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

Дисциплина «Методы оптимизации»

ЛАБОРАТОРНАЯ РАБОТА 4

Методы многомерной оптимизации

Графическая интерпретация целевой функции. Классификация основных методов. Методические указания для образовательной программы 6B06103 «Математическое и компьютерное моделирование». Продолжительность — 5 академических часов.

1. Цель работы и порядок выполнения

Цель работы — научиться исследовать геометрию функции двух переменных, строить её поверхность и линии уровня, определять направления изменения функции и объяснять, какие сведения о функции используют разные методы оптимизации. Основное внимание уделяется пониманию задачи до запуска численного алгоритма. Здесь исследуется отдельный шаг; полный градиентный метод изучается в отдельной теме программы.

После выполнения работы студент должен уметь записать многомерную задачу минимизации; различать поверхность и линии уровня; объяснять геометрический смысл минимума, максимума и седловой точки; вычислять градиент и производную по направлению; выбирать и проверять длину шага на квадратичном примере; классифицировать методы по используемым производным; оформлять вычислительный эксперимент с проверкой результата.

Необходимы знания производных, векторов и матриц, базовые навыки Python. Используются NumPy и Matplotlib. Параметры вариантов находятся в variants.json.

Этап	Содержание	Время, мин
Подготовка	Изучить геометрическую интерпретацию и классификацию методов	35
Ручной расчёт	Разобрать общий пример, вычислить направления и длины шагов	40
Работа с кодом	Выполнить пример и построить графики своего варианта	60
Эксперимент	Сравнить направления и исследовать влияние кривизны	50
Отчёт и защита	Объяснить результаты, ответить на контрольные вопросы	40

Итого — 225 минут, то есть 5 академических часов по 45 минут. Преподаватель может распределить эти этапы между занятиями, сохранив общую трудоёмкость.

2. Методические сведения

2.1. Постановка задачи и геометрическая модель

Многомерная задача минимизации имеет вид

$$f(x) \to \min, \qquad x = (x_1, \dots, x_n)^{\mathsf{T}} \in D.$$

Здесь x — вектор управляемых переменных, f — целевая функция, D — допустимое множество. При $D = \mathbb{R}^n$ задача является безусловной. Если на переменные наложены

дополнительные условия, например $x_1 + x_2 \le 5$, задача относится к условной оптимизации. В данной работе рассматриваются безусловные задачи с двумя переменными.

Поверхность $z = f(x_1, x_2)$ показывает значение функции как высоту над плоскостью переменных. Линия уровня со значением ℓ представляет множество точек этой плоскости, в которых функция принимает одинаковое значение:

$$L_\ell = \{(x_1, x_2) : f(x_1, x_2) = \ell\}.$$

Поверхность удобна для общего представления формы функции, а линии уровня — для сравнения точек и направлений движения. В отличие от трёхмерной проекции, плоский рисунок не скрывает точки за поверхностью. На рисунке необходимо подписывать значения уровней и соблюдать одинаковый масштаб обеих координатных осей [1, § 1; 2, § 1.7].

