МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з курсу «Системний аналіз та математичне моделювання в турбінобудуванні» для студентів напрямків підготовки 6.050604 «Енергомашинобудування» – спеціалізації 6.05060402 «Турбіни», 6.05060406 «Газотурбінні установки та компресорні станції» та 6.050601 «Теплофізика» – спеціалізація 6.05060102 «Теплофізика»

Затверджено
редакційно-видавничою
радою університету,
протокол 2 від 25.06.2015

Харків
НТУ «ХПІ»
2015

Методичні вказівки до лабораторних робіт з курсу «Системний аналіз та математичне моделювання в турбінобудуванні» для студентів напрямків підготовки 6.050604 «Енергомашинобудування» – спеціалізації 6.05060402 «Турбіни», 6.05060406 «Газотурбінні установки та компресорні станції» та 6.050601 «Теплофізика» – спеціалізація 6.05060102 «Теплофізика» / уклад. В. М. Савченко, В. П. Суботович. – Х. : НТУ «ХПІ», 2015. – 40 с.

Укладачі: В. М. Савченко
В. П. Суботович

Рецензент О. І. Тарасов

Кафедра турбінобудування

ВСТУП

Системний аналіз та математичне моделювання в галузі турбінобудування є важливими способами отримання нових знань, технологічних та конструкторських рішень. Імітаційне та математичне моделювання суттєво зменшують час впровадження нових наукових розробок у виробництво [1, 2].

Сучасний фахівець в галузі турбінобудування має володіти апаратом аналітичного та числового аналізу, мати навички програмної реалізації основних методів та алгоритмів, серед яких оптимізація функцій однієї та багатьох змінних, обробка експериментальних даних, апроксимація функціональних залежностей тощо. Ці знання та навички допомагають аналізувати отримані за допомогою пакетів прикладних програм CAD/CAM/CAE результати моделювання процесів та пристроїв, приймати правильні конструкторські та технологічні рішення, обирати оптимальні алгоритми розв'язання різних класів задач.

Методичні вказівки містять необхідні теоретичні відомості про методи, що вивчаються, та алгоритми побудови обчислювального процесу за обраним методом [3–7]. Для програмної реалізації методів, що вивчаються, обрано мову програмування GNU Octave (MATLAB) [8–12] та інтегроване середовище (IDE) GNU Octave. Стандартна бібліотека GNU Octave надає перевірені реалізації поширених числових методів та засоби візуалізації результатів, а IDE GNU Octave, починаючи з версії 4.0, має зручний графічний віконний інтерфейс.

Методичні вказівки містять приклади розв'язання задач та їх окремих фрагментів, що повинно допомогти студентам в процесі розробки власних обчислювальних алгоритмів та програм на їх основі. Для всіх тем наведено орієнтовний перелік стандартних функцій та операторів GNU Octave (додаток А), що надає необхідні інструменти для безпосереднього розв'язання задач та перевірки результатів, отриманих за допомогою власних програм, написаних мовою програмування GNU Octave (MATLAB).

GNU Octave є вільним програмним забезпеченням, тому отримувані за його використанням результати моделювання можна використовувати в інженерній практиці та науковій діяльності. Рекомендується користуватися найсвіжішими версіями пакету GNU Octave.

Лабораторна робота 1

РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ

Метою є ознайомлення студентів з методами розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) аналітичними та числовими методами із використанням GNU Octave.

Завдання

Варіанти завдання обирають із додатка Б.

1. Розв'язати СЛАР методом Гаусса.
2. Розв'язати СЛАР за допомогою вбудованих функцій GNU Octave.
3. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Задано систему з N лінійних алгебраїчних рівнянь (1.1). Знайти розв'язок цієї системи у вигляді $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$, де $\boldsymbol{A}$ – матриця коефіцієнтів розміром $N \times N$, $\boldsymbol{b}$ – вектор правих частин розміром N, $\boldsymbol{x}$ – шуканий вектор розміром N.

$$\begin{cases} a_{11}x_1 & a_{12}x_2 & \dots & a_{1N}x_N & = & b_1, \\ a_{21}x_1 & a_{22}x_2 & \dots & a_{2N}x_N & = & b_1, \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{N1}x_1 & a_{N2}x_2 & \dots & a_{NN}x_N & = & b_N. \end{cases} \tag{1.1}$$

В турбінобудуванні необхідність розв'язання СЛАР високих порядків виникає в різних оптимізаційних задачах, наприклад, профілювання лопаток турбін за допомогою поліномів та сплайнів [1, 2]. Частина методів (Крамера, Гаусса) розв'язання СЛАР є зручними для «ручного» застосування, інша (ітераційні методи, розкладання матриць) – для реалізації у вигляді комп'ютерної програми [3–6]. За допомогою GNU Octave СЛАР розв'язують як у напівавтоматичному режимі, так і використовуючи велику кількість вбудованих функцій [7, 8].

Метод Гаусса складається з прямого та зворотного проходів. Під час прямого проходу за допомогою елементарних перетворень послідовно виключають змінні з (1.1) та формують верхню трикутну матрицю (1.2), а під час зворотного – обчислюють корені $x_i, i \in [1, N]$, починаючи з x_N. Методу Гаусса властиво накопичувати похибки при використанні десяткових дробів на проміжних кроках методу. Реалізація методу для роботи із простими дробами є нетривіальною задачею, оскільки зазвичай мови програмування не підтримують такий формат даних на внутрішньому рівні.

$$\begin{cases} c_{11}x_1 & c_{12}x_2 & \dots & c_{1N-1}x_{N-1} & c_{1N}x_N & = & d_1, \\ 0 & c_{22}x_2 & \dots & c_{2N-1}x_{N-1} & c_{2N}x_N & = & d_2, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & c_{NN}x_N & = & d_N. \end{cases} \quad (1.2)$$

Функції GNU Octave, які використовуються в роботі: *chol*( ), *lu*( ), *qr*( ), *norm*( ), *eig*( ), оператор «\».

Порядок виконання роботи

1. Згідно з варіантом привести СЛАР до вигляду $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$.
2. Розв'язати СЛАР за методом Гаусса.
3. Знайти власні значення матриць $\boldsymbol{A}$ та $\boldsymbol{b}$.
4. Знайти розв'язок у вигляді $\boldsymbol{x} = \boldsymbol{A} \backslash \boldsymbol{b}$ та відхил $\boldsymbol{b} - \boldsymbol{Ax}$.
5. Розв'язати СЛАР за допомогою розкладання матриць Холецького, *LU*- та *QR*-розкладань.
6. Задати матриці `A=(rand(5).*25).+10` та `b=(rand(1,5).*20)'`. Повторити п. 2. Прокоментувати результати.

Приклад розв'язання СЛАР у GNU Octave.

```
A = [5 2 3; 2 6 1; 3 1 7]; b = [10; 20; 30]
[L]=chol(A)       % Розклад Холецького
err= A-L'*L       % L'- транспонована матриця
y=L'\b; x=L\y     % x=L\(L'\b)
norm(A*x-b)       % Похибка розв’язку, p-норма (p=2)
[L U P]=lu(A)     % LU-розклад
err=A-L*U         % Похибка розкладання
y = L\b; x = U\y  % Розв'язок x = U\(L\b)
norm(A*x-b)       % Похибка, p-норма (p=2)
[Q R] = qr(A)     % QR-розклад
x = R\(Q'*b)      % Розв'язок СЛАР
norm(A*x-b)       % Похибка, p-норма (p=2)
```

Питання для самоперевірки

1. Назвіть приклади застосування СЛАР у турбінобудуванні.
2. Які недоліки у методу Гаусса? Як їх подолати?
3. Назвіть основні прийоми роботи з матрицями у GNU Octave.
4. Задати матриці `A=[2,3; 4,1]`, `b=[13; 17]` та знайти розв'язок СЛАР числовим і графічним методами за допомогою GNU Octave.
5. Задати матриці `A=[2,3; 4,1]`,`b=[10; 8]` та знайти розв'язок СЛАР за методом Крамера та графічним методом за допомогою GNU Octave.

Лабораторна робота 2

ЛІНІЙНИЙ РЕГРЕСІЙНИЙ АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Метою є ознайомлення студентів з лінійним методом найменших квадратів (МНК) та аналіз експериментальних даних засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка В.

1. За допомогою МНК та вбудованих функцій GNU Octave визначити параметри кривої, яка найкращим чином описує експериментальні дані.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. В результаті експериментальних досліджень отримано набір точок (табл. 2.1), які пов'язані деякою функціональної залежністю $y = f(x)$. Побудувати аналітичну залежність $y = f(x, a_1, a_2, \dots, a_n)$, яка найбільш точно описує результати експерименту.

Таблиця 2.1 – Набір вхідних даних

x	x_1	x_2	x_3	…	x_n
y	y_1	y_2	y_3	…	y_n

Лінійний МНК. Для розв'язання цієї задачі обирають вид аналітичної залежності $y = f(x, a_1, a_2, \dots, a_n)$ та визначають коефіцієнти a_i. У найпростішому випадку шукають параметри прямої $y = a_2 x + a_1$

Відповідно до *методу найменших квадратів* [3, 4] у кожній точці мінімізують суму квадратів відхилів значень експериментальної (y_i) та розрахованої (Y_i) функцій

$$\varphi(a_1, a_2, \dots, a_n) = \sum_{i=1}^{n} |y_i - Y_i|^2 \to \min. \qquad (2.1)$$

Необхідною умовою існування мінімуму функції (2.1) є $\partial\varphi/\partial a_i = 0$. Якщо параметри a_i входять у (2.1) лінійно, отримують систему з n лінійних рівнянь з n невідомими

$$
\begin{cases}
\sum_{i=1}^{n} \partial\varphi(a_1, a_2, \dots, a_n)/\partial a_1 = 0, \\
\dots \\
\sum_{i=1}^{n} \partial\varphi(a_1, a_2, \dots, a_n)/\partial a_n = 0.
\end{cases} \quad (2.2)
$$

Лінійний випадок, $Y = a_1 + a_2 x$. Підстановкою Y в (2.1) та (2.2) отримують систему двох рівнянь з двома невідомими (2.3)

$$
\begin{cases}
a_1 n + a_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i, \\
a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i,
\end{cases} \quad (2.3)
$$

розв’язком якої є

$$
\begin{aligned}
a_2 &= \left[n \sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i \right] \Big/ \left[n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2 \right], \\
a_1 &= \frac{1}{n} \left[\sum_{i=1}^{n} y_i - a_2 \sum_{i=1}^{n} x_i \right].
\end{aligned} \quad (2.4)
$$

Коефіцієнт кореляції r (2.6) використовують для оцінки степеня залежності між виміряними величинами. Якщо $|r| \to 1$, то існує лінійна залежність між даними, а якщо $|r| \to 0$, то між даними немає лінійного зв’язку.

$$
r = \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}) \Big/ \sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}, \quad (2.5)
$$

де $\overline{x}$, $\overline{y}$ – середні арифметичні значення.

В разі коректного розв’язку задачі *t*-критерій Ст’юдента (2.6) має бути більше табличного значення розподілу Ст’юдента для заданого *n* [3]

$$
t = r\sqrt{(n-2)/(1-r^2)}\,. \quad (2.6)
$$

Функції GNU Octave, які використовуються в роботі: *polyfit*( ), *polyval*( ), *sum*( ), *mean*( ), *corr*( ), *sqp*( ), *plot*( ), оператор «\».

Порядок виконання роботи

1. Визначити параметри лінії регресії за МНК (2.1–2.6):

- задати вектори експериментальних значень x та y;
- за допомогою *plot*(x,y) побудувати графік виміряних значень;
- сформувати систему рівнянь (2.3). Розв'язати отриману систему за допомогою команди «\» (див. лаб. роб. 1);
- розрахувати коефіцієнт кореляції (2.5);
- побудувати на одному графіку експериментальні точки та розраховану лінію регресії.

