НАО «Восточно-Казахстанский технический университет имени Д. Серикбаева»

Дисциплина «Методы оптимизации»

ЛАБОРАТОРНАЯ РАБОТА 6

Градиентный метод, наискорейший и покоординатный спуск, метод Ньютона

Методические указания с примерами расчётов и программами на Python

Дисциплина: «Методы оптимизации». **Образовательная программа:** 6B06103 «Математическое и компьютерное моделирование». **Лабораторная работа 6 — 5 академических часов.**

1. Цель и организация работы

Цель работы — освоить четыре метода безусловной минимизации функции нескольких переменных, проверить условия их применения и сравнить результаты при одинаковом критерии точности. Работа продолжает изучение геометрии квадратичных функций и классических условий экстремума.

Сначала следует изучить методические пояснения и выполнить первые шаги вручную. Затем нужно запустить общий программный пример и только после этого перейти к своему варианту. Программа используется для проверки и исследования, поэтому в отчёте должны быть представлены вывод формул, промежуточные расчёты и объяснение полученных результатов.

Этап	Содержание	Время
Подготовка	Постановка задачи, градиент, матрица Гессе, аналитическое решение	1 академический час
Ручные расчёты	Первый шаг каждого метода; один полный координатный цикл	1 академический час
Реализация	Запуск кода, разбор обновлений и условий остановки	1 академический час
Исследование	Индивидуальный вариант, влияние точности и поворота осей	1 академический час
Оформление и защита	Графики, сравнительная таблица, выводы и ответы	1 академический час

Требуемая подготовка: частные производные, градиент, матрица Гессе, положительная определённость, решение линейных систем, базовые массивы и циклы Python. Используются библиотеки NumPy и Matplotlib. Полные программы находятся в отдельном архиве; в методичке приведены ключевые фрагменты и объяснения.

2. Постановка задачи и область применимости программ

Рассматривается задача $f(\mathbf{x}) \to \min$ на всей плоскости. Для вычислительного сравнения используется строго выпуклая квадратичная функция:

$$f(\mathbf{x}) = c + \frac{1}{2}(\mathbf{x}-\mathbf{h})^T H(\mathbf{x}-\mathbf{h}), \qquad H = H^T > 0.$$

Здесь $\mathbf{h}$ — фиксированный вектор, c — константа, а H — постоянная симметричная положительно определённая матрица. Условие $H > 0$ означает, что $\mathbf{v}^T H\mathbf{v} > 0$ для любого ненулевого вектора $\mathbf{v}$.

Градиент, матрица Гессе и точное решение имеют вид

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = H(\mathbf{x} - \mathbf{h}), \qquad \nabla^2 f(\mathbf{x}) = H,$$

$$\mathbf{x}^* = \mathbf{h}, \qquad f^* = c.$$

Знание точного решения необходимо для независимой проверки алгоритмов. При этом итерации используют значение функции, градиент и матрицу Гессе по формулам метода. Нельзя заменять исследование командой, которая сразу возвращает известный вектор $\mathbf{h}$.

Обозначим наименьшее и наибольшее собственные значения через μ и L. Тогда $0 < \mu \le L$, а число обусловленности равно $\kappa = L/\mu$. Большая разница собственных значений соответствует вытянутым эллипсам уровня и может замедлять методы спуска.

Границы реализации. Приложенный код предназначен для положительно определённых квадратичных задач без ограничений. Формулы точного шага и однократного решения методом Ньютона опираются на эту структуру. Общие нелинейные функции требуют других вычислений производных, поиска шага и дополнительных проверок; простая замена строки целевой функции не превращает программу в универсальный решатель.

3. Общая схема методов спуска

Во многих методах новая точка строится по формуле

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k.$$

Здесь $\mathbf{d}^k$ задаёт направление, а $\alpha_k > 0$ — длину шага. Эти решения следует различать. Даже правильное направление спуска при слишком большом шаге может привести к росту функции.

Для дифференцируемой функции условие направления спуска имеет вид

$$\nabla f(\mathbf{x}^k)^T \mathbf{d}^k < 0.$$

Действительно, производная функции $\varphi(\alpha) = f(\mathbf{x}^k + \alpha\mathbf{d}^k)$ в нуле равна этому скалярному произведению. Если оно отрицательно, функция уменьшается при достаточно малых положительных шагах.

В градиентных методах выбирают $\mathbf{d}^k = -\mathbf{g}_k$. Тогда $\mathbf{g}_k^T \mathbf{d}^k = -\| \mathbf{g}_k \|_2^2 < 0$, если градиент ненулевой. В евклидовой норме нормированное направление $-\mathbf{g}_k / \| \mathbf{g}_k \|_2$ даёт наибольшую скорость локального уменьшения среди единичных направлений. Это утверждение относится к бесконечно малому перемещению и не означает, что путь к решению будет самым коротким.

Общая последовательность действий: вычислить градиент, проверить остановку, выбрать направление и шаг, получить новую точку, записать диагностические величины и повторить. Обязательны ограничение числа итераций и отдельный признак успешного завершения.

