

Київський національний університет імені Тараса
Шевченка

Факультет комп'ютерних наук та кібернетики

Лабораторна робота №1
Варіант 9

Студент: Кіщук Ярослав Ярославович

Київ - 2025

Зміст

| | |
|--|-----------|
| 1 Постановка задачі | 2 |
| 2 Теоретичні відомості | 2 |
| 2.1 Методи спуску та вибір напряму | 2 |
| 2.2 Backtracking із умовою Арміхо | 3 |
| 2.3 Правило Голдштейна | 3 |
| 2.4 Градієнтний спуск | 3 |
| 2.5 Класичне поділення кроку | 3 |
| 2.6 Правило Голдстейна | 3 |
| 3 Програмна реалізація | 4 |
| 4 Обчислювальні експерименти | 6 |
| 4.1 Налаштування | 6 |
| 4.2 Перерізи поверхні рівня | 6 |
| 4.3 Числові результати | 7 |
| 4.4 Візуальний аналіз траєкторій | 8 |
| 5 Висновки | 16 |

1. Постановка задачі

Розглядаємо квадратичну функцію варіанта 9

$$f(x_1, x_2, x_3) = 200x_1^2 + 5x_2^2 + 144x_3^2 - 24x_1x_2 - 48x_1x_3 + 24x_2x_3 + 5, \quad (1)$$

що може бути записана у матричному вигляді $f(x) = x^\top Qx + 5$, де

$$Q = \begin{bmatrix} 200 & -12 & -24 \\ -12 & 5 & 12 \\ -24 & 12 & 144 \end{bmatrix}, \quad x = (x_1, x_2, x_3)^\top.$$

Постановка завдання містить дві частини:

1. Побудувати графічні уявлення $f(x_1, x_2, x_3)$: перерізи при фіксованих змінних та ізоповерхні.
2. Розв'язати задачу мінімізації $\min_{x \in \mathbb{R}^3} f(x)$ градієнтним спуском, застосовуючи **два** варіанти лінійного пошуку: (i) поділ кроку (критерій простого зменшення), (ii) правило Голдстейна; виконати порівняння збіжності.

Градієнт. Градієнт має вигляд

$$\nabla f(x) = 2Qx = \begin{bmatrix} 400x_1 - 24x_2 - 48x_3 \\ 10x_2 - 24x_1 + 24x_3 \\ 288x_3 - 48x_1 + 24x_2 \end{bmatrix}.$$

Глобальний мінімум. Матриця Q симетрична і додатно визначена, тому функція є строго опуклою й має єдиний мінімум у точці $x^* = \mathbf{0}$, де $f(x^*) = 5$ (див., напр., [1]: розд. 3 «Convex functions» — критерій через гессіан $\nabla^2 f \succ 0$ та приклади 3.2 і 3.22 для квадратичних функцій; також розд. 4.4 про квадратичні задачі та єдиність мінімізатора за $Q \succ 0$).

2. Теоретичні відомості

2.1. Методи спуску та вибір напряму

Загальна форма одного кроку методу спуску [1, §9.2]:

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}, \quad t^{(k)} > 0, \quad (2)$$

де $\Delta x^{(k)}$ — обраний напрямок. Для коректності спуску вимагаємо

$$\nabla f(x^{(k)})^\top \Delta x^{(k)} < 0, \quad (3)$$

тобто напрямок утворює гострий кут з антиградієнтом (*descent direction*). Далі виконуємо *line search* для вибору кроку $t^{(k)}$ вздовж променя $\{x^{(k)} + t \Delta x^{(k)} \mid t \geq 0\}$, після чого оновлюємо точку; див. також схему *General descent method* у [1, Алгоритм 9.1].

2.2. Backtracking із умовою Арміхо

Класичний backtracking [1, Алгоритм 9.2] перевіряє умову Арміхо

$$f(x + t \Delta x) \leq f(x) + \alpha t \nabla f(x)^\top \Delta x, \quad \alpha \in (0, 0.5), \beta \in (0, 1), \quad (4)$$

починаючи з $t \leftarrow 1$ та звужуючи крок $t \leftarrow \beta t$, доки нерівність не виконається. Геометрично це означає, що графік $f(x + t\Delta x)$ лежить нижче за лінійну апроксимацію з нахилом, помноженим на α (див. ілюстрацію до [1, Алг. 9.2]). У нашій реалізації ми також використовуємо спрощений критерій простого зменшення $f(x + t\Delta x) < f(x)$, який відповідає підходу “дроблення кроку” з матеріалах *Методи оптимізації. Частина 2 (2020)* [3].

2.3. Правило Голдштейна

Правило Голдштейна [2, еq. (3.11)] задає двосторонню умову для $c \in (0, \frac{1}{2})$:

$$f(x) + (1 - c)t \nabla f(x)^\top p \leq f(x + tp) \leq f(x) + ct \nabla f(x)^\top p. \quad (5)$$

Ліва межа гарантує достатнє зменшення (не надто малий приріст), а права — запобігає надто малим крокам. У порівнянні з Арміхо правило Голдштейна краще керує масштабом кроку, але може виключати деякі мініуми лінійного пошуку; теорії збіжності обох правил подібні.

2.4. Градієнтний спуск

Метод градієнтного спуску [1, §9.3] обирає $\Delta x = -\nabla f(x)$, тобто

$$x_{k+1} = x_k - t_k \nabla f(x_k), \quad (6)$$

де t_k визначається одним з описаних лінійних пошуків. Типовий критерій зупинки — $\|\nabla f(x_k)\|_2 \leq \eta$ для малого $\eta > 0$.

2.5. Класичне поділення кроку

Практична реалізація лінійного пошуку використовує *backtracking*: починаємо з t_0 та зменшуємо $t \leftarrow \beta t$ (типово $\beta \approx 0.5$), поки не виконається умова прийнятності [3].