2. Визначити коефіцієнти лінії регресії за допомогою стандартних функцій GNU Octave. Порівняти результати, побудувавши на одному графіку експериментальні точки та розраховану лінію регресії.

3. Визначити коефіцієнти лінії регресії за допомогою функції *sqp*( ). Порівняти результати з попередніми.

Питання для самоперевірки

1. Назвіть приклади використання МНК у турбінобудуванні.
2. Що дає знання аналітичної залежності між даними, які були отримані під час експерименту?
3. Назвіть основні етапи методу найменших квадратів?
4. Для чого використовують коефіцієнт кореляції у МНК?
5. Як сформувати СЛАР для визначення коефіцієнтів регресії?

Лабораторна робота 3

НЕЛІНІЙНИЙ РЕГРЕСІЙНИЙ АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Метою є ознайомлення студентів з нелінійним методом найменших квадратів та аналізу експериментальних даних засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка В.

1. За допомогою МНК та вбудованих функцій GNU Octave визначити параметри кривої, яка найкращим чином описує експериментальні дані.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. В результаті експериментальних досліджень отримано ряд точок (табл. 2.1). В результаті лінійного МНК з'ясували, що між точками не існує лінійної залежності. Побудувати аналітичну залежність $y = f(x, a_1, a_2, \dots, a_n)$, яка найбільш точно описує результати експерименту.

Нелінійний МНК. В нелінійному регресійному аналізі перевіряють гіпотезу про те, що експериментальні дані пов'язані деякою нелінійною залежністю. У ряді випадків за допомогою додаткових перетворень можна позбутися нелінійності функції Y (2.1) та застосувати лінійний МНК або підібрати функціональну залежність (3.1)–(3.3) та оцінити її адекватність (3.4).

Поліном k-го степеня $Y = \sum_{i=1}^{k+1} a_i x^{i-1}$. Коефіцієнти полінома визначають розв'язанням СЛАР (3.1), матриці коефіцієнтів якої визначають відповідно до (3.2)–(3.3).

$$\boldsymbol{C} \cdot \boldsymbol{a} = \boldsymbol{b}. \tag{3.1}$$

$$C_{ij} = \sum_{k=1}^{n} x_k^{i+j-2}, i = 1, \dots, k+1, j = 1, \dots, k+1. \tag{3,2}$$

$$b_{ij} = \sum_{k=1}^{n} y_k x_k^{i-1}, i = 1, \dots, k+1. \tag{3.3}$$

Функції, які можна привести до лінійного вигляду:

- $Y = ax^b \mathrm{e}^{cx}, Y = ax^b, Y = a\mathrm{e}^{xb}$. За умови $y > 0$, після логарифмування та замін $Z = \ln Y$ та $A = \ln a$ отримують функцію $Z = A + b \cdot \ln x + cx$. Після підстановки Z в (2.1) отримують СЛАР;
- $Y = 1/(ax + b)$ – заміни $Z = 1/Y$ та $X = 1/x$;
- $Y = 1/(a\mathrm{e}^x + b)$ – заміни $Z = 1/Y$ та $X = \mathrm{e}^x$.

У разі адекватно підібраної функції сумарна квадратична помилка наближається до нуля, а індекс кореляції R близький до одиниці

$$\sum_{i=1}^{n} [y_i - Y_i]^2 \to 0, R = \sqrt{1 - \sum_{i=1}^{n} (y_i - Y_i)^2 \Big/ \sum_{i=1}^{n} (y_i - \overline{y})^2}. \tag{3.4}$$

Функції GNU Octave, які використовуються в роботі: *polyfit*( ), *polyval*( ), *spline*( ), *sqp*( ), *plot*( ).

Порядок виконання роботи

1. Задати вектори x та y вихідних точок та побудувати їх графік.
2. За допомогою функції $polyfit(\,)$ обчислити коефіцієнти для декількох поліномів різних степенів (наприклад, 3, 5, 6, 7). Оцінити адекватність підібраної функції за допомогою (3.4). Побудувати графіки, порівняти результати з експериментальними даними, зробити висновки.
3. Обчислити коефіцієнти полінома, використовуючи (3.1)–(3.3) та

вилучивши один-два коефіцієнти полінома, наприклад, два старші парні степеня. Оцінити адекватність підібраної функції, побудувати відповідні графіки, зробити висновки.

4. Визначити коефіцієнти полінома із п. 2 за допомогою $sqp($ $)$.

5. Визначити аналітичну функцію, що найкраще відповідає експериментальним даним (типи функцій див. додаток Б).

6. Визначити сплайн, який найкраще відповідає експериментальним даним, побудувати графік, зробити висновок.

Питання для самоперевірки

1. В яких задачах у галузі турбінобудування виникає необхідність застосування нелінійного МНК?
2. Опишіть алгоритм обчислення коефіцієнтів полінома k-го степеня.
3. Як оцінити адекватність обраної функціональної залежності?
4. Як нелінійну функцію привести до лінійного вигляду?
5. Як степінь полінома впливає на криву регресії? Що відбувається за значного підвищення степеня полінома ($k > 7 \ldots 10$)?

Лабораторна робота 4

ПРЯМІ МЕТОДИ ПОШУКУ МІНІМУМУ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ

Метою є ознайомлення студентів з прямими методами пошуку мінімуму однієї змінної засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка Г.

1. Знайти мінімум функції згідно з варіантом: а) за допомогою програми на мові GNU Octave; б) за допомогою вбудованих функцій.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Нехай цільова функція $y = f(x)$ є функцією однієї змінної x та має мінімум на заданому інтервалі. Визначити мінімум цільової функції із заданою похибкою ε.

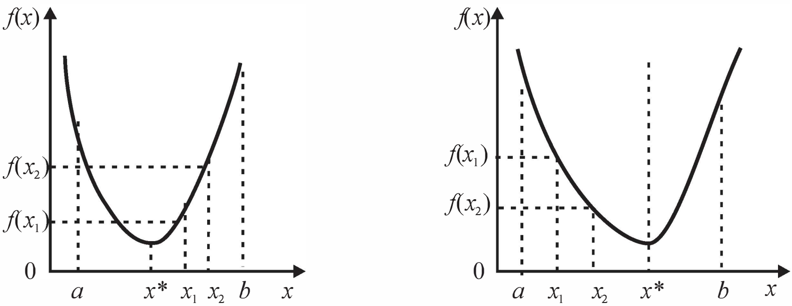
В галузі турбінобудування подібні задачі виникають доволі часто, наприклад, під час обчислення параметрів ідеальної течії в ступені газової турбіни цільова функція відхилу масової витрати газу через соплову ($G_S(p_1)$) та робочу решітки ($G_R(p_1)$) $f(p_1) = [G_S(p_1) - G_R(p_1)]^2$ є функцією однієї змінної (p_1) або в процесі оптимізації профілів перетинів лопаток турбін, де цільова функція відхилу площі ($f(\omega_1)$) від заданої (f) $F = |f(\omega_1) - f|$ також є функцією однієї змінної (ω_1) [1, 2].

Загальна схема прямих методів пошуку мінімуму на інтервалі.

Функцію $f(x)$, $x \in D = [a, b]$ називають унімодальною на D, якщо існує така точка $x^* \in D$, що

$$f(x_1) > f(x_2),\ x^* > x_2 > x_1,\ f(x_1) < f(x_2),\ x^* < x_1 < x_2,\ \ x_1, x_2 \in D. \quad (4.1)$$

Якщо унімодальна функція є неперервною, то вона має єдину точку мінімуму на D, яка співпадає з x^*(рис.4.1 *а*).



а *б*

Рисунок 4.1 – Локалізація інтервалу пошуку мінімуму

Для *локалізації інтервалу, що містить мінімум, о*бчислюють значення функції у двох точках x_1 та x_2 на інтервалі $[a, b]$ ($a < x_1 < x_2 < b$). Із властивості унімодальності (4.1) можна зробити висновок, що мінімум розташовано або на $[a, x_2]$, або на $[x_1, b]$.

Якщо $f(x_1) < f(x_2)$, то мінімум не може бути на $[x_2, b]$ (рис. 4.1 *а*), відповідно, його виключають, прийнявши $b = x_2$. Якщо $f(x_1) > f(x_2)$, то мінімум не може знаходитися на $[a, x_1]$ (рис. 4.2 *б*), тому $a = x_1$. У випадку $f(x_1) = f(x_2)$ мінімум буде на $[x_1,\ x_2]$, тому $a = x_1$, $b = x_2$.

Коли довжина інтервалу, що містить мінімум, стає меншою за задану похибку ε, обчислення припиняють.

Алгоритм локалізації інтервалу, що містить мінімум.

Задати точку x_0 і деяку додатну величину Δ.

Крок 1. Якщо $f(x_0) > f(x_0 + \Delta)$, то $k = 1$, $x_1 = x_0 + \Delta$, $h = \Delta$. Інакше, якщо $f(x_0) > f(x_0 - \Delta)$, то $x_1 = x_0 - \Delta$, $h = -\Delta$.

Крок 2. $h = 2h$, $x_{k+1} = x_k + h$.

Крок 3. Якщо $f(x_k) > f(x_{k+1})$, то $k = k + 1$ та повертаються до кроку 2, інакше відрізок $[x_{k-1}, x_{k+1}]$ містить точку мінімуму.

Методи одновимірного пошуку відрізняються способом вибору точок $x_1,\ x_2$. Ефективність відповідних алгоритмів оцінюють за кількістю обчислень функції, що необхідні для досягнення заданої точності.

Метод рівномірного пошуку. Інтервал $[a, b]$, де знаходиться мінімум, ділять на однакові відрізки розміром $\Delta x = \varepsilon$. Для N-точок $(N = (b - a)/\Delta x)$ обчислюють значення функції та знаходять найменше з них, аргумент якого і є шуканим оптимальним значенням x.

Метод дихотомії (перший варіант). Задають $\varepsilon > 0$, визначають початкове наближення $x_0 = (a + b)/2$.

Крок 1. Якщо $b - a < \varepsilon$, то $x^* = (a + b)/2$ – точка мінімуму, закінчують обчислення, інакше $x_1 = (x_0 + a)/2$ та $x_2 = (x_0 + b)/2$.

Крок 2. Якщо $f(x_1) < f(x_0)$, то присвоєннями $b = x_0$ та $x_0 = x_1$ виключають інтервал $[x_0, b]$ та повертаються на крок 1, інакше – крок 3.

Крок 3. Якщо $f(x_2) < f(x_0)$, то присвоєннями $a = x_0$ та $x_0 = x_2$ виключають інтервал $[a, x_0]$ та повертаються на крок 1, інакше $a = x_1$, $b = x_2$ та повертаються на крок 1.

Метод дихотомії (другий варіант). Точки $x_1,\ x_2$ ділять розташовано на відстані $\Delta < \varepsilon$ від середини $[a_i, b_i]$

$$x_1 = \frac{a_i + b_i - \Delta}{2}, x_2 = \frac{a_i + b_i + \Delta}{2}. \tag{4.2}$$

За одну *ітерацію* (один прохід алгоритму) інтервал невизначеності зменшується вдвічі (4.2). За n ітерацій його довжина стане $(b - a)/2^n$. Задля досягнення заданої точності потрібно $n \geq \ln((b - a)/\varepsilon)/\ln 2$ ітерацій. На кожній ітерації цільову функцію обчислюють двічі.

Алгоритм *методу дихотомії* (*другий варіант*). Задати $\varepsilon > 0, \Delta < \varepsilon$, визначити відрізок локалізації мінімуму $[a, b]$.

Крок 1. $x_0 = (a + b)/2$. За формулою (4.2) обчислюють x_1, x_2. Якщо $f(x_1) > f(x_2)$, то $a = x_0$, інакше $b = x_0$

Крок 2. Якщо $b - a < \varepsilon$, то $x^* = (a + b)/2$ – точка мінімуму, закінчують обчислення, інакше – крок 1.