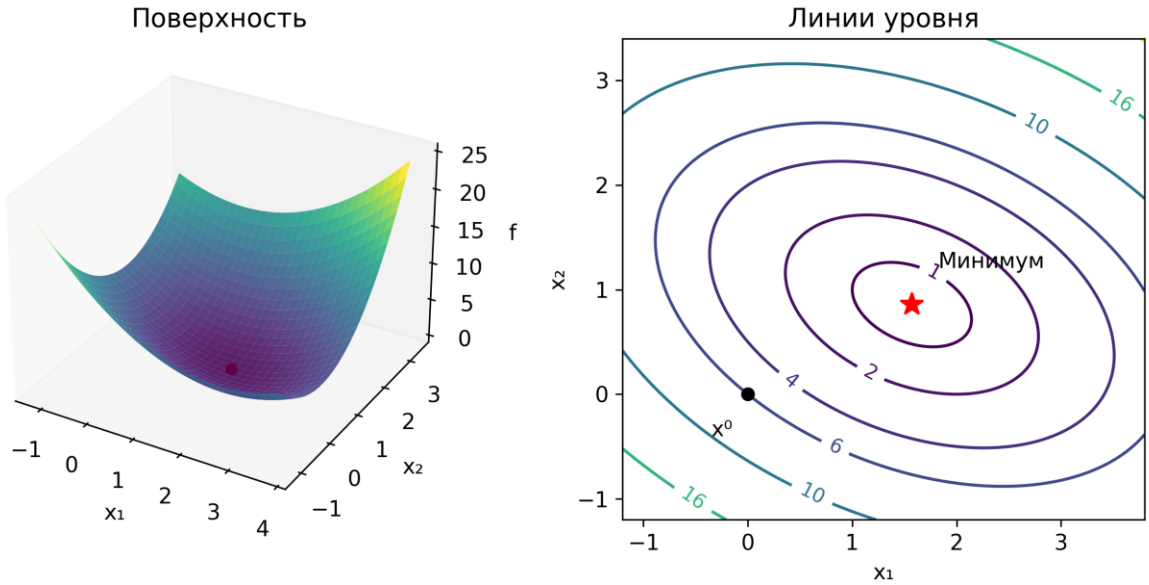


Рисунок 1 — Два способа изображения одной целевой функции. Красная отметка обозначает её минимум.

Окно построения графика, например $-1 \le x_1 \le 4$, само по себе не является ограничением исходной задачи. Оно определяет только видимый участок. Минимальная точка среди узлов сетки также не доказывает существование глобального минимума: сетка конечна, а задача задана на непрерывном множестве.

2.2. Минимум, максимум и седловая точка

Точка x^* является локальным минимумом, если в некоторой её окрестности все допустимые точки удовлетворяют неравенству $f(x) \ge f(x^*)$. Для глобального минимума это неравенство должно выполняться на всём допустимом множестве. В определении максимума знак неравенства меняется. Локальный минимум может совпадать с глобальным, но из локальной информации это следует не всегда.

Рассмотрим три функции:

$$f_1(x_1, x_2) = x_1^2 + x_2^2, \qquad f_2(x_1, x_2) = -x_1^2 - x_2^2, \qquad f_3(x_1, x_2) = x_1^2 - x_2^2.$$

В начале координат первая функция имеет глобальный минимум, вторая — глобальный максимум, а третья — седловую точку. У третьей функции вдоль оси x_1 значения возрастают от нуля, а вдоль оси x_2 убывают. Во всех трёх случаях градиент в начале координат равен нулю.

Поэтому равенство нулю первых производных само по себе не определяет вид стационарной точки.



Рисунок 2 — Геометрические различия стационарных точек. Круговые линии уровня встречаются и у минимума, и у максимума: необходимо учитывать значения уровней.

Проверка только двух координатных сечений тоже может оказаться недостаточной. Для функции $q(x_1, x_2) = x_1^2 + 4x_1x_2 + x_2^2$ оба сечения через начало координат, полученные при $x_1 = 0$ или $x_2 = 0$, имеют минимум в нуле. Однако вдоль прямой $x_2 = -x_1$ получается $q = -2x_1^2$. Следовательно, начало координат является седловой точкой. Этот пример показывает, зачем исследовать направления, отличные от координатных.

2.3. Градиент, линии уровня и направление спуска

Для дифференцируемой функции градиент — вектор частных производных:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)^{\mathsf{T}}.$$

Если вектор d задаёт направление, то изменение функции вдоль прямой описывается функцией одной переменной $\varphi(\alpha) = f(x + \alpha d)$. Её производная при $\alpha = 0$ равна скалярному произведению

$$\varphi'(0) = \nabla f(x)^{\mathsf{T}} d.$$

При отрицательном значении этого произведения направление является направлением спуска: для достаточно малого положительного α значение функции уменьшится. Положительное значение соответствует первоначальному возрастанию. При нулевом значении линейное приближение не позволяет сделать вывод; требуется дополнительное исследование.

Пусть гладкая кривая $x(t)$ проходит по линии уровня. Тогда $f(x(t)) = \ell$, и дифференцирование даёт $\nabla f(x)^{\mathsf{T}} x'(t) = 0$. Следовательно, в регулярной точке, где градиент не равен нулю, он перпендикулярен касательной к линии уровня. Антиградиент $-\nabla f$ направлен в сторону наиболее быстрого первоначального уменьшения функции в евклидовой метрике.

Последнее утверждение относится к сравнению направлений единичной длины. Из неравенства Коши — Буняковского при $\| d \| = 1$ следует

$$\nabla f(x)^{\mathsf{T}} d \geq -\| \nabla f(x) \|.$$

Равенство достигается при $d = -\nabla f(x)/\| \nabla f(x) \|$. Оно характеризует локальную скорость изменения, но не гарантирует, что большой шаг в этом направлении будет удачным или сразу приведёт к глобальному минимуму.

