1+\sqrt{5}}{2}$
2. Если $f(x_1) < f(x_2)$, то поиск на отрезке (a, x_2)
3. Если $f(x_1) \geq f(x_2)$, то поиск на отрезке (x_1, b)

Алгоритм продолжается до тех пор, пока не будет достигнута заданная точность. На каждом шаге можно использовать значение функции, вычисленное ранее, из-за чего происходит только одно вычисление функции за шаг.

Метод одномерного поиска по правилу Вольфе, алгоритм:

1. Выбор стартового значения коэффициента α, стартовой точки x, направления поиска p и коэффициентов $\varepsilon_1,\ \varepsilon_2,\ \theta_1,\ \theta_2$.
2. Проверка условий Армихо $(f(x + \alpha p) \leq f(x) + \varepsilon_1 a \cdot p^T \nabla f(x))$ и кривизны $(p^T \nabla f(x + \alpha p) \geq \varepsilon_2 p^T \nabla f(x))$:
   1) Если условие Армихо не выполняется, то переход:
      $\alpha \to \alpha * \theta_1, (0 < \theta_1 < 1)$
   2) Если условие Армихо выполняется, но не выполняется условие кривизны, то переход:
      $\alpha \to \alpha * \theta_2, (\theta_2 > 1)$
   3) Если условия Армихо и кривизны выполняются, то конец поиска, возвращение α.

Метод Ньютона и квазиньютоновские методы

Метод Ньютона, алгоритм:

1. Выбор стартовой точки $x_{(0)}$
2. Вычисление $x_{(i+1)} = x_{(i)} - \lambda_i \left[\nabla^2 f(x_{(i)})\right]^{-1} \nabla f(x_{(i)})$, где λ_i - вычисленный шаг (методом одномерного поиска по направлению, либо просто постоянный шаг)
3. Проверка условия остановки – достижения заданной точности.

Квазиньютоновские методы основаны на накоплении информации о кривизне целевой функции по наблюдениям за изменением градиента. На каждом шаге алгоритма вместо использования реального гессиана вычисляется его приближенное значение B_i.

Например, для метода BFGS:

$$B_{i+1} = B_i - \frac{B_i s_i s_i^T B_i^T}{s_i^T B_i s_i} + \frac{y_i y_i^T}{y_i^T s_i},$$

Где $s_i = x_{i+1} - x_i, y_i = \nabla f(x_{(i+1)}) - \nabla f(x_{(i)})$. В качестве начального значения B_0 обычно берется единичная матрица

Часть 2. Реализация методов

Ссылка на реализацию методов

При исследовании используется алгоритмы Newton-CG и BFGS, реализованные в библиотеке scipy.optimize.

Помимо используемого метода библиотека scipy.optimize предоставляет и другие функции для минимизации (или максимизации) целевых функций, возможно, с учетом ограничений. Она включает в себя решатели нелинейных задач (с поддержкой алгоритмов локальной и глобальной оптимизации), линейное программирование, ограниченный и нелинейный метод наименьших квадратов, поиск корней и подгонку кривой.

Методы минимизации вызывается с помощью функции scipy.optimize.minimize, где аргумент method равен названию вызываемого метода (в данном случае «Newton-CG» или «BFGS»).

Функция scipy.optimize.minimize так же принимает минимизируемую функцию, стартовую точку и другие гиперпараметры. Можно установить стратегию вычисления градиента и гессиана функции: передать явную формулу и конечно-разностную схему для численной оценки (2-point, 3-point, cs). Для вычисления гессиана можно указать стратегию вычисления (для квазиньютоновских методов, например).

Для метода Newton-CG и BFGS есть также гиперпараметры c_1 и c_2 – коэффициенты для проверки правил Армихо и кривизны (ε_1 и ε_2 в моем описании условия Вольфе). По умолчанию c_1 = 1e-4, c_2 = 0.9 (эти же значения будут использованы в дальнейшем для сравнения)

При вызове метода Newton-CG в качестве параметра необходимо также передавать градиент минимизируемой функции.