Метод золотого перетину. Точки x_1 та x_2 ділять інтервал $[a, b]$ за пропорцією *золотого перетину* (4.3)

$$x_1 = a_i + \frac{(3 - \sqrt{5})}{2}(b_i - a_i), \qquad x_2 = b_i - \frac{(3 - \sqrt{5})}{2}(b_i - a_i). \tag{4.3}$$

За одну ітерацію інтервал невизначеності зменшується приблизно в 1.618… разів. На наступній ітерації функцію обчислюють один раз, а для досягнення заданої точності потрібно $n \geq \ln((b-a)/\varepsilon)/\ln\left(\left(\sqrt{5}-1\right)/2\right)$ ітерацій.

Метод чисел Фібоначчі. Як відомо, числа Фібоначчі визначені рекурентними співвідношеннями

$$F_{n+2} = F_{n+1} + F_n, \ \ n = 1,2,3\ldots, \ \ F_1 = F_2 = 1.$$

За методом чисел Фібоначчі на початковому інтервалі $[a_0, b_0]$ обчислюють точки

$$x_1 = a_0 + \frac{F_n}{F_{n+2}}(b_0 - a_0),\ x_2 = a_0 + \frac{F_{n+1}}{F_{n+2}}(b_0 - a_0), \tag{4.4}$$

де n обчислюють з $(b_0 - a_0)/\varepsilon < F_{n+2}$, а числа Фібоначчі можна обчислити за формулою Біне $F_n \cong \left[\left((1+\sqrt{5})/2\right)^n - \left((1-\sqrt{5})/2\right)^n\right]/\sqrt{5}$.

На k-му кроці методу точки x_1, x_2 розташовано симетрично відносно середини відрізка $[a_k, b_k]$

$$x_1 = a_k + \frac{F_{n-k+1}}{F_{n+2}}(b-a),\ x_2 = a_k + \frac{F_{n-k+2}}{F_{n+2}}(b-a). \tag{4.5}$$

Наприкінці процесу $k = n$, а точки x_1 та x_2 співпадають і ділять відрізок $[a_k, b_k]$ навпіл. Зі збільшенням n, з огляду на те, що F_n/F_{n+2} – нескінченна десяткова дріб, симетрія інтервалів порушується, що призводить до спотворення методу.

Для методів золотого перетину і чисел Фібоначчі дотримуються тієї ж стратегії виключення інтервалів, що і в методі дихотомії.

Функції GNU Octave, які використовуються в роботі: *fminbnd*( ), *fminunc*( ), *fminsearch*( ), *sqp*( ), *plot*( ).

Порядок виконання роботи

1. Для досліджуваної функції визначити функцію користувача та побудувати її графік засобами GNU Octave.

2. Визначити інтервал локалізації мінімуму.

3. Уточнити значення мінімуму за допомогою заданого методу. Додати на графік проміжні точки мінімуму і границі інтервалів для перших кроків алгоритму.

4. Порівняти результати за п.п. 3–4 з аналогічними результатами, отриманими за допомогою вбудованих функцій GNU Octave.

Приклад (без пунктів 2 і 3, які слід виконати самостійно)

```
function [f]=myf(x)          %_1 функція користувача
  for i=1:length(x) f(i)=-sin(x(i)); endfor
endfunction
format long g                % 15 знаків після крапки
x=0:0.1:3; plot(x, myf(x))   %_2 побудова графіка
grid on                      % сітка на графіку
hold on     % подальший вивід на той самий графік
%_3, %_4 виконати самостійно
%_5 оптимізація різними методами
[X, Y, INFO, OUTPUT]=fminbnd(@myf, 0,3)
plot(X,Y,'o',"markersize",7); % точка мінімуму
[X, Y]=fminsearch(@myf, 2)    % plot(X,FVAL)
[X, Y, INFO, OUTPUT, GRAD, HESS] = fminunc(@myf, 2)
[X, Y, INFO, ITER]=sqp(2, @myf)
```

Питання для самоперевірки

1. Сформулюйте задачі одновимірної оптимізації, які виникають в процесі моделювання процесів та пристроїв в галузі турбінобудування.
2. Дайте визначення унімодальної функції на інтервалі.
3. Опишіть загальний алгоритм звуження інтервалу невизначеності.
4. Опишіть методи чисел Фібоначчі, дихотомії, золотого перетину.
5. Назвіть характерні недоліки методів нульового порядку?

Лабораторна робота 5

ПРЯМІ МЕТОДИ ПОШУКУ МІНІМУМУ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ. МЕТОДИ ТОЧКОВОГО ОЦІНЮВАННЯ

Метою є ознайомлення студентів з методами точкового оцінювання для пошуку мінімуму однієї змінної засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка Г.

1. Знайти мінімум функції згідно з варіантом: а) за допомогою програми на мові GNU Octave; б) за допомогою вбудованих функцій.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Нехай унімодальна цільова функція $y = f(x)$ є функцією однієї змінної x та має мінімум на заданому інтервалі. Визначити мінімум цільової функції із заданою похибкою ε.

Методи точкового оцінювання. Розглянуті в попередній лабораторній роботі методи дозволяють знайти малий інтервал ($\Delta x \leq \varepsilon$), в якому знаходиться мінімум функції, властивості якої взагалі не враховують. За іншим підходом використовують кілька значень функції в певних точках для її апроксимації звичайним поліномом в обмеженій області. Основна ідея методу – можливість апроксимації гладкої функції та оцінювання точки екстремуму поліномом високого степеня. Якість цієї оцінки можна покращити двома способами – *збільшенням степеня* полінома або *зменшенням інтервалу* апроксимації. При цьому, характер поведінки функції, яку мінімізують, враховується при виборі вигляду полінома.

Метод квадратичної апроксимації (*метод Пауелла*). Нехай в обмеженому інтервалі значень x функцію $f(x)$ можна апроксимувати квадратичним поліномом. Нехай для заданої послідовності точок x_1, x_2, x_3 відомі значення функції в цих точках $f(x_1) = f_1, f(x_2) = f_2, f(x_3) = f_3$, тоді можна визначити коефіцієнти полінома a_0, a_1 та a_2

$$\varphi(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2). \qquad (5.1)$$

Із (5.1) постійні коефіцієнти полінома визначають, як

$$a_0 = f_1, a_1 = \frac{f_2 - f_1}{x_2 - x_1}, a_2 = \frac{1}{x_3 - x_2}\left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right). \qquad (5.2)$$

Якщо точність апроксимації з допомогою квадратичного полінома є досить високою, то, враховуючи унімодальність функції $f(x)$ та застосувавши необхідну умову існування екстремуму $\partial\varphi(x)/\partial x = 0$ до (5.1), отримують оцінку мінімуму

$$x^* = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2}. \qquad (5.3)$$

Алгоритм методу Пауелла заснований на послідовному наближенні до мінімуму за допомогою (5.1)–(5.3).

Крок 1. Задають x_1, похибку $\varepsilon > 0$ та крок $h > 0$.

Крок 2. $x_2 = x_1 + h$.

Крок 3. Якщо $f(x_1) > f(x_2)$, то $x_3 = x_1 + 2h$, інакше $x_3 = x_1 - h$.

Крок 4. Обчислюють значення $F_{min} = \min\{f(x_1), f(x_2), f(x_3)\}$, яке відповідає точці x_{min}

Крок 5. За (5.2)–(5.3) визначають x^*. Якщо $|x^* - x_{min}| < \varepsilon$, то пошук закінчують; інакше якщо $f(x^*) < F_{min}$, то $x_1 = x^*$, інакше $x_1 = x_{min}$ та переходять на крок 2.

Метод квадратичної апроксимації зазвичай застосовують після локалізації точки мінімуму одним з методів, розглянутих у лабораторній роботі 4, так як для функції, що двічі диференціюється, поліном другого порядку достатньо добре апроксимує функцію в околі точки мінімуму.

Алгоритм інтерполяційного методу Девіса – Свена – Кемпі знаходження мінімуму функції на інтервалі.

Задають довільну точку x_0, похибку $\varepsilon > 0$ та крок $h > 0$.

Крок 1. Встановлюють напрямок убування цільової функції. Для цього беруть $x_1 = x_0 + h$. Якщо $f(x_1) \geq f(x_0)$, то $h = -h$, $x_1 = x_0 + h$, $h = 2h$.

Крок 2. Повторюють:

- $x_3 = x_1 + h$;
- якщо $f(x_3) \leq f(x_1)$, то $h = 2h$, $x_0 = x_1$, $x_1 = x_3$, інакше $h = h/2$, $x_2 = x_1 + h$ та переходят на крок 3.

Крок 3. Вилучають найвіддаленішу від мінімуму точку, а з решти трьох b є центральною точкою, $a = b - h$, $c = b + h$.

Крок 4. Проводять квадратичну інтерполяції для визначення мінімуму

$$x^* = b + \frac{h}{2} \cdot \frac{f(a) - f(c)}{f(a) - 2f(b) + f(c)}.$$

Якщо $f(c) < f(x^*)$, то $x_0 = c$, $h = h/2$ та повертаються на крок 1, інакше $x_0 = x^*$ та переходять на крок 1. Обчислення повторюють до тих пір, поки різниця двох послідовних оцінок $|x_{k-1}^* - {x_k}^*| > \varepsilon$.

Функції GNU Octave, які використовуються в роботі: *powell*( ), *plot*( ).

Порядок виконання роботи

1. Для заданої функції (додаток Г) визначити функцію користувача та побудувати її графік засобами GNU Octave.
2. Реалізувати методи точкового оцінювання.
3. Показати на графіку перші кроки роботи алгоритму.
4. Порівняти результат за п. 3 з аналогічними результатами, отриманими за допомогою вбудованих функцій GNU Octave.

Приклад мінімізації за методом Пауелла.

```
clear all; more off; myf=@(x) -sin(x) % функція
X=0:0.1:3; plot(X, myf(X))          % графік
```

```
x=1.0;                                  % початкова точка
o = optimset('MaxIter', 100, 'TolFun', 1E-10); % параметри
[x,y,convergence,iters,nevs] = powell (myf,x,o)
```

Питання для самоперевірки

1. У яких випадках застосовують методи точкового оцінювання?
2. Отримайте розрахункові співвідношення (5.2) і (5.3).
3. Чому бажано визначити початковий інтервал локалізації мінімуму?
4. Опишіть метод Пауелла.
5. Опишіть метод Девіса – Свена – Кемпі.

Лабораторна робота 6

МЕТОДИ ПЕРШОГО ТА ДРУГОГО ПОРЯДКІВ ДЛЯ ПОШУКУ МІНІМУМУ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ

Метою є ознайомлення студентів з методами першого та другого порядків для пошуку мінімуму функції однієї змінної засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка Г.

1. Знайти мінімум функції згідно з варіантом: а) за допомогою програми на мові GNU Octave; б) за допомогою вбудованих функцій.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Нехай цільова функція $y = f(x)$ є унімодальною, безперервною та такою, що диференціюється (або двічі диференціюється) на інтервалі пошуку мінімуму. Визначити мінімум цільової функції із заданою похибкою ε.

Методи оптимізації першого та другого порядків. Методам нульового порядку властиво накопичувати помилки, що призводить до виникнення додаткових похибок. Якщо функція на всьому інтервалі $[a, b]$ є унімодальною та безперервно диференційованою, то точка мінімуму x^* є коренем рівняння $f'(x) = 0$. Для відшукання x^* використовують методи розв'язання нелінійних рівнянь, а найбільш критичною операцією, що призводить до додаткових похибок та грубих помилок, є числове диференціювання.

Метод середньої точки побудований аналогічно прямим методам виключення інтервалу, тільки на кожній ітерації перевіряють виконання умов $f'(x) = 0$ та (або) $f'(x) < ε$.

Крок 1. Задають $\varepsilon > 0$ та дві точки a_1 і b_1 такі, що $f'(a_1) < 0$ та $f'(b_1) > 0$. Приймають $x_1 = (a_1 + b_1)/2$, $k = 1$.