4. Градиентный метод с постоянным шагом

4.1. Формула и условие сходимости

Метод использует одно и то же положительное значение α:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \mathbf{g}_k.$$

Для рассматриваемой квадратичной функции введём ошибку $\mathbf{e}_k = \mathbf{x}^k - \mathbf{h}$. Подстановка градиента даёт

$$\mathbf{e}_{k+1} = (I - \alpha H)\mathbf{e}_k.$$

В собственных направлениях матрицы каждая компонента ошибки умножается на $1 - \alpha\lambda_i$. Для сходимости из любой начальной точки достаточно и необходимо, чтобы модуль каждого множителя был меньше единицы. Получаем условие

$$0 < \alpha < \frac{2}{L}.$$

При $\alpha = 1/L$ коэффициент сжатия в евклидовой норме не превосходит $1 - \mu/L$. Если $к$ велико, этот коэффициент близок к единице и уменьшение ошибки может быть медленным. Для квадратичной задачи оптимальный в смысле наихудшего спектрального коэффициента постоянный шаг равен $2/(L + \mu)$; это не универсальная формула для произвольной нелинейной функции.

4.2. Почему функция уменьшается

Точное квадратичное разложение даёт

$$f(\mathbf{x} - \alpha\mathbf{g}) = f(\mathbf{x}) - \alpha \parallel \mathbf{g} \parallel_2^2 + \frac{\alpha^2}{2}\mathbf{g}^T H\mathbf{g}.$$

Так как $\mathbf{g}^T H\mathbf{g} \le L \parallel \mathbf{g} \parallel_2^2$, то

$$f(\mathbf{x} - \alpha\mathbf{g}) \le f(\mathbf{x}) - \alpha\left(1 - \frac{\alpha L}{2}\right) \parallel \mathbf{g} \parallel_2^2.$$

При $0 < \alpha < 2/L$ и ненулевом градиенте правая часть меньше текущего значения. Если шаг слишком мал, метод устойчив, но делает очень короткие перемещения. Если шаг слишком велик, компоненты ошибки могут возрастать.

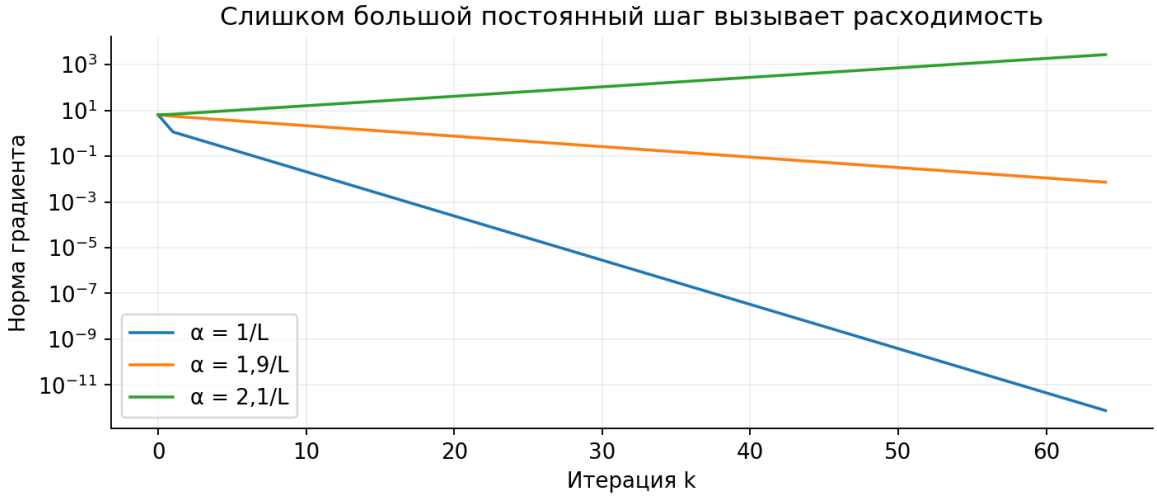


Рисунок 1 — Три шага на общей задаче из раздела 8; при $\alpha = 2{,}1/L$ наблюдается расходимость.

Приложенный решатель отклоняет постоянный шаг вне интервала устойчивости. Рисунок с расходящейся последовательностью построен отдельным демонстрационным расчётом. В неквадратичных задачах неизвестное безопасное значение шага часто подбирают уменьшением пробного шага до выполнения условия достаточного убывания [1, § 6.1].

5. Метод наискорейшего спуска

5.1. Отличие от постоянного шага

Направление остаётся антиградиентным, но на каждой итерации шаг выбирается из одномерной задачи:

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}^k - \alpha \mathbf{g}_k).$$

В данной работе под наискорейшим спуском понимается градиентный метод с **точной минимизацией вдоль антиградиента**. Для общей функции эту одномерную задачу решают численно. Для квадратичной функции формула выводится явно [1, § 6.2].

Положим $\varphi(\alpha) = f(\mathbf{x} - \alpha \mathbf{g})$, где $\mathbf{g}$ зафиксирован в текущей точке. Тогда

$$\varphi'(\alpha) = -\mathbf{g}^T \mathbf{g} + \alpha \mathbf{g}^T H \mathbf{g}.$$

Поскольку $H > 0$, при ненулевом градиенте вторая производная положительна. Поэтому

$$\alpha_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T H \mathbf{g}_k}, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{g}_k.$$

Порядок действий: сначала проверить норму градиента, затем вычислять дробь. В точке решения и числитель, и знаменатель равны нулю, поэтому вычисление шага до проверки остановки приводит к неопределённости.