2.4. Направление и длина шага

Один переход из точки $x^{(0)}$ в новую точку записывается так:

$$x^{(1)} = x^{(0)} + \alpha d, \qquad \alpha \geq 0.$$

В этой формуле нужно принять два решения: выбрать направление d и длину шага α. Если d единичный, параметр α равен евклидовой длине перемещения. Если направление не нормировано, длина перемещения равна $\alpha \| d \|$. Поэтому значения параметра α для по-разному нормированных направлений нельзя непосредственно сравнивать.

Для квадратичной функции с симметричной положительно определённой матрицей H изменение вдоль направления вычисляется точно:

$$f(x^{(0)} + \alpha d) = f(x^{(0)}) + \alpha g^{\mathsf{T}} d + \frac{\alpha^2}{2} d^{\mathsf{T}} H d, \qquad g = \nabla f(x^{(0)}).$$

Минимум на луче $\alpha \geq 0$ достигается при

$$\alpha_* = \max\left(0, -\frac{g^{\mathsf{T}} d}{d^{\mathsf{T}} H d}\right).$$

Формула получена приравниванием производной по α к нулю с последующим учётом ограничения $\alpha \geq 0$. Она применима к рассматриваемой квадратичной модели. Для произвольной нелинейной функции поиск шага обычно требует отдельного одномерного метода. При направлении подъёма минимум на луче данной выпуклой квадратичной функции достигается при $\alpha = 0$, то есть переход не выполняется.

2.5. Классификация методов оптимизации

Выбор метода зависит от того, какие сведения о функции доступны и насколько дорого их вычислять [1, § 4]. Основная классификация по порядку используемых производных приведена в таблице.

Класс	Используемая информация	Примеры и особенности
Методы нулевого порядка	Значения функции в пробных точках	Поиск Хука — Дживса, метод Нелдера — Мида. Производные не требуются; качество результата зависит от структуры задачи и правил поиска.
Методы первого порядка	Градиент, обычно вместе со значением функции	Градиентный метод, методы сопряжённых градиентов. Направление выбирается с учётом локальных скоростей изменения.
Методы второго порядка	Градиент и матрица вторых производных	Метод Ньютона. Дополнительно учитывается кривизна; вычисление и обработка матрицы могут быть дорогими.

Класс	Используемая информация	Примеры и особенности
Квазиньютоновские методы	Значения функции и градиента; обновляемое приближение матрицы кривизны	Например, BFGS. Аналитическое вычисление матрицы вторых производных не требуется.

Первые три строки образуют классификацию по порядку производных. Квазиньютоновские методы выделены отдельно, чтобы показать отличие использования точной матрицы вторых производных от построения её приближения. Численное приближение градиента по значениям функции также требует учитывать дополнительные вычисления и ошибку разностной формулы.

Методы классифицируют и по другим признакам: для условной или безусловной оптимизации; локального или глобального поиска; детерминированные или стохастические. Эти признаки независимы от порядка производных. Например, случайный выбор пробных точек сам по себе не доказывает нахождение глобального минимума.

Название «симплекс» встречается в разных алгоритмах. В методе Нелдера — Мида симплекс — это набор из $n+1$ вершин в пространстве переменных, который перемещается и деформируется при поиске. В симплекс-методе линейного программирования выполняются переходы между базисными планами. Совпадение слова в названиях не означает совпадения алгоритмов.

3. Подробный пример с ручным расчётом

3.1. Исходная функция и точное решение

Рассмотрим задачу

$$f(x_1, x_2) = x_1^2 + x_1 x_2 + 2x_2^2 - 4x_1 - 5x_2 + 6 \to \min.$$

Её градиент и матрица вторых производных имеют вид

$$\nabla f(x) = \begin{pmatrix} 2x_1 + x_2 - 4 \\ x_1 + 4x_2 - 5 \end{pmatrix}, \qquad H = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

Решение системы $2x_1 + x_2 = 4, x_1 + 4x_2 = 5$ даёт точку $h = (11/7, 6/7)^{\mathsf{T}}$. Подстановка в целевую функцию приводит к значению $f(h) = 5/7$. Для проверки глобального характера минимума введём $z = x - h$. Тогда

$$f(x) = \frac{5}{7} + z_1^2 + z_1 z_2 + 2z_2^2 = \frac{5}{7} + \left(z_1 + \frac{z_2}{2}\right)^2 + \frac{7}{4} z_2^2.$$

Оба квадратичных слагаемых неотрицательны и одновременно обращаются в нуль только при $z = 0$. Поэтому найденный минимум является единственным и глобальным. Такое представление служит независимой проверкой вычислительной программы, а не заменяет исследование направлений.