Крок 2. Якщо $f'(x_k) = 0$ або $f'(x_k) < \varepsilon$, то $x^* = (a_k + b_k)/2$ є шуканою точкою мінімуму, обчислення закінчують.

Крок 3. Якщо $f'(x_k) < 0$, то $a_{k+1} = x_k$, $b_{k+1} = b_k$, інакше $a_{k+1} = a_k$, $b_{k+1} = x_k$, $x_k = (a_k + b_k)/2$ та повертаються на крок 2.

Даний метод швидше за прямі збігається до шуканого значення x^*.

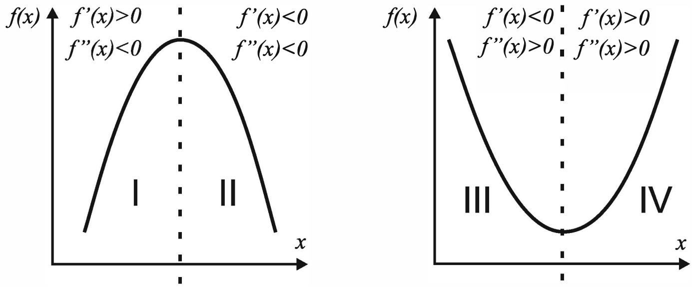


Рисунок 6.1 – Знаки першої та другої похідних

За *методом хорд* функцію замінюють хордою на інтервалі $[a, b]$. Нехай значення першої похідної на кінцях інтервалу є різними за знаком $f'(a) < 0$ и $f'(b) > 0$, тобто функція є опуклою на інтервалі (рис. 4.1, рис. 6.1). За початкове наближення x_0 обирають той кінець інтервалу $[a, b]$, де $f(x_0) \cdot f''(x_0) < 0$. Послідовність наближень до мінімуму визначається рекурентною формулою $x_k = b - f'(b)/(f'(b) - f'(a)) \cdot (b - a)$, $k = 0,1,2, \ldots$. Алгоритм звуження інтервалу невизначеності аналогічний до попереднього. Якщо $f'(x_k) = 0$ або $f'(x_k) \le \varepsilon$ обчислення закінчують, $x^* = x_k$ є точкою мінімуму.

За методом дотичних (*Ньютона*) дугу кривої $y = f(x)$ на інтервалі $[a, b]$ замінюють її дотичною $f'(x) \cong f'(x_0) - f''(x_0)(x - x_0)$. За початкове наближення x_0 обирають той кінець $[a, b]$, де $f(x_0) \cdot f''(x_0) > 0$. Послідовність наближень до мінімуму визначається рекурентною формулою $x_{k+1} = x_k - f'(x_k)/f''(x_k)$, $k = 0,1,2, \ldots$. Основна проблема – правильне обрання точки x_0. Умовою зупинення обчислень є $|x_{k+1} - x_k| < \varepsilon$.

Метод квадратичної апроксимації. Нехай для унімодальної опуклої та двічі безперервно диференційованої на відрізку $[a, b]$ функції $f(x)$ відомими є $y_1 = f(a)$, $y_2 = f(b)$, $y_{11} = f'(a)$, $y_{12} = f'(b)$.

Якщо $y_{11} \cdot y_{12} < 0$, можна побудувати єдиний інтерполяційний поліном третього степеня (кубічний інтерполяційний поліном Ерміта)

$$P_3(x) = c_0 + c_1(x-a) + c_2(x-a)(x-b) + c_3(x-a)^2(x-b) \qquad (6.1)$$

де $c_0 = y_1$; $c_1 = (y_2 - y_1)/(b-a)$; $c_2 = (y_2 - y_1)(b-a)^{-2} - y_{11}/(b-a)$; $c_3 = (y_{12} - y_{11})(b-a)^{-2} - 2(y_2 - y_1)(b-a)^{-3}$.

Із необхідної умови існування екстремуму $\partial P_3(x)/\partial x = 0$, отримують рівняння $c_3(x-a)^2 - 2(y_{11} + y_{12} - 3c_1)/(b-a) + y_{11} = 0$, розв'язком якого є $x_0 = a + \mu(b-a)$, $\mu = (\omega + z - y_{11})/(2\omega - y_{11} + y_{12})$,
$z = y_{11} + y_{12} - 3(y_2 - y_1)/(b-a)$, $\omega = \sqrt{z^2 - y_{11}y_{12}}$, $\mu \in (0{,}1)$, $x_0 \in (a,b)$.

Алгоритм методу.

Задають $\varepsilon > 0$, за умови $f'(a) \cdot f'(b) < 0$ знаходять дві точки a та b.

Крок 1. Обчислюють функцію та її перші похідні в точках a та b. За формулою (6.1) обчислюють коефіцієнти полінома P_3.

Крок 2. Обчислюють x_0 та $f'(x_0)$. Якщо $|f'(x_0)| < \varepsilon$, то обчислення закінчують, а точка $x^* = x_0$. Якщо $f'(x_0) < 0$, то $a = x_0$, інакше $b = x_0$ та повторюють з кроку 1.

Функції GNU Octave, які використовуються в роботі: *fminunc*( ), *diff*( ), *deriv*( ), *polyder*( ), *gradient*( ), *nrm*( ), *nonlin_min*( ), *plot*( ).

Порядок виконання роботи

1. Для функції (додаток В) визначити функцію користувача та побудувати її графік засобами GNU Octave.
2. Реалізувати методи оптимізації згідно із завданням. Порівняти результати для похідних, визначених аналітичними та числовими методами.
3. Показати на графіку перші кроки роботи алгоритму.
4. Порівняти результати за п. 2 та результати, які отримані за допомогою $fminunc$( ).
5. Побудувати графік залежності кількості ітерацій від заданої точності. Для побудови графіка провести 10 вимірювань, початкова $\varepsilon = 0.1$.

Приклад 1. Обчислення похідної в точці (символьне)

```
clear all; pkg load symbolic; pkg load optim
function f=myf(x)   f=sin(x); % похідна -> cos(x) end
syms s; dx=diff(sin(s),s)     % похідна в символьному вигляді
vpa(subs(dx, s,1))            % розраховане значення
deriv(@myf, 1 ) % h=1e-7
gradient(@myf, 1, 0.00001)    % gradient(@sin,1)-помилка, h=1
```

Приклад 2. Обчислення похідної в точці (числове)

```
f=@(x) sin(x); a=-3; b=3; h=0.1; x=a:h:b;
```

```
dx=gradient(f(x), h)      % похідна
plot(x,f(x), x,dx)
```

Приклад 3. Пошук мінімуму функції методом середньої точки

```
clear all; format long g %  15 знаків після крапки
function [f]=myf(x) f=-sin(x); endfunction
x=0:0.1:3;                % дані для побудови графіку
plot(x, @myf(x)); hold; grid ; a=0; b=3; eps=0.001; k=1;
do
  y11=deriv(@myf, a, 1);  y12=deriv(@myf, b, 1);
  plot(0, y11,"r:o", 3, y12,"g-x");
  x1=(b+a)/2.
  dx1=deriv(@myf, x1, 1)
  plot(x1, dx1,"b:+", a, @myf(a),"g:o", b,@myf(b),"c:o" )
  if (dx1<0) a=x1;  else  b=x1;  endif; k++;
until (abs(dx1)<eps)
plot(x1, @myf(x1), "m:o", x1, dx1,"m:o"); hold off
printf("error is: %g, iter=%d\n", 1-sin(x1), k)
```

Приклад 4. Використання функції *fminunc*.

```
function bstop = showJ_history(x, optv, state)
    plot(optv.iter, optv.fval, 'x'); bstop = false;
endfunction
function [f,g] = myfun(x)
  f = 3*x(1)^2 + 2*x(1)*x(2) + x(2)^2;  % cost function
  if nargout > 1
    g(1) = 6*x(1)+2*x(2); g(2) = 2*x(1)+2*x(2);
   endif
endfunction
options = optimset('OutputFcn', @showJ_history,
'GradObj','on');
x0 = [1,1]; xlabel("iteration"); ylabel("cost function");
hold on; [x,fval] = fminunc('myfun',x0',options); hold off
```

Питання для самоперевірки

1. Які недоліки прямих методів усувають методи вищих порядків?
2. Яким критеріям повинна задовольняти функція, щоб можна було застосувати методи першого і другого порядків?
3. Дайте порівняльну характеристику методів хорд і дотичних.
4. Основний недолік методу Ньютона?
5. Переваги та недоліки методу квадратичної апроксимації в порівнянні з іншими методами?

Лабораторна робота 7

ПРЯМІ МЕТОДИ ПОШУКУ МІНІМУМУ ФУНКЦІЇ БАГАТЬОХ ЗМІННИХ

Метою є ознайомлення студентів з прямими методами безумовної оптимізації функції багатьох змінних засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка Д.

1. Знайти мінімум функції згідно з варіантом: а) за допомогою програми на мові GNU Octave; б) за допомогою вбудованих функцій.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Нехай задано функцію $f(\vec{x}): \mathbb{R}^n \to \mathbb{R}$, де $\vec{x}$ – вектор $(x_1, \ldots, x_n)$. Знайти мінімум цієї функції $f(\vec{x}) \to \min\limits_{x \in \mathbb{R}^n}$ з заданою похибкою $\varepsilon > 0$.

Загальна характеристика методів прямого пошуку. У методах прямого пошуку для визначення напрямку спуску до точки мінімуму не використовують інформація про градієнт функції або її матрицю Гессе. Напрямок мінімізації повністю визначається послідовністю обчислених значень функції. Як правило, безумовна мінімізація за методами першого і другого порядків забезпечує вищу швидкість збіжності, але, на практиці, аналітичне визначення похідних функції багатьох змінних не завжди можливе, а застосування числових методів веде до похибок і помилок. В інших випадках, критерій оптимальності може бути заданий не в явному вигляді, а системою рівнянь, що ускладнює або унеможливлює визначення похідних.

Перевагою методів прямого пошуку є простота реалізації у вигляді комп'ютерної програми, при цьому не потрібно знати цільову функцію в явному вигляді та легко врахувати обмеження, як на окремі змінні, так і на область пошуку.

У прямих методах будують послідовності векторів $\vec{x}^0, \vec{x}^1, \ldots, \vec{x}^n$, таких, що $f(\vec{x}^0) > f(\vec{x}^1) > \cdots > f(\vec{x}^n)$. Початкову точку $\vec{x}^0$ обирають або довільною, або таку, що розташована найближче до очікуваної точки мінімуму. Перехід (ітерація) від точки $\vec{x}^k$ до точки $\vec{x}^{k+1}$, $k = 0, 1, 2, \ldots, n$, складається з двох етапів – вибору напрямку руху з точки $\vec{x}_k$ та визначення кроку вздовж цього напрямку.

Метод покоординатного спуску (*підйому*) застосовують у разі, якщо досліджувана функція є унімодальною.

Алгоритм методу. Задати вектор $\vec{x}^0 \in \mathbb{R}^n$ та похибку $\varepsilon > 0$. В якості критерію зупинення вибирати $\|\vec{x}^{k+1} - \vec{x}^k\|_2 \leq \varepsilon$ або $|f(\vec{x}^{k+1}) - f(\vec{x}^k)| \leq \varepsilon$.

Крок 1. Повторюють для $i = 1 \dots n$, поки не виконаний критерій зупинення:

- фіксують значення всіх змінних, крім x_i;
- одновимірна оптимізація отриманої функції.

Результат $\vec{x}^k = (x_1, x_2, \dots, x_n)$, де k – крок оптимізації.