2.6. Правило Голдстейна

Використовуємо також правило Голдстейна [2], яке задає двосторонню умову на прийнятність кроку для $c \in (0, 0.5)$:

$$f(x_k) + (1 - c)t \nabla f(x_k)^\top p_k \leq f(x_k + tp_k) \leq f(x_k) + ct \nabla f(x_k)^\top p_k.$$

Нижня межа гарантує достатнє зменшення, а верхня — уникає надто малих кроків.

3. Програмна реалізація

Архітектура пакета

Реалізацію оформлено як пакет Python `lab_module` з такими модулями:

- **objectives.py**: інтерфейси (Protocol) та квадратична ціль `QuadraticObjective`; фабрика `build_variant9_objective()` створює завдання варіанта 9.
- **line_search.py**: умови прийнятності кроку `split_step_cond` (просте зменшення) і `goldstein_cond`; універсальний `backtracking back_tracking(...)`.
- **optimizers.py**: градієнтний спуск `gd_back_tracking(...)` з поверненням повної історії.
- **visualization.py**: візуалізації `surface_for_slice`, `show_isosurface_with_path`, `plot_metrics`, `save_video_xyz`.

Усі вектори передаються як `numpy.ndarray` форми $(n,)$ (у нашій роботі $n = 3$), значення функцій — як `float`.

Модуль objectives.py

- **Інтерфейс.** Оголошено Protocol `DifferentiableObjective` з методами `value(x): float`, `gradient(x): np.ndarray` та властивістю `dimension`.
- **Квадратична ціль.** `QuadraticObjective(Q, b, c)` реалізує $f(x) = x^\top Qx + b^\top x + c$ і $\nabla f(x) = 2Qx + b$. Для варіанта 9 $b=0$, $c=5$.
- **Фабрика.** `build_variant9_objective()` повертає готовий об'єкт із матрицею Q з (1).

Модуль line_search.py

Умови прийнятності.

- `split_step_cond(f, x, p, t, f_x, grad_f, c)`: перевіряє лише $f(x + tp) < f(x)$ (просте зменшення), без додаткових констант типу Арміхो.
- `goldstein_cond(..., c)`: перевіряє нерівності Гольдштейна для $c \in (0, 0.5)$:

$$f(x) + (1 - c)t \nabla f(x)^\top p \leq f(x + tp) \leq f(x) + ct \nabla f(x)^\top p.$$

Backtracking. Функція `back_tracking(f, grad_f, x, p, cond, t0, beta, c)`:

- Автоматично гарантує напрям спуску: якщо $\nabla f(x)^\top p \geq 0$, замінює $p \leftarrow -\nabla f(x)$; за нульового градієнта повертає крок 0.
- Для `goldstein_cond` виконує двохфазний пошук: спочатку розширення кроку (t/β) , поки не виконається ліва межа, далі звуження $(t^*\beta)$, поки не виконається права межа.
- Для інших умов — стандартне звуження $t \leftarrow \beta t$ до першого виконання умови.

Модуль optimizers.py

Градієнтний спуск. Функція `gd_back_tracking(f, grad_f, x0, cond, max_iters, tol, t0, beta, c)` реалізує ітерації $x_{k+1} = x_k - t_k \nabla f(x_k)$, де t_k обирається через `back_tracking`. Повертає пару $(x_{\min}, \text{history})$, де

- `history['x']`: список точок x_k ;
- `history['f']`: значення $f(x_k)$;
- `history['t']`: прийняті кроки t_k ;
- `history['grad_norm']`: $\|\nabla f(x_k)\|_2$.

Збіжність контролюється порогом `tol` за нормою градієнта або `max_iters`.

Модуль visualization.py

- `surface_for_slice(f, grid, slice_var, slice_val)`: 3D поверхня перерізу для фіксованої змінної (x_1, x_2 або x_3).
- `show_isosurface_with_path(f, history, grid_range, vol_n, level, opacity)`: об'ємна сітка та ізоповерхня з накладеною траєкторією оптимізації.
- `plot_metrics(history, ...)`: інтерактивні графіки $f(x_k), t_k, \|\nabla f(x_k)\|$ (логшкили за потреби).
- `save_video_xyz(history, filename, fps)`: анімація траєкторії у 3D (потребує `ffmpeg`).

Використання та верифікація

Мінімальний приклад.

```
import numpy as np
from lab_module import build_variant9_objective, gd_back_tracking, split_step_cond

obj = build_variant9_objective()
x0 = np.array([2.0, 1.0, 0.5])
x_min, history = gd_back_tracking(
    obj.value, obj.gradient, x0,
    cond=split_step_cond, max_iters=2000, tol=1e-4,
)
print('x*=' , x_min, 'f*=' , obj(x_min))
```

Залежності. Візуалізації використовують Plotly/Matplotlib; відео потребує `ffmpeg`. Список залежностей у `requirements.txt`.

Демонстраційний ноутбук `demo_module_usage.ipynb` містить усі сценарії експериментів і буде інтерактивні графіки, з яких у звіті розміщено плейсхолдери для подальших скрінів.

4. Обчислювальні експерименти

4.1. Налаштування

Використано три початкові точки:

$$A : (2, 1, 0.5), \quad B : (-10, 10, 1), \quad C : (0.5, -0.8, 0.2).$$

Для кожної точки застосовано дві стратегії пошуку кроку:

1. *Simple backtracking* (умова простого зменшення);
2. *Goldstein backtracking* (правило Голдстейна, $c = 0.3$).

Початковий крок $t_0 = 1$, коефіцієнт скорочення $\beta = 0.5$, критерій зупинки — $\|\nabla f(x_k)\|_2 < 10^{-4}$.

4.2. Перерізи поверхні рівня

На рис. 1–3 подано перерізи функції (1), коли одна зі змінних зафікована на нулі.