Линии уровня при $\ell > 5/7$ являются эллипсами с центром h. Из-за смешанного слагаемого $x_1 x_2$ их оси повёрнуты относительно координатных осей. При $\ell = 5/7$ множество уровня состоит из одной точки, а при $\ell < 5/7$ оно пусто.

3.2. Один шаг по антиградиенту

Возьмём начальную точку $x^{(0)} = (0,0)^{\mathsf{T}}$. Имеем $f(x^{(0)}) = 6$ и $g = (-4,-5)^{\mathsf{T}}$. Выберем ненормированный антиградиент $d = (4,5)^{\mathsf{T}}$. Тогда

$$g^{\mathsf{T}}d = -41, \qquad d^{\mathsf{T}}Hd = 172.$$

Функция на соответствующем луче равна

$$\varphi(\alpha) = 6 - 41\alpha + 86\alpha^2.$$

Её производная $\varphi'(\alpha) = -41 + 172\alpha$ обращается в нуль при $\alpha_* = 41/172 \approx 0{,}238372$. Поэтому новая точка и значение функции равны

$$x^{(1)} = \left(\frac{41}{43}, \frac{205}{172}\right)^{\mathsf{T}} \approx (0{,}953488;\, 1{,}191860)^{\mathsf{T}}, \qquad f(x^{(1)}) = \frac{383}{344} \approx 1{,}113372.$$

Значение уменьшилось с 6 до приблизительно 1,113372, но остаётся больше $5/7 \approx 0{,}714286$. Минимизация вдоль одного направления не равносильна минимизации по всей плоскости. В новой точке производная вдоль выбранного направления равна нулю; сам градиент в общем случае не равен нулю.

3.3. Почему слишком большой шаг не подходит

Уменьшение функции относительно начальной точки требует $-41\alpha + 86\alpha^2 < 0$. Для положительных шагов это условие выполняется при $0 < \alpha < 41/86 \approx 0{,}476744$.

Параметр шага α	Значение $f(x^{(0)} + \alpha d)$	Результат относительно $f(x^{(0)}) = 6$
0,10	2,760000	Уменьшение
0,25	1,125000	Уменьшение
0,50	7,000000	Увеличение

Таким образом, правильный выбор направления не отменяет необходимости подбирать длину шага. При $\alpha = 0{,}5$ движение по антиградиенту проходит слишком далеко по выбранному лучу.

3.4. Сравнение с координатными направлениями

Из той же начальной точки рассмотрим $d_1 = (1,0)^{\mathsf{T}}$ и $d_2 = (0,1)^{\mathsf{T}}$. В первом случае $\varphi_1(\alpha) = 6 - 4\alpha + \alpha^2$, а во втором $\varphi_2(\alpha) = 6 - 5\alpha + 2\alpha^2$. Результаты одномерной минимизации приведены ниже.

Направление	Оптимальный параметр α_*	Полученная точка	Значение функции
$(4,5)^{\mathsf{T}}$	$41/172$	$(41/43, 205/172)^{\mathsf{T}}$	$383/344 \approx 1{,}113372$
$(1,0)^{\mathsf{T}}$	2	$(2,0)^{\mathsf{T}}$	2
$(0,1)^{\mathsf{T}}$	$5/4$	$(0{,}5/4)^{\mathsf{T}}$	$23/8 = 2{,}875$