За методом Хука – Дживса напрямок спуску вибирається шляхом дослідження околу точки, знайденої на попередньому кроці. Алгоритм методу складається з послідовності кроків дослідницького пошуку (вибору точки) навколо базисної точки та пошуку за зразком. Результуючий напрямок на k-й ітерації є вектором $\vec{x}^k - \vec{x}^{k-1}$. На етапі дослідження використовують вектор переміщень, множник прискорення і масштабний коефіцієнт, на який ділять координати вектора в тому випадку, якщо в результаті дослідницького пошуку алгоритм не зміг зрушити з попередньої точки. Під час пошуку за зразком використовують дані, що були отримані в процесі дослідження, а мінімізація функції завершується пошуком у напрямку, заданому зразком.

За методом деформованого багатогранника (*метод Нелдера – Міда*) для мінімізації функції $f(\vec{x})$ в n-вимірному просторі будують багатогранник, що містить $(n + 1)$ вершину, яка відповідає деякому вектору. Обчислюють значення цільової функції $f(\vec{x})$ в кожній з вершин багатогранника, визначають максимальне з цих значень та відповідну йому вершину $\vec{x}_h$. Через цю вершину і центр ваги інших вершин проводять пряму, на якій знаходиться точка $\vec{x}_q$ з меншим значенням цільової функції, ніж у вершині $\vec{x}_h$. Надалі вершину $\vec{x}_h$ вилучають, а з решти вершин і точки $\vec{x}_q$ будують новий багатогранник, з яким повторюють описану процедура. У процесі виконання даних операцій багатогранник змінює свої розміри, що й зумовило назву методу.

За методом Розенброка пошук оптимальної точки здійснюють в кожному напрямку. Величина кроку в процесі пошуку безперервно змінюється залежно від рельєфу поверхні рівня. Поєднання обертання координат з регулюванням кроку робить метод Розенброка ефективним для розв'язання складних завдань оптимізації.

За методом Пауелла використовують властивість квадратичної функції таку, що будь-яка пряма, яка проходить через точку мінімуму функції, перетинає під рівними кутами дотичні до поверхонь рівного рівня функції в точках перетину.

Для функції двох змінних зручно представити її графік у вигляді ліній рівня. Лінією рівня функції $f\ (\vec{x}), \vec{x} = (x_1, x_2)$ називають геометричне місце точок площини, в яких функція $f\ (\vec{x}) = \text{const}$. Через звичайну точку площини проходить тільки одна лінія рівня. Точки максимуму і мінімуму можуть бути оточені замкнутими лініями рівня, в сідлових точках перетинаються дві або більше лінії рівня (див. додаток Е).

Функції GNU Octave, які використовуються в роботі: *fminsearch*( ), *powell*( ), *nonlin_opt*( ), *surf*( ), *contour*( ).

Порядок виконання роботи

1. Побудувати графіки досліджуваної функції за допомогою функцій *surf*( ) та *contour*( ).

2. Написати програму оптимізації за методом покоординатного спуску.

3. Реалізувати та дослідити метод Хука – Дживса [10].

4. Порівняти результати за п.п. 2 і 3 з результатами роботи функції GNU Octave *fminsearch*( ), в якій реалізований метод деформованого багатогранника. Кілька разів змінити $\vec{x}^0$, зменшити ε, зменшити крок, порівняти результати.

Приклад мінімізації за допомогою функції *fminsearch*( ).

```
opt=optimset('Display','iter','TolX', 1e-15,'MaxIter', 1000)
function [y]=f(x, c)
  c=[-2 -1 1];
  y=-(c(1)*x(1).^2+c(2)*x(2).^2+c(3)*x(1).*x(2));
endfunction
x=[-30;30]; [x, fval]=fminsearch (@f, x, opt)
```

Питання для самоперевірки

1. Опишіть алгоритм методу покоординатного спуску. Як буде себе вести цій метод для функцій з малою кривизною в області мінімуму?

2. Опишіть алгоритм методу Хука – Дживса.

3. Опишіть алгоритм методу Нелдера – Міда.

4. Сформулюйте переваги і недоліки методів прямого пошуку.

5. Яким чином впливає вибір $\vec{x}^0$ на результат та кількість обчислень?

6. Чому при побудові обчислювального процесу оптимізації функції зазвичай обмежують кількість ітерацій?

7. Побудуйте графік залежності $n = f(\varepsilon)$. Як пов’язані похибка та кількість обчислень?

Лабораторна робота 8

ГРАДІЄНТНІ МЕТОДИ ПОШУКУ МІНІМУМУ ФУНКЦІЇ БАГАТЬОХ ЗМІННИХ

Метою є ознайомлення студентів градієнтними методами безумовної оптимізації функції багатьох змінних засобами GNU Octave.

Завдання

Варіанти завдання обирають із додатка Д.

1. Знайти мінімум функції згідно з варіантом: а) за допомогою програми на мові GNU Octave; б) за допомогою вбудованих функцій.

2. Звіт про роботу повинен містити короткий опис роботи, текст програми і результати її виконання.

Методичні вказівки до виконання завдання

Постановка задачі. Нехай задано функцію $f(\vec{x})\colon \mathbb{R}^n \to \mathbb{R}$, де $\vec{x}$ – вектор $(x_1, \dots, x_n)$. Функція $f(\vec{x})$ повинна відноситься до класу безперервно диференційованих (або двічі безперервно диференційованих) функцій. Знайти мінімум цієї функції $f(\vec{x}) \to \min\limits_{x \in \mathbb{R}^n}$ із заданою похибкою $\varepsilon > 0$.

Загальні відомості про градієнтні методи. Градієнтом функції $f(\vec{x})$ є вектор $(\partial f(x)/\partial x_1, \dots, \partial f(x)/\partial x_n)$

$$\nabla f = \frac{\partial f(\vec{x})}{\partial x_1}\vec{e}_1 + \frac{\partial f(\vec{x})}{\partial x_2}\vec{e}_2 + \dots + \frac{\partial f(\vec{x})}{\partial x_n}\vec{e}_n, \tag{8.1}$$

де $\vec{e}_i = (e_1, e_2, \dots, e_n), i = 1 \dots n$ – одиничний вектор.

Вектор, протилежний градієнту, називають антиградієнтом. Градієнт перпендикулярний до ліній рівня та спрямований убік найшвидшого зростання функції в даній точці. У точці мінімуму(максимуму) градієнт функції дорівнює нулю. На властивостях градієнта засновані методи оптимізації першого порядку, які дозволяють визначити точку локального мінімуму функції. У разі цільових функцій з сильно витягнутими, вигнутими або такими, що мають гострі кути, лініями рівня, методи першого-другого порядку можуть виявитися нездатним забезпечити просування до точки мінімуму.

Загальний алгоритм градієнтних методів.

1. Задають початкову точку $\vec{x}^k \in \{X\}$ $(k = 0)$, визначають градієнт (∇^k), обирають напрямок руху з цієї точки до наступної точки

$$\vec{x}^{k+1} = \vec{x}^k - \gamma^k \nabla f(\vec{x}^k) \tag{8.2}$$

де γ^k – величина кроку, k – номер ітерації.

2. Перевіряють умову закінчення обчислень. Для завдань безумовної оптимізації ($x \in \mathbb{R}^n$) в точці екстремуму норма градієнта не перевищує задану похибку ε: $\|\nabla f(\vec{x}^k)\|_2 = \sqrt{\sum_{j=1}^{n}\left(\partial f(\vec{x}^k)/\partial x_j\right)^2} < \varepsilon$. У разі точного розв'язку в точці екстремуму $\|\nabla f(\vec{x}^k)\|_2 \equiv 0$. Якщо допустиму множину $\{X\}$ задано, то в точці розв'язку (якщо вона на границі) норма вектора-градієнта може бути будь-якою, а критерій зупинення обирають, обмежуючи збільшення цільової функції на двох послідовних ітераціях.

В задачах безумовної оптимізації за способом руху до екстремуму градієнтні методи можна розділити на дві групи:

- на $(k+1)$-й ітерації послідовно, одна за одною, обчислюють координати вектора $(k+1)$-ї точки (методи покоординатного спуску);
- всі координати $(k+1)$-ї точки обчислюють відразу.

Крок методу може бути:

- *постійним,* задають до початку розрахунків;
- *адаптивним*, використовують у разі неможливості визначити мінімум с заданим кроком γ^0, тобто $\|\nabla f(\vec{x}^k)\|_2$ починає збільшуватися з кроку k;
- *обчислюваним* на кожному кроці за кожним напрямком відповідно до визначеного критерію.

Метод покоординатного спуску (*Гаусса – Зейделя*, рис. Д Е.1). На черговій ітерації поступово здійснюють спуск уздовж кожної з координат, а крок γ^k обчислюють *n*-разів на кожній ітерації.

Алгоритм методу. Задають вектор $\vec{x}^0$ та похибку $\varepsilon > 0$.

Крок 1. Повторюють:

- розраховують крок $\gamma_i^k = \arg\min_{\gamma_i^k \geq 0} f\left(\vec{x}_i^k - \gamma_i^k \nabla f(\vec{x}_i^k)\right)$,

$i = 1 \dots n$ – відповідний елемент вектора $\vec{x}^k$;

- розраховують $\vec{x}_i^{k+1} = \vec{x}_i^k - \gamma_i^k \nabla f(\vec{x}_i^k)$;
- якщо $\|\vec{x}_n^k - \vec{x}_0^k\|_2 > \varepsilon$, то $\vec{x}_0^{k+1} = \vec{x}_n^k$, $k = k+1$; інакше закінчують обчислення, а точка $\vec{x}^* \cong \vec{x}^k$ є шуканою точкою мінімуму.

Градієнтний метод з постійним кроком (рис. Д Е.2).

Крок 1. Задають вектор $\vec{x}^0$, крок $\gamma > 0$, похибку $\varepsilon > 0$ та $k = 0$.

Крок 2. Повторюють:

- розраховують вектор градієнта $\nabla f(\vec{x}^k)$;
- якщо $\|\nabla f(\vec{x}^k)\|_2 \leq \varepsilon$, то $\vec{x}^* \cong \vec{x}^k$ – припиняють обчислення; інакше обчислюють $\vec{x}^{k+1} = \vec{x}^k - \gamma^k \nabla f(\vec{x}^k)$.

Більш гнучким є *градієнтний метод із адаптивним кроком.*

Задають вектор $\vec{x}^0$, крок $\gamma > 0$, похибку $\varepsilon > 0$ та $k = 0$.

Крок 1. Розраховують вектор градієнта $\nabla f(\vec{x}^k)$.

Крок 2. Повторюють:

- розраховують норму градієнта $\|\nabla f(\vec{x}^k)\|_2$;
- пробний крок $\vec{x}^{k+1} = \vec{x}^k - \gamma^k \nabla f(\vec{x}^k)$;
- якщо $\|\nabla f(\vec{x}^{k+1})\|_2 < \|\nabla f(\vec{x}^k)\|_2$ та $\|\nabla f(\vec{x}^k)\|_2 < \varepsilon$, $k = k + 1$; інакше, якщо $\|\nabla f(\vec{x}^k)\| < \varepsilon$, то $\vec{x}^* \cong \vec{x}^k$ – точка мінімуму, припиняють обчислення; інакше, якщо $\|\nabla f(\vec{x}^{k+1})\|_2 > \|\nabla f(\vec{x}^k)\|_2$, то відновлюють значення вектору $\vec{x}^{k+1} = \vec{x}^k + \gamma^k \nabla f(\vec{x}^k)$ та змінюють крок $\gamma = \gamma \cdot \delta$, де δ – масштабний множник, наприклад 0,5.

Метод найшвидшого спуску (рис. Д Е.3).

Крок 1. Задають початковий вектор $\vec{x}_0$, похибку $\varepsilon > 0$ та $k = 0$.

Крок 2. Визначають вектор градієнта $\nabla f(\vec{x}^k)$ і нормований вектор градієнта $\nabla f(\vec{x}^k)/\|\nabla f(\vec{x}^k)\|_2$.

Перевіряють умову закінчення обчислень $\|\nabla f(\vec{x}^k)\|_2$.