Surface of f with $x_3 = 0.109$

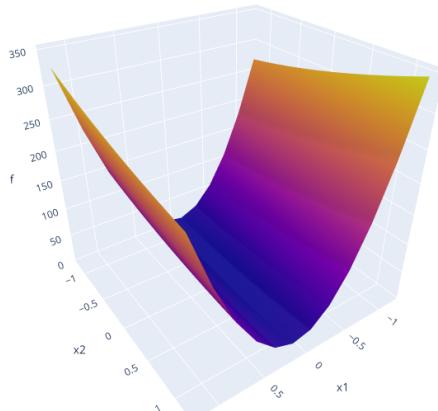
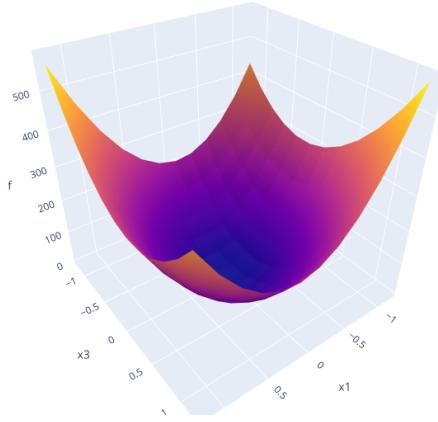
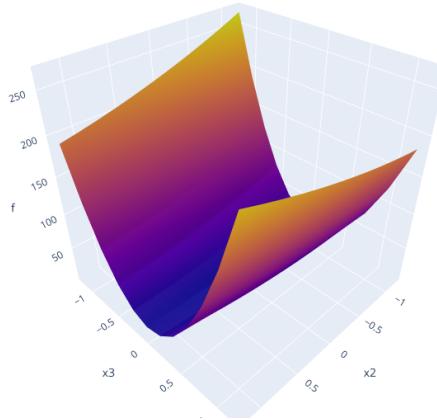


Рис. 1: Переріз при $x_3 = 0$.

Surface of f with $x_2 = 0.109$ Рис. 2: Переріз при $x_2 = 0$.Surface of f with $x_1 = 0.109$ Рис. 3: Переріз при $x_1 = 0$.

4.3. Числові результати

Результати ітераційних процесів зведенено в табл. 1. Для кожної пари “початкова точка + метод” подано кількість ітерацій, фінальне значення функції, норму градієнта та статистику кроків. Дані збігаються з інтерактивними візуалізаціями у ноутбуці.

Табл. 1: Зведення метрик для різних стартових точок.

| Старт | Метод | Ітерації | $f(x_{\text{fin}})$ | $\ \nabla f\ $ | \bar{t} | σ_t |
|-------|-----------|----------|---------------------|----------------------|-----------------------|-----------------------|
| A | Simple | 319 | 5.000000 | 9.6×10^{-5} | 5.12×10^{-3} | 1.81×10^{-3} |
| A | Goldstein | 317 | 5.000000 | 8.7×10^{-5} | 5.13×10^{-3} | 1.83×10^{-3} |
| B | Simple | 380 | 5.000000 | 9.9×10^{-5} | 5.13×10^{-3} | 1.81×10^{-3} |
| B | Goldstein | 375 | 5.000000 | 9.6×10^{-5} | 5.14×10^{-3} | 1.83×10^{-3} |
| C | Simple | 313 | 5.000000 | 9.7×10^{-5} | 5.13×10^{-3} | 1.81×10^{-3} |
| C | Goldstein | 308 | 5.000000 | 9.3×10^{-5} | 5.14×10^{-3} | 1.83×10^{-3} |

4.4. Візуальний аналіз траєкторій

Нижче наведено узагальнені графіки:

- Метрики збіжності ($f(x_k)$, t_k , $\|\nabla f(x_k)\|$) дляожної стартової точки й кожного правила.
- Порівняльні діаграми: кількість ітерацій.
- Ізоповерхні з траєкторіями дляожної комбінації.

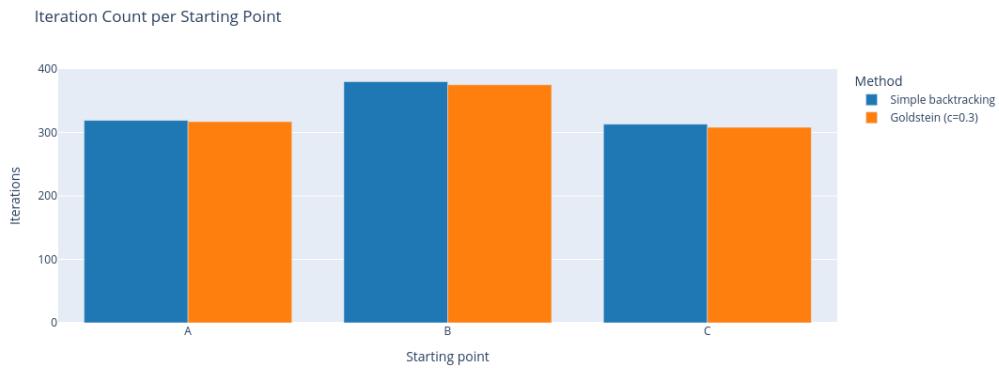


Рис. 4: Агреговане порівняння за кількістю ітерацій.

Агреговані порівняння.

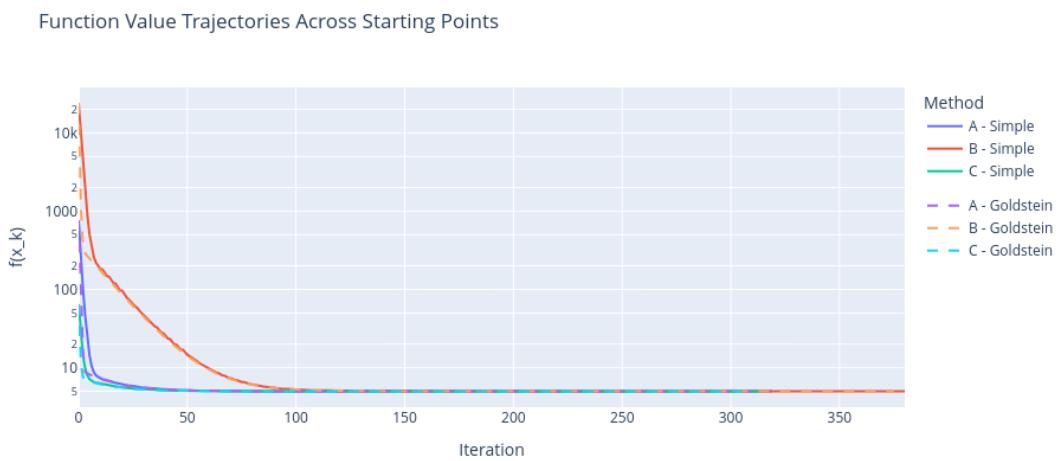


Рис. 5: Еволюція $f(x_k)$ для всіх стартів і правил.