5.2. Свойство ортогональности и его смысл

Для точного шага на квадратичной функции

$$\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k H \mathbf{g}_k,$$

$$\mathbf{g}_{k+1}^T \mathbf{g}_k = \mathbf{g}_k^T \mathbf{g}_k - \alpha_k \mathbf{g}_k^T H \mathbf{g}_k = 0.$$

Соседние ненулевые градиенты ортогональны. Это полезная независимая проверка реализации. В численных расчётах скалярное произведение сравнивают с допуском, учитывающим длины векторов, а не требуют буквального машинного нуля.

На вытянутых эллипсах уровня метод может двигаться зигзагом. Точный минимум вдоль текущей прямой не гарантирует наименьшего общего числа итераций среди всех алгоритмов. В частности, метод, учитывающий матрицу Гессе, способен выбрать более подходящее направление.

6. Покоординатный спуск

6.1. Один цикл и точная минимизация по координате

В покоординатном методе изменяют одну координату, сохраняя остальные. В данной работе реализуется циклический вариант с точной минимизацией по каждой координате. В учебнике [1, § 6.4] такой вариант рассматривается как метод Гаусса — Зейделя; в § 6.3 также описывается покоординатный спуск с подбором шага.

Пусть $\mathbf{e}_i$ — i-й координатный вектор. Одномерная функция имеет вид

$$\psi(\delta) = f(\mathbf{x} + \delta \mathbf{e}_i) = f(\mathbf{x}) + \delta g_i + \frac{1}{2} H_{ii} \delta^2.$$

Из условия $\psi'(\delta) = 0$ получаем

$$\delta_i = -\frac{g_i}{H_{ii}}, \qquad x_i \leftarrow x_i - \frac{g_i}{H_{ii}}.$$

Положительность H_{ii} следует из положительной определённости. После обновления одной координаты градиент необходимо пересчитать или корректно обновить. Следующая координата использует **уже изменённую точку**. Одновременное вычисление всех новых координат по старому вектору задаёт другую итерационную схему.

Один полный цикл в двумерной задаче состоит из изменения x_1, затем x_2. В программе число итераций этого метода означает число полных циклов. Промежуточные точки сохраняются для построения ломаной траектории.

6.2. Убывание функции и влияние осей

Подстановка найденного перемещения даёт

$$f(\mathbf{x} + \delta_i \mathbf{e}_i) - f(\mathbf{x}) = -\frac{g_i^2}{2H_{ii}} \leq 0.$$

Если соответствующая компонента градиента ненулевая, значение строго уменьшается. После минимизации по координате её частная производная равна нулю. Однако следующее изменение другой координаты может снова сделать первую производную ненулевой, если в функции имеются смешанные члены.

Для положительно определённой квадратичной функции циклическая точная минимизация сходится к единственному решению. Скорость зависит от ориентации эллипсов относительно координатных осей. Если матрица диагональна, каждая координата независима от остальных и одного полного цикла достаточно. При повёрнутых эллипсах может потребоваться много циклов.

7. Метод Ньютона

7.1. Построение направления

Метод использует квадратичную модель функции в текущей точке:

$$m_k(\mathbf{d}) = f(\mathbf{x}^k) + \mathbf{g}_k^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T H_k \mathbf{d}.$$

Если $H_k > 0$, минимум модели определяется системой

$$H_k \mathbf{d}_k = -\mathbf{g}_k, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}_k.$$

В программе нужно **решать линейную систему**, например функцией np.linalg.solve. Явное вычисление обратной матрицы для этой цели не требуется и обычно является менее удачным вычислительным решением.

Для нашей квадратичной функции $H_k = H$, а модель совпадает с самой функцией. Поэтому

$$\mathbf{d}_k = -H^{-1}H(\mathbf{x}^k - \mathbf{h}) = \mathbf{h} - \mathbf{x}^k.$$

После одного полного шага получается точное решение в арифметике без округления. При машинных вычислениях остаётся малая численная ошибка. Если начальная точка уже удовлетворяет критерию остановки, шаг вообще не выполняется.

7.2. Что меняется для неквадратичной функции

Для общей функции матрица Гессе меняется от точки к точке. Положительная определённость H_k гарантирует, что ненулевое ньютоновское направление является направлением спуска:

$$\mathbf{g}_k^T \mathbf{d}_k = -\mathbf{g}_k^T H_k^{-1} \mathbf{g}_k < 0.$$

Однако это **не гарантирует уменьшения исходной нелинейной функции при полном шаге**. Направление построено по локальной модели; далеко от текущей точки модель может быть неточной. Поэтому применяют демпфирование и поиск шага, например проверку условия Армихо:

$$f(\mathbf{x}^k + \alpha \mathbf{d}_k) \le f(\mathbf{x}^k) + \sigma \alpha \mathbf{g}_k^T \mathbf{d}_k, \qquad 0 < \sigma < 1.$$

При невыполнении условия положительный шаг уменьшают. Если матрица Гессе не положительно определена, дополнительно меняют направление или регуляризуют матрицу. При гладкости, невырожденном положительном Гессиане в минимуме и достаточно близком начальном приближении классический метод Ньютона имеет локальную квадратичную сходимость. Эти условия нельзя заменять утверждением «Ньютон всегда сходится быстрее».

Пример опасного направления. Для $f(x) = x^4 - x^2$ в точке $x = 0{,}1$ получаем $f' = -0{,}196$, $f'' = -1{,}88$. Ньютоновское перемещение $d = -f'/f'' \approx -0{,}104255$. Произведение $f'd > 0$, то есть направление не является направлением спуска. Этот пример не входит в класс задач приложенного решателя и рассматривается аналитически.