Крок 3. Повторюють:

- розраховують $\gamma^k = \arg\min_{\gamma^k \geq 0} f(\vec{x}_k - \gamma^k \nabla f(\vec{x}^k))$ – крок в обраному напрямку – одним з методів одновимірної оптимізації.
- розраховують $\vec{x}^{k+1} = \vec{x}^k - \gamma^k \nabla f(\vec{x}^k)$;
- якщо $\|\nabla f(\vec{x}^k)\|_2 > \varepsilon$, то $k = k + 1$, інакше $\vec{x}^* \cong \vec{x}^k$ – точка мінімуму, припиняють обчислення.

За методом Ньютона (метод другого порядку) використовують точні або наближені значення других часткових похідних цільової функції для обчислення матриці Гессе $H(\vec{x}^k)$ та вектору $\vec{x}^k$

$$\vec{x}^{k+1} = \vec{x}^k - \gamma^k H^{-1}(\vec{x}^k)\, \nabla f(\vec{x}^k), \tag{8.3}$$

де $H(\vec{x}^k) = \nabla^2 f(\vec{x}^k) \in \mathbb{R}^{nxn}$ – матриця Гессе, а H^{-1} – зворотна матриця Гессе.

Алгоритм методу. Задають вектор $\vec{x}_0$, похибку $\varepsilon > 0$ та $k = 0$.

Крок 1. Повторюють:

- обчислюють $\nabla f(\vec{x}^k)$ і якщо $\|\nabla f(\vec{x}^k)\|_2 < \varepsilon$, то $\vec{x}^* \cong \vec{x}^k$ – шукана точка мінімуму, припиняють обчислення;
- обчислюють матрицю Гессе $H(\vec{x}^k)$. Якщо не всі її власні значення є додатними, то припиняють обчислення;
- обчислюють напрямок пошуку $\vec{p}^k = -H^{-1}(\vec{x}^k)\nabla f(\vec{x}^k)$;
- методами одновимірної мінімізації шукають крок $\gamma^k = \arg\min_{\gamma \geq 0} f(\vec{x}^k + \gamma \vec{p}^k)$ та за (8.3) обчислюють $\vec{x}^{k+1}$, $k = k + 1$.

Функції GNU Octave, які використовуються в роботі: *nonlin_min*( ), *minimize*( ) *d2_min*( ), *bfgsmin*( ), *nrm*( ), *surf*( ), *contour*( ).

Порядок виконання роботи

1. Побудувати графік досліджуваної функції за допомогою функцій *surf*( ) та *contour*( ).

2. Дослідити програми (додаток Е), змінити початкові умови $\vec{x}_0$, ε, γ та порівняти результати.

3. Змінити програми п. 2 для роботи з заданою функцією, дослідити поведінку програми за різних початкових умовах

4. Дослідити функції GNU Octave, в яких реалізовано методи першого-другого порядків. Порівняти з результатами, отриманими в п. 3.

Приклад використання вбудованих функцій мінімізації.

```
function cost = foo (xx) % min=(1,1,1), f=3.0
  xx--;  cost = sum (-cos(xx)+xx.^2/9); endfunction
function dc = diffoo (x) % перші часткові похідні
  x = x(:)' - 1;  dc = sin (x) + 2*x/9; endfunction
function [c, dc, d2c] = d2foo (x) % другі часткові похідні
  c = foo(x); dc=diffoo(x);  d2c=diag(cos(x(:)-1) + 2/9);
end
x0 = [-15; -15; -15]; % вектор-стовпець
[x,v,n] = minimize ("foo", x0) % метод Нелдера - Міда
[x]=nrm(@foo,x0) % метод Ньютона - Рафсона
[x,v,n] = minimize ('foo',x0,'ndiff') %l-bfgs
[x,v,n] = minimize ("foo",x0',"d2f","d2foo") % d2_min
[x,objf,cvg,outp] = nonlin_min (@foo, x0)
function [d2c]=d2foo(x)d2c=diag(cos(x(:)-1) + 2/9); end
opt=optimset("algorithm","d2_min","objf_grad",@diffoo,
"objf_hessian", @d2foo)
[x,objf,cvg,outp]=nonlin_min (@foo,x0,opt)
% пошук мінімуму для відомих цільових функцій
pbl = optim_problems ( ).general.schittkowski_281;
nonlin_min (pbl.f, pbl.init_p, optimset ("algorithm",
"d2_min", "objf_grad", pbl.dfdp, "objf_hessian", pbl.hessian))
```

Питання для самоперевірки

1. Опишіть метод покоординатного спуску.
2. Опишіть метод градієнтного спуску з постійним кроком.
3. Опишіть метод градієнтного спуску з адаптивним кроком.
4. Опишіть метод найшвидшого спуску.
5. Опишіть метод Ньютона – Рафсона? Як обчислити матрицю Гессе?

ДОДАТКИ

Додаток А
ФУНКЦІЇ GNU OCTAVE

Робота з матрицями

1. $[R, p, Q] = \boldsymbol{chol}(A)$ – розклад Холецького; $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{A}$; $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{Q}^T\boldsymbol{A}\,\boldsymbol{Q}$; $p = 0$, якщо матрицю визначено як додатну.
2. $[L, U, P] = \boldsymbol{lu}(A)$ – LU-розклад матриці на верхню ($\boldsymbol{L}$) та нижню ($\boldsymbol{U}$) трикутні та матрицю перестановок ($\boldsymbol{P}$) такий, що $\boldsymbol{P} \cdot \boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}$.
3. $[Q, R, P] = \boldsymbol{qr}(A)$ – QR-розклад матриці на унітарну ($\boldsymbol{Q}$), верхню трикутну ($\boldsymbol{R}$) та матрицю перестановок ($\boldsymbol{P}$) такий, що $\boldsymbol{A} \cdot \boldsymbol{P} = \boldsymbol{Q} \cdot \boldsymbol{R}$.
4. $\boldsymbol{norm}(A, p)$ – норма матриці. Опція p={0,1,2,'*inf*','*fro*'} є відповідною метрикою. Без аргументів повертає евклідову норму $\|\boldsymbol{b}\|_2 = \sqrt{\sum_i |b_i|^2}$, а у формі $norm(A,'fro')$ повертає норму Фробеніуса $\|\boldsymbol{A}\|_2 = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|a_{i,j}\right|^2}$;
5. $[R, lambda] = \boldsymbol{eig}(A)$ обчислює діагональну матрицю власних значень $\boldsymbol{lambda}$ та матрицю правих власних векторів $\boldsymbol{R}$, що задовольняють рівності $\boldsymbol{A} \cdot \boldsymbol{R} = \boldsymbol{R} \cdot \boldsymbol{lambda}$. Ці вектори є нормованими таким чином, що норма кожного з них дорівнює одиниці. Для пошуку лівих власних векторів – $[L, lambda] = \boldsymbol{eig}(A')$.
6. $\boldsymbol{sum}(A)$, $\boldsymbol{mean}(A)$ – сума та середнє арифметичне елементів.
7. $\boldsymbol{corr}(x, y)$ – обчислення коефіцієнта кореляції двох векторів.

Графіки функцій

1. $\boldsymbol{plot}(x_1, y_1, \dots)$ – побудова графіку функцій, x_i, y_i – одновимірні матриці, y – може бути функцією, наприклад,

```
function [r]=f(x)
  r=sin(x.^2);
endfunction;
y=@(x) cos(x);
x=linspace(-3,3); Y=y(x) ; plot(x, sin(x), x, Y, x, f(x)).
```

2. $\boldsymbol{surf}(X, Y, Z)$, $\boldsymbol{mesh}(X, Y, Z)$ – побудова графіку поверхні $\boldsymbol{Z}=f(\boldsymbol{X}, \boldsymbol{Y})$, *а* $[X, Y] = \boldsymbol{meshgrid}(x, y)$, x, y – вектори значень.
3. $\boldsymbol{contour}\,(X, Y, Z)$ – побудова ліній рівня.

Поліноми, сплайни

1. $\boldsymbol{polyfit}(x, y, k)$. Параметри x, y – масиви вхідних даних, k – степінь полінома. Функція $polyfit$ повертає матрицю коефіцієнтів $\boldsymbol{a}$, починаючи з найбільшого степеня.

2. $\boldsymbol{polyval}(p, x)$ – обчислення значення полінома p в точці x.
3. $\boldsymbol{spline}\ (x, y, breaks)$ – апроксимація сплайнами.
4. $\boldsymbol{polyder}\ (c)$ – обчислення похідної від полінома, заданого вектором коефіцієнтів c.

Диференціювання

1. $dx = \boldsymbol{deriv}\ (f, x0, h, O, N)$ числове диференціювання (*optim*), f – ім'я функції (або вказівник), h – крок, x_0 – точка, $O = 2(4)$, $N = 1 \dots 4$ – порядок похідної.
2. $dx = \boldsymbol{diff}(x)$ – обчислення кінцевих різниць. Якщо x – одновимірний масив вигляду, то $diff(x)$ – вектор різниць сусідніх елементів. Апроксимацією похідної *n*-го порядку є відношення $diff(y, n)./diff(x, n)$.
3. $df = \boldsymbol{diff}(f(x, y, \dots), n)$ – символьне диференціювання (*symbolic*).
4. $[fx, fy, fz, \dots] = \boldsymbol{gradient}(f, h)$ – обчислення градієнта. Для обчислення похідної в точці x_0: $\boldsymbol{gradient}(@f, x_0, h)$, де h – точність.

Оптимізація

Для задач оптимізації використовують вбудовані функції (аналоги функцій *Matlab*) та пакет розширень *optim.* Функція з пакету *optim* ***nonlin_min*** – є інтерфейсом для задач даного класу (замінює ***minimize***). У пакет включено файл *optim_problems.m*, в якому є приклади задач оптимізації. У всіх функціях:

- fun – функція, мінімум якої шукають (або покажчик на неї); *x*0 – початкова точка (вектор);
- $control$ –структура $optimset$ для установки параметрів оптимізації;
- x – знайдена мінімальна точка (або вектор);
- y – значення функції в точці мінімуму;
- $info$ – додаткова інформація про результати обчислень;
- cvg – індикатор успішності процесу оптимізації.

Вбудовані функції оптимізації

1. $[x, y, info, iter, nf, lambda] = \boldsymbol{sqp}(\mathrm{x}0, fun)$ – повертає результат розв'язання задачі квадратичного програмування. У загальному випадку – це багатовимірна оптимізація з обмеженнями (g, h, lb, ub), які задають додатковими параметрами функції *sqp*( ) у форматі $[\dots] = \boldsymbol{sqp}\ (x0, fun, g, h, lb, ub, maxiter, tol)$. Параметри $maxiter$ та tol відповідно обмежують кількість ітерацій та встановлюють точність.

2. $[x, y, cvg, info] = \boldsymbol{fminbnd}\ (fun, a, b, control)$ – пошук мінімуму унімодальної функції на заданому інтервалі методом золотого перетину і параболічної інтерполяції. $control$ може приймати значення *“$FunValCheck$”*, *“$OutputFcn$”*, *“$TolX$”*, *“$MaxIter$”*, *“$MaxFunEvals$”*.

3. $[x, y] = \boldsymbol{fminsearch}\ (fun, x0, control)$ – метод безумовної оптимізації нульового порядку Нелдера-Міда (*Nelder-Mead*).

4. $[x, fval] = \boldsymbol{fminunc}('myfun', x0, control)$ – безумовна мінімізація функції. Для використання методу першого порядку задають опцію $optimset('GradObj', 'on')$ (див. приклад), $control$ встановлює наступні параметри "$FunValCheck$", "$OutputFcn$", "$TolX$", "$TolFun$", "$MaxIter$", "$MaxFunEvals$", "$GradObj$", "$FinDiffType$", "$TypicalX$", "$AutoScaling$".

Функції оптимізації з пакету *optim*

1. $[x, v, nev, \ldots] = \boldsymbol{minimize}\ (f, args, \ldots)$ – пошук мінімуму, інтерфейс до ряду методів оптимізації, в т.ч., $d2\_min$ та $bfgsmin$.