Траєкторії $f(x_k)$ у логмасштабі.

Стартова точка А

Метрики: старт A (Simple backtracking)

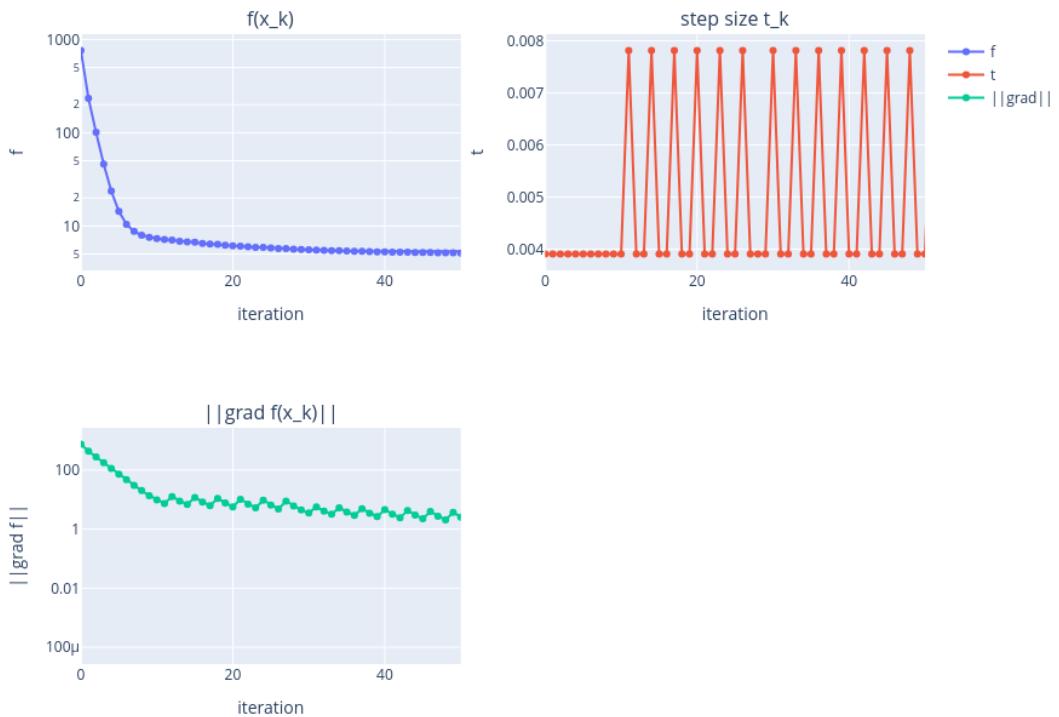


Рис. 6: Старт A — Метрики (Simple).

Метрики: старт A (Goldstein ($c=0.3$))

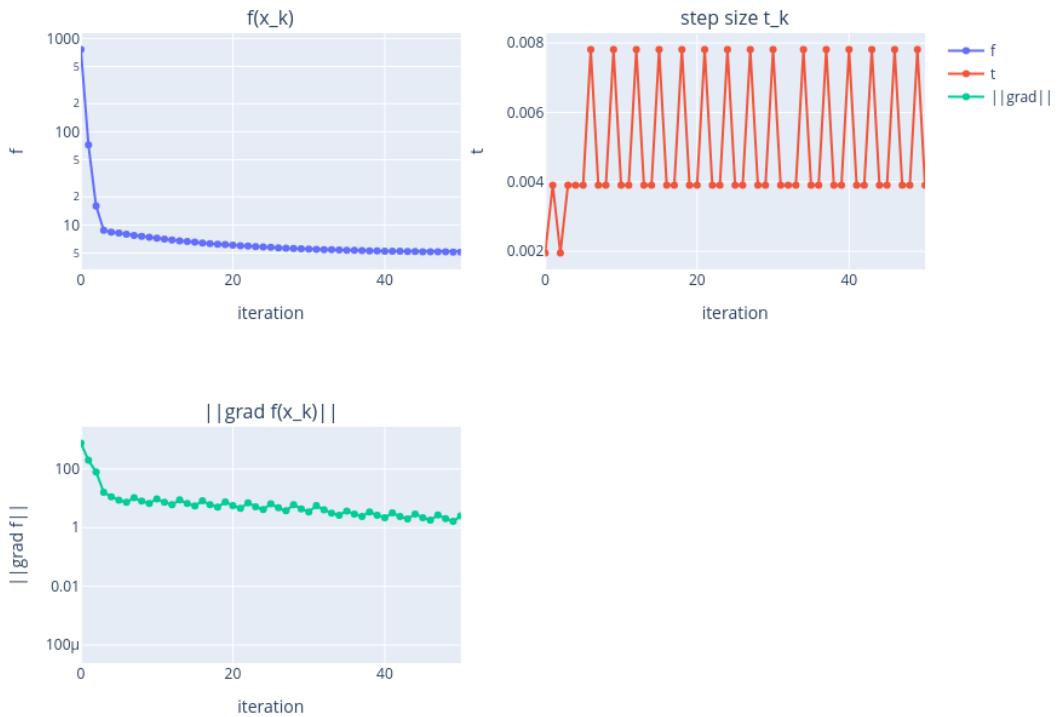


Рис. 7: Старт A — Метрики (Goldstein).