8. Подробный общий пример

8.1. Исследование функции

Рассмотрим функцию

$$f(x, y) = x^2 + xy + 2y^2 - 4x - 5y + 6, \qquad \mathbf{x}^0 = (0; 0)^T.$$

Её градиент и матрица Гессе равны

$$\nabla f(x, y) = \begin{pmatrix} 2x + y - 4 \\ x + 4y - 5 \end{pmatrix}, \qquad H = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

Главные угловые миноры: $2 > 0$ и $\det H = 7 > 0$. По критерию Сильвестра матрица положительно определена. Следовательно, функция строго выпукла и единственная стационарная точка является глобальным минимумом.

Из системы $2x + y = 4, x + 4y = 5$ получаем

$$x^* = \frac{11}{7}, \qquad y^* = \frac{6}{7}, \qquad f^* = \frac{5}{7}.$$

Собственные значения равны $3 - \sqrt{2}$ и $3 + \sqrt{2}$, поэтому $\mu \approx 1{,}585786$, $L \approx 4{,}414214$, $\kappa \approx 2{,}783612$. В начальной точке $f_0 = 6, \mathbf{g}_0 = (-4; -5)^T$ и $\| \mathbf{g}_0 \|_2 = \sqrt{41}$.

8.2. Первый шаг градиентного метода

Для общего примера выберем $\alpha = 1/4$. Шаг допустим, поскольку $1/4 < 2/\left(3 + \sqrt{2}\right) \approx 0{,}453082$.

$$\mathbf{x}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} -4 \\ -5 \end{pmatrix} = \begin{pmatrix} 1 \\ 5/4 \end{pmatrix}.$$

Подстановка даёт $f(\mathbf{x}^1) = 9/8 = 1{,}125$. Новый градиент равен $(-3/4; 1)^T$, его норма — $5/4$. Значение функции и норма градиента уменьшились, но точность 10^{-6} ещё не достигнута.

8.3. Первый шаг наискорейшего спуска

Вычислим произведение матрицы на начальный градиент:

$$H\mathbf{g}_0 = \begin{pmatrix} -13 \\ -24 \end{pmatrix}, \qquad \mathbf{g}_0^T H \mathbf{g}_0 = 172, \qquad \mathbf{g}_0^T \mathbf{g}_0 = 41.$$

Следовательно,

$$\alpha_0 = \frac{41}{172}, \qquad \mathbf{x}^1 = \begin{pmatrix} 41/43 \\ 205/172 \end{pmatrix} \approx \begin{pmatrix} 0{,}953488 \\ 1{,}191860 \end{pmatrix}.$$

Функция принимает значение $383/344 \approx 1{,}113372$. Новый градиент равен $(-155/172; 124/172)^T$. Проверка ортогональности:

$$\mathbf{g}_1^T \mathbf{g}_0 = \frac{620 - 620}{172} = 0.$$

Значение после этого шага ниже, чем после градиентного шага $1/4$, потому что выбран точный минимум вдоль той же прямой. Этот вывод сравнивает именно первые шаги из одной точки, а не общую эффективность любых двух алгоритмов.

8.4. Первый полный цикл покоординатного спуска

Сначала изменим x при $y = 0$. Условие $2x - 4 = 0$ даёт $x = 2$, поэтому промежуточная точка равна $(2; 0)$ и $f = 2$.

Затем изменим y уже при $x = 2$. Условие $2 + 4y - 5 = 0$ даёт $y = 3/4$. Конец первого цикла:

$$\mathbf{x}^1 = (2; 3/4)^T, \qquad f(\mathbf{x}^1) = 7/8 = 0{,}875.$$

В этой точке градиент равен $(3/4; 0)^T$. Вторая компонента нулевая, поскольку только что выполнена точная минимизация по y. Первая снова ненулевая из-за смешанного члена xy. Следовательно, одного цикла в данном примере недостаточно.

8.5. Первый шаг Ньютона

Нужно решить систему

$$\begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}.$$

Её решение $\mathbf{d} = (11/7; 6/7)^T$. После прибавления к нулевой начальной точке получаем точный минимум. Нулевой градиент и положительная определённость подтверждают результат.