Часть 3. Исследование методов на эффективность

В качестве функций, на которых будет проводиться исследование эффективности методов поиска, были взяты функции:

- $e^{10x^2+y^2}$(рис. 1)
- $\ln(1+100x^2+y^2)$ (рис. 2)
- $\cos(e^x)\cdot\cos(e^y)$ (рис. 3)
- $(1-x)^2 + 100(y-x^2)^2$ – функция Розенброка (рис. 4)

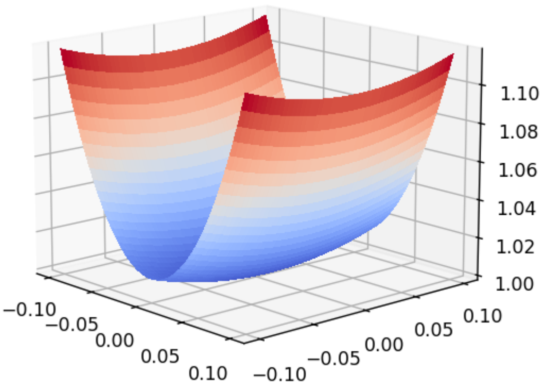


Рис. 1. График функции $e^{10x^2+y^2}$

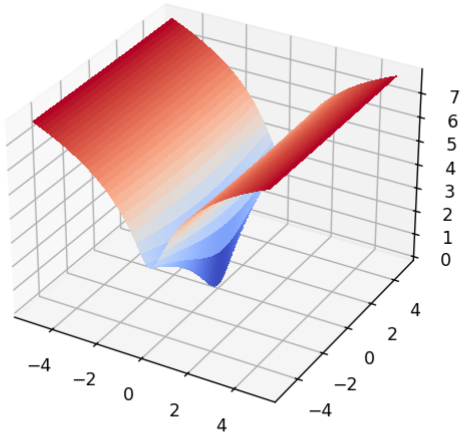


Рис. 2. График функции $\ln(1+100x^2+y^2)$

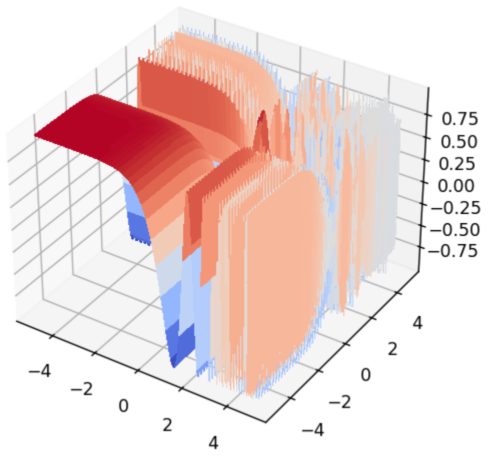


Рис. 3. График функции $\cos(e^x)\cdot\cos(e^y)$

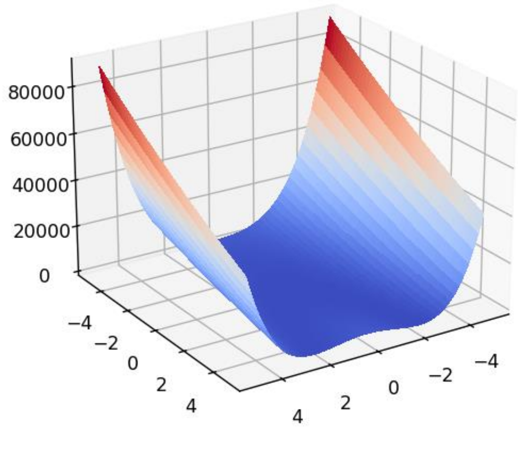


Рис. 4. График функции $(1-x)^2 + 100(y-x^2)^2$

Количество итераций в зависимости от выбора метода, функции и стартовой точки (точность = 1e-9*)