Isosurface - Start A (Simple backtracking)

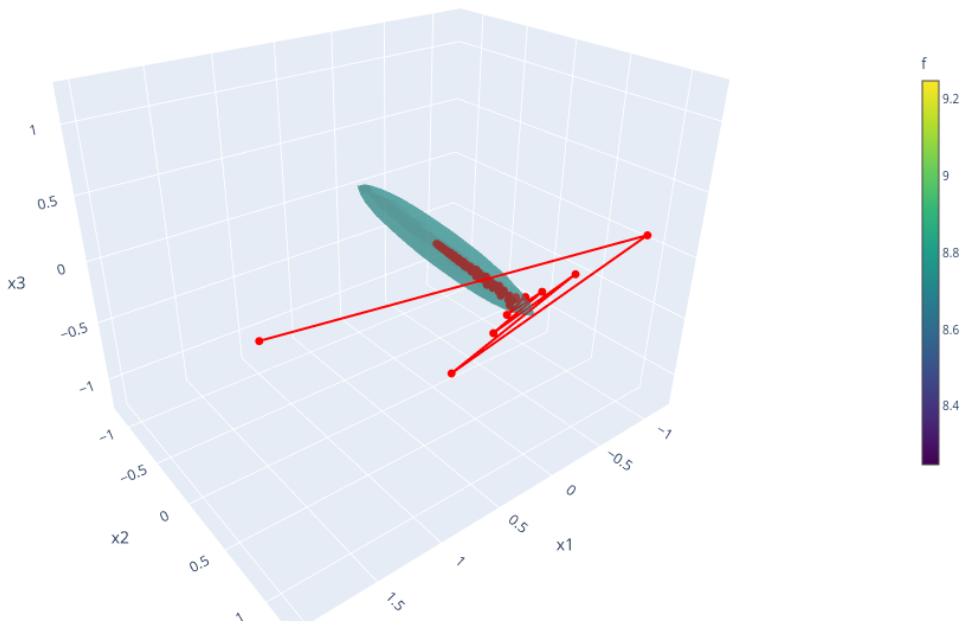


Рис. 8: Старт А — Ізоповерхня (Simple).

Isosurface - Start A (Goldstein ($c=0.3$))

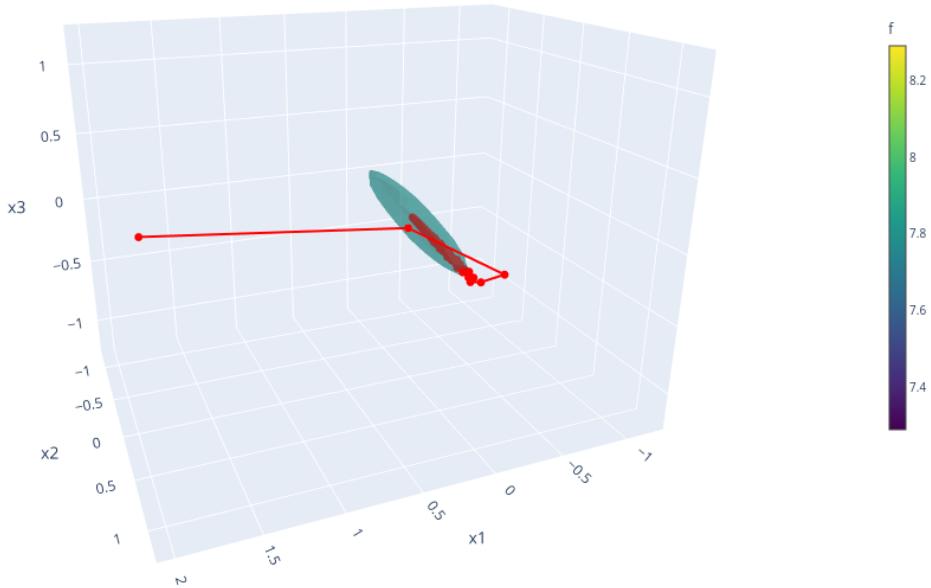


Рис. 9: Старт А — Ізоповерхня (Goldstein).

Стартова точка В

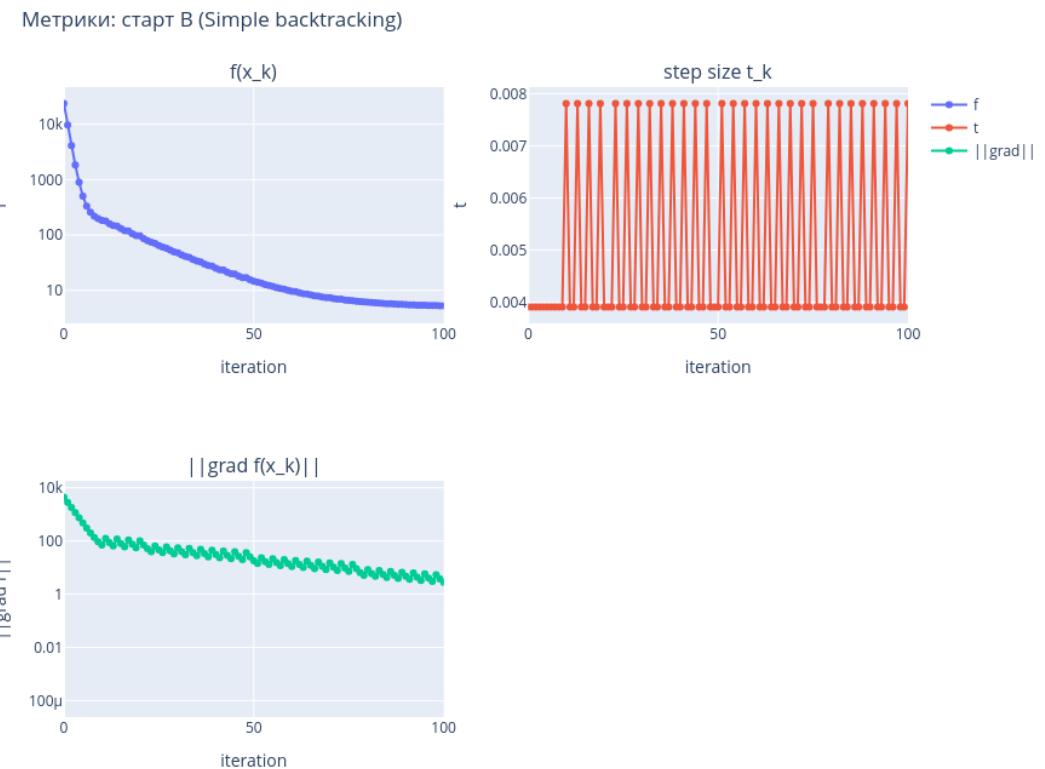


Рис. 10: Старт В — Метрики (Simple).