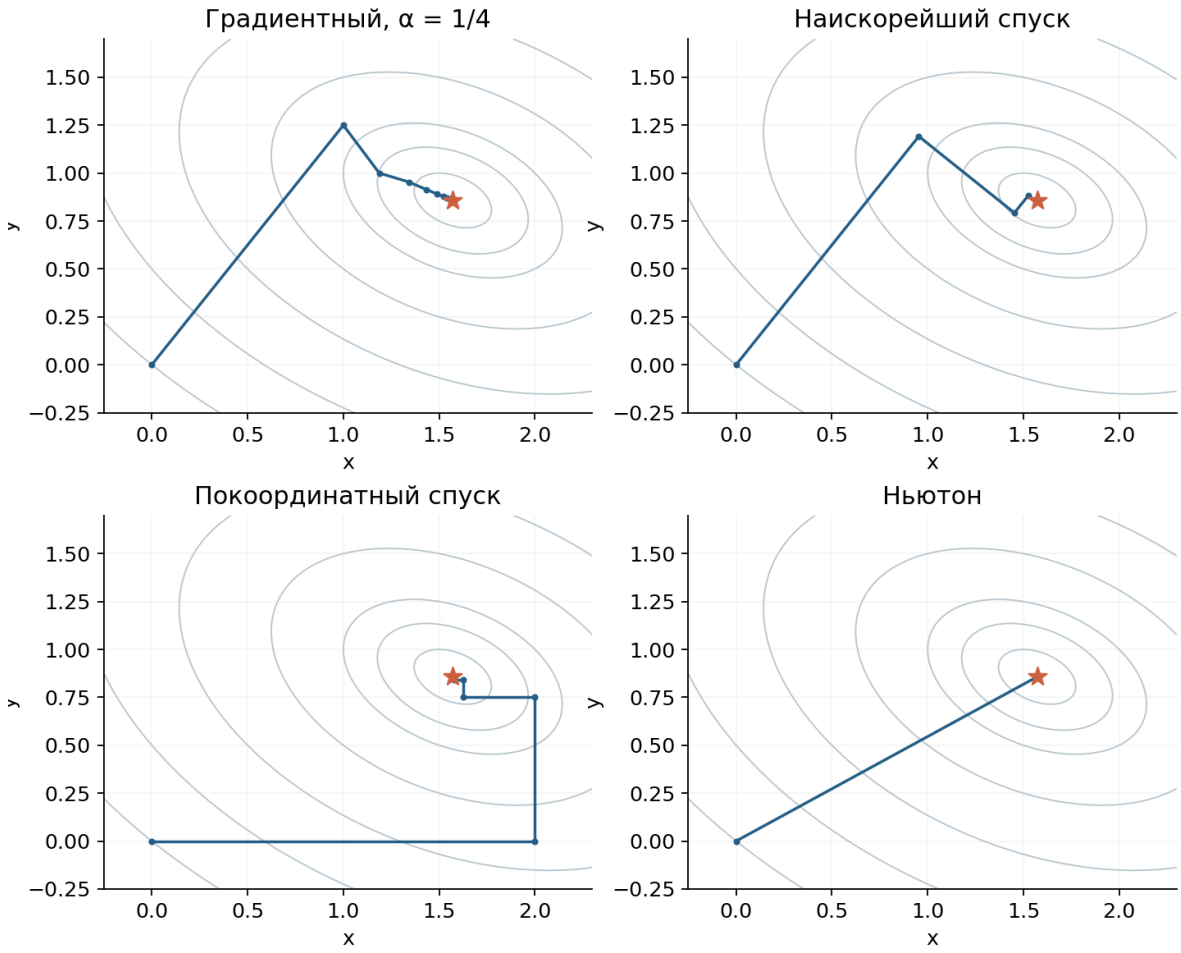


Рисунок 2 — Траектории на линиях уровня общей функции. Звёздочкой отмечен минимум; координатная траектория включает промежуточные точки внутри циклов.

9. Критерий остановки и оценка точности

Для всех методов используется один основной критерий:

$$\| \nabla f(\mathbf{x}^k) \|_2 \leq \varepsilon_g.$$

В основной серии расчётов $\varepsilon_g = 10^{-6}$. Для положительно определённой квадратичной задачи этот критерий имеет количественный смысл. Так как $\mathbf{g} = H(\mathbf{x} - \mathbf{h})$, то

$$\| \mathbf{x} - \mathbf{h} \|_2 \leq \frac{\| \mathbf{g} \|_2}{\mu}, \qquad 0 \leq f(\mathbf{x}) - c \leq \frac{\| \mathbf{g} \|_2^2}{2\mu}.$$

Первая оценка следует из нижней границы собственного значения. Для второй используем равенство $f - c = 1/2\, \mathbf{g}^T H^{-1} \mathbf{g}$ и верхнюю оценку собственного значения обратной матрицы $1/\mu$.

Малая длина шага сама по себе не является надёжным критерием: при искусственно маленьком α перемещения могут быть малы далеко от минимума. Аналогично, небольшая разность соседних значений функции может означать медленный прогресс, а не найденное решение.

Статус программы	Интерпретация
converged	Норма градиента не превосходит заданный допуск.
max_iter	Достигнут предел итераций; заданная точность не подтверждена.
stagnation	Представление точки перестало изменяться, но градиент ещё слишком велик.
nonfinite	Появилось неконечное значение: требуется проверка данных и численной устойчивости.

Вычисленный минимум следует принимать только после проверки статуса и остаточной нормы. Статус достижения предела итераций нельзя автоматически переименовывать в успешное завершение.

10. Программная реализация

10.1. Состав архива и запуск

В архив входят descent_methods.py — четыре алгоритма и модель задачи; run_variant.py — запуск примера или варианта; variants.json — параметры 18 вариантов; one_dim_basics.py — демонстрация правила исключения интервала из лекции; requirements.txt и README.txt.

Распакуйте архив в одну папку. В терминале этой папки установите зависимости и выполните общий пример:

```
python -m pip install -r requirements.txt
python run_variant.py --example
python run_variant.py --variant 6 --eps 1e-6
```

Для общего примера постоянный шаг равен 1/4. Для индивидуальных вариантов программа автоматически выбирает $1/L$. Числовой номер после --variant заменяют номером своего варианта. Результаты сохраняются в отдельную подпапку: таблица CSV, история CSV и два рисунка PNG. Название подпапки содержит номер варианта и точность; повторный запуск тех же параметров обновляет эти результаты.