Стартовая точка	Мой метод Ньютона			Newton-CG	BFGS (разн. схемы)	BFGS (аналит. выражения)	Нелдер-Мид	Град. спуск (с золотым сечением) до eps = 1e-6*
	Постоянный шаг	Золотое сечение	Условия Вольфе					
$e^{10x^2+y^2}$								
(0.02, 0.3)	4	3	4	4	7	8	70	50
(0.1, 0.1)	4	3	4	6	5	5	71	11
(1, 1)	18	2	18	19	30	30	78	-
$\ln(1 + 100x^2 + y^2)$								
(0.001, 0.1)	4	3	4	6	6	8	67	372
(0.01, 0.01)	4	3	4	4	6	7	66	92
(0.05, 0)	5	3	5	4	6	5	76	4
$\cos(e^x) \cdot \cos(e^y)$								
$(\ln(\pi + 0.5), \ln(2\pi + 0.5))$	5	5	5	6	11	11	66	20
$(\ln(\pi - 0.5), \ln(2\pi + 0.5))$	6	4	6	6	7	8	66	17
$(\ln(\pi + 1), \ln(2\pi))$	4	5	4	4	8	6	67	3
$(1 - x)^2 + 100(y - x^2)^2$								
(0, 0)	3	10	6	32	21	21	118	6025
(1, 0)	2	2	2	30	34	34	117	5931
(0, 1)	6	11	9	29	23	23	137	6026

Количество вычислений значений функции | градиента | гессиана в зависимости от метода, функции и стартовой точки

Стартовая точка	Мой метод Ньютона			Newton-CG	BFGS (разн. схемы)	BFGS (ан. выр.)	Нелдер-Мид	Град. спуск (с зол. сеч.) до eps = 1e-6*
	Постоянный шаг	Золотое сечение	Условия Вольфе					
$e^{10x^2+y^2}$								
(0.02, 0.3)	0 | 4 | 4	45 | 3 | 3	8 | 14 | 4	5 | 15 | 0	252 | 80 | 0	11 | 11 | 0	139 | 0 | 0	625 | 50 | 0
(0.1, 0.1)	0 | 4 | 4	45 | 3 | 3	8 | 14 | 4	7 | 19 | 0	27 | 9 | 0	9 | 9 | 0	143 | 0 | 0	146 | 11 | 0
(1, 1)	0 | 18 | 18	34 | 2 | 2	36 | 70 | 18	20 | 49 | 0	99 | 33 | 0	32 | 32 | 0	156 | 0 | 0	-
$\ln(1+100x^2+y^2)$								
(0.001, 0.1)	0 | 4 | 4	45 | 3 | 3	8 | 12 | 4	9 | 37 | 0	27 | 9 | 0	50 | 38 | 0	135 | 0 | 0	4092 | 372 | 0
(0.01, 0.01)	0 | 4 | 4	45 | 3 | 3	8 | 14 | 4	5 | 13 | 0	27 | 9 | 0	48 | 36 | 0	131 | 0 | 0	1057 | 92 | 0
(0.05, 0)	0 | 5 | 5	44 | 3 | 3	10 | 18 | 5	31 | 24 | 0	150 | 46 | 0	8 | 8 | 0	155 | 0 | 0	44 | 4 | 0
$\cos(e^x)\cdot\cos(e^y)$								
$(\ln(\pi+0.5), \ln(2\pi+0.5))$	0 | 5 | 5	79 | 5 | 5	10 | 18 | 5	6 | 17 | 0	69 | 23 | 0	14 | 14 | 0	130 | 0 | 0	230 | 20 | 0
$(\ln(\pi-0.5), \ln(2\pi+0.5))$	0 | 6 | 6	60 | 4 | 4	12 | 20 | 6	7 | 22 | 0	198 | 62 | 0	10 | 10 | 0	131 | 0 | 0	195 | 17 | 0
$(\ln(\pi+1), \ln(2\pi))$	0 | 4 | 4	79 | 5 | 5	8 | 14 | 4	4 | 7 | 0	114 | 34 | 0	9 | 9 | 0	132 | 0 | 0	38 | 3 | 0
$(1-x)^2 + 100(y-x^2)^2$								
(0, 0)	0 | 3 | 3	171 | 10 | 10	12 | 22 | 6	50 | 201 | 0	78 | 26 | 0	26 | 26 | 0	222 | 0 | 0	66276 | 6025 | 0
(1, 0)	0 | 2 | 2	30 | 2 | 2	4 | 6 | 2	41 | 117 | 0	117 | 39 | 0	39 | 39 | 0	222 | 0 | 0	65250 | 5931 | 0
(0, 1)	0 | 6 | 6	189 | 11 | 11	18 | 32 | 9	39 | 98 | 0	87 | 29 | 0	29 | 29 | 0	260 | 0 | 0	66290 | 6026 | 0