Метрики: старт В (Goldstein ($c=0.3$))

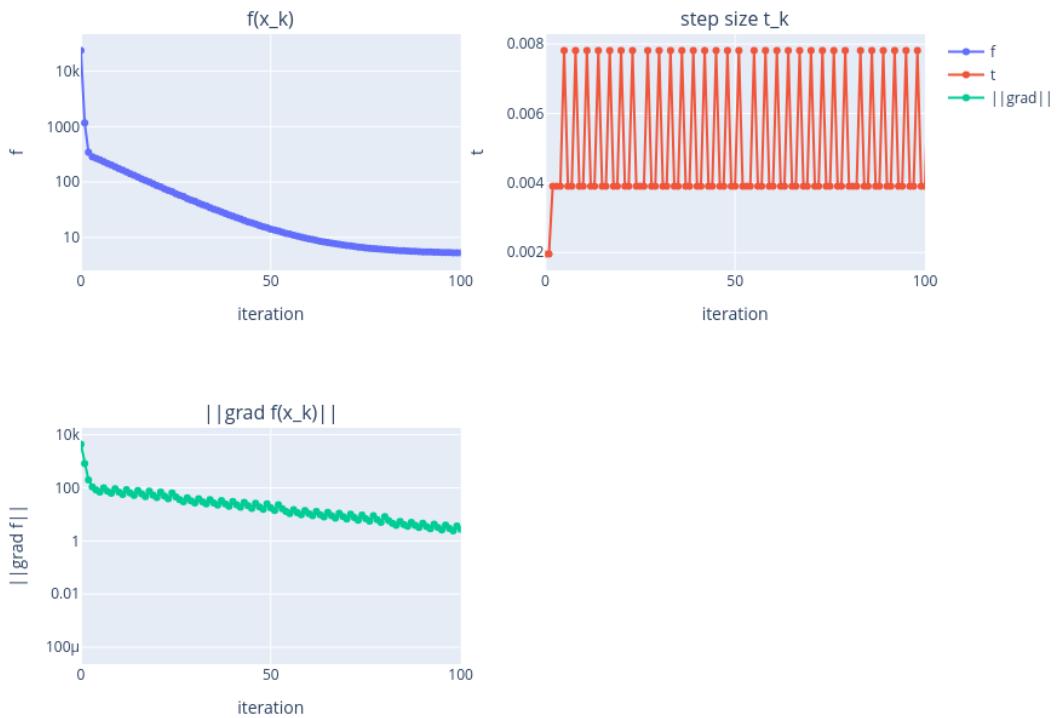


Рис. 11: Старт В — Метрики (Goldstein).

Isosurface - Start B (Simple backtracking)

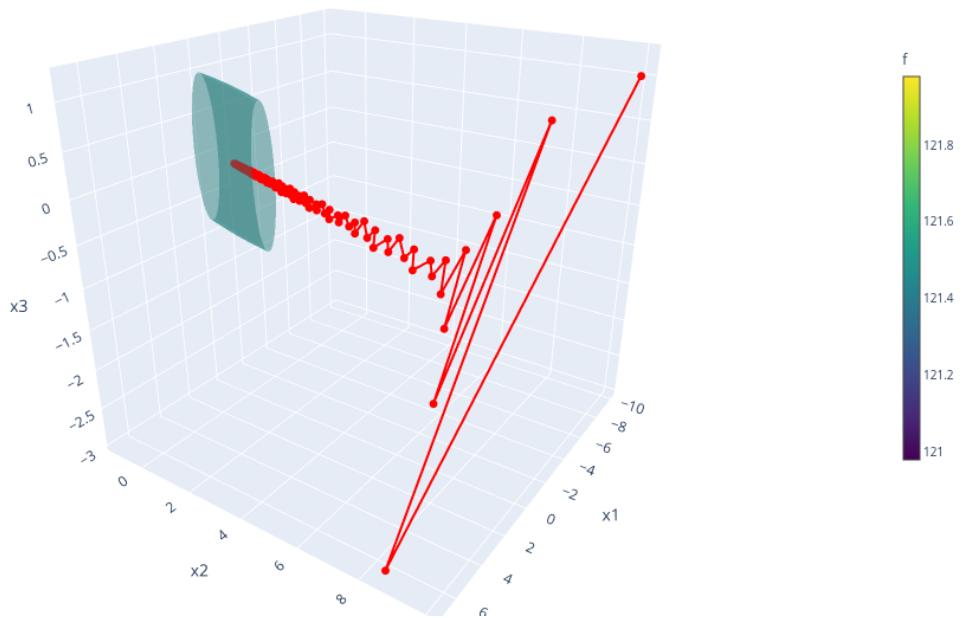


Рис. 12: Старт В — Изоповерхня (Simple).

Isosurface - Start B (Goldstein ($c=0.3$))