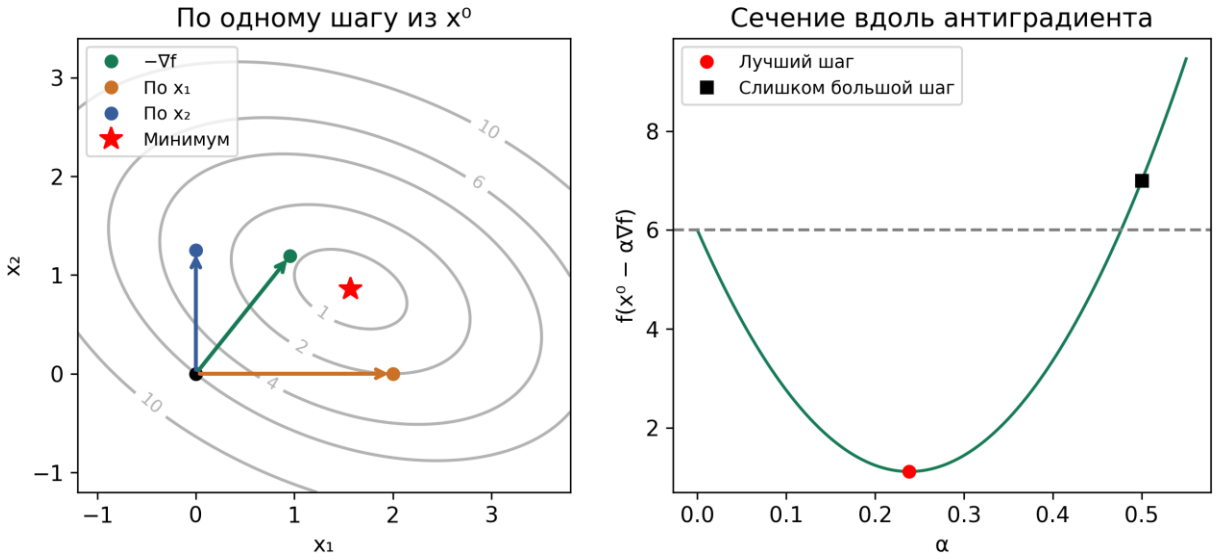


Рисунок 3 — Отдельные шаги из общей начальной точки и сечение целевой функции вдоль антиградиента.

В этом примере после точного выбора шага антиградиентное направление даёт наименьшее из трёх полученных значений. Этот результат относится к конкретной функции и точке; он не устанавливает универсальное превосходство одного метода. На рисунке и в ручной таблице антиградиент не нормирован. Программа индивидуальных вариантов нормирует направления: конечная точка на том же луче сохраняется, а параметр шага меняется.

Рассмотрим также $d = (5, -4)^{\mathsf{T}}$. Имеем $g^{\mathsf{T}}d = 0$: направление касается линии уровня в начальной точке. При этом $d^{\mathsf{T}}Hd > 0$, поэтому любой ненулевой шаг вдоль этого направления увеличивает рассматриваемую функцию. Нулевая производная по направлению не означает постоянства функции на всей прямой.

4. Кривизна и форма линий уровня

Для индивидуальных заданий используется семейство функций

$$f(x) = c + \frac{1}{2}(x-h)^{\mathsf{T}}H(x-h), \qquad H = R\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}R^{\mathsf{T}}, \qquad 0 < \lambda_1 \le \lambda_2,$$

где h — центр, c — значение минимума, а матрица поворота задаётся углом θ:

$$R = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}.$$

В повёрнутых координатах $u = R^{\mathsf{T}}(x-h)$ функция имеет вид $f = c + (\lambda_1 u_1^2 + \lambda_2 u_2^2)/2$. При уровне $\ell = c + \rho$, $\rho > 0$, полуоси эллипса равны

$$a = \sqrt{\frac{2\rho}{\lambda_1}}, \qquad b = \sqrt{\frac{2\rho}{\lambda_2}}.$$

Отношение $\kappa = \lambda_2/\lambda_1$ является спектральным числом обусловленности положительно определённой матрицы H. Отношение полуосей равно $a/b = \sqrt{\kappa}$. При одинаковой кривизне уровни круговые. При большом различии кривизн появляются узкие вытянутые эллипсы; для

итерационного поиска такая геометрия может затруднить подбор шагов и направлений. По одному рисунку нельзя определить точное число итераций конкретного метода.