10.2. Как хранится функция

Используется центрированная запись, которая особенно удобна для проверки известных минимумов:

```
H = np.array([[2.0, 1.0], [1.0, 4.0]])
h = np.array([11.0/7.0, 6.0/7.0])
q = Quadratic(H, h, c=5.0/7.0)
x0 = np.array([0.0, 0.0])
result = minimize(q, x0, method="fixed", alpha=0.25)
print(result["status"], result["x"], result["norm"])
```

Конструктор проверяет размеры, конечность данных, симметрию и положительную определённость матрицы. Матрица, не удовлетворяющая этим требованиям, отклоняется с объяснением. Вектор h является частью учебной модели, позволяющей заранее знать точное решение каждого варианта.

Градиент вычисляется аналитически, без конечных разностей:

```
z = x - q.h
value = q.c + 0.5 * z @ q.H @ z
g = q.H @ z
```

Оператор @ означает матричное или скалярное произведение. Обычное умножение * для массивов NumPy выполняется поэлементно и не заменяет эти операции.

10.3. Ключевые обновления

После проверки ненулевого градиента постоянный и точный шаги задаются так:

```
# Градиентный метод: 0 < alpha < 2/L
x = x - alpha * g

# Наискорейший спуск для квадратичной функции
step = (g @ g) / (g @ q.H @ g)
x = x - step * g
```

Эти фрагменты показывают две альтернативы; их не следует выполнять последовательно как одну итерацию. В полном файле выбор выполняется по параметру method.

Покоординатный цикл содержит немедленный пересчёт градиента:

```
for i in range(x.size):
    x[i] -= g[i] / q.H[i, i]
    g = q.gradient(x)
```

Если пересчитать градиент только после завершения цикла, следующая координата будет использовать устаревшие данные и получится другой алгоритм. В более эффективной реализации градиент можно обновлять по формуле $\mathbf{g} \leftarrow \mathbf{g} + H\mathbf{e}_i\delta_i$, но в учебном коде оставлен полный пересчёт для прозрачности.

Шаг Ньютона реализуется без явной обратной матрицы:

```
d = np.linalg.solve(q.H, -g)
x = x + d
g = q.gradient(x)
```

Общий решатель возвращает историю значений, норму градиента, путь, статус и счётчики. Для градиентного метода и Ньютона одна итерация означает одно векторное обновление; для координатного метода — полный цикл по всем координатам.

10.4. Учёт вычислительной работы

Показатель	Что именно считает программа
iterations	Векторные шаги либо полные координатные циклы.
updates	Векторные обновления у трёх методов; отдельные скалярные обновления у координатного метода.
n_f	Значение в начале и после каждой внешней итерации, включая запись истории.
n_g	Начальный полный градиент и все последующие пересчёты; в координатном методе — после каждой координаты.
n_linear_solves	Решения линейной системы при шагах Ньютона.

Поскольку функция квадратична, Гессиан задан один раз. Проверка собственных значений, построение графиков и подготовка исходных данных не включены в перечисленные счётчики.

Число updates у разных методов также не означает одинаковую стоимость: скалярное и векторное обновления различаются.

Даже совпадение числа вычислений градиента не делает полную стоимость одинаковой. Наискорейший спуск дополнительно вычисляет произведение $H\mathbf{g}$, а Ньютон решает систему. При больших плотных матрицах решение системы существенно дороже отдельного векторного шага. В данной работе сравниваются диагностические показатели и геометрия поведения, а не производительность промышленной реализации.

11. Ожидаемые результаты общего примера

При $\varepsilon_g = 10^{-6}$, начальной точке $(0; 0)$ и постоянном шаге $1/4$ получены следующие результаты. Небольшие различия последних знаков возможны из-за арифметики с плавающей точкой.

Метод	Итерации / циклы	Обновления	Nf	Ng	Норма градиента
Градиентный	29	29	30	30	7.793e-07
Наискорейший	13	13	14	14	2.138e-07
Координатный	8	16	9	17	3.576e-07
Ньютон	1	1	2	2	0.000e+00

Все методы приближают точку $(11/7; 6/7)$ и значение $5/7 \approx 0{,}7142857143$. Для Ньютона выполнено одно решение линейной системы; для остальных методов — ни одного. Числа получены приложенным кодом, а точное решение — независимо, из системы стационарности.

У координатного метода восемь циклов означают шестнадцать перемещений по координатам. Поэтому сравнение «8 меньше 13» без учёта структуры итерации не является достаточным выводом о скорости. На рисунке 3 по горизонтали отложено число вычислений полного градиента; эта шкала полезнее простого номера цикла, но тоже не учитывает всю стоимость.