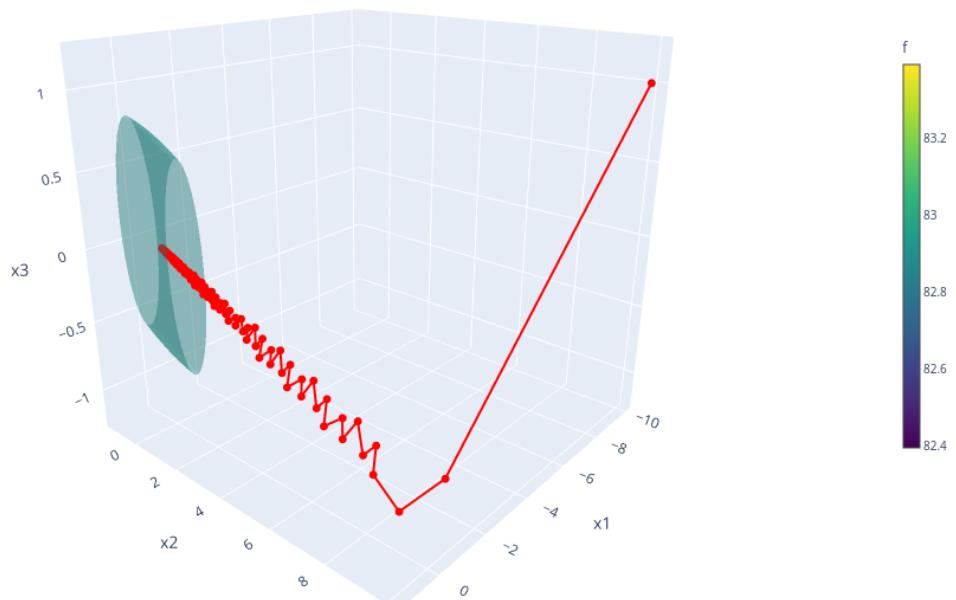


Рис. 13: Старт В — Ізоповерхня (Goldstein).

Стартова точка С

Метрики: старт С (Simple backtracking)

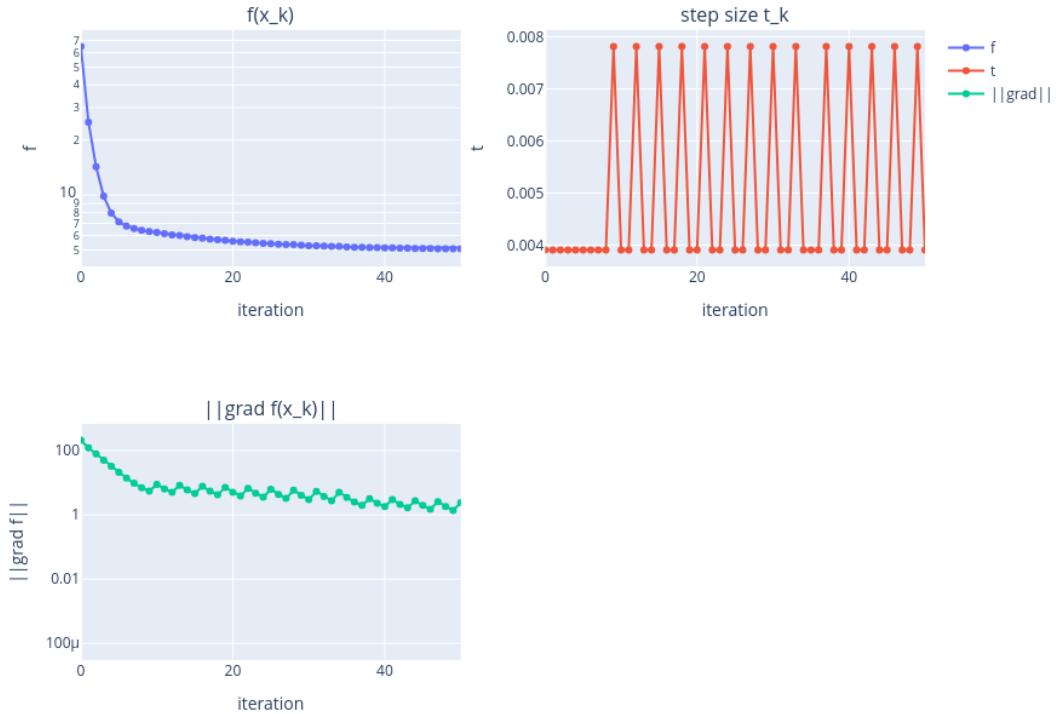


Рис. 14: Старт С — Метрики (Simple).

Ці візуалізації підтверджують, що правило Голдстейна дозволяє робити трохи більші кроки та дає незначне скорочення кількості ітерацій; обидва підходи сходяться до одного мінімуму.

Метрики: старт С (Goldstein ($c=0.3$))

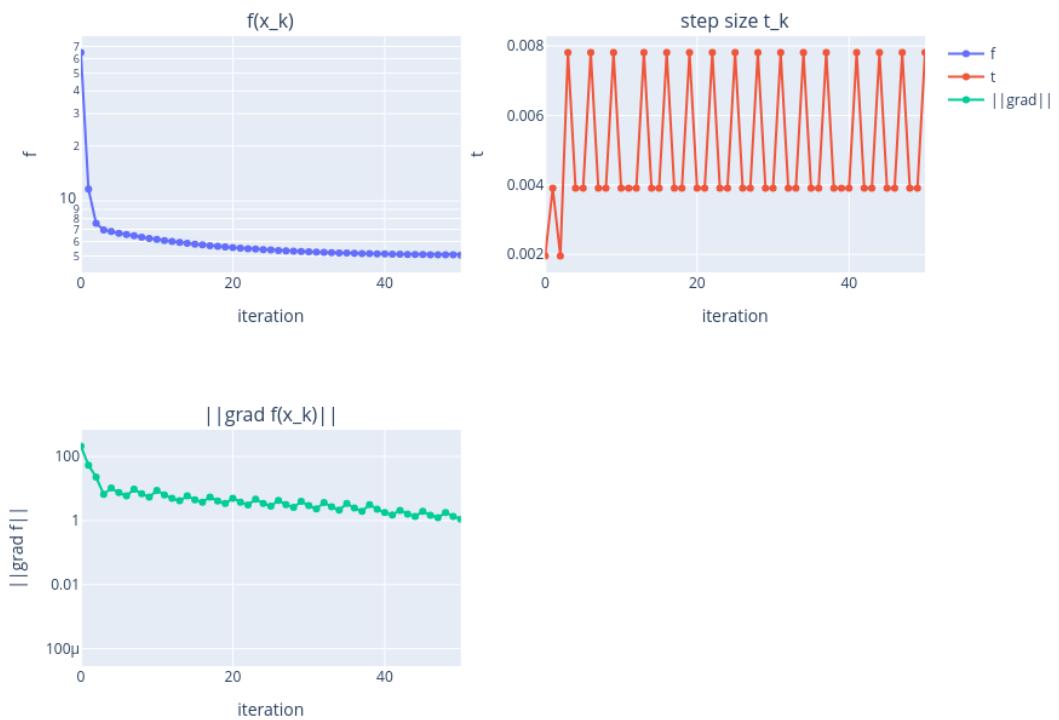


Рис. 15: Старт С — Метрики (Goldstein).