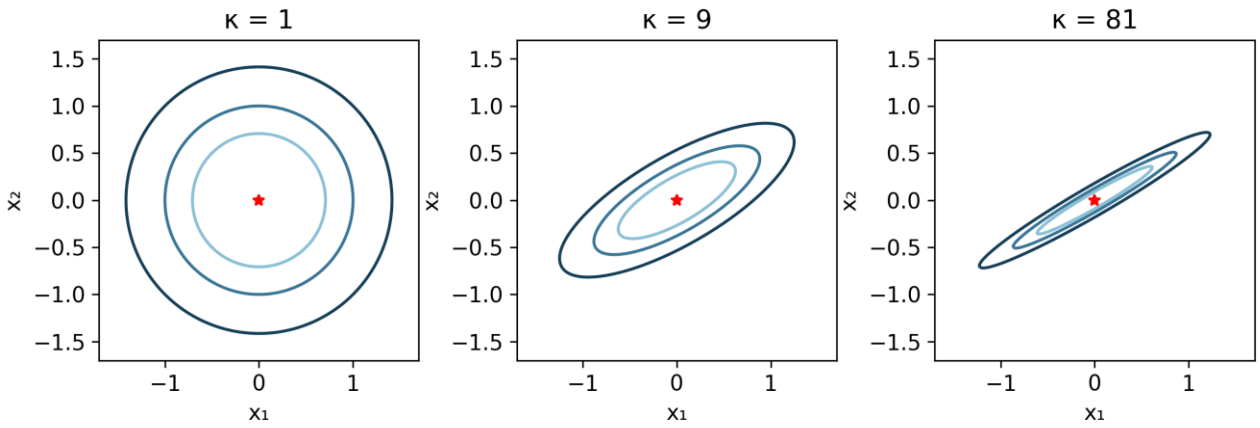


Рисунок 4 — Изменение формы линий уровня при $\kappa = 1, 9$ и 81. На всех панелях используется одинаковый масштаб координат.

Сдвиг h меняет расположение центра, добавление c меняет значения уровней, а угол θ меняет ориентацию эллипсов. Поворот не меняет собственных значений H. При отображении графиков с разными масштабами осей истинное отношение полуосей визуально искажается; для линий уровня в программе используется set_aspect('equal').

5. Программная реализация и проверка

5.1. Состав файлов и запуск

В архиве к работе находятся файлы geometry.py, run_geometry.py и variants.json. Первый содержит функции модели и вычисления отдельного шага, второй выполняет выбранный вариант и строит иллюстрации, третий хранит параметры 18 вариантов. Файл simplex_tableau.py относится к лекции 4 и для выполнения данной лабораторной работы не требуется.

Распакуйте архив в отдельную папку, откройте терминал в этой папке и выполните команды:

```
python -m pip install numpy matplotlib
python geometry.py
python run_geometry.py 1
```

В последней команде замените 1 номером своего варианта. Команда python geometry.py выводит результаты общего ручного примера. Команда python run_geometry.py 1 создаёт папку geometry_variant_01 с исходными параметрами, таблицей направлений и рисунками. Повторный запуск того же варианта обновляет результаты в его папке. При настроенной среде с установленными библиотеками повторная установка не требуется.

5.2. Основные функции модели

В следующем фрагменте показаны функции из файла geometry.py. Массив H хранит матрицу кривизны, h — центр, a constant — значение функции в центре. Оператор @ выполняет матричное умножение или скалярное произведение в зависимости от размеров массивов.

```
import numpy as np

def model(lam1, lam2, angle_deg, center, constant=0.0):
    if not 0 < lam1 <= lam2:
```

```
        raise ValueError("Требуется 0 < lam1 <= lam2")
    phi = np.deg2rad(angle_deg)
    R = np.array([[np.cos(phi), -np.sin(phi)],
                  [np.sin(phi), np.cos(phi)]])
    H = R @ np.diag([lam1, lam2]) @ R.T
    h = np.asarray(center, dtype=float)
    if h.shape != (2,):
        raise ValueError("Центр должен иметь две координаты")
    return H, h, float(constant), R


def value(x, H, h, constant=0.0):
    z = np.asarray(x, dtype=float) - h
    return float(constant + 0.5 * z @ H @ z)


def gradient(x, H, h):
    return H @ (np.asarray(x, dtype=float) - h)
```

Функция value вычисляет значение в одной точке. В файле дополнительно определена grid_values, которая обрабатывает координатную сетку для построения поверхности и линий уровня. Аналитический градиент позволяет проверить условие направления спуска без разностного приближения.

5.3. Выбор шага по направлению

```
def best_ray_step(x, d, H, h, constant=0.0):
    """Минимизация квадратичной функции на луче x+alpha*d, alpha>=0."""
    x, d = np.asarray(x, dtype=float), np.asarray(d, dtype=float)
    if np.linalg.norm(d) == 0:
        raise ValueError("Направление не должно быть нулевым")
    g = gradient(x, H, h)
    curvature = float(d @ H @ d)
    if curvature <= 0:
        raise ValueError("Для этого примера нужна положительная кривизна")
    slope = float(g @ d)
    alpha = max(0.0, -slope / curvature)
    point = x + alpha*d
    return dict(slope=slope, curvature=curvature, alpha=alpha,
                point=point, value=value(point, H, h, constant))
```

Здесь slope — значение $g^{\mathrm{T}}d$, curvature — значение $d^{\mathrm{T}}Hd$, а point — новая точка. Проверка положительной кривизны соответствует предположениям принятой модели. Полученная формула не является универсальным решателем произвольных задач оптимизации.