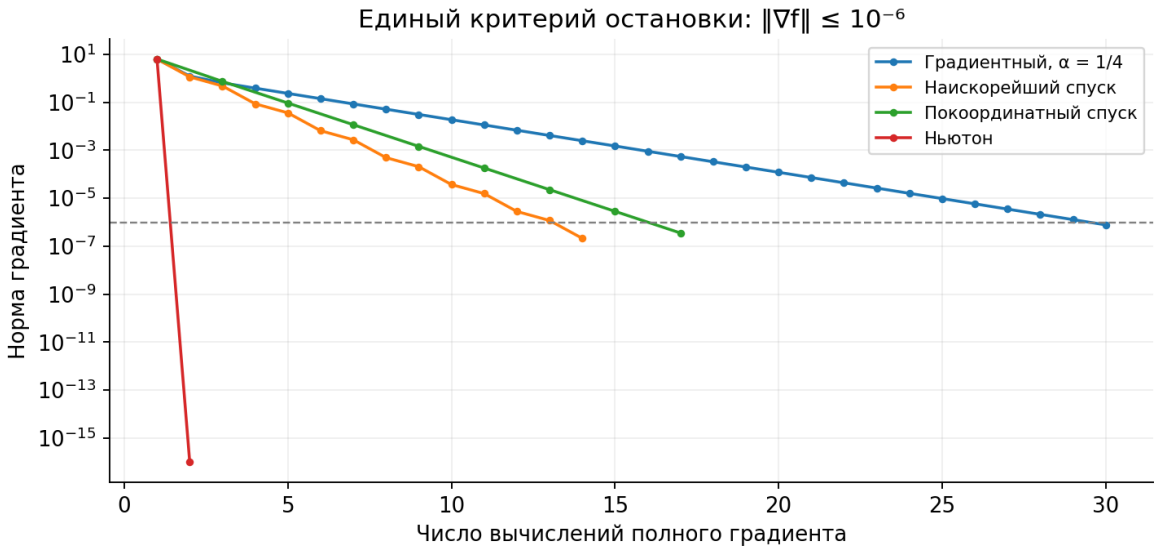


Рисунок 3 — Сопоставление методов по остаточной норме и числу вычислений градиента. Нижнее значение для Ньютона на логарифмической шкале ограничено уровнем 10^{-16} для отображения нуля.

12. Индивидуальные варианты

Для каждого варианта задаются собственные значения, угол поворота в градусах, центр и минимальное значение. Параметры согласованы с семейством квадратичных функций лабораторной работы 4, что позволяет использовать ранее исследованную геометрию.

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \qquad H = R \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} R^T.$$

$$\mathbf{h} = (h_1; h_2)^T, \qquad \mathbf{x}^0 = \mathbf{h} + R(1{,}5; 0{,}7)^T.$$

В вычислениях Python градусы переводятся в радианы функцией np.deg2rad. Начальная точка специально задана в повёрнутой системе, чтобы при изменении угла сохранялось её положение относительно главных осей эллипсов.

№	λ_1	λ_2	$\theta, ^\circ$	h_1	h_2	c
1	1	1	0	-1.5	1.0	-1
2	2	8	15	-1.3	0.9	0
3	3	48	30	-1.1	0.8	1
4	1	36	45	-0.9	0.7	2
5	2	128	60	-0.7	0.6	-1
6	3	300	75	-0.5	0.5	0
7	1	1	0	-0.3	0.4	1
8	2	8	15	-0.1	0.3	2
9	3	48	30	0.1	0.2	-1
10	1	36	45	0.3	0.1	0
11	2	128	60	0.5	0.0	1
12	3	300	75	0.7	-0.1	2
13	1	1	0	0.9	-0.2	-1
14	2	8	15	1.1	-0.3	0
15	3	48	30	1.3	-0.4	1
16	1	36	45	1.5	-0.5	2
17	2	128	60	1.7	-0.6	-1
18	3	300	75	1.9	-0.7	0

Для проверки раскрытия матрицы используйте формулы

$$H_{11} = \lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta,$$

$$H_{22} = \lambda_1 \sin^2\theta + \lambda_2 \cos^2\theta, \qquad H_{12} = H_{21} = (\lambda_1 - \lambda_2)\sin\theta\cos\theta.$$

Если обозначить $z_1 = x_1 - h_1$, $z_2 = x_2 - h_2$, то

$$f = c + \frac{1}{2} H_{11} z_1^2 + H_{12} z_1 z_2 + \frac{1}{2} H_{22} z_2^2.$$

Коэффициент смешанного члена равен H_{12}, а не $H_{12}/2$: в матричном произведении симметричные смешанные слагаемые встречаются дважды.

13. Порядок выполнения индивидуального задания

1. Выпишите параметры варианта, постройте R, H и начальную точку. Запишите функцию в координатной форме. Покажите положительную определённость, найдите μ, L, κ, точный минимум и минимальное значение.
2. Выведите градиент и Гессиан. Вычислите в начальной точке значение функции и норму градиента. В ручных расчётах сохраняйте не менее шести значащих цифр либо используйте точные выражения.
3. Выполните первый шаг градиентного метода с $\alpha = 1/L$, первый шаг наискорейшего спуска, один полный координатный цикл и один шаг Ньютона. Для каждого промежуточного результата вычислите функцию. Проверьте ортогональность двух градиентов наискорейшего спуска.
4. Запустите все методы при $\varepsilon_g = 10^{-4}$ и 10^{-6}, пределе 100000 внешних итераций и одинаковой начальной точке. Внесите в таблицу статус, точку, функцию, норму градиента и все счётчики.
5. Сравните найденную точку с аналитическим решением. Вычислите фактическую ошибку аргумента и верхнюю оценку $\| \mathbf{g} \|/\mu$. Проверьте, что оценка не меньше фактической ошибки с учётом округления.
6. Постройте траектории на линиях уровня и графики нормы градиента. Покажите промежуточные координатные перемещения. Объясните связь формы эллипсов и поведения методов.
7. Проведите дополнительный опыт: замените только угол на 0°, затем на 45°, сохраняя спектр, центр, константу и правило задания начальной точки. Используйте параметр --angle. Сравните результаты с исходным углом.
8. Сделайте содержательный вывод: какой метод оказался удобнее для данной задачи; какие расходы скрываются за одной итерацией; как поворот повлиял на координатный метод; почему результат для квадратичной модели нельзя автоматически переносить на любую функцию.