2. $[x, y, cvg, iters, nevs] = \boldsymbol{powell}\ (fun, x0, control)$ – метод Пауелла (багатовимірний випадок), $control$ приймає "$TolX$", "$MaxFunEvals$", "$MaxIter$", "$Display$".

3. $[x, y, cvg, info] = \boldsymbol{nonlin\_min}\ (fun, x0, control)$ – інтерфейс до методів $lm\_feasible$ ($Levenberg - Marquardt$), $octave\_sqp$, $siman$, $d2\_min$.

4. $[x, v, nev, h, args] = \boldsymbol{d2\_min}(fun, d2f, args, control, code)$ – відноситься до групи ньютонівських методів, метод функції $nonlin\_min$.

5. $[x, y, cvg, iters] = \boldsymbol{bfgsmin}(fun, args, control)$ – метод Бройдена – Флетчера – Гольдфарба – Шано (*BFGS*).

6. $x = \boldsymbol{nrm}\ (f, x0)$ – метод Ньютона – Рафсона.

Додаток Б

Завдання до лабораторної роботи 1

Знайти розв’язок системи рівнянь згідно варіанта.

1. $\begin{cases} x_1 - 2x_2 + 3x_3 = 3 \\ 2x_1 + x_2 + 7x_3 = 1 \\ x_1 + 4x_2 + 2x_3 = 2 \end{cases}$

2. $\begin{cases} 2x_1 - 3x_2 + x_3 = 10 \\ 3x_1 - x_2 - x_3 = 3 \\ x_1 + x_2 - 7x_3 = 8 \end{cases}$

3. $\begin{cases} 5x_1 - 3x_2 + 9x_3 = 3 \\ 6x_1 + 2x_2 - x_3 = 2 \\ 3x_1 - x_2 + x_3 = -1 \end{cases}$

4. $\begin{cases} 4x_1 - x_2 + x_3 = 2 \\ 2x_1 + 2x_2 - x_3 = 8 \\ 3x_1 + 5x_3 - x_3 = 7 \end{cases}$

Продовження додатка Б

5. $\begin{cases} 7x_1 + x_2 - 5x_3 = 6 \\ 4x_1 - 4x_2 - 5x_3 = -12 \\ x_1 - 2x_2 + x_3 = -4 \end{cases}$

6. $\begin{cases} 4x_1 - 2x_2 + 3x_3 = 2 \\ 2x_2 - x_2 + 2x_3 = 8 \\ 3x_1 + 5x_2 - x_3 = 7 \end{cases}$

7. $\begin{cases} 7x_1 - x_2 + 2x_3 = -1 \\ x_1 + 2x_2 - 3x_3 = 5 \\ 5x_1 - 4x_2 - 2x_3 = -4 \end{cases}$

8. $\begin{cases} 9x_1 - 3x_2 + 11x_3 = -2 \\ 5x_1 + 2x_2 - 7x_3 = 6 \\ 3x_1 - 4x_2 + x_3 = -8 \end{cases}$

9. $\begin{cases} 2x_1 - 2x_2 + x_3 = 2 \\ 2x_2 + x_2 + x_3 = 5 \\ 7x_1 - x_2 + 2x_3 = 6 \end{cases}$

10. $\begin{cases} 3x_1 - x_2 - x_3 = -6 \\ 5x_1 - 2x_2 + x_3 = -2 \\ 6x_1 - x_2 + 2x_3 = 5 \end{cases}$

11. $\begin{cases} 4x_1 - x_2 + 2x_3 = 7 \\ x_1 + 4x_2 + x_3 = 6 \\ 2x_1 - x_2 + x_3 = 3 \end{cases}$

12. $\begin{cases} 4x_1 - x_2 + x_3 = -2 \\ 5x_1 + 2x_2 - x_3 = 6 \\ x_1 - 4x_2 + x_3 = -8 \end{cases}$

13. $\begin{cases} x_1 - 4x_2 + x_3 = 2 \\ 5x_1 + x_2 - 4x_3 = 5 \\ 2x_1 - x_2 + 2x_3 = 8 \end{cases}$

14. $\begin{cases} 7x_1 - x_2 + x_3 = 1 \\ 5x_1 + 2x_2 - x_3 = -2 \\ 3x_1 - 7x_2 + 2x_3 = 3 \end{cases}$

15. $\begin{cases} x_1 + 3x_2 + 7x_3 = 8 \\ 2x_1 - x_2 + 3x_3 = 10 \\ x_1 - 3x_2 + x_3 = 6 \end{cases}$

16. $\begin{cases} 6x_1 - x_2 + x_3 = 2 \\ 2x_1 - 4x_2 + x_3 = -3 \\ x_1 + x_2 - 2x_3 = 1 \end{cases}$

Додаток В

ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ 2–3

В результаті експериментальних досліджень ступені турбіни отримано набір точок. В *лабораторній роботі* 2 визначити коефіцієнти лінії регресії за МНК. В *лабораторній роботі* 3 підібрати степеневий поліном та аналітичну формулу, які найкращим чином описують експериментальну залежність (рис. Д В.1), використовуючи нелінійний МНК.

Варіанти завдань:

1.

x	1	2	3	4	5
y	2,2	2,8	3,2	3,4	3,8

2.

x	1	2	3	4	5
y	3,15	4,65	5,1	5,25	5,4

3.

x	2	3	4	5	6
y	11,25	9,3	8,25	5,25	4,5

9.

x	1	2	3	4	5
y	2,2	0,4	0,22	0,16	0,12

10.

x	1	2	3	4	5
y	0,25	0,09	0,07	0,05	0,04

11.

x	1	2	3	4	5
y	3	3,5	3,67	3,75	3,8

Продовження додатка В

4.

x	1	2	3	4	5
y	5,63	4,65	4,13	2,63	2,25

5.

x	1	2	3	4	5
y	8,2	5,9	4,9	4	3,2

6.

x	1	2	3	4	5
y	7,2	5,9	4,9	4	3,2

7.

x	1	2	3	4	5
y	4,81	5,76	6,91	8,31	9,95

8.

x	1	2	3	4	5
y	2	6,2	8,6	10,4	11,65

12.

x	1	2	3	4	5
y	2,57	7,15	13	19	27,3

13.

x	1	2	3	4	5
y	3,83	2,55	2,4	2,03	1,91

14.

x	1	2	3	4	5
y	5,1	4,4	3,2	2,7	2,55

15.

x	1	2	3	4	5
y	1,1	1,55	1,9	2,25	2,5

16.

x	1	2	3	4	5
y	1,1	0,2	0,11	0,08	0,06

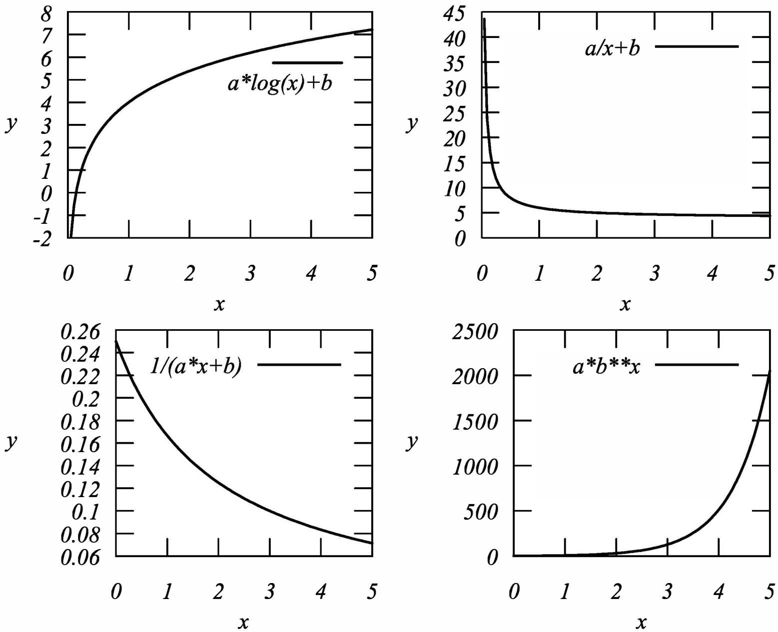


Рисунок Д В.1 – Варіанти функцій для лабораторної роботи 3

Додаток Г
ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ 4–6

Знайти мінімум функції $f(x)$ на визначеному інтервалі.

1. $f(x) = x^4 - 48x^2 + 22x + 5,\ x \in [-8; 0]$.
2. $f(x) = -x^4 + 48x^2 - 22x + 1,\ x \in [-5; 5]$.
3. $f(x) = x^5 + 10x^2 - 10x + 5,\ x \in [-1{,}5; 2]$.
4. $f(x) = 0{,}75x^3 + 2{,}5x^2 - 8{,}25x + 0{,}5, x \in [-2{,}5; 7]$.
5. $f(x) = 0{,}25x^3 - 5x^2 - 8{,}25x + 3,\ x \in [-3; 25]$.
6. $f(x) = 3x^4 - 0{,}8x^3 - 1{,}2x - 15x, x \in [-4; 4]$.
7. $f(x) = 0{,}01x^6 + 2x^4 - 55x^2,\ x \in [-7{,}5; -2]$.
8. $f(x) = x^3 - 5x^2 + x,\ x \in [1; 5]$.
9. $f(x) =-x^3 + 7x^2 + x,\ x \in [-6; 3]$.
10. $f(x) = x^5 + 25x^4 + 7x^3,\ x \in [-11; 10]$.
11. $f(x) = x^2 \ln 0{,}25x + 5,\ x \in [1; 6]$.
12. $f(x) = \sin 0{,}5x^2 \log 0{,}5x,\ x \in [2{,}5; 3.6]$.
13. $f(x) = \mathrm{arctg}\, x \cos 0{,}25x,\ x \in [-7; 1]$.
14. $f(x) = \mathrm{e}^x \sin\cos 0{,}75x,\ x \in [-5; 0]$.
15. $f(x) = 3x \sin \mathrm{e}^{x/3},\ x \in [-10; 2]$.
16. $f(x) = 0{,}5/x - x^2 + 5x,\ x \in [0; 2]$.

Додаток Д
ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ 7–8

Знайти мінімум функції $f(x, y)$ на визначеному інтервалі. Для функцій, що приймають більше двох аргументів, рекомендується під час виконання роботи вважати їх за функції двох змінних [11, 12].

1. Функція *Розенброка* $f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$,
$\min|_{n=2} = f(1; 1) = 0,\ -\infty \le x_i \le \infty, 1 \le i \le n$.

2. Функція *De Jong* (*сфера*) $f(x) = \sum_{i=1}^{n} x_i^2$,
$f(x_1; \dots; x_n) = f(0; \dots; 0) = 0,\ -\infty \le x_i \le \infty, 1 \le i \le n$.

3. Функція *Beale* $f(x, y) = (1{,}5 - x + xy)^2 + (2{,}25 - x + xy^2)^2 +$
$+(2{,}625 - x + xy^3)^2, f(3; 0{,}5) = 0,\ -4{,}5 \le x, y \le 4{,}5$.

4. *Функція Goldstein-Price*
$f(x, y) = [1 + (x + y + 1)^2(19 - 14x + +3x^2 - 14y + 6xy + 3y^2)] \times$
$\times [30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$,
$f(0; -1) = 3,\ -2 \le x, y \le 2$.

5. Функція *Booth* $f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2$, $f(1;3) = 0$, $-10 \leq x, y \leq 10$.

6. Функція *Matyas* $f(x,y) = 0{,}26(x^2 + y^2) - 0{,}48xy$, $f(0;0) = 0$, $-10 \leq x, y \leq 10$.

7. Функція *McCormick* $f(x,y) = \sin(x + y) + (x - y)^2 - 1{,}5x + 2{,}5y + 1$, $f(-0{,}54719; -1{,}54719) = -1{,}9133$, $-1{,}5 \leq x \leq 4, -3 \leq y \leq 4$.