В общем примере программа должна получить $x^* = (11/7,6/7)^{\mathrm{T}}$, $f^* = 5/7$, градиент в начале $(-4,-5)^{\mathrm{T}}$ и результаты из таблицы раздела 3.4. Направление $d = g$ даёт положительное скалярное произведение $g^{\mathrm{T}}d$; в этом случае возвращается $\alpha = 0$ и сохраняется начальная точка.

5.4. Как читать результаты индивидуального варианта

Файл parameters.json содержит матрицу H, минимум h, значение c, начальную точку, градиент, собственные значения и число обусловленности. Файл directions.csv содержит три направления и результаты отдельных шагов. Столбцы d1, d2 — компоненты направления; g_dot_d — производная вдоль него; alpha — длина шага для единичного направления; x1, x2 — координаты полученной точки; f — значение функции.

Программа сравнивает единичный антиградиент и два координатных направления. Знак каждого координатного направления выбирается так, чтобы соответствующая компонента градиента не увеличивала функцию в первом приближении. Если эта компонента равна нулю,

для соответствующей оси используется положительное направление; формула шага возвращает нуль.

Файл surface_and_levels.png показывает поверхность, линии уровня и три независимых перехода из одной начальной точки. Отрезки на этом рисунке не образуют последовательность итераций. Файл direction_sections.png показывает три функции $f(x^{(0)} + \alpha d)$ при единичных направлениях. На каждой кривой отмечен минимум на луче.

Проверьте программу по четырём независимым признакам: $f(h) = c$; $\nabla f(h) = 0$; ни один из выбранных оптимальных шагов не увеличивает функцию; при $\alpha_* > 0$ выполняется $\nabla f(x^{(1)})^{\mathsf{T}} d \approx 0$. Последняя проверка относится только к выбранному направлению. Для чисел с плавающей точкой используйте небольшой допуск, например 10^{-8} для этих учебных данных.

6. Индивидуальные задания

6.1. Параметры вариантов

В таблице угол θ указан в градусах, $h = (h_1, h_2)^{\mathsf{T}}$. Функция строится по формулам раздела 4. Начальная точка для всех вариантов задаётся общим правилом

$$x^{(0)} = h + R\begin{pmatrix}1{,}5\\0{,}7\end{pmatrix}.$$

Вариант	λ_1	λ_2	$\theta, ^\circ$	h_1	h_2	c
1	1	1	0	-1.5	1	-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	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

Такой выбор начальной точки задаёт одинаковое смещение в собственных координатах квадратичной формы. Поэтому можно осмысленно сравнивать влияние различий кривизны и поворота. При $\lambda_1 = \lambda_2$ поворот не меняет саму функцию относительно её центра, но меняет заданную этим правилом начальную точку.

6.2. Обязательные задания

1. Запишите целевую функцию своего варианта в матричной и развёрнутой координатной форме. Укажите допустимое множество, центр h, значение c, матрицу H и начальную точку. Переведите угол в радианы при ручных вычислениях тригонометрических функций.
2. Выведите формулу градиента. Обоснуйте, почему h — единственный глобальный минимум, используя положительность λ_1 и λ_2. Вычислите $f(x^{(0)})$ и $\nabla f(x^{(0)})$.
3. Выберите уровень $\ell = c + \lambda_1$. Вычислите полуоси эллипса и отношение их длин. Найдите две точки этого уровня по правилу $h + R(a, 0)^\top$ и $h + R(0, b)^\top$; проверьте значения функции в них.
4. Постройте поверхность и не менее пяти линий уровня. Подпишите оси, значения уровней, центр и начальную точку. Объясните, почему выбранное окно изображения не является ограничением задачи.
5. Запишите три единичных направления, используемых программой. Для каждого вручную вычислите $g^\top d$, $d^\top H d$, α_* , новую точку и значение функции. Достаточно привести промежуточные вычисления с шестью значащими цифрами, сохранив полную точность в программе.
6. Сопоставьте ручные вычисления с directions.csv. Если величины различаются, сначала проверьте нормировку направления, единицы угла и порядок матричного умножения. Проведите четыре проверки из раздела 5.4.
7. Для единичного антиградиента сравните шаги $0{,}5\alpha_*$, α_*, $2\alpha_*$ и $2{,}5\alpha_*$. Объясните результаты по квадратичной формуле: при $2\alpha_*$ функция возвращается к начальному значению, а при большем шаге возрастает. Это утверждение здесь относится к квадратичной модели и ненулевому направлению спуска.
8. Сохранив λ_1, h, c и угол, дополнительно исследуйте отношения кривизн $\kappa = 1, 9$ и 81. Для этого используйте функцию model с $\lambda_2 = \kappa\lambda_1$. Постройте линии одинаковых уровней в одном масштабе и объясните изменение формы. В отчёте отделите эти дополнительные модели от исходного варианта.
9. Составьте небольшую сравнительную таблицу методов нулевого, первого и второго порядков: необходимые данные, характер направления поиска, возможные вычислительные затраты. Объясните место квазиньютоновского подхода.
10. Сделайте вывод о том, какие сведения о функции даёт геометрическое исследование и почему выбор направления необходимо рассматривать совместно с выбором шага. Не распространяйте результат трёх отдельных переходов на скорость сходимости полных итерационных методов.