Подсказка для опыта с поворотом. При ортогональном повороте и согласованном начальном векторе постоянный градиентный метод и точный наискорейший спуск воспроизводят повёрнутые траектории с теми же нормами в точной арифметике. Ньютон по-прежнему решает квадратичную задачу одним шагом. Координатные направления привязаны к осям и при повороте ведут себя иначе. Если $\lambda_1 = \lambda_2$, эллипсы становятся окружностями и поворот не меняет функцию; отсутствие различий в таком варианте является ожидаемым результатом.

Пример команд для исследования шестого варианта:

```
python run_variant.py --variant 6 --eps 1e-4
python run_variant.py --variant 6 --eps 1e-6
python run_variant.py --variant 6 --angle 0 --eps 1e-6
python run_variant.py --variant 6 --angle 45 --eps 1e-6
```

Дополнительное задание повышенной сложности. Для фиксированной матрицы сравните шаги $1/L$ и $2/(L+\mu)$. Выведите спектральные коэффициенты сжатия и объясните различие. Затем для $f(x) = x^4 - x^2$ в точке 0,1 вручную проверьте, почему ньютоновское направление не является направлением спуска. Этот нелинейный пример не передаётся в класс Quadratic.

14. Требования к отчёту и критерии оценивания

В отчёт включаются название и цель работы, номер варианта, исходные параметры, аналитическое исследование, ручные первые шаги, описание кода, таблицы для двух точностей, рисунки, опыт с изменением угла, выводы и ответы на контрольные вопросы. Достаточно привести ключевые фрагменты программы с пояснениями; полный код можно приложить отдельным файлом.

Числа в таблицах следует печатать с достаточной точностью. Нулевое значение, полученное только после округления до нескольких знаков, не является доказательством выполнения критерия. Для нормы градиента удобно научное представление, например $7{,}79 \cdot 10^{-7}$.

Проверяемый результат	Баллы
Правильная постановка, производные, положительная определённость и точное решение	20
Четыре ручных расчёта с промежуточными проверками	25
Корректный запуск, статусы, единый критерий и сравнительные таблицы	25
Графики и объяснение влияния поворота и точности	20
Оформление выводов и ответы на защите	10
Итого	100

15. Типичные ошибки

Потеря множителя два. В выражении $x^2 + xy + 2y^2$ матрица Гессе равна $\begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$; её диагональ состоит из вторых производных, а не исходных коэффициентов квадратов.

Неверный шаг. Условие устойчивости проверяют по наибольшему собственному значению L, а не по следу матрицы или наименьшему собственному значению. Допустимый интервал открыт: граничное значение $2/L$ не даёт общей гарантии сходимости.

Устаревший градиент. В координатном методе после первого перемещения нужно обновить данные перед второй координатой. Иначе нарушается схема, выведенная в методичке.

Деление на ноль. Формула точного шага применяется после проверки нормы градиента. Нулевой градиент — основание остановиться, а не вычислять отношение нулей.

Несопоставимые счётчики. Полный координатный цикл, одно перемещение по оси и один ньютоновский шаг — разные операции. Вывод о качестве метода должен учитывать точность, статус, стоимость и характер задачи.

Обобщение без условий. Один шаг Ньютона достаточен здесь благодаря положительно определённой квадратичной модели. Для произвольной нелинейной функции это утверждение неверно.

16. Контрольные вопросы

1. Что является направлением спуска и как проверить это с помощью градиента?

2. Почему отрицательный градиент связан с наибольшей локальной скоростью уменьшения в евклидовой норме?
3. Как из спектра матрицы Гессе получается условие $0 < \alpha < 2/L$?
4. Чем постоянный градиентный шаг отличается от точного шага наискорейшего спуска?
5. Выведите формулу $\alpha_k = (\mathbf{g}_k^T \mathbf{g}_k)/(\mathbf{g}_k^T H \mathbf{g}_k)$.
6. Почему при точном линейном поиске соседние градиенты ортогональны?
7. В чём различие координатного перемещения и полного цикла?
8. Почему после изменения второй координаты первая частная производная может снова стать ненулевой?
9. Почему координатный метод чувствителен к повороту главных осей эллипсов?
10. Как строится ньютоновское направление и почему не нужно явно вычислять обратную матрицу?
11. При каких условиях Ньютон решает задачу одним шагом? Что может нарушиться для нелинейной функции?
12. Почему малая длина шага не является достаточным признаком найденного минимума?
13. Как оценить ошибку аргумента по норме градиента и μ?
14. Что означает завершение со статусом max_iter?
15. Какие вычисления не отражает счётчик числа итераций?

17. Литература

[1] Пантелеев А. В., Летова Т. А. Методы оптимизации в примерах и задачах: учебное пособие. 4-е изд., испр. СПб.: Лань, 2015. 512 с. Для данной работы: § 6.1, с. 161–165; § 6.2, с. 165–169; § 6.3–6.4, с. 170–178; § 7.1, с. 196–198.

Методические пояснения, общий пример, индивидуальные варианты, рисунки и программный код подготовлены для учебной работы. Программы проверены на всех 18 вариантах для четырёх методов при допуске 10^{-6}; дополнительно проверены первые ручные шаги, монотонность значений, оценка ошибки, ортогональность градиентов и обработка недопустимых параметров.