8. Функція *Powell* $f(\vec{x}) = \sum_{i=1}^{4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$, $x_i \in [-4; 5], i = 1 \ldots d$, $f(\vec{x}^*) = 0$, $x^* = (0, \ldots, 0)$. Прийняти $i = 1$, графіки побудувати для фіксованих значеннях $x_1 = x_2 = \text{const}$ або $x_3 = x_4 = \text{const}$.

9. Функція *Styblinski-Tang* $f(\vec{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$, $f(\vec{x}^*) = -39{,}16599d$, $x^* = (-2{,}90534; \ldots; -2{,}90534)$.

10. Функція *Himmelblau* $f(x,y) = (x^2 + y - 11)^2 + (x + y^2 + 7)^2$, $x \in [-4; 4], y \in [-4; 4]$, $f(3{,}2) = 0$, $f(-2{,}8051; 3{,}1313) = 0$, $f(-3{,}7793; -3{,}2832) = 0$, $f(3.5844; 1.8481) = 0$.

11. Функція з двома глобальними мінімумами (*two–hump camel*) $f(x,y) = 1{,}0316285 + 4x^2 - 2{,}1x^4 + x^6/3 + xy - 4y^2 + 4y^4$, $\vec{x}^* = (0{,}898; -0.7126), (-0{,}0898;\ 0{,}7126)$.

12. Функція *Branin-Hoo*:
$f(\vec{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t)\cos x_1 + s + 5x_1$,
$a = 1, b = 5.1\ /\ (4\pi^2), c = 5\,/\,\pi, r = 6, s = 10, t = 1/(8\pi)$,
$x_1 \in [-5, 10]$, $x_2 \in [0, 15]$,
$\vec{x}^* = \big((8{,}8893; 1{,}8893), (-3{,}6891; 13{,}6296), (\ 2{,}5939;\ 2{,}7409)\ \big)$.

13. Функція з трьома глобальними мінімумами (*three–hump camel*): $f(x,y) = 2x^2 - 1.05x^4 + x^6/6 + xy + y^2$, $x \in [-5, 5]$, $y \in [-5; 5]$, $f(0;0) = 0$, $f(-1{,}74753;\ 0{,}87381) = 0{,}29864$, $f(1{,}74753;\ -0{,}87381) = 0{,}29864$.

14. Функція суми різних степенів $f(\vec{x}) = \sum_{i=1}^{n}|x_i|^{i+1}$, $x_i \in [-1;\ 1], i = 1 \ldots n$, , $f(0;0) = 0$.

15. Функція *Eason*: $f(\vec{x}) = -\cos x_1 \cos x_2 \mathrm{e}^{-(x_1-\pi)^2-(x_2-\pi)^2}$, $x_i \in [-100; 100]$, $i = 1{,}2$, $f(0;0) = 0$.

Далі наведено приклад реалізації на мові GNU Octave функції Розенброка, яка приймає вектор $\vec{x}$ та повертає значення, градієнт і матрицю Гессе. (https://github.com/probml/pmtk3/blob/master/demos/rosen2d.m). Інші функції потрібно реалізувати аналогічно, також див. додаток Е.

```
function [f g H] = rosen2d(x)
  if nargin == 0;
     [f g H] = rosen2d(randn(100, 1));  return;  endl;
  if isvector(x) % градієнт
    f = 100*(x(2) - x(1)^2)^2 + (1-x(1))^2;
  else
    f = 100*(x(:,2) - x(:,1).^2).^2 + (1-x(:,1)).^2;
  if nargout > 1 % градієнт
    g = [-400*(x(2)-x(1)^2)*x(1)-2*(1-x(1)); 200*(x(2)-
x(1)^2)]; end
  if nargout > 2 % матриця Гессе
    H = [1200*x(1)^2-400*x(2)+2, -400*x(1); -400*x(1), 200];
  end
end
```

Додаток Е

ПРИКЛАД РЕАЛІЗАЦІЇ ГРАДІЄНТНИХ МЕТОДІВ

Д Е.1 Метод покоординатного спуску

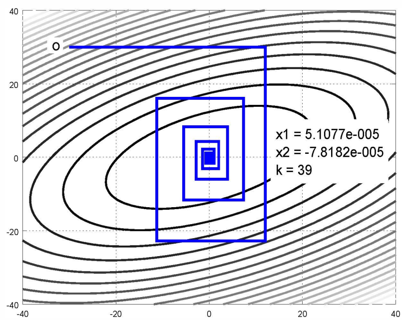


Рисунок Д Е.1 – Візуалізація методу покоординатного спуску

```
clear all; more off;  graphics_toolkit ("gnuplot");
function [z, df]=func(c, x, y)
  if (nargin==2)
     z=c(1)*x(1).^2 + c(2)*x(2).^2 + c(3)*x(1).*x(2);
  endif
  if (nargin==3)
     z=c(1)*x.^2 + c(2)*y.^2 + c(3)*(x.*y);
```

```
    endif
    if (nargout ==2) && (nargin ==2) % градієнт
      df(1)=2*c(1)*x(1) + c(3)*x(2);  df(2)=2*c(2)*x(2) +
c(3)*x(1); endif
    z=-z;
  endfunction
  c=[-2 -1 1]; x=[-30 30];% початкові значення
  % c=[-2 -5 3] %!->1 % c=[1 -1 1]  %!->2
  % c=[1 -1 -1] %!->3 % c=[-2 -2 5] %!->4
  g = 0.2;              % постійна кроку
  d = 0.00001;          % похибка
  k = 1; kmax = 100;  % лічильник ітерацій
  x1t = [x(1)]; x2t = [x(2)]; % проміжні точки
  i = 2;
  while k < kmax
    [temp, gr] = func(c, x); % спуск за першою координатою
    x(1) = x(1) + g*gr(1);
    x1t(i) = x(1); x2t(i) = x(2); i = i + 1;
    [temp, gr] = func(c, x); % спуск за другою координатою
    x(2) = x(2) + g*gr(2);
    x1t(i) = x(1);  x2t(i) = x(2);  i = i + 1;
    if sqrt(gr(1).^2 + gr(2).^2) <= d;% умова зупинення
      break;
    end
    k = k + 1;
  end
  xx = linspace(-40,40); yy = linspace(-40,40); % ==>графік
  [X, Y] = meshgrid(xx, yy);   Z=func(c, X, Y)
  [C, h] = contour(X, Y, Z); % surf(X,Y,Z)
  %clabel(C, h); % мітки на лініях рівня
  hold on; plot(x2t, x2t, '-', 'LineWidth', 3);
  text(x2t(1) -2 , x2t(1) + 0.75, 'O' );
  text(x(1) + 2, x(2), strvcat(['x1 = ' num2str(x(1))], ...
        ['x2 = ' num2str(x(2))], ['k = ' num2str(k)]));
  hold off; % ==>графік
```

Д Е.2. Цикл, в якому реалізовано градієнтний метод

```
  while k < kmax
    [temp, gr] = func(c, x); % спуск за обома координатою
    x = x + g*gr;
    x2t(i) = x(1); x2t(i) = x(2); i = i + 1;
    if norm(gr, 'fro') <= d; % умова зупинення
      break;
```

```
  end
  k = k + 1;
end
```

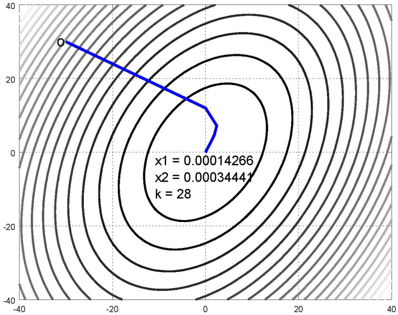


Рисунок Д Е.2 – Візуалізація градієнтного методу

Д Е.3. Цикл, в якому реалізовано метод найшвидшого спуску

```
while k < kmax
  [temp , gr] = func(c, x);
  g=-sumsq(gr)/(2*c(1)*gr(1)^2 + 2*c(2)*gr(2)^2 +
2*c(3)*gr(1)*gr(2));
  x = x + g*gr; x2t(i) = x(1);  x2t(i) = x(2);  i = i + 1;
  if norm(gr, 'fro') <= d;% умова зупинення
    break; %
  end
  k = k + 1;
end
```

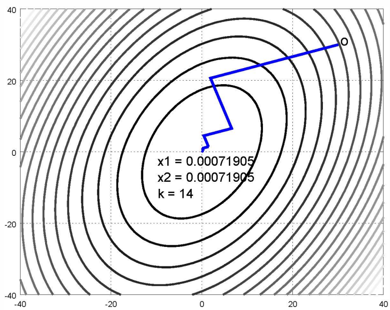


Рисунок Д Е.3 – Візуалізація методу найшвидшого спуску

Список літератури

1. Бойко А. В. Аэродинамика проточной части паровых и газовых турбин: расчеты, исследования, оптимизация, проектирование / А. В. Бойко, А. В. Гаркуша. – Х. : ХГПУ, 1999. – 360 с.
2. Бойко А. В. Многокритериальная многопараметрическая оптимизация проточной части осевых турбин с учетом режимов эксплуатации / А. В. Бойко, А. П. Усатый, А. С. Руденко. – Х. : НТУ «ХПІ», 2014. – 220 с.
3. Бронштейн И. Н. Справочник по математике для инженеров и учащихся втузов / И. Н. Бронштейн, К. А. Семендяев. – М. : Наука, 1986. – 544 с.
4. Каханер Д. Численные методы и математическое обеспечение / Д. Каханер, К. Моулер, С. Нэш. – М. : Мир, 1998. – 575 с.
5. Нефьодов Ю. М. Методи оптимізації в прикладах і задачах : навч. посіб. для студ. вищ. навч. закл. / Ю. М. Нефьодов, Т. Ю. Балицька. – К. : Кондор, 2011. – 324 с.
6. Жалдак М. І. Основи теорії і методів оптимізації : навч. посіб. для студ. мат. спец. вищ. навч. закл. / М. І. Жалдак, Ю. В. Триус. – Черкаси : Брама-Україна, 2005. – 608 с.
7. Лінійна алгебра та аналітична геометрія : навч. посіб. / [В. В. Булдигін, І. В. Алєксєєва, В. О. Гайдей та ін.]. – К. : ТВіМС, 2011. – 224 с.
8. Алексеев Е. Р. Введение в Octave для инженеров и математиков / Е. Р. Алексеев, О. В. Чеснокова. – М. : ALT Linux, 2012. – 368 с.
9. Мэтьюз Дж. Г. Численные методы. Использование MATLAB / Дж. Г. Мэтьюз, К. Д. Финк. – М. : Издат. дом «Вильямс», 2001. – 720 с.
10. MATLAB Source Codes [Електронний ресурс]. – Режим доступу : http://people.sc.fsu.edu/~jburkardt/m_src/m_src.html
11. Virtual Library of Simulation Experiments: Test Functions and Datasets [Електронний ресурс]. – Режим доступу : http://www.sfu.ca/~ssurjano/index.html
12. Global Optimization Test Problems [Електронний ресурс]. – Режим доступу : http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm

ЗМІСТ

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт з курсу «Системний аналіз та математичне моделювання в турбінобудуванні» для студентів напрямків підготовки 6.050604 «Енергомашинобудування» – спеціалізації 6.05060402 «Турбіни», 6.05060406 «Газотурбінні установки та компресорні станції» та 6.050601 «Теплофізика» – спеціалізація 6.05060102 «Теплофізика»

Укладачі: САВЧЕНКО Володимир Миколайович
СУБОТОВИЧ Валерій Петрович

Відповідальний за випуск проф. А. В. Бойко

Роботу до видання рекомендував проф. О.В. Потетенко

В авторській редакції

План 2015 р., поз. 238

Підп. до друку 18.11.2015. Формат 60х84 1/16. Папір офсетний.
Друк – ризографія. Гарнітура Times New Roman. Ум. друк. арк. 2,33
Наклад 50 прим. Ціна договірна.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК №3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.