6.3. Дополнительные задания

Исследуйте функцию $q(x_1, x_2) = x_1^2 + 4x_1x_2 + x_2^2$: постройте координатные сечения и сечение $x_2 = -x_1$ на одном рисунке. Объясните, почему первые два сечения не обнаруживают седловой характер начала координат.

Для своего варианта возьмите направление, перпендикулярное градиенту в начальной точке. Проверьте равенство $g^{\mathsf{T}}d = 0$ и исследуйте изменение функции при ненулевых шагах. Сопоставьте линейное приближение с точной квадратичной формулой.

7. Требования к отчёту и оценивание

Отчёт должен содержать номер варианта и постановку задачи; развёрнутую функцию и градиент; обоснование точного минимума; расчёт полуосей выбранного уровня; ручное сравнение трёх направлений; таблицу программных результатов; рисунки с подписями и одинаковым масштабом осей; исследование длины шага и кривизны; вывод. Полный листинг программы можно вынести в приложение. Скриншот запуска без объяснения вычислений не заменяет аналитическую часть.

Для воспроизводимости укажите команду запуска, используемые файлы, версии Python и библиотек из своей среды. Подпишите каждый рисунок и приведите в тексте объяснение того, что он показывает. Округляйте числа только при оформлении таблиц, сохраняя исходную точность вычислений.

Критерий	Баллы
Правильная постановка, функция и исходные данные	10
Градиент, обоснование минимума, полуоси уровней	20
Ручной расчёт и проверка трёх направлений	20
Работоспособный код и независимые проверки	20
Графики и исследование шага и кривизны	20
Выводы, оформление и защита	10
Итого	100

8. Контрольные вопросы

1. Чем поверхность функции двух переменных отличается от её линии уровня?
2. Почему одинаковая форма линий уровня ещё не позволяет отличить минимум от максимума?
3. В чём различие локального и глобального минимума?
4. Почему равенство градиента нулю не гарантирует минимум?
5. Может ли минимум в двух координатных сечениях оказаться седловой точкой на плоскости?
6. Как вычисляется производная функции вдоль направления d?
7. Какое условие гарантирует уменьшение функции при достаточно малом положительном шаге?
8. В каком смысле антиградиент задаёт наиболее быстрое уменьшение функции?
9. Почему при сравнении первоначальных скоростей изменения нормируют направления?
10. Как вывести формулу оптимального шага на луче для положительно определённой квадратичной функции?

11. Почему после точного одномерного поиска градиент в новой точке может не равняться нулю?
12. Как собственные значения матрицы H связаны с полуосями линий уровня?
13. Какие свойства функции меняются при повороте, сдвиге центра и добавлении константы?
14. Какие данные используют методы нулевого, первого и второго порядков?
15. В чём различие симплекса метода Нелдера — Мида и базисных планов линейного программирования?
16. Почему минимум на конечной сетке и три удачных шага не доказывают глобальную сходимость метода?

9. Литература

1. Пантелеев А. В., Летова Т. А. Методы оптимизации в примерах и задачах: учебное пособие. 4-е изд., испр. — Санкт-Петербург: Лань, 2015. — 512 с. § 1: постановка задачи и геометрическая интерпретация; § 4: общие принципы численного поиска и классификация методов.
2. Kochenderfer M. J., Wheeler T. A. Algorithms for Optimization. — Cambridge, MA: MIT Press, 2019. § 1.7: визуализация; гл. 2: производные; гл. 5–7: методы, использующие различную информацию о функции.

Формулировки примеров, индивидуальные варианты и программные фрагменты подготовлены для этой работы. Библиографические ссылки указывают на источники теоретической основы.