Выводы

1. Реализованный мной метод Ньютона и Newton-CG, а также квазиньютоновские методы работают примерно одинаково эффективно.

2. Квазиньютоновский метод с аналитическим вычислением градиента и разностным работают примерно за одинаковое количество итераций, однако методу с разностным вычислением градиента требуется больше вычислений значений функции (как раз для поиска градиента).

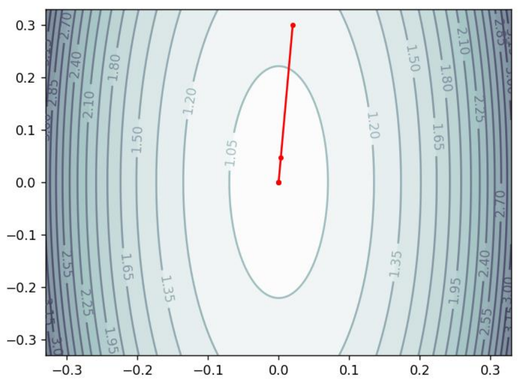
3. Метод Ньютона с постоянным шагом и поиском с помощью условий Вольфе на выбранных функциях работают одинаково эффективно (вероятно, из-за удачного выбора значения постоянного шага). Метод Ньютона с золотым сечением требует меньше итераций, так как более точно находит оптимум по направлению. Однако из-за этого он требует и большего количества вычислений значений функции.

4. Методы Ньютона и квазиньютоновские методы работают более эффективно на функциях с легко вычисляемым гессианом, чем методы нулевого и первого порядков (градиентный спуск и метод Нелдера-Мида). Это связано с более «стратегическим» выбором направления и более тщательным анализом поведения и кривизны функции. Но за эту эффективность мы платим поиском и вычислением гессиана функции (и из-за этого метод Ньютона может быть менее эффективным на некоторых функциях)

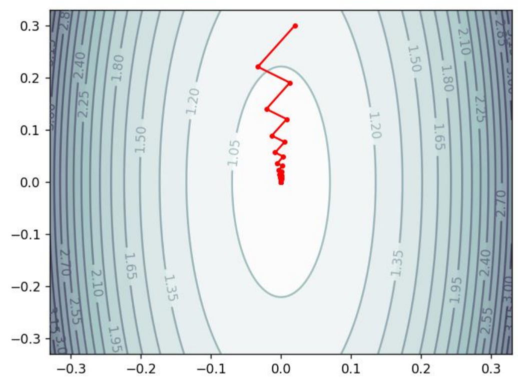
   Пример различия эффективности метода Ньютона и градиентного спуска:

$$e^{10x^2+y^2}, \text{ старт} = (0.02, 0.3)$$

Метод Ньютона

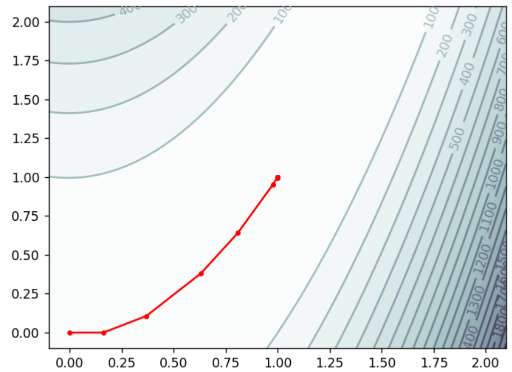


Градиентный спуск

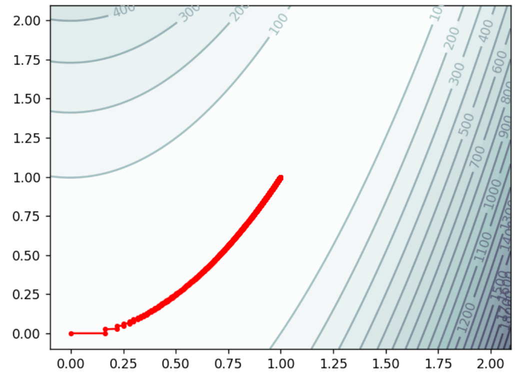


$(1 - x)^2 + 100(y - x^2)^2$, старт = (0, 0)

Метод Ньютона



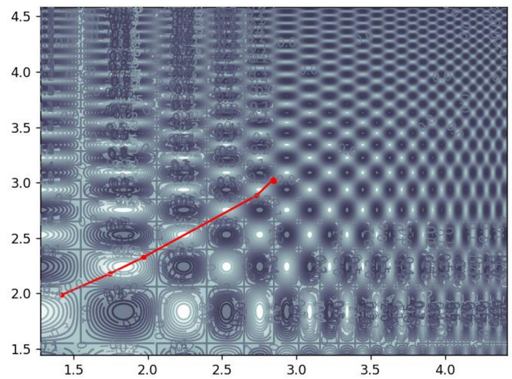
Градиентный спуск



Дополнительное задание 2 (п. 1, 2)

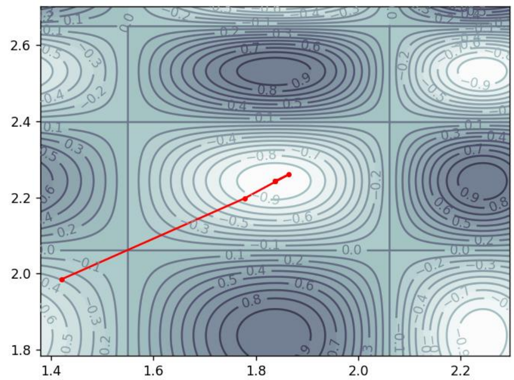
При рассмотрении функции $\cos(e^x) \cdot \cos(e^y)$ можно заметить, что метод Ньютона не всегда работает так, как ожидается. Рассмотрим результат выполнения поиска из точки $(\ln(\pi + 1), \ln(2\pi + 1))$:

Метод Ньютона (с постоянным шагом)



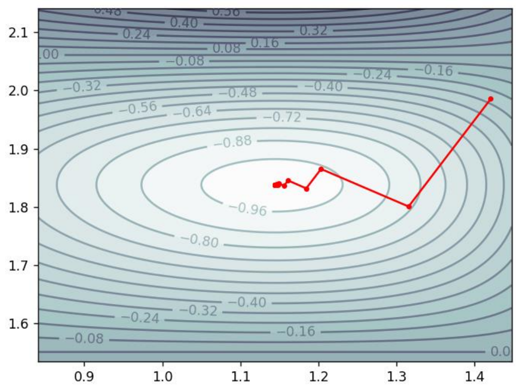
Сошелся в седловой точке

Метод Ньютона (с условиями Вольфе)



Сошелся в точке $(\ln(2\pi), \ln(3\pi))$

Градиентный спуск



Сошелся в точке

$$(\ln(\pi), \ln(2\pi))$$

Метод сошелся не в ожидаемой точке оптимума. Это связано с тем, что выбранная начальная точка не была достаточно близка к ожидаемой. Поэтому необходимо следить за тем, чтобы выполнялись необходимые условия схождения метода Ньютона.