Isosurface - Start C (Simple backtracking)

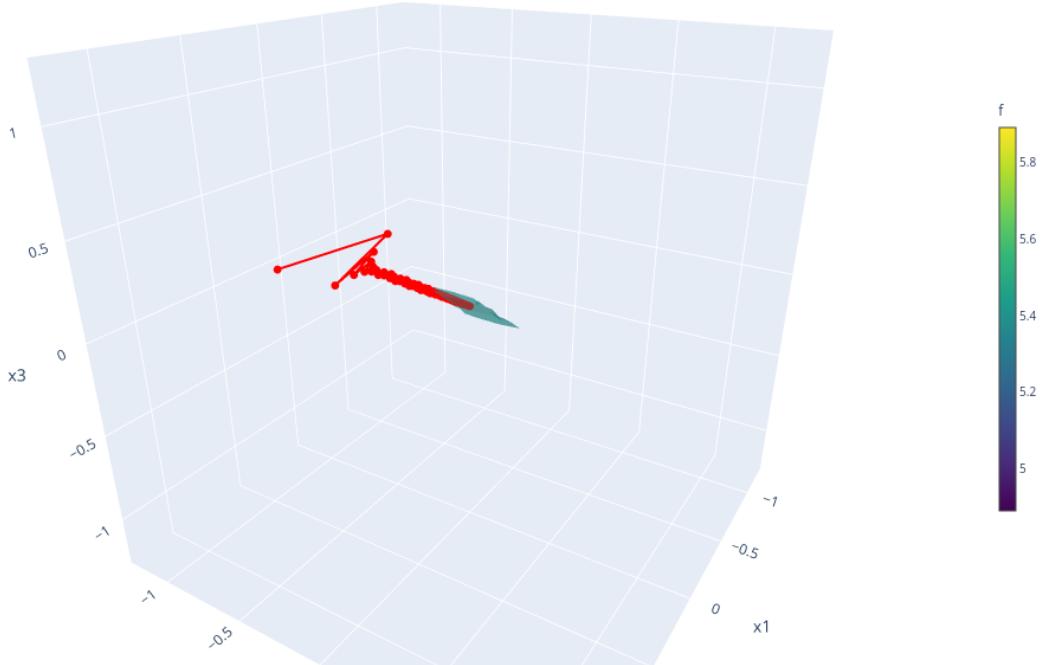


Рис. 16: Старт С — Изоповерхня (Simple).

Isosurface - Start C (Goldstein (c=0.3))

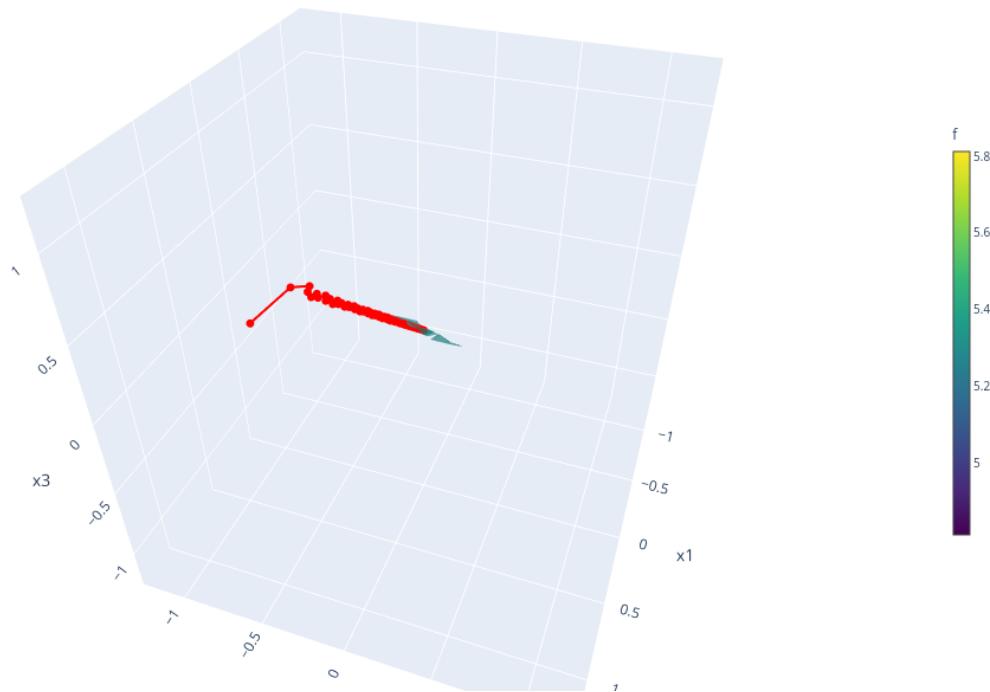


Рис. 17: Старт С — Ізоповерхня (Goldstein).

5. Висновки

- Квадратична функція варіанта 9 є строго опуклою; підтверджено, що всі траєкторії збігаються до $x^* = 0$.
- Проста умова зменшення та правило Голдстейна дають подібну швидкість збіжності; останній трохи зменшує кількість ітерацій, контролюючи надто короткі кроки.
- Побудовані перерізи та ізоповерхні наочно демонструють форму функції та шлях оптимізації, що полегшує інтерпретацію числових результатів.

Література

- [1] Stephen Boyd та Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] Jorge Nocedal та Stephen J. Wright. *Numerical Optimization*. 2-е вид. Springer, 2006.
- [3] Навчальні матеріали кафедри оптимізації. *Методи оптимізації. Частина 2*. Лекційні конспекти, локальна копія у репозиторії. 2020.