

Міністерство освіти та науки України
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
кафедра електромеханіки



**МЕТОДИЧНІ ВКАЗІВКИ
ДО САМОСТІЙНОЇ РОБОТИ**

з дисципліни «Математичні моделі та програмні засоби розв'язання комплексних
практичних задач електротехніки та електромеханіки»
першого (бакалаврського) рівня вищої освіти

Галузь знань: G «Інженерія, виробництво та будівництво»

Спеціальність: G3 «Електрична інженерія»

Укладачі:

д-р. техн. наук, професор, професор кафедри електромеханіки В. Титюк
старший викладач кафедри електромеханіки В. Власенко

Відповідальний за випуск:

д-р. техн. наук, професор, зав. кафедри електромеханіки С. Толмачов

Рецензент:

канд. техн. наук, доцент кафедри електромеханіки М. Барановська

Методичні вказівки до самостійної роботи з дисципліни «Математичні моделі та програмні засоби розв'язання комплексних практичних задач електротехніки та електромеханіки» призначені для підготовки студентів спеціальності G3 «Електрична інженерія» (141 «Електроенергетика, електротехніка та електромеханіка»). Вони містять короткі теоретичні відомості, поетапні інструкції для проведення самостійної роботи студентів, контрольні запитання для перевірки засвоєних знань, а також перелік рекомендованої літератури.

Розглянуто
на засіданні кафедри
електромеханіки
Протокол №1 від 29.08.2025 р.

Схвалено
на вченій раді
електротехнічного факультету
Протокол № 1 від 29.08.2025 р.

З М І С Т

Самостійна робота №1

ОСНОВИ РОБОТИ У GNU OCTAVE/MATLAB. БАЗОВІ ОПЕРАЦІЇ У GNU

OCTAVE/MATLAB 4

Самостійна робота №2

ПОБУДУВАННЯ ГРАФІКІВ В OCTAVE /MATLAB 11

Самостійна робота №3

АНАЛІТИЧНИЙ РОЗРАХУНОК ЕКСТРЕМУМУ ФУНКЦІЇ ОДНІЙ

ЗМІННОЇ..... 18

Самостійна робота №4

КІЛЬКІСНІ МЕТОДИ ПОШУКУ ЕКСТРЕМУМУ ФУНКЦІЇ ОДНІЄ ЗМІННОЇ.

МЕТОД ЗВОРОТНОГО ЗМІННОГО КРОКУ 23

Самостійна робота №5

КІЛЬКІСНІ МЕТОДИ ОДНОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД ПОЛОВИННОГО

ДІЛЕННЯ 28

Самостійна робота №6

МЕТОДИ БЕЗУМОВНОЇ БАГАТОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД ХУКА-

ДЖИВСА (HOOKE – JEEVES)..... 34

Самостійна робота №7

МЕТОДИ БЕЗУМОВНОЇ БАГАТОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД НЕЛДЕРА-

МІДА (A. NELDER AND R. MEAD)..... 39

Список рекомендованих джерел..... 45

САМОСТІЙНА РОБОТА №1 ОСНОВИ РОБОТИ У GNU OCTAVE/MATLAB. БАЗОВІ ОПЕРАЦІЇ У GNU OCTAVE/MATLAB

Ціль роботи : отримати уявлення про основні принципи програмування на m - мові програмування в GNU Octave/MATLAB, навчитися створювати вектори та матриці вихідних даних необхідного розміру та виконувати базові операції над ними.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Ознайомлення з GNU Octave.

Сучасну інженерну діяльність неможливо уявити без широкого застосування обчислювальної техніки та сучасного програмного забезпечення.

«Золотим» стандартом програм для виконання інженерних розрахунків у сфері моделювання процесів і систем, а також у ряді інших галузей інженерної діяльності в даний час, на наш погляд, є програма MATLAB (від слів (MATrix LABoratory – матрична лабораторія), розроблена американською компанією MathWorks. Компанія MathWorks створена в 1984 році Моулером (Cleve Moler) та Стівом Бангертом (Steve Bangert).

Домінуюче положення MATLAB на ринку інженерних обчислень пов'язане з тим, що ця потужна обчислювальна система має вбудовані засоби для вирішення надзвичайно широкого кола математичних, інженерних та економічних завдань, що динамічно розвивається, залишаючись на передньому краї сучасних інформаційних технологій, пов'язаних з машинним навчанням, штучними нейронними мережами та інше.

Працюючи в середовищі MATLAB, користувач, що навіть не є програмістом, може легко і швидко вирішувати обчислювальні завдання у різних галузях науки і техніки (лінійна алгебра, теорія управління, обробка сигналів тощо). Просте командне середовище дозволяє вводити вирази у формі, близькі до природного математичного запису.

Найбільшою мірою система спрямована виконання інженерних розрахунків. Математичний апарат оптимізований для обчислень, що проводяться з матрицями та комплексними числами. MATLAB містить безліч вбудованих функцій, необхідних інженеру та науковцю для виконання складних чисельних розрахунків, а також моделювання поведінки технічних систем та фізичних процесів. Потужна графічна система MATLAB дозволяє візуалізувати подання даних, що уможливорює графічний аналіз результатів. Всі функціональні можливості об'єднані зручним інтерфейсом користувача.

Істотним недоліком MATLAB є висока вартість ліцензійних продуктів, що призводить до необхідності пошуку альтернативних рішень. Саме такою альтернативою програмі MATLAB є GNU Octave.

GNU Octave — програмна система для математичних обчислень, що вільно розповсюджується за ліцензією GNU General Public License, яка використовує сумісну з MATLAB мову високого рівня.

Octave представляє інтерактивний командний інтерфейс для вирішення лінійних та нелінійних математичних завдань, а також проведення інших чисельних

експериментів. Octave можна використовувати для пакетної обробки. Крім роботи з програмами (функціями), обчислення можна виконувати в режимі калькулятора, тобто отримувати результат відразу після введення чергової команди.

Мова Octave оперує арифметикою речових і комплексних скалярів, векторів і матриць, має розширення для вирішення лінійних алгебраїчних завдань, знаходження коренів систем нелінійних алгебраїчних рівнянь, роботи з поліномами, розв'язання різних диференціальних рівнянь, інтегрування систем диференціальних і диференціальних на кінцевих та нескінченних інтервалах. Цей список можна легко розширити, використовуючи мову Octave (або використовуючи модулі, що динамічно завантажуються, створені мовами C, C++, Фортран та ін.).

Саме сумісність GNU Octave з MATLAB за мовою високого рівня, подібність інтерфейсу користувача роблять GNU Octave відмінною альтернативою MATLAB, допомагають з незначними витратами підготуватися до роботи в MATLAB.

1.2 Головне вікно програми GNU Octave

Після запуску програми Octave на екран виводиться основне вікно, яке іноді називають робочий стіл (desktop) (рис. 1.1).

Інтерфейс користувача GNU Octave локалізовано і може відображатися російською мовою, що вигідно відрізняє GNU Octave від MATLAB .

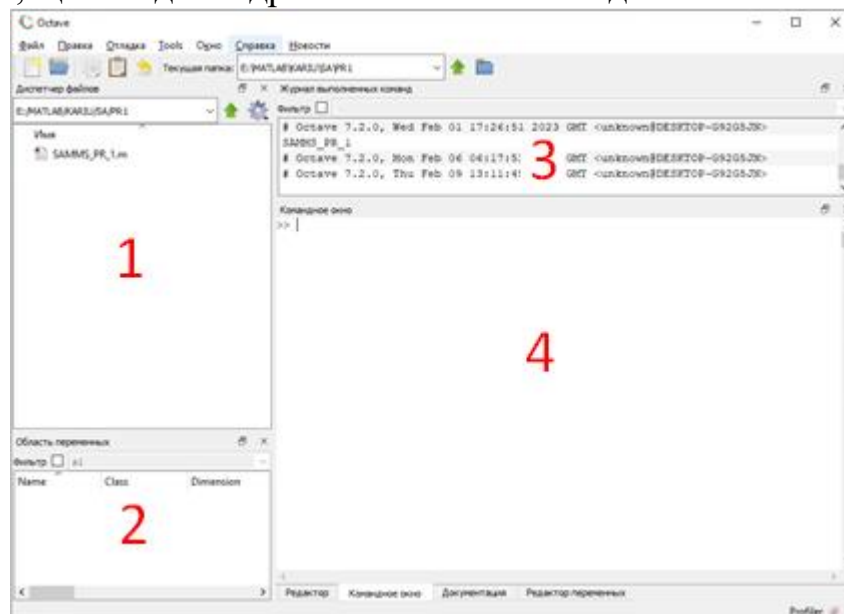


Рисунок 1.1 – Головне вікно програми GNU Octave

У верхній частині головного вікна розташовані стандартні елементи керування:

- рядок головного меню, в якому знаходяться пункти меню *Файл*, *Правка*, *Налагодження*, *Tools*, *Вікно*, *Довідка* та *Новини* ;
- панель інструментів з кнопками, що дозволяють виконати деякі найпоширеніші операції, які можна вибрати через меню.

У лівій частині головного вікна розташовані два важливі елементи управління:

1. **Диспетчер** файлів дозволяє встановити поточну папку і відображати файли, що знаходяться в ній. Аналог в інтерфейсі MATLAB

називається **Current Folder** ;

2. **Область змінних** , призначена для зберігання та керування змінними, створеними для повторного використання поточними проектами. Аналог в інтерфейсі MATLAB називається **Workspace** ;

У правій частині головного вікна розташовані такі елементи керування:

3. У верхній частині розташовано вікно **Журнал виконаних команд** , призначене для перегляду та повторного виклику раніше введених команд. Аналог в інтерфейсі MATLAB називається **Command History** (історія команд).

4. У нижній частині праворуч розташоване найбільше і ймовірно найважливіше вікно - **Командне вікно**. Аналог в інтерфейсі MATLAB називається **Command Window** . В інтерфейсі GNU OCTAVE **Командне вікно** містить чотири вкладки. У вкладці **Редактор** розташований текстовий редактор для створення та редагування текстів програм (скриптів) внутрішньою мовою програмування. У вкладці **Командне вікно** відображаються дані, команди та результати їх виконання, а також повідомлення про помилки. У вкладці **Документація** розташована довідкова система з окремих компонентів GNU OCTAVE . У вкладці **Редактор змінних** відображаються та редагуються змінні з **Області змінних** після подвійного клацання на ім'я потрібної змінної.

Усі змінні, що використовуються в поточному сеансі, зберігаються в пам'яті комп'ютера в області, яка називається **Область змінних** . Повний перелік заданих змінних відображається у вікні **Область змінних** . Тут міститься повний перелік змінних та його розміри, але з значення змінних. Подвійне клацання на ім'я змінної викликає відкриття вкладки **Редактор змінних** , яке можна використовувати для редагування окремих елементів векторів та матриць.

Регулювати розміри вікон можна за допомогою миші.

1.3 Основні команди головного меню GNU Octave

Робота з меню в GNU Octave подібна до роботи з меню програм Microsoft Office. Багато елементів меню мають для швидкого виклику команд гарячі клавіші (їх можна бачити праворуч від команди, наприклад *Copy*→*Ctrl+C* тощо).

Розглянемо склад та призначення деяких команд меню.

Робота із файлами. Пункт меню *Файл* містить наступні команди (рис. 1.2, а):

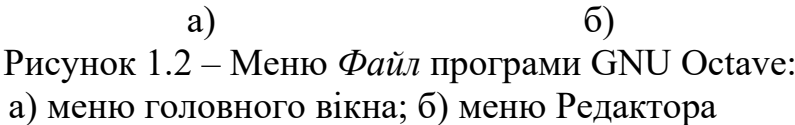
Створити – дозволяє створити новий *m*- файл зі сценарієм (програмою, скриптом) у редакторі GNU Octave у вкладці **Редактор** ;

Відкрити – дозволяє відкрити наявний *m*- файл;

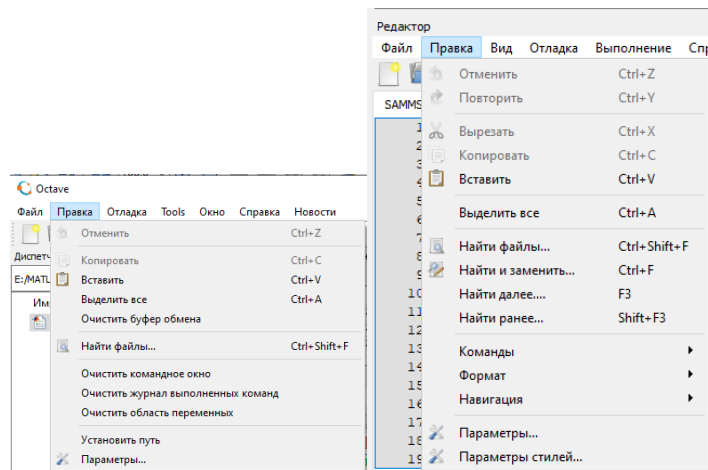
Останні файли – дозволяє отримати швидкий доступ до останніх файлів, якими ви недавно працювали;

Завантажити область змінних – дозволяє завантажувати із зазначеного файлу з розширенням *mat* змінні, що зберігаються в ньому;

Зберегти область змінних як ... – дозволяє зберігати вибрані змінні робочого простору у файл з розширенням *mat* для повторного використання після вимкнення GNU Octave.



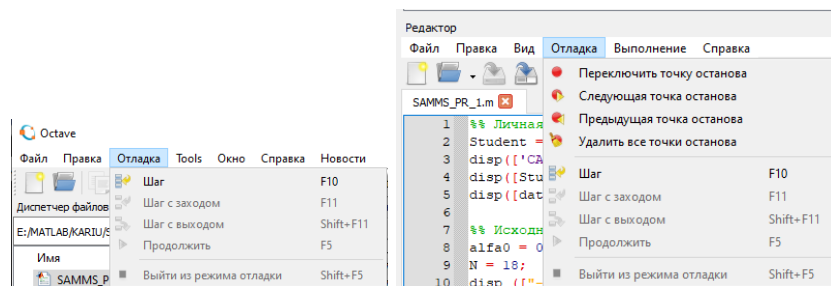
Налагодження сценаріїв. Пункт меню *Налагодження* містить традиційні команди для контрольованого виконання сценаріїв (програм): для покрокового виконання програм (*Крок*, *Крок із заходом*, *Крок з виходом*, *Продовжити*), рис. 1.4 а. Пункт меню *Налагодження* у вкладці **Редактор** містить додатково команди управління точками переривання (breakpoints) (*Переключити точку зупинки*, *Наступна точка зупинки*, *Попередня точка зупинки*, *Видалити всі точки зупинки*), призначеними для переривання виконання ходу сценаріїв у заданих точках, рис. 1.4 б.



а)

б)

Рисунок 1.3 – Меню *Правка* програми GNU Octave:
а) меню головного вікна; б) меню Редактора



а)

б)

Рисунок 1.4 – Меню *Налагодження* програми GNU Octave:
а) меню головного вікна; б) меню Редактора

Tools. Меню *Tools* містить команди керування Профайлером програми (Profile) для підвищення продуктивності розробленого сценарію.

Управління конфігурацією робочого столу. Пункт меню *Вікно* (рис. 1.5) дозволяє при необхідності керувати конфігурацією робочого столу (наявністю вікон можна керувати, ставлячи або забираючи галочки біля вибраних команд меню).

Пункт меню *Вікно* служить для роботи з окремими вікнами програми. Тут відображаються команди, які відповідають вікнам та програмам, які наразі відкриті. Вибір однієї з цих команд дозволяє зробити активним потрібне вікно.

Пункт меню *Довідка* надає доступ до довідкової системи програми GNU Octave .

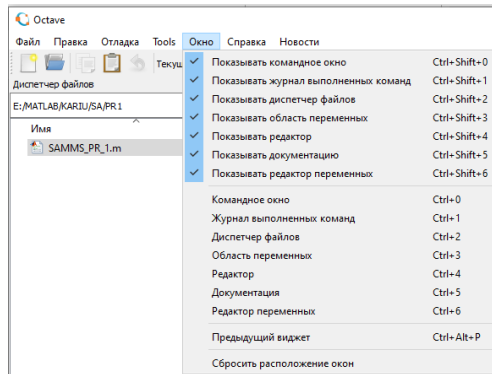


Рисунок 1.5 - Меню *Вікно* програми GNU Octave

Командне вікно (Command Window). Це вікно призначене для введення чисел, змінних, виразів та команд. Тут же виводяться результати роботи та повідомлення про помилки. Про готовність системи до введення свідчить символ `>>`, який знаходиться у текстовому полі командного вікна (у командному рядку). Щоб ввести команду, натисніть клавішу `<Enter>`. Поки вона не натиснута, вираз може бути відредаговано або видалено. Перегляд вмісту вікна можна здійснити за допомогою смуги прокручування, а також клавіші `<PgUp>`, `<PgDown>`, `<Ctrl+Home>`, `<Ctrl+End>`, `<Home>`, `<End>`.

Клавіші керування курсором `↑` та `↓` у командному вікні GNU Octave мають зовсім інше призначення, ніж у текстових редакторах. Використання цих клавіш дозволяє відображати в командному рядку раніше введені з клавіатури команди та вирази для їх повторного використання. Це можливо тому, що всі виконані команди зберігаються у спеціальній області пам'яті. Щоб стерти вміст командного вікна, достатньо набрати та виконати команду `clc`.

Область змінних (Workspace). У GNU OCTAVE всі змінні, використовуються в поточному сеансі роботи, зберігаються в області, що називається Області змінних або робочим простором. Повний перелік змінних відображається у вікні *Область змінних*, де можна бачити список поточних змінних та їх розміри. Ця інформація представлена у вигляді таблиці, що складається з наступних шпальт: *Name* – вказується ім'я змінної; *Class* – тип змінної; *Dimension* – розмірність змінної.

Якщо двічі клацнути по рядку, який відповідає будь-якій змінній, інформація про неї відобразиться у вкладці *Редактор змінних*.

Журнал виконаних команд. (Command History) - тут відображаються дата та час сеансу роботи в GNU OCTAVE, а також міститься перелік команд, введених протягом поточного сеансу. Їх можна знову виконати, двічі клацнувши по команді. Можна виконати кілька команд, що знаходяться в цьому вікні. Для цього потрібно виділити команди за допомогою миші, утримуючи клавішу `<Shift>`, а потім натиснути `<Enter>`. Якщо ж потрібні команди розташовуються не поспіль, їх треба виділити, утримуючи клавішу `<Ctrl>`. Для повторного використання команд також можна скористатися стрілками `↑↓`, поміщаючи ці команди в командний рядок. При натисканні правої клавіші миші на будь-якій команді в цьому вікні відображається контекстне меню, що містить команди копіювання буфер обміну виділеної команди, її видалення, створення *m*-файлу і т.п.

Більш повну та детальну інформацію ви можете отримати в [1].

1.4 Практичне завдання

1. Як номер Вашого варіанта **N** використовуйте Ваш порядковий номер в академічному журналі.

2. Створення матриць та векторів. Наведіть приклади операторів **zeros**, **eye**, **ones** для створення спеціальних матриць розмірністю $[3 \times 3]$. Виконайте ті самі операції для створення вектора-стовпця і вектора-рядки довжиною 3.

3 Створіть квадратну матрицю та вектор стовпець за даними табл.1.1. відповідно до останньої цифри номера Вашого варіанта **N** .

Таблиця 1 Значення елементів матриць та векторів.

Номер завдання	Квадратна матриця A			Вектор стовпець b
1	2			3
1	1,84	2,25	2,53	-6,09
	2,32	2,60	2,82	-6,98
	1,83	2,06	2,24	-5,52
2	2,58	2,93	3,13	-6,66
	1,32	1,55	1,58	-3,58
	2,09	2,25	2,34	-5,01
3	2,18	2,44	2,49	-4,34
	2,17	2,31	2,49	-3,91
	3,15	3,22	3,17	-5,27
4	1,54	1,70	1,62	-1,97
	3,69	3,73	3,59	-3,74
	2,45	2,43	2,25	-2,26
5	1,53	1,61	1,43	-5,13
	2,35	2,31	2,07	-3,69
	3,83	3,73	3,45	-5,98
6	2,36	2,37	2,13	1,48
	2,51	2,40	2,10	1,92
	2,59	2,41	2,06	2,16
7	3,43	3,38	3,09	5,52
	4,17	4,00	3,65	6,93
	4,30	4,10	3,67	7,29
8	3,88	3,78	3,45	10,41
	3,00	2,79	2,39	8,36
	2,67	2,37	1,96	7,62
9	3,40	3,26	2,90	13,05
	2,64	2,39	1,96	10,30
	4,64	4,32	3,85	17,89
0	2,53	2,36	1,93	12,66
	3,95	4,11	3,66	21,97
	2,78	2,43	1,94	13,93

4. Наведіть приклад використання стандартних математичних функцій: квадратний корінь, синус, косинус, тангенс, експонент, десятковий та натуральний логарифм. Як аргумент використовуйте елемент матриці **A(2, 2)** . Перевірте результати на калькуляторі вашого смартфона 😊 ! Наведіть у звіті значення здійснених обчислень.

5. Покажіть застосування операції транспонування до матриці **A** та вектора-стовпця **b** . Наведіть у звіті значення вихідних та транспонованих змінних.

6. Інтерпретуючи матрицю **A** як матрицю коефіцієнтів, а вектор-стовпець **b** – як вектор вільних членів отримайте рішення системи лінійних рівнянь алгебри за допомогою операції лівого поділу матриць. Виконайте перевірку рішення шляхом встановлення рішення у вихідну СЛАУ. Наведіть у звіті значення отриманого рішення та результати перевірки.

7. Виконайте налагоджений сценарій в онлайн-сервісі GNU OCTAVE <https://octave-online.net/>

8. Зробити висновки щодо виконаної роботи.

Контрольні запитання.

1. Перерахуйте основні елементи головного вікна Octave/Matlab та вкажіть їх призначення.

2. Що означає назву програми MATLAB?

3. Яку операцію виконує функція ones?

4. Яку операцію виконує функція zeros?

5. Для вектора-рядка [1 4 8] виконайте транспонування та запишіть результат.

6. Як виконується поелементне множення матриць? Яка умова має виконуватися для матриць для виконання поелементного множення?

7. Наведіть приклад використання функції linspace.

Зміст звіту

1. Титульний лист

2. Цілі, завдання роботи

3. Текст завдання

4. Результати роботи (скриншоти даних, формул, діаграми)

5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №2 ПОБУДУВАННЯ ГРАФІКІВ В OCTAVE /MATLAB

Ціль роботи : отримати необхідні навички для візуалізації експериментальних досліджень та математичного моделювання, маніпулювання елементами графіків.

ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 Загальні відомості про підсистему побудови графіків у MATLAB

MATLAB має виключно потужну систему для побудови різних двовимірних та тривимірних графіків, а також їх налаштування, редагування та форматування. Типи

та підтипи графіків MATLAB дуже різноманітні. Список функцій двовимірної графіки можна одержати командою `help graph2d`, тривимірною -`help graph3d`.

Графіки виводяться у окремих графічних вікнах з допомогою команди виду `figure(n)`, де n – номер графічного вікна. На одному графіку можна побудувати кілька кривих, що відрізняються кольором та типами ліній та точок. Графіки можуть бути скопійовані та вставлені до інших програм: Word, Excel, Power Point і т.д. Для цього використовується команда `Edit/Copy Figure` вікна графіки.

команди, що часто використовуються при побудові графіків:

```
plot(t,y)% Графік безперервної функції y(t)
plot (x1, y1, x2, y2)% Графіки залежностей y1 від x1 та y2 від x1
stem(x, y)% Графік дискретної функції (сигналу) y(x)
stairs (x, y) % Графік у вигляді ступінчастої лінії
loglog(f,Y)% Графік з логарифмічними масштабами по x та y
semilogx(f,Y)% Логарифмічний масштаб x і лінійний y
polar (phi, r) % Графік у полярних координатах
title ('назва')% Висновок заголовка графіка
xlabel ('час') % Мітка по осі x
ylabel ( 'Напруга')% Мітка по осі y
legend ( 'АЧХ системи')% Висновок напису, що пояснює
axis ([xmin, xmax, ymin, ymax]) % Встановлення масштабів по осях x та y
xlim ([xmin, xmax])% Встановлення масштабу по осі x
ylim([ymin, ymax])% Встановлення масштабу по осі y
figure (n)% Встановлює фігуру (вікно) n активною
subplot (r, c, n) % розбиває графічне вікно на r*c підвіконня і
subplot (rcn)% встановлює підокно n як активне.
grid on% до графіка додається сітка
hold on% дозволяє побудувати кілька графіків у вікні
hold off% скасовує hold on для поточного графіка
text% дозволяє розмістити текст на графіку
zoom on/off% увімкнення/вимкнення можливості збільшення
% Фрагментів графіка з використанням
% Лівую та правою кнопок миші
```

Для побудови графіка функції в системі декартової координат використовується команда **plot**. Для побудови двох векторів даних x і y як графічної залежності $y = f(x)$ потрібно використовувати команду **plot(x,y)**.

Виведення графічних даних у новому вікні проводиться командою **figure**. Команда **figure(n)** визначає номер вікна із графічними даними. За допомогою команди **grid** виконується нанесення сітки та графіка. Для виведення додаткових до існуючих графічних даних у поточному вікні виконується команда **hold on**. Підпис координатних осей здійснюється відповідними командами **xlabel('ім'я осі')** та **ylabel('ім'я осі')**. Назва графіка можна присвоїти за допомогою команди **title('ім'я графіка')**. Текстовий напис у сфері побудови графіка виконується шляхом команди **text(x, y, текст написи)**, де x і y – відносні координати написи у сфері графіка.

Побудова графіків у системі Matlab допускається різними кольорами, типами ліній та маркерами точок. Для цього команда побудови графіка може задаватися в наступній розширеній формі: **plot(x, y, '<колір> <тип лінії> <маркер>')**. Для того, щоб задати певний колір графіка, використовуються відповідні літери: c – блакитний; m – рожевий; y – жовтий; r – червоний; g – зелений; b – синій; w – білий; k – чорний. Типи ліній задаються певними символами: - - - безперервна лінія; "--" - Штрихова; ":" - Пунктирна; «-.» - Штрих-пунктирна; "none" - без лінії. Знаки, які можна використовувати для завдання маркерів точок на графіку: +; "o"; "x"; "*".

Досить часто графіки будують за зміни аргументу функції в заданому діапазоні на сітці з постійним кроком.

Для завдання вектора, що містять аргумент досліджуваної функції, служать спеціальні команди, що згадувалися у самостійній роботі №1. Наведемо ці команди ще раз.

Один із методів створення вектора полягає у використанні виразу типу арифметичної прогресії – **[початкове значення: збільшення: кінцеве значення]** .

Наприклад:

```
>> x = [0: 0.2: 1]% Послідовність від 0 до 1 з кроком 0.2
```

```
x =  
0 0.20 0.40 0.60 0.80 1.00
```

```
>>k=0:10; % Створюється вектор 1×11 з елементами 0, 1, 2, ..., 10.
```

Крок змінної може бути негативним

```
>> a = [10: -2.5: 0]  
a =  
10.0000 7.5000 5.0000 2.5000 0
```

Інший метод створення вектора-рядка - команда `linspace (x1, x2, n)` . Ця команда створює n рівновіддалених чисел у діапазоні від x_1 до x_2 , наприклад,

```
>> x1 = linspace (0, 4, 5)
```

```
x1 =  
0 1 2 3 4
```

2.2 Практичне завдання

У процесі виконання цієї роботи буде корисним ознайомитися з довідковими відомостями по командах `plot`, `stem`, `hold`, `grid`, `subplot`, `figure`, `title`, `axis`, `xlabel`, `ylabel`, `legend`, `randn` .

1. Взяти вихідні дані з табл.2.1 відповідно до Вашого порядкового номера в академічному журналі.

№ пп	Функція	a	b	№ пп	Функція	a	b
1	$f(x) = 2x^3 + 3x^2$	1,0	3,2	16	$f(x) = 2x^2 + 3\ln x$	1,0	3,2
2	$f(x) = 3\tan x - 0,1$	1,0	1,8	17	$f(x) = 21x^2 + 14x - 5$	1,0	1,8
3	$f(x) = e^x + 2x$	2,0	4,0	18	$f(x) = x^{0,4} + x^{0,2}$	2,0	4,0
4	$f(x) = \sin x + x^2$	1,0	2,8	19	$f(x) = 0,1^x + e^x$	1,0	2,8
5	$f(x) = x^4 + x^2 + 1$	2,0	4,4	20	$f(x) = \sin x - \sqrt{x}$	2,0	4,4
6	$f(x) = e^{1,5} + x^{1,5}$	2,2	3,8	21	$f(x) = \sqrt{\pi^x + e^x}$	2,2	3,8
7	$f(x) = \log x - x^2$	2,0	4,0	22	$f(x) = e^x + \tan(\pi x)$	2,0	4,0
8	$f(x) = \sin^2 x + \cos^2 x$	1,0	5,0	23	$f(x) = x^2 + 2\ln x$	1,0	5,0
9	$f(x) = 45x^2 - 8x + 16$	1,0	1,8	24	$f(x) = e^x - \cos^2 x$	1,0	1,8
10	$f(x) = \cos^2(\pi x) - 1,5$	1,0	1,2	25	$f(x) = \pi\sqrt{x} + \log x$	1,0	1,2

11	$f(x) = 2\sqrt{x} + x^2$	2,0	3,6	26	$f(x) = \ln x - \sqrt{\tan x}$	2,0	3,6
12	$f(x) = \arcsin x + x^3$	1,8	3,2	27	$f(x) = 2x^2 + 3,5x - 1,8$	1,8	3,2
13	$f(x) = \sin x - e^x$	1,0	5,0	28	$f(x) = e^{1,5} + x^{1,5}$	1,0	2,0
14	$f(x) = x^2 - \pi \cos x$	1,0	1,8	29	$f(x) = \arctan x - e^x$	1,5	2,5
15	$f(x) = 5x^2 + 2x - 17$	2,0	3,2	30	$f(x) = e^x / \sin^2 x$	2,0	4,0

2. Побудувати графік функції **f(x)** у межах від **a** до **b** . Крок побудови графіка вибрати самостійно. Графік необхідно оформити, задавши колір та тип лінії. На графіку мають бути підписані осі та нанесена сітка. У назві діаграми вкажіть своє прізвище та ініціали. На вільному місці у галузі побудови графіка необхідно побудувати легенду графіка та підписати побудовану функцію.

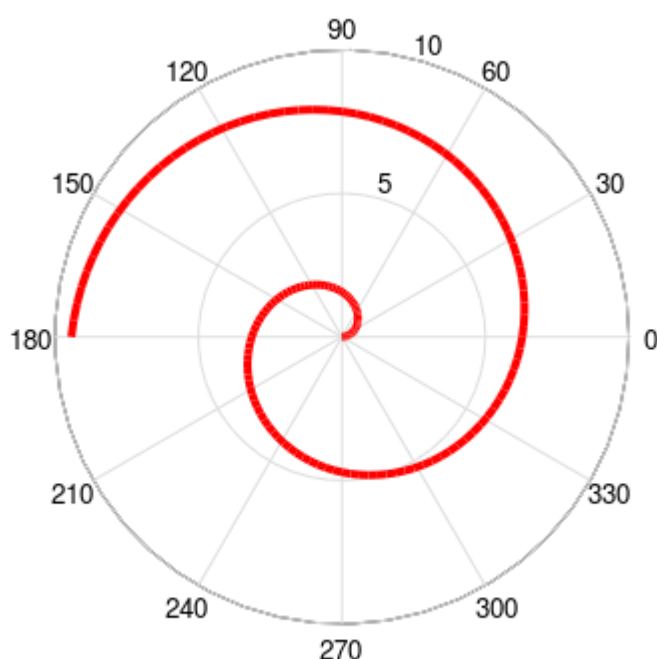
2.3 Графіки у полярній системі координат.

Функція в полярній системі координат визначає залежність відстані між фіксованою точкою (початком координат) і поточною точкою графіка від полярного кута.

приклад. Побудова архімедової спіралі $p((p)-(p))$ (див. малюнок на наступній сторінці).

```
fi=0:pi/50:3*pi;
ro = fi;
h=polar(fi,ro);
set (h, 'linewidth1, 3, 'color1,'r1)
```

Пояснення. Функція `ro` повертає деяке речове число (дескриптор графіка) , використовуючи яке можна встановлювати властивості цього графіка за допомогою команди `set` .



Спіраль Архімеда.

Завдання 2. Побудувати в полярних координатах графік функції діапазоні зміни аргументу $[0; 2\pi]$. Використовуючи команду `set`, вкажіть наведені параметри графіка.

Варіант	Функція	Товщина лінії	Колір
1	$\rho(\varphi) = 1 - e^{\varphi/2}$	1	Синій
2	$\rho(\varphi) = \frac{\varphi}{\varphi^2 + 1}$	2	Червоний
3	$\rho(\varphi) = \cos(3 \cdot \varphi)$	3	Зелений
4	$\rho(\varphi) = 3 + \cos(4 \cdot \varphi)$	4	Фіолетовий
5	$\rho(\varphi) = 1 + \cos(3 \cdot \varphi) + (\sin(3 \cdot \varphi))^2$	1	Синій
6	$\rho(\varphi) = 1 - (\cos(4 \cdot \varphi))^3$	2	Червоний
7	$\rho(\varphi) = 1 - e^{\varphi}$	3	Зелений
8	$\rho(\varphi) = \frac{2 + \varphi}{1 + \varphi}$	4	Фіолетовий
9	$\rho(\varphi) = 2 + \cos(3 \cdot \varphi)$	1	Синій
10	$\rho(\varphi) = 1 - 4 \cdot \cos(3 \cdot \varphi)$	2	Червоний
11	$\rho(\varphi) = 1 + (\cos(3 \cdot \varphi))^2$	3	Зелений
12	$\rho(\varphi) = 1 + (\cos(3 \cdot \varphi))^3$	4	Фіолетовий
13	$\rho(\varphi) = \frac{1 + \varphi}{3 + \varphi}$	1	Синій
14	$\rho(\varphi) = \cos(5 \cdot \varphi)$	2	Червоний

2.4 Тривимірні графіки у декартовій системі координат

Для підготовки даних та побудови графіків-поверхень використовуються команди:

`meshgrid` (х-діапазон, у-діапазон) - повертає двовимірний масив - матрицю $[X, Y]$, яка містить координати вузлових точок прямокутної сітки на площині Oxy ,

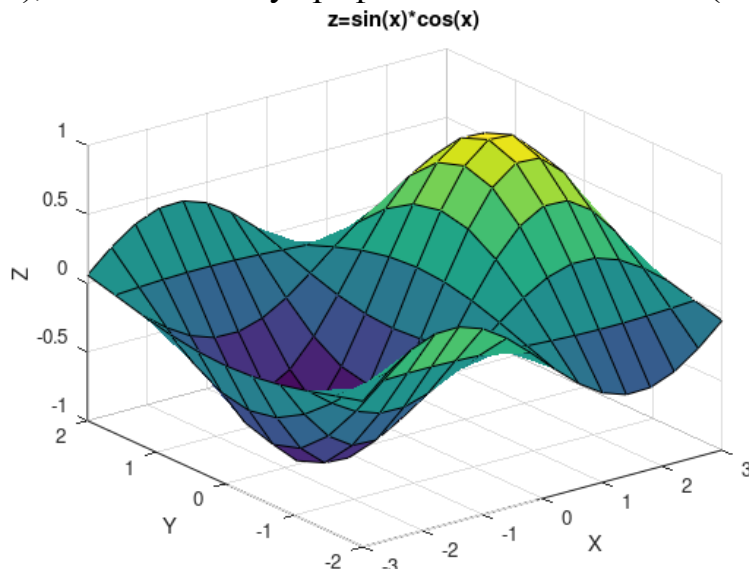
`surf(X, Y, Z)` — будує поверхню функції $Z = f(X, Y)$ з використанням палітри кольорів за замовчуванням.

приклад. Побудувати графік функції двох змінних $z = \sin x * \cos x$ якщо x змінюється від -3 до 3 з кроком 0,4, а y змінюється від -2 до 2 з кроком 0,4.

Рішення

```
[X, Y]=meshgrid (-3:0.4:3,-2:0.4:2);
Z = sin (X) . * cos (Y);
surf (X, Y, Z);
xlabel('X');
ylabel('Y');
zlabel ('Z');
title('z=sin(x)*cos(x)')
```

Зауваження. Команди xlabel (' текст '), ylabel (' текст '), zlabel (' текст ') дозволяють підписати осі координат, однак слід пам'ятати, що Octave може некоректно відображати символи кирилиці. Те саме стосується і команди title (' Текст '), яка виводить у графічне вікно заголовок (назва) графіка.



Побудовану поверхню можна повернути, щоб розглянути її "з усіх боків". Для цього слід вибрати інструмент "Повернути" (найліва кнопка на панелі графічного вікна), при цьому курсор змінює свій вигляд. Після цього потрібно переміщати мишку з лівою натиснутою кнопкою.

Завдання 3. Побудувати графік функції двох змінних відповідно до номера варіанта.

Варіант	Функція	Вихідні дані
1	$z = \frac{3+y}{x^2+1}$	$x_1 = -2$; $x_2 = 8$; $h_x = 0.2$; $y_1 = 0$; $y_2 = 15$; $h_y = 0.3$.
2	$z = \frac{y^2}{x(3+y)}$	$x_1 = 2$; $x_2 = 7$; $h_x = 0.1$; $y_1 = -2$; $y_2 = 8$; $h_y = 0.2$.
3	$z = \frac{x^2+5}{2y-1}$	$x_1 = -5$; $x_2 = 5$; $h_x = 0.2$; $y_1 = 3$; $y_2 = 18$; $h_y = 0.3$.
4	$z = \frac{xy}{x^2+y^2}$	$x_1 = 1$; $x_2 = 26$; $h_x = 0.5$; $y_1 = -4$; $y_2 = 6$; $h_y = 0.2$.
5	$z = \frac{2x-y}{y(x+5)}$	$x_1 = -3$; $x_2 = 12$; $h_x = 0.3$; $y_1 = 2$; $y_2 = 12$; $h_y = 0.2$.
6	$z = \frac{y-7}{5x}$	$x_1 = 1$; $x_2 = 6$; $h_x = 0.1$; $y_1 = -1$; $y_2 = 14$; $h_y = 0.3$.
7	$z = \frac{x+y}{3x+1}$	$x_1 = 1$; $x_2 = 11$; $h_x = 0.2$; $y_1 = -10$; $y_2 = 15$; $h_y = 0.5$.

8	$z = \frac{y^2}{1+x^2}$	$x_1 = -3; x_2 = 12; h_x = 0.3;$ $y_1 = -4; y_2 = 1; h_y = 0.1.$
9	$z = x + \frac{1}{2y}$	$x_1 = -3; x_2 = 17; h_x = 0.4;$ $y_1 = 1; y_2 = 6; h_y = 0.1.$
10	$z = \frac{x}{8+y^2}$	$x_1 = -4; x_2 = 6; h_x = 0.2;$ $y_1 = 0; y_2 = 15; h_y = 0.3.$
11	$z = \frac{y-3}{2x^2}$	$x_1 = 1; x_2 = 16; h_x = 0.3;$ $y_1 = -6; y_2 = 14; h_y = 0.4.$
12	$z = y + \frac{x}{y-1}$	$x_1 = -2; x_2 = 8; h_x = 0.2;$ $y_1 = -5; y_2 = 0; h_y = 0.1.$
13	$z = \frac{y^2+1}{xy}$	$x_1 = 1; x_2 = 21; h_x = 0.4;$ $y_1 = 1; y_2 = 11; h_y = 0.2.$
14	$z = \frac{x-y}{x^2+2}$	$x_1 = -3; x_2 = 2; h_x = 0.1;$ $y_1 = -5; y_2 = 10; h_y = 0.3.$

2.5 Побудова поверхні відгуку та ліній рівня тривимірної функції

Для кожного студента задана функція двох змінних видів:

$$z = a \cdot x^2 + b \cdot y^2 + 0.1 \cdot c \cdot x \cdot y$$

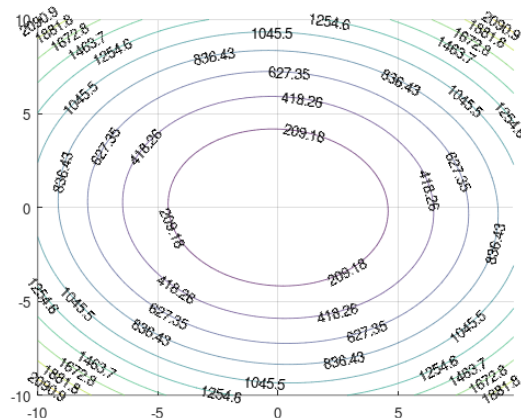
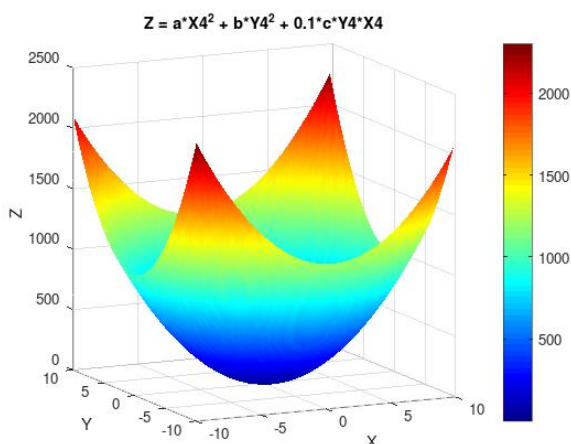
де а – b – с- порядковий номер в алфавіті першої літери прізвища, імені та по батькові студента відповідно.

Самостійно підберіть область побудови цієї функції.

Користуючись технікою, викладеною в розд. 2.4. побудуйте поверхню відгуку цієї функції.

За допомогою функції contour 3(X , Y , Z) побудуйте лінії рівня заданої функції.

Приклад результатів, отриманих під час розділу 2.5 наведено на наступних малюнках.



Контрольні запитання.

1. Яка команда створює нове вікно для побудови графіків?

2. Що робить команда `grid on` ?
3. Що робить команда `hold on` ?
4. За допомогою якої команди будується кругова діаграма?
5. За допомогою якої команди будується стовпчаста діаграма?
6. Якою командою виводиться заголовок діаграми?
7. Які дії виконує команда `meshgrid` ?

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скріншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №3 АНАЛІТИЧНИЙ РОЗРАХУНОК ЕКСТРЕМУМУ ФУНКЦІЇ ОДНІЙ ЗМІННОЇ

Ціль роботи: отримати необхідні навички для реалізації чисельних методів вирішення одновимірних оптимізаційних завдань.

ТЕОРЕТИЧНІ ВІДОМОСТІ

3.1 Загальні відомості про одновимірне оптимізаційне завдання

Постановка задачі.

Для заданої цільової функції $f(x)$ у заданій області визначення $[x_1, x_2]$ необхідно знайти екстремальне (мінімальне або максимальне) значення цільової функції, а також відповідне значення аргументу x .

Завдання пошуку мінімального чи максимального значення не утворюють окремих завдань. З елементарних міркувань ясно, що пошук мінімуму функції $f(x)$ тотожний пошуку максимуму функції $g = -f(x)$. Тому надалі ми розглядатимемо завдання пошуку екстремуму як завдання відшукування саме максимуму цільової функції.

3.2 Аналітичне визначення положення точки екстремуму. Побудова графіка функції. Візуалізація результату.

Для кожного студента задана одновимірна функція однієї змінної форми:

$$y = a \cdot x^2 + b \cdot x + 0.1 \cdot c$$

де a – b – c – порядковий номер в алфавіті першої літери прізвища, імені та по батькові студента відповідно.

Так як функція задана аналітично, ми можемо визначити точне положення екстремуму (максимум або мінімум) цієї функції. Для того, щоб знайти екстремум функції, необхідно знайти першу похідну заданої функції та прирівняти її до нуля.

Знаходимо рівняння першої похідної заданої функції за правилами диференціювання з математики.

$$\frac{dy}{dx} = 2a \cdot x + b$$

Прирівнюючи похідну до нуля, знайдемо положення екстремуму заданої функції по осі Ox :

$$0 = 2a \cdot x + b \rightarrow x_0 = -\frac{b}{2a}$$

Побудуйте графік функції $y = f(x)$ у діапазоні зміни аргументу від $a = x_0 - 5$ до $b = x_0 + 5$.

На отриманому графіку відобразіть точку екстремуму, використовуючи виділення кольором та розміром маркера.

3.3. Наближене обчислення похідних.

Похідної функції називається межа відношення межа збільшення функції до збільшення аргументу, коли збільшення аргументу прагне до нуля:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x}, \Delta y = f(x + \Delta x) - f(x)$$

У численних розрахунках похідну приблизно знаходять, вважаючи збільшення аргументу рівним деякому кінцевому числу, отримуючи наближене рівняння для обчислення похідної:

$$f'(x) = \frac{\Delta y}{\Delta x}.$$

Це співвідношення називається апроксимацією (наближенням) похідною за допомогою кінцевих різниць. Це співвідношення можна використовувати для наближеного обчислення похідної від функції, заданої як аналітичним виразом, і таблично.

Якщо функція задана таблично, величина дорівнює різниці між сусідніми вузлами таблиці в околиці яких обчислюється похідна. При цьому якщо кількість вузлів невелика і вузли розташовані на великій відстані один від одного, формула кінцевих різниць може давати істотну похибку.

Загальний підхід до завдання чисельного диференціювання функції, заданої таблицею значень, ґрунтується на використанні поліномів інтерполяції.

На практиці рідко вдаються до апроксимації функції інтерполяційним поліномом, використовуючи методи глобальної інтерполяції через велику похибку. Як правило, використовують локальну інтерполяцію. При цьому в околиці точки, для якої потрібно обчислити похідну, інтерполують функцію поліномом невисокого ступеня.

У разі інтерполяції функції $f(x)$ поліномом четвертого ступеня $P_4(x)$ можна отримати наступні формули (центральні різницеві співвідношення) для першої похідної:

$$\left. \frac{df}{dx} \right|_{x=x_i} \approx \frac{1}{12h} (y_{i-2} - 8y_{i-1} + 8y_{i+1} - y_{i+2})$$

Джерелами похибки чисельного диференціювання є:

1. Похибка апроксимації, яка при зменшенні кроку h , як правило, зменшується.
2. Похибки обчислення значень функції y_i у вузлах, що зростають із зменшенням кроку h .

3.4 Рівняння дотичної до кривої у заданій точці.

Якщо відомо значення першої похідної y' у певній точці з координатами (x_0, y_0) , то рівняння дотичної можна записати в наступному вигляді:

$$y = y_0 + b \cdot (x - x_0)$$

Необхідно побудувати графіки дотичної лінії у точках $a_1 = x_0 - 3$; $a_2 = x_0$; $a_3 = x_0 + 2$.

3.5 Приклад сценарію вирішення самостійної роботи

```
disp( 'Особиста інформація студента' );
Student = 'Титюк В.К., група ЕПА-78-1' ; %%=====
disp( 'ММУ Самостійна робота №3.' )
disp(Student)
disp(date())

disp( 'Завдання 1' );
disp( 'Вихідні дані варіанта' );
disp( '--- Вихідні дані ---' );
N = 5; %%=====
disp([ 'Номер варіанта за журналом = ' , num2str(N)])
disp( '-----' )

disp( '--- Вихідні дані варіанта ---' );
a = 3; %% Номер першої літери імені %%=====
b = 13; %% Номер першої літери прізвища %%=====
c = 19; %% Номер першої літери по батькові %%=====
disp([ 'Номер першої літери імені ' num2str(a)]);
disp([ 'Номер першої літери прізвища ' num2str( b )]);
disp([ 'Номер першої літери по-батькові ' num2str( c )]);

disp( 'Задана функція : ' );
disp([ 'y(x) = ' num2str(a) '*x^2 + ' num2str(b) '*x + 0.1*' num2str(c)]);

disp( 'Завдання 2' );
disp( 'Аналітичне знаходження екстремуму' )
x0 = -b/(2*a);
disp([ 'Абсциса екстремуму (координата по осі X) = ' num2str(x0)])
y0 = a * x0 ^ 2 + b * x0 + 0.1 * c;
disp([ 'Ордината екстремуму (координата по осі Y) = ' num2str(y0)])
disp( 'Обчислюємо межі побудови графіка заданої функції' )
X1 = x0 - 5;
X2 = x0 + 5;
disp([ 'кордони побудови графіка заданої функції -> від ' num2str(X1) ' до ' num2str(X2)])
M = 101;
disp ([ 'Задаємо кількість точно на графіку N = ' num2str(M)])
disp( 'Створюємо масив аргументів функції x' );
x = linspace(X1, X2, M);
disp( 'Обчислюємо масив значень функції y' );
y = a * x. ^ 2 + b * x + 0.1 * c;
figure(1);
plot(x, y, 'k' , 'linewidth' , 2);
grid on ; % будуємо сітку
hold on ; % Дозволяємо будувати ще графіки
plot(x0, y0, 'or' , 'MarkerFaceColor' , 'red' , 'MarkerSize' , 8)
% Назва діаграми
title ([ Student 'Варіант' num 2 str ( N )]);
% Підпис до осі X
xlabel( 'X \rightarrow' );
% Підпис до осі Y
ylabel( 'Y \rightarrow' );
% Легенда функції
legend ( Student , 'Мінімум' );
```

```

disp ( 'Завдання 3' );
disp( 'Будуємо дотичні до заданої функції у трьох точках:' );
xd1 = x0-3;
xd2 = x0;
xd3 = x0+2;
disp([ 'Перша точка ' num2str(xd1)]);
disp([ 'Друга точка ' num2str(xd2)]);
disp([ 'Третя точка ' num2str(xd3)]);

xabs = abs(x - xd1);
idx1 = find(xabs <= min(xabs));
xabs = abs(x - xd2);
idx2 = find (xabs <= min (xabs));
xabs = abs(x - xd3);
idx3 = find(xabs <= min(xabs));

% Похідна у першій точці
hx = x(idx1) - x(idx1-1);
fb2 = y(idx1-2);
fb1 = y(idx1-1);
ff2 = y(idx1+2);
ff1 = y(idx1+1);
derX1 = (fb2-8 * fb1 + 8 * ff1-ff2) / (12 * hx);
% Похідна у другій точці
hx = x(idx2) - x(idx2-1);
fb2 = y(idx2-2);
fb1 = y(idx2-1);
ff2 = y(idx2+2);
ff1 = y(idx2+1);
derX2 = (fb2-8 * fb1 + 8 * ff1-ff2) / (12 * hx);

% Похідна у третій точці
hx = x(idx3) - x(idx3-1);
fb2 = y(idx3-2);
fb1 = y(idx3-1);
ff2 = y(idx3+2);
ff1 = y(idx3+1);
derX3 = (fb2-8*fb1+8*ff1-ff2)/(12*hx);

disp([ 'Виробна в першій точці ' num2str(derX1)]);
disp([ 'Виробна в другій точці ' num2str(derX2)]);
disp([ 'Вироблена в третій точці ' num2str(derX3)]);

% Розраховуємо масиви аргументів та значень дотичних в окремих точках
x11 = linspace(xd1-2, xd1+2, 21);
x22 = linspace(xd2-2, xd2+2, 21);
x33 = linspace(xd3-2, xd3+2, 21);
% Значення дотичних у першій точці
y10 = a * xd1 ^ 2 + b * xd1 + 0.1 * c;
y11 = y10 + derX1 * (x11-xd1);
% Значення дотичних у другій точці
y20 = a * xd2 ^ 2 + b * xd2 + 0.1 * c;
y22 = y20 + derX2 * (x22-xd2);

% Значення дотичних у третій точці
y30 = a * xd3 ^ 2 + b * xd3 + 0.1 * c;
y33 = y30 + derX3 * (x33-xd3);

figure(2);
plot(x, y, 'k' , 'linewidth' , 2);
grid on ; % будуємо сітку
hold on ; % Дозволяємо будувати ще графіки
plot(x11, y11, '-b' , 'LineWidth' , 0.5)
plot(x22, y22, '-r' , 'LineWidth' , 0.5)
plot(x33, y33, '-m' , 'LineWidth' , 0.5)
plot(xd1, y10, 'or' , 'MarkerFaceColor' , 'blue' , 'MarkerSize' , 8)
plot(xd2, y20, 'or' , 'MarkerFaceColor' , 'red' , 'MarkerSize' , 8)
plot(xd3, y30, 'or' , 'MarkerFaceColor' , 'magenta' , 'MarkerSize' , 8)

```

```

% Назва діаграми
title ([ Student 'Варіант' num 2 str ( N )]);
% Підпис до осі X
xlabel( 'X \rightarrow' );
% Підпис до осі Y
ylabel( 'Y \rightarrow' );
% Легенда функції
legend(Student, 'Дотик 1' , 'Дотик 2' , 'Дотик 3' );

```

3.6 Приклад результатів самостійної роботи

Приклад протоколу роботи сценарію у Командному вікні.

Особиста інформація студента

ММУ Самостійна робота №3.

Титюк В.К., група ЕПА-78-1

01-Oct-2023

Завдання 1

Вихідні дані варіанта

--- Вихідні дані ---

Номер варіанта за журналом = 5

--- Вихідні дані варіанта ---

Номер першої літери імені 3

Номер першої літери прізвища 3

Номер першої літери по-батькові 3

Задана функція:

$y(x) = 3x^2 + 13x + 0.1 \cdot 19$

Завдання 2

Аналітичне знаходження екстремуму

Абсцис екстремуму (координата по осі X) = -2.1667

Ордината екстремуму (координата з осі Y) = -12.1833

Обчислюємо межі побудови графіка заданої функції

межі побудови графіка заданої функції -> від -7.1667 до 2.8333

Задаємо кількість точну на графіку N = 101

Створюємо масив аргументів функції x

Обчислюємо масив значень функції y

Завдання 3

Будуємо дотичні до заданої функції у трьох точках:

Перша точка -5.1667

Друга точка -2.1667

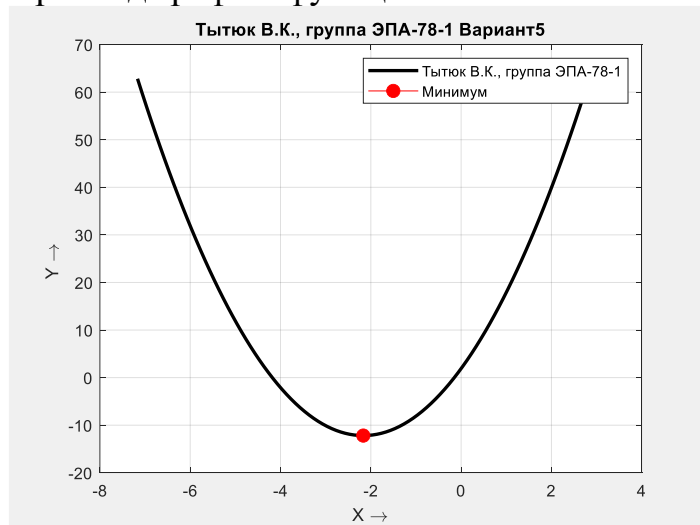
Третя точка -0.16667

Похідна у першій точці -18

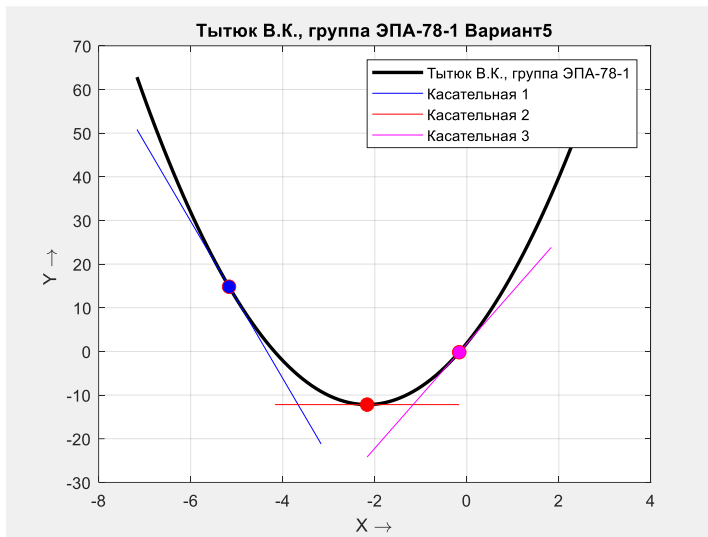
Похідна у другій точці -1.4803e-15

Похідна у третій точці 12

Приклад графіка функції та аналітично обчисленим мінімумом.



Приклад графіка функції та графіків дотичних у трьох точках.



Контрольні запитання.

1. Дайте визначення першої похідної функції.
2. Який загальний підхід до задачі чисельного диференціювання функції, заданої таблицею значень?
3. Чому дорівнює значення першої похідної функції у точці її мінімуму?
4. Чому дорівнює значення першої похідної функції у точці її максимуму?

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скриншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №4 КІЛЬКІСНІ МЕТОДИ ПОШУКУ ЕКСТРЕМУМУ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ. МЕТОД ЗВОРОТНОГО ЗМІННОГО КРОКУ

Ціль роботи : вивчити алгоритми та отримати необхідні навички для реалізації розв'язання одновимірних оптимізаційних задач методом зворотного змінного кроку.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Хоча практично завдання, у яких критерій заданий функцією однієї змінної, зустрічаються досить рідко, аналіз таких завдань займає центральне місце у оптимізаційних дослідженнях. Це пояснюється тим, що багато одновимірних методів досить зрозумілі, легко можуть бути проілюстровані графічно, що дозволяє глибше зрозуміти сутність завдань оптимізації та сприяє набуттю навичок їх вирішення.

Постановка задачі одновимірної оптимізації полягає у знаходженні точки x^* , в якій цільова функція $f(x^*)$ набуває мінімального (максимального) значення. Функція $f(x)$ має локальний мінімум у точці x^* , якщо при $\varepsilon > 0$ існує околиця $[x - \varepsilon; x + \varepsilon]$, така, що для всіх значень x в околиці $f(x)$ більше $f(x^*)$.

Класифікація чисельних методів. Для визначення екстремуму функції однієї змінної можна застосовувати пошукові методи нульового порядку (потрібне обчислення лише значень функції оптимізації) та методи першого та старших порядків, що вимагають обчислення похідних відповідних порядків. Серед методів пошуку виділяють методи послідовного скорочення інтервалу (сканування, локалізації оптимуму, половинного поділу, золотого перерізу, Фібоначчі) та методи точкового оцінювання (Пауелла, квадратичної апроксимації, зворотного змінного кроку).

Метод зворотного змінного кроку належить до групи методів точкового оцінювання.

Припускає завдання початкової координати x_1 і збільшення Δx . Сутність методу ось у чому. Розраховується координата $x_{k+1} = x_k + \Delta x$. Якщо $f(x_{k+1}) < f(x_k)$, то крок вважається "вдалим" і його значення збільшується $\Delta x = \alpha \Delta x$, $\alpha > 1$. Якщо $f(x_{k+1}) \geq f(x_k)$, то крок вважається "невдалим". Тому напрям змінюється на протилежне, а значення кроку зменшується на величину $\beta < 1$ і дорівнюватиме $\Delta x = -\beta \Delta x$. Потім перевіряється умова закінчення $|x_{k+1} - x_k| \leq \varepsilon$ та пошук триває.

Алгоритм методу зворотного змінного кроку.

Початковий етап. Вибрати початкову точку x_1 та приріст Δx . Визначити коефіцієнти деформації кроку $\alpha > 1$, $0 < \beta < 1$ і точність пошуку $\varepsilon > 0$. Покласти $k = 1$ і перейти до основного етапу.

Основний етап. Крок 1 Якщо $\Delta x < \varepsilon$ то зупинитися; точкою мінімуму є точка x_k , інакше вирахувати $x_{k+1} = x_k + \Delta x$ і перейти до кроку 2.

Крок 2 Якщо $f(x_k) > f(x_{k+1})$, то $\Delta x = \alpha \Delta x$. В іншому випадку покласти $\Delta x = -\beta \Delta x$. Замінити $k = k + 1$ та перейти до кроку 1.

Приклад розрахунку екстремуму функції шляхом зворотного змінного кроку.

Постановка задачі. Розрахувати мінімум функції $f(x) = x^2 - 10x + 3$ з точністю $\varepsilon = 0,1$. Початкову точку прийняти рівною $x_1 = -1$.

Вибираємо такі параметри методу: початковий крок - $\Delta x = 1$, коефіцієнти розтягування та стиснення кроку - $\alpha = 1,5$; $\beta = 0,4$. Результати розрахунку представлені у таблиці 4.1.

Таблиця 4.1

Розрахунок мінімуму функції $f(x) = x^2 - 10x + 3$ шляхом зворотного змінного кроку.

№	x_k	Δx	$f(x_k)$	Критерій
1	-1.000	1.000	14.000	не досягнуто
2	0.000	1.500	3.000	не досягнуто
3	1.500	2.250	-9.750	не досягнуто
4	3.750	3.375	-20.438	не досягнуто
5	7.125	-1.350	-17.484	не досягнуто
6	5.775	-2.025	-21.399	не досягнуто
7	3.750	0.810	-20.438	не досягнуто

8	4.560	1.215	-21.806	не досягнуто
9	5.775	-0.486	-21.399	не досягнуто
10	5.289	-0.729	-21.916	не досягнуто
11	4.560	0.292	-21.806	не досягнуто
12	4.852	0.437	-21.978	не досягнуто
13	5.289	-0.175	-21.916	не досягнуто
14	5.114	-0.262	-21.987	не досягнуто
15	4.852	0.105	-21.978	не досягнуто
16	4.957	0.157	-21.998	не досягнуто
17	5.114	-0.063	-21.987	досягнуто

Таким чином, в результаті реалізації методу зворотного змінного кроку за шістнадцять ітерацій визначено мінімальну точку $x^* = 5,114$ з точністю 0,063.

4.1. Практичне завдання

Для кожного студента задано одновимірну функцію виду:

$$y = a \cdot x^2 + b \cdot x + 0.1 \cdot c$$

де а, b, с- порядковий номер в алфавіті першої літери прізвища, імені та по батькові студента відповідно.

Завдання 1. Друкувати особисту інформацію студента. Визначити значення коефіцієнтів а, b, с заданої одновимірної функції, вивести протокол роботи їх значення та подання функції у формі рядка.

Завдання 2. Вибрати початкове наближення екстремуму x^* , початкове значення кроку h , значення коефіцієнтів α_i β надрукувати їх значення в протокол роботи. Побудувати графік цільової функції, показати положення початкової точки пошуку та точного значення мінімуму цільової функції.

Як початкове наближення екстремуму задайте значення

$$x^* = x_0 - \max(a, b, c)$$

де x_0 - точне значення положення екстремуму на осі X, обчислене за формулою, наведеною у самостійній роботі №3

Завдання 3. Виконати пошук екстремуму шляхом зворотного змінного кроку. Виводьте результати розрахунків протокол роботи на кожному кроці розрахунків.

Завдання 4. Побудувати графік зміни величини кроку у процесі пошуку екстремуму.

4.2 Приклад сценарію вирішення самостійної роботи

```
disp('Особиста інформація студента');
Student = 'Титюк В.К., група ЕПА-78-1'; %%=====
disp('ММУ Самостійна робота №3.')
disp(Student)
disp(date())

disp(' Завдання 1');
disp('Вихідні дані варіанта');
disp('--- Вихідні дані ---');
N = 5; %%=====
disp(['Номер варіанта за журналом = ', num2str(N)])
disp('-----')

disp('--- Вихідні дані варіанта ---');
a = 3; %% Номер першої літери імені %%=====
b = 13; %% Номер першої літери прізвища %%=====
c = 19; %% Номер першої літери по батькові %%=====
disp(['Номер першої літери імені ' num2str(a)]);
disp(['Номер першої літери прізвища ' num2str(b)]);
```

```

disp(['Номер першої літери по батькові' num2str(c)]);

disp(' Задана функція :');
disp(['y(x) = ' num2str(a) ' *x^2 + ' num2str(b) ' *x + 0.1*' num2str(c)]);

disp('Завдання 2');
% Аналітичне знаходження екстремуму
x0 = -b/(2*a);
y0 = a * x0 ^ 2 + b * x0 + 0.1 * c;
disp(['Точне значення координати екстремуму на осі X = ' num2str(x0)])
% Задаємо Початкове наближення екстремуму x* до змінної xstart
xstart = x0 - max([abc]);
ystart = a*xstart^2 + b*xstart + 0.1*c;
disp(['Початкове наближення екстремуму x* на осі X = ' num2str(xstart)])
% Початкове значення кроку h
h = min ([abc]);
disp(['Початкове значення кроку пошуку h = ' num2str(h)])
% значення коефіцієнтів alfa та beta
alfa = 1;
beta = 0.7;
disp(['Значення коефіцієнтів зміни кроку пошуку alfa = ' num2str(alfa)' . beta = ' num2str(beta)])
% Точність розрахунку
eps = 0.1;
disp(['Точність розрахунку eps = ' num2str(eps)])

% Візуалізація початкового стану пошуку
X1 = xstart-2;
X2 = x0 + max([abc])+2;
disp(['кордони побудови графіка заданої функції -> від ' num2str(X1)' до ' num2str(X2)'])
M = 101;
x = linspace(X1, X2, M);
y = a * x. ^ 2 + b * x + 0.1 * c;
figure(1);
plot(x, y, 'k', 'linewidth', 2);
grid on; % будуємо сітку
hold on; % Дозволяємо будувати ще графіки
plot(x0, y0, 'ob', 'MarkerFaceColor', 'blue', 'MarkerSize', 8)
plot(xstart, ystart, 'or', 'MarkerFaceColor', 'red', 'MarkerSize', 8)
% Назва діаграми
title ([ Student ' Варіант = ' num 2 str ( N )]);
% Підпис до осі X
xlabel('X \rightarrow');
% Підпис до осі Y
ylabel('Y \rightarrow');
% Легенда функції
legend ( Student , 'Мінімум', 'Старт');

disp ('Завдання 3');
% Лічильник ітерацій
iter = 0;

X0 = xstart;
Y0 = ystart;
Sol = [];

while (abs(h)>eps)
X1 = X0 + h;
Y1 = a * X1 ^ 2 + b * X1 + 0.1 * c;

disp([num2str(iter) '|' num2str(X0) '|' num2str(h) '|' num2str(Y0) '|' ' Не досягнуть '])
if (Y1 > Y0)
h = -beta * h;
else
h = alfa * h;
end
Sol = [Sol; [Y0-Y1, h]];
X0 = X1; Y0 = Y1;
iter = iter + 1;
if iter > 150
disp('Перевищено максимальну кількість ітерацій')
end

```

```

        break;
end
end

if iter <= 150
disp([num2str(iter) '|' num2str(X0) '|' num2str(h) '|' num2str(Y0) '|' ' Досягнуто '])
end

disp(' Завдання 4);
figure(2);
plot(Sol(:,1)/100, 'LineWidth', 2); grid on;hold on;
plot(Sol(:,2), '-r', 'LineWidth', 2);
xlabel('Ітерація');
ylabel('0.01*Відхилення, Крок')

```

4.3 Приклад результатів практичної роботи

Приклад протоколу роботи сценарію у Командному вікні.

Особиста інформація студента

ММУ Самостійна робота №3.

Титюк В.К., група ЕПА-78-1

07- Ост -2023

Завдання 1

Вихідні дані варіанта

--- Вихідні дані ---

Номер варіанта за журналом = 5

--- Вихідні дані варіанта ---

Номер першої літери імені 3

Номер першої літери прізвища 13

Номер першої літери по-батькові 19

Задана функція:

$y(x) = 3 \cdot x^2 + 13 \cdot x + 0.1 \cdot 19$

Завдання 2

Точне значення координати екстремуму на осі X = -2.1667

Початкове наближення екстремуму x * на осі X = -21.1667

Початкове значення кроку пошуку h = 3

Значення коефіцієнтів зміни кроку пошуку $\alpha = 1$. beta = 0.7

Точність розрахунку eps = 0.1

межі побудови графіка заданої функції -> від -23.1667 до 18.8333

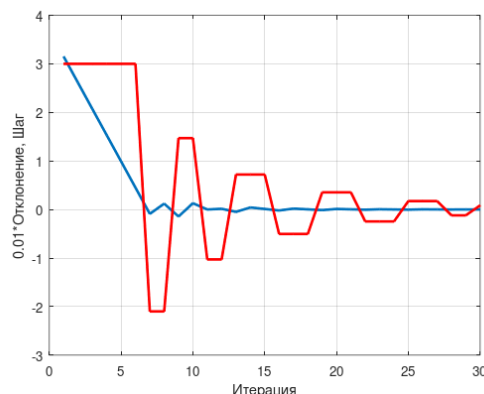
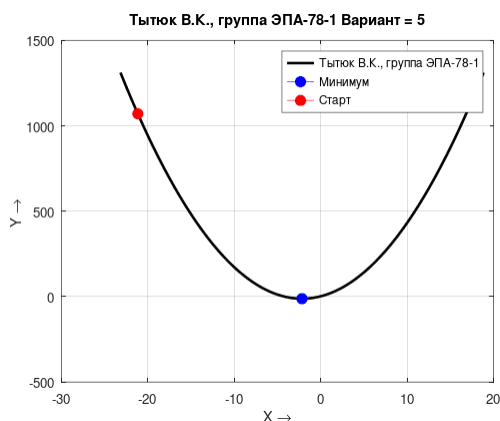
Завдання 3

```

0|-21.1667|3|1070.8167|Не досягнуто
1|-18.1667|3|755.8167|Не досягнуто
2|-15.1667|3|494.8167|Не досягнуто
3|-12.1667|3|287.8167|Не досягнуто
4|-9.1667|3|134.8167|Не досягнуто
5|-6.1667|3|35.8167|Не досягнуто
6|-3.1667|3|-9.1833|Не досягнуто
7|-0.1667|-2.1|-0.18333|Не досягнуто
8|-2.2667|-2.1|-12.1533|Не досягнуто
9|-4.3667|1.47|2.3367|Не досягнуто
10|-2.8967|1.47|-10.5846|Не досягнуто
11|-1.4267|-1.029|-10.5405|Не досягнуто
12|-2.4557|-1.029|-11.9328|Не досягнуто
13|-3.4847|0.7203|-6.972|Не досягнуто
14|-2.7644|0.7203|-11.1116|Не досягнуто
15|-2.0441|0.7203|-12.1382|Не досягнуто
16|-1.3238|-0.50421|-10.0519|Не досягнуто
17|-1.828|-0.50421|-11.8392|Не досягнуто
18|-2.3322|-0.50421|-12.1011|Не досягнуто
19|-2.8364|0.35295|-10.8377|Не досягнуто
20|-2.4834|0.35295|-11.8823|Не досягнуто
21|-2.1305|0.35295|-12.1794|Не досягнуто
22|-1.7776|-0.24706|-11.7291|Не досягнуто
23|-2.0246|-0.24706|-12.1228|Не досягнуто
24|-2.2717|-0.24706|-12.1502|Не досягнуто
25|-2.5187|0.17294|-11.8115|Не досягнуто
26|-2.3458|0.17294|-12.0871|Не досягнуто

```

27 | -2.1729 | 0.17294 | -12.1832 | Не досягнуто
 28 | -1.9999 | -0.12106 | -12.0999 | Не досягнуто
 29 | -2.121 | -0.12106 | -12.1771 | Не досягнуто
 30 | -2.242 | 0.084743 | -12.1663 | Досягнуто



Контрольні запитання.

1. Що означає слово "оптимальний"?
2. Чому неприпустимі вирази «найоптимальніший», «оптимальніший», «оптимальніший»?
3. Дайте визначення цільової функції.
4. У чому полягає завдання оптимізації?
5. Дайте визначення розв'язання задачі оптимізації.
6. Дайте визначення мінімуму та максимуму функції однієї змінної.
7. Які функції називаються строго спадаючими і строго зростаючими?
8. Дайте визначення унімодалльної функції.

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скріншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №5 КІЛЬКІСНІ МЕТОДИ ОДНОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД ПОЛОВИННОГО ДІЛЕННЯ.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Завдання пошуку екстремуму функції однієї змінної виникає при оптимізації цільової функції, яка залежить від однієї скалярної змінної. Такі завдання входять складовою в багато ітераційних методів вирішення задач багатовимірної оптимізації.

Наприклад, чисельні методи пошуку екстремуму мають особливість у тому, що їх застосування не дозволяє визначити точне значення координат, при якому досягається екстремум функції. У цьому випадку визначають інтервал невизначеності, в якому локалізується екстремум функції.

Розмір цього інтервалу – Δ , визначається з вимог точності результату рішення при постановці завдання (швидкість, точність тощо.). Таким чином, чисельне вирішення задачі пошуку екстремуму функції зводиться до зменшення інтервалу невизначеності від вихідного до Δ .

Багато методів оптимізації застосовні лише тоді, коли цільова функція $f(x)$ є унімодальною, тобто будь-який локальний мінімум ЦФ одночасно є і глобальним.

Безліч функцій, унімодальних на відрізку $[a, b]$, позначимо $Q[a, b]$.

Для перевірки унімодальності функції $f(x)$ на практиці використовують такі критерії:

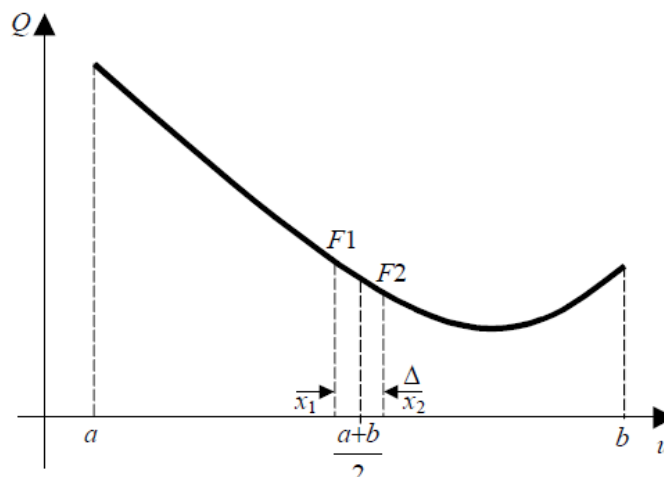
– якщо функція $f(x)$ диференційована на відрізку $[a, b]$ і похідна $f'(x)$ не зменшується на цьому відрізку, то $f(x)$ унімодальна на відрізку $Q[a, b]$;

– якщо функція $f(x)$ двічі диференційована на відрізку $[a, b]$ і друга похідна $f''(x) \geq 0$ при $x \in [a, b]$, то $f(x)$ унімодальна на відрізку $Q[a, b]$.

Якщо ці критерії не виконуються, функція $f(x)$ є мультимодальною або багатоекстремальною.

Природним і найпоширенішим методом пошуку екстремуму функції однієї змінної є метод послідовного поділу відрізка навпіл. Цей метод був відомий ще в Стародавній Греції як метод дихотомії.

Нехай потрібно визначити мінімум унімодальної функції $Q(u)$ на відрізку $[a, b]$ з точністю Δ :

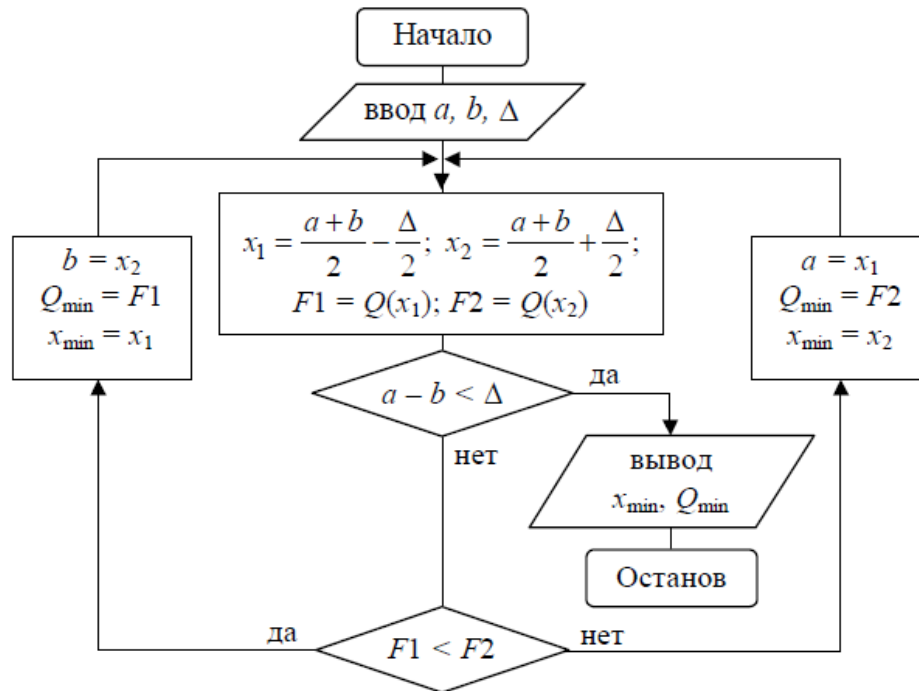


Малюнок 5.1 Геометрична інтерпретація методу половинного поділу одновимірної оптимізації

Відрізок $[a, b]$ ділиться навпіл і обчислюються значення функції $F1 = Q(x_1)$ і $F2 = Q(x_2)$ у точках

$$x_{1,2} = \frac{a + b \pm \Delta}{2}$$

На основі аналізу значень функції $F1$ і $F2$ вдвічі зменшується інтервал невизначеності і процес повторюється доки виконується умова $(b - a) > \Delta$. Блок-схема цього наведено нижче малюнку.



Малюнок 5.2 Блок схема методу половинного поділу одновимірної оптимізації

Оцінка швидкості збіжності методу половинного поділу включає аналіз того, як швидко метод наближається до оптимального рішення. Швидкість збіжності може залежати від кількох факторів:

1. Вихідний інтервал: Ширина початкового інтервалу істотно впливає швидкість збіжності. Чим менший початковий інтервал, тим швидше метод може зійтися до оптимуму. Початковий інтервал можна вибирати так, щоб він містив оптимум.
2. Вибір критерію зупинки: Швидкість збіжності залежить також від того, який критерій зупинки вибраний. Наприклад, метод може завершитись при досягненні заданої точності або після певної кількості ітерацій.
3. Властивості функції: Швидкість збіжності може сильно варіювати в залежності від форми функції, що оптимізується. Гладкі та монотонні функції можуть сходитися швидше, ніж функції з піками, западинами та нестійкими точками.
4. Точність апроксимації: Точність апроксимації функції кожної ітерації також впливає швидкість збіжності. Точніші апроксимації можуть прискорити збіжність, але можуть бути обчислювально більш витратними.
5. Вибір початкової точки: Початкова точка, з якої починається метод половинного поділу, може впливати на швидкість збіжності. Вибір невідповідної початкової точки може сповільнити збіжність.
6. Реалізація алгоритму: Ефективність реалізації алгоритму та вибір методів обчислень (наприклад, яким чином обчислюються значення функції) можуть вплинути на швидкість збіжності.

Зазвичай швидкість збіжності методу половинного поділу оцінюється термінами кількості ітерацій, необхідні досягнення заданої точності. Важливо пам'ятати, що

метод половинного поділу сходить лінійно, тобто розмір інтервалу зменшується вдвічі на кожній ітерації.

5.1. Практичне завдання

Для кожного студента задано одновимірну функцію виду:

$$y = a \cdot x^2 + b \cdot x + 0.2 \cdot c$$

де а, b, с- порядковий номер в алфавіті першої літери прізвища, імені та по батькові студента відповідно.

Завдання 1. Друкувати особисту інформацію студента. Визначити значення коефіцієнтів а, b, с заданої одновимірної функції, вивести протокол роботи їх значення та подання функції у формі рядка.

Завдання 2. Вибрати інтервал унімодальності функції [X₁ , X₂] , і надрукувати значення протоколу роботи. Побудувати графік цільової функції, показати положення кінців інтервалу унімодальності та точного значення мінімуму цільової функції.

Як початкове наближення екстремуму задайте значення

$$X_1 = x_0 - \max(a, b, c) \\ X_2 = x_0 + 2 * \max(a, b, c)$$

де x₀ - точне значення положення екстремуму на осі X, обчислене за формулою, наведеною у самостійній роботі №3

Завдання 3. Виконати пошук екстремуму шляхом половинного поділу. Виводьте результати розрахунків протокол роботи на кожному кроці розрахунків.

Завдання 4. Розрахувати відносну похибку визначення мінімального значення функції.

Побудувати графік зміни величини інтервалу відділення екстремуму у процесі пошуку.

5.2 Приклад сценарію вирішення самостійної роботи

```
disp('Особиста інформація студента');
Student = 'Титюк В.К., група ЕПА-78-1'; %%=====
disp('ММУ Самостійна робота №4. Пошук мінімуму методом половинного поділу')
disp(Student)
disp(date())

disp(' Завдання 1');
disp('Вихідні дані варіанта');
disp('--- Вихідні дані ---');
N = 5; %%=====
disp(['Номер варіанта за журналом = ', num2str(N)])
disp('-----')

disp('--- Вихідні дані варіанта ---');
a = 3; %% Номер першої літери імені %%=====
b = 13; %% Номер першої літери прізвища %%=====
c = 19; %% Номер першої літери по батькові %%=====
disp(['Номер першої літери імені ' num2str(a)]);
disp(['Номер першої літери прізвища ' num2str(b)]);
disp(['Номер першої літери по батькові ' num2str(c)]);

disp(' Задана функція :');
disp(['y(x) = ' num2str(a) '*x^2 + ' num2str(b) '*x + 0.2*' num2str(c)]);

disp('Завдання 2');
% Аналітичне знаходження екстремуму
x0 = -b/(2*a);
y0 = a * x0 ^ 2 + b * x0 + 0.2 * c;
```

```

disp(['Точне значення координати екстремуму на осі X = 'num2str(x0)])
% Задаємо Початкове наближення екстремуму x* до змінної xstart
xstart = x0 - max([abc]);
ystart = a*xstart^2 + b*xstart + 0.2*c;
xend = x0 + 2 * max ([abc]);
yend = a * xend ^ 2 + b * xend + 0.2 * c;
disp(['Початковий інтервал відділення екстремуму = 'num2str(xstart)' до 'num2str(xend)])
% Точність розрахунку
eps = 0.15;
disp(['Точність розрахунку eps = 'num2str(eps)])

% Візуалізація початкового стану пошуку
X1 = xend + 2;
X2 = xstart - 2;
disp(['кордони побудови графіка заданої функції -> від 'num2str(X1)' до 'num2str(X2)'])
M = 101;
x = linspace(X1, X2, M);
y = a * x. ^ 2 + b * x + 0.2 * c;
figure(1);
plot(x, y, 'k', 'linewidth', 2);
grid on; % будуємо сітку
hold on; % Дозволяємо будувати ще графіки
plot(x0, y0, 'ob', 'MarkerFaceColor', 'blue', 'MarkerSize', 8)
plot(xstart, ystart, 'or', 'MarkerFaceColor', 'red', 'MarkerSize', 8)
plot(xend, yend, 'om', 'MarkerFaceColor', 'magenta', 'MarkerSize', 8)
% Назва діаграми
title ([ Student ' Варіант = ' num 2 str ( N )]);
% Підпис до осі X
xlabel('X \rightarrow');
% Підпис до осі Y
ylabel('Y \rightarrow');
% Легенда функції
legend(Student, 'Min', 'X1', 'X2', 'Location', 'NorthWest');

disp('Завдання 3');
% Лічильник ітерацій
iter = 0;

A = xstart;
B = xend;
Sol = [];

while (abs(A - B) >= 1.1*eps)
X1 = (A + B - eps)/2;
X2 = (A + B + eps) / 2;
F1 = a * X1 ^ 2 + b * X1 + 0.2 * c;
F2 = a * X2 ^ 2 + b * X2 + 0.2 * c;

disp([num2str(iter) ' | h= ' num2str(abs(AB)) ' | досягнуть '])
if (F1 < F2)
B = X2;
Qmin = F1;
Xmin = X1;
else
A = X1;
Qmin = F2;
Xmin = X2;
end
Sol = [Sol; [iter, Xmin, Qmin]];
iter = iter + 1;
if iter > 150
disp('Перевищено максимальну кількість ітерацій')
break;
end
end

if iter <= 150
disp([num2str(iter) ' | h=' num2str(abs(AB)) ' | Xmin=' num2str(Xmin) ' | Ymin=' num2str(Qmin) ' | '
Досягнуто '])
end

```



```

disp(' Завдання 4');
disp([' Відносна похибка обчислення мінімуму = ' num2str(abs(y0-Qmin)/abs(y0))])
figure(2);
plot(Sol(:,1), Sol(:,2), 'LineWidth', 2); grid on;hold on;
plot(Sol(:,1), Sol(:,3), 'LineWidth', 2); grid on;hold on;
xlabel(' Ітерація ');
ylabel('Xmin, Ymin')

```

5.3 Приклад результатів самостійної роботи

Приклад протоколу роботи сценарію у Командному вікні.

Особиста інформація студента

ММУ Самостійна робота №4. Пошук мінімуму методом половинного поділу

Титюк В.К., група ЕПА-78-1

21-Окт-2023

Завдання 1

Вихідні дані варіанта

--- Вихідні дані ---

Номер варіанта за журналом = 5

--- Вихідні дані варіанта ---

Номер першої літери імені 3

Номер першої літери прізвища 13

Номер першої літери по-батькові 19

Задана функція:

$y(x) = 3x^2 + 13x + 0.2 \cdot 19$

Завдання 2

Точне значення координати екстремуму на осі X = -2.1667

Початковий інтервал відділення екстремуму = -21.1667 до 35.8333

Точність розрахунку eps = 0.15

межі побудови графіка заданої функції -> від 37.8333 до -23.1667

Завдання 3

0 | h = 57 | Не досягнуто

1 | h = 28.575 | Не досягнуто

2 | h = 14.3625 | Не досягнуто

3 | h = 7.2563 | Не досягнуто

4 | h = 3.7031 | Не досягнуто

5 | h = 1.9266 | Не досягнуто

6 | h = 1.0383 | Не досягнуто

7 | h = 0.59414 | Не досягнуто

8 | h = 0.37207 | Не досягнуто

9 | h = 0.26104 | Не досягнуто

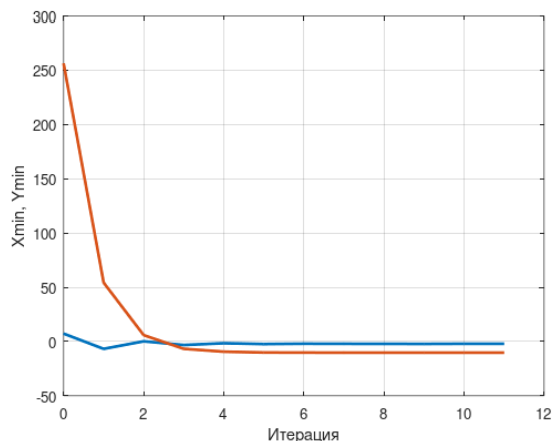
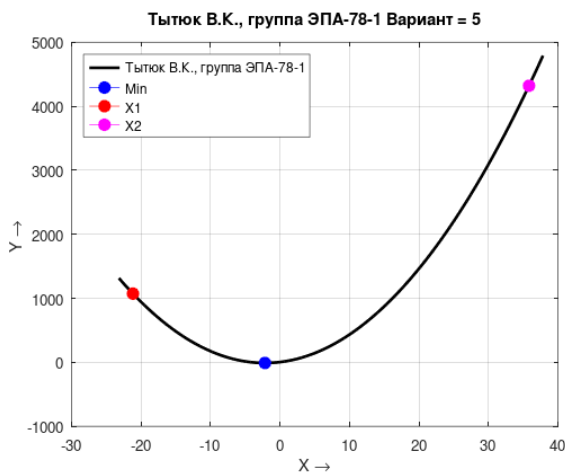
10 | h = 0.20552 | Не досягнуто

11 | h = 0.17776 | Не досягнуто

12 | h = 0.16388 | Xmin = -2.0991 | Ymin=-10.2696|Досягнуто

Завдання 4

Відносна похибка обчислення мінімуму = 0.0013337



Контрольні запитання.

1. Дайте визначення унімо дальньої функції.
2. Дайте визначення абсолютної похибки наближених обчислень.
3. Дайте визначення відносної похибки наближених обчислень.
4. Що означає термін «дихотомія»?
5. Як працює метод половинного поділу? Опишіть основні кроки.
6. Як вибрати початковий інтервал методу половинного поділу, щоб гарантувати збіжність до оптимуму?
7. Як можна оцінити швидкість збіжності методу половинного поділу та які фактори можуть вплинути на цю швидкість?

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скриншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №6 МЕТОДИ БЕЗУМОВНОЇ БАГАТОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД ХУКА-ДЖИВСА (HOOKE – JEEVES).

Мета роботи: Ознайомлення із загальними характеристиками чисельних методів вирішення багатовимірних оптимізаційних завдань без обмежень: загальна формула, збіжність, умови зупинки. Дослідження прямих методів пошуку, що використовують для формування рішення лише відомості про значення функції.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Як правило, досліджуване завдання є досить громіздким. У цьому випадку використання аналітичних підходів до її вирішення не завжди є доцільним, а іноді просто неможливим. Тоді для її вирішення застосовується апарат чисельних методів.

Будь-який чисельний алгоритм розв'язання задачі оптимізації ґрунтується на точному чи наближеному обчисленні значень цільової функції, функцій, що задають допустиму множину (для завдань умовної оптимізації) та їх похідних. З отриманої інформації будується наближення до розв'язання завдання.

Для кожної конкретної задачі питання про те, які характеристики необхідно вибрати для обчислення вирішується залежно від властивостей функції, що мінімізується, обмежень і наявних можливостей зі зберігання та обробки інформації. Так, для не диференційованої функції не можна скористатися алгоритмом, який передбачає можливість обчислення похідної у точці градієнта функції.

Алгоритми, які використовують інформацію лише про значення мінімізованої функції, називаються алгоритмами нульового порядку; алгоритми, що використовують також інформацію про значення перших похідних – алгоритмами першого порядку тощо.

Робота алгоритмів, як правило, складається із двох етапів. У першому етапі обчислюються передбачені алгоритмом характеристики завдання. На другому етапі отриманої інформації будується наближення до рішення.

Надалі для запису методів мінімізації використовуватимемо скорочення виду:

$$x^{k+1} = x^k + \alpha_k \cdot h^k.$$

Вектор h^k визначає напрямок $(k+1)$ -го кроку оптимізації, а коефіцієнт α_k - довжину цього кроку.

Збіжність методів оптимізації.

Важливою характеристикою чисельних методів є їхня збіжність.

Говорять, що метод оптимізації сходиться, якщо виконується таке:

$$x^k \xrightarrow[k \rightarrow \infty]{} x^*$$

Більшість теорем про збіжність методів оптимізації доводять у припущенні про опуклість цільової функції, оцінки швидкості збіжності встановлюють при обмежувальному припущенні сильної опуклості.

Встановлення факту збіжності та оцінка швидкості збіжності дають суттєву інформацію про обраний метод мінімізації. Насамперед вимоги, які доводиться накладати на функцію, що мінімізується, показують сферу застосування методу.

Умови зупинення.

Для завдання конкретного обчислювального алгоритму нескінченно кроковий метод необхідно доповнити умовою зупинки.

Умова зупинки може визначатися наявними обчислювальними ресурсами. Зупинка може здійснюватися після досягнення заданої точності. Однак при вирішенні реального завдання складно оцінити справжню точність. Тому про досягнення точності необхідно судити за непрямыми ознаками.

Насправді часто використовують такі умови зупинки:

$$\begin{aligned} \|x^{k+1} - x^k\| &\leq \varepsilon_1; \\ \|f(x^{k+1}) - f(x^k)\| &\leq \varepsilon_2; \\ \|f'(x^{k+1})\| &\leq \varepsilon_3. \end{aligned}$$

До початку обчислень вибирається одна з наведених умов зупинки та відповідна точність ε .

6.1 Методи розв'язання задач нульового порядку. Метод Хука-Джівса

Цей метод належить до методів нульового порядку (або методів прямого пошуку) визначення мінімуму функції n змінних. Це означає, що при пошуку мінімуму (максимуму) використовуються лише значення функції.

Цей метод є комбінацією двох видів пошуку: дослідницького пошуку та прискорюючого пошуку за зразком.

Дослідницький пошук спрямовано виявлення локального поведінки цільової функції і визначення напрямку її спадання вздовж "ярів". Отримана інформація використовується при пошуку за зразком, орієнтованим на прискорення пошуку за рахунок просування вздовж передбачуваного напрямку мінімуму.

Метод складається з наступних кроків.

Попередня (нульова ітерація). Задати початкову точку, крок для кожної координати та точність.

Основна ітерація включає дослідницький пошук і, якщо він виявився вдалим, то пошук за зразком.

Метод Хука-Дживса характеризується нескладною стратегією пошуку, відносною простотою обчислення та невисоким рівнем вимог до обсягу пам'яті, який виявляється нижчим, ніж при використанні методу пошуку за симплексом. Завдяки цьому алгоритм Хука-Дживса знаходить широке застосування практично.

6.2 Практичне завдання

З таблиці 6.1 вибрати двовимірну функцію, на яку потрібно знайти мінімум, задаючи початкову точку пошуку з тієї ж таблиці.

Таблиця 6.1 Двовимірні функції та початкова точка пошуку мінімуму

Варіант	Функція	Початкова точка пошуку
1.	$Y = (1 - X_1)^2 + (2 - X_2)^2$	[0, 0]
2.	$Y = 9 - 2 * (5 * X_1 + 2 * X_2) + X_1^2 + X_2^2$	[0.2, 0.1]
3.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - 0.01 * X_1 * X_2$	[-1, 0]
4.	$Y = (10 - 2 * X_1)^2 + (12 - 5 * X_2)^2$	[0.3, 0.5]
5.	$Y = (8 - X_1)^2 - (7 - X_2)^2 + 3X_2^4$	[-3, 5]
6.	$Y = 19 - 15 * X_1 - 8 * X_2 + 3 * X_1^2 + X_2^2$	[0, 0]
7.	$Y = X_1^6 - (1 - X_1)^2 + (2 - X_2)^2$	[10 , 20]
8.	$Y = 2 * X_1^2 + 4 * X_1 * X_2^3 - 10 * X_1 * X_2 + X_2^3$	[0, 0]
9.	$Y = 9 - 25 * X_1 + X_1^2 - 22 * X_2 + X_2^2$	[0, 1]
10.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - 3 * X_1 * X_2$	[-6 , 7]
11.	$Y = X_1^2 - X_1^3 * X_2^2 - 9 * X_1 * X_2 + X_2^3$	[20 , -10]
12.	$Y = 18 - 20 * X_1 - 8 * X_2 + 2 * X_1^2 + 2 * X_2^2$	[0 , 0]
13.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - X_1 * X_2$	[-1 , 6]
14.	$Y = 3 - 3.3 * X_1 - 1.1 * X_2 + 2 * X_1^2 + 2 * X_2^2$	[0 , 0]
15.	$Y = 4 * X_1^2 + 3 * X_2^2 - 4 * X_1 * X_2 + X_1$	[5 , 3]
16.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - X_1 - X_2$	[2 , 8]
17.	$Y = 6 - 12 * X_1 + 12 * X_2 + 8 * X_1^2 + 6 * X_2^2$	[6 , 1]
18.	$Y = 0.5 * X_1^2 + 0.75 * X_1 * X_2^3 - 2 * X_1 * X_2$	[3, 6]
19.	$Y = 2.5 * X_1^2 + 1.5 * X_1 * X_2 + 2 * X_1^2$	[5, 2]
20.	$Y = (6 - 1.5 * X_1)^2 - (4 - 2.5 * X_2)^2 + 0.5X_2^3$	[7, 9]

Реалізувати обчислення заданої функції у наданому зразку сценарію GNU Octave, виконати сценарій та отримати результати пошуку мінімуму. Проведіть ці розрахунки

ЗМІНИВШИ ПОЧАТКОВУ ТОЧКУ ПОШУКУ НА ВЛАСНИЙ РОЗСУД.

6.3 Приклад сценарію вирішення самотійної роботи

```
% OCTAVE version

1;

function stop = myoutput(x,optimvalues,state);
stop = false;
if isequal(state,'iter')
SP = evalin('base', 'SearchPath');
SP = [SP; x];
assignin('base','SearchPath',SP);
end
end

clear;clc;close all;

SearchPath = [];

disp('Особиста інформація студента');
Student = 'Титюк В.К., група ЕПА-78-1'; %%=====
disp('ММУ Самостійна робота №7. Пошук мінімуму методом Нелдера-Міда')
disp(Student)
disp(date())

disp('Завдання 1');
disp('Вихідні дані варіанта');
disp('--- Вихідні дані ---');
N = 5; %%=====
disp(['Номер варіанта за журналом = ', num2str(N)])
disp('-----')

% Minimize Rosenbrock's function, a notoriously difficult optimization problem для багатьох algorithms:
%func = @(x)100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;

func = @(x)(1-x(1)).^2 + (2-x(2)).^2;

options = optimset('Display', 'iter', 'OutputFcn', @myoutput);
x0 = [12, 5];
y0 = func(x0);
[endpt, zmin, exitflag] = fminsearch(func,x0, options);
% 1 Функція спрямована на рішення x.
% 0 Number of iterations exceeded options.MaxIter or number of function evaluations exceeded options.MaxFunEvals.
% -1 Цей алгоритм був виконаний з виконання функції.

xmin = endpt(1);
ymin = endpt(2);

disp(['Результат пошуку мінімуму. Xmin = ', num2str(xmin), '. Ymin = ', num2str(ymin), '. Zmin = ', num2str(zmin)])

disp('Тривимірні графіки в декартовій системі координат');
x1 = -2; x2 = 14; hx = 0.1;
y1 = -3; y2 = 8; hy = 0.1;
[X,Y]=meshgrid(x1:hx:x2, y1:hy:y2);

for i = 1: size (X) (1)
for j = 1: size (X) (2)
Z(i, j) = func([X(i, j), Y(i, j)]);
end
end

figure(1); hold on; grid on;
colormap ("summer");
surf (X, Y, Z);

shading interp;
```

```

colorbar;
xlabel('X');
ylabel('Y');
zlabel('Z');
scatter3(x0(1), x0(2), y0, 80, 'filled', 'markerfacecolor', 'blue');
scatter3(xmin, ymin, zmin, 80, 'filled', 'markerfacecolor', 'red');

figure(2); hold on; grid on;
contour3(X,Y,Z);
xlabel('X');
ylabel('Y');
zlabel('Z');
scatter3(x0(1), x0(2), y0, 80, 'filled', 'markerfacecolor', 'blue');
scatter3(xmin, ymin, zmin, 80, 'filled', 'markerfacecolor', 'red');

```

6.4 Приклад результатів практичної роботи

Особиста інформація студента

ММУ Самостійна робота №6. Пошук мінімуму методом Хука-Джівса

Титюк В.К., група ЕПА-78-1

04-Nov-2023

Завдання 1

Вихідні дані варіанта

--- Вихідні дані ---

Номер варіанта за журналом = 5

 $f(x) = 45.00000000$ at $x = -5.00000000$ $y = 5.00000000$

$f(x) = 2.50000000$ at $x = 2.50000000$ $y = 2.50000000$

$f(x) = 0.31250000$ at $x = 1.25000000$ $y = 2.50000000$

$f(x) = 0.07812500$ at $x = 1.25000000$ $y = 1.87500000$

$f(x) = 0.01953125$ at $x = 0.93750000$ $y = 1.87500000$

$f(x) = 0.00488281$ at $x = 0.93750000$ $y = 2.03125000$

$f(x) = 0.00122070$ at $x = 1.01562500$ $y = 2.03125000$

$f(x) = 0.00030518$ at $x = 1.01562500$ $y = 1.99218750$

$f(x) = 0.00007629$ at $x = 0.99609375$ $y = 1.99218750$

$f(x) = 0.00001907$ at $x = 0.99609375$ $y = 2.00195312$

$f(x) = 0.00000477$ at $x = 1.00097656$ $y = 2.00195312$

$f(x) = 0.00000119$ at $x = 1.00097656$ $y = 1.99951172$

$f(x) = 0.00000030$ at $x = 0.99975586$ $y = 1.99951172$

$f(x) = 0.00000007$ at $x = 0.99975586$ $y = 2.00012207$

$f(x) = 0.00000002$ at $x = 1.00006104$ $y = 2.00012207$

$f(x) = 0.00000000$ at $x = 1.00006104$ $y = 1.99996948$

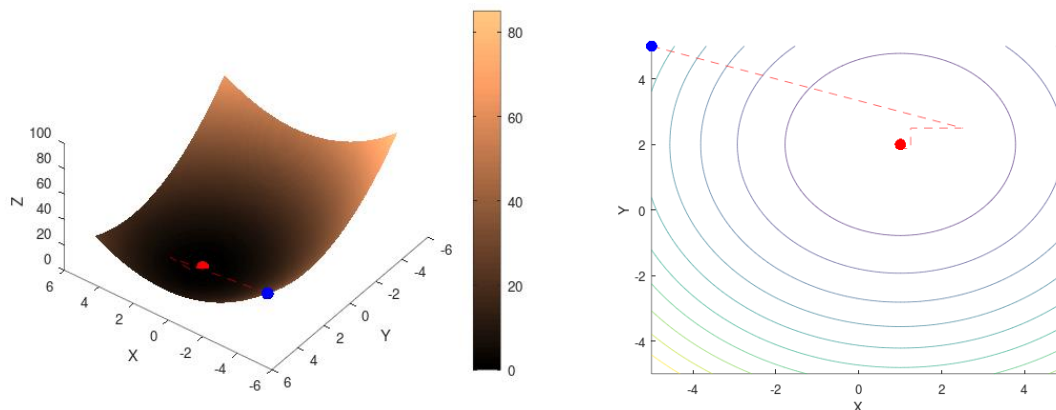
$f(x) = 0.00000000$ at $x = 0.99998474$ $y = 1.99996948$

$f(x) = 0.00000000$ at $x = 0.99998474$ $y = 2.00000763$

$f(x) = 0.00000000$ at $x = 1.0000381$ $y = 2.00000763$

Результат пошуку мінімуму. $X_{min} = 1$. $Y_{min} = 2$. $Z_{min} = 1.819e-11$

Тривимірні графіки у декартовій системі координат



Контрольні запитання.

1. Що таке багатовимірна оптимізація і в чому її відмінність від одновимірної оптимізації?
2. Які основні завдання вирішують із використанням методів багатовимірної оптимізації?
3. Який принцип лежить в основі методу Хука-Джівса, і які етапи включає цей метод?
4. Який критерій зупинки використовується у методі Хука-Джівса і чому він важливий?
5. Як вибрати початкове наближення у методі Хука-Джівса, і як воно впливає на збіжність методу?
6. У чому полягає роль кроку (крокового вектора) у методі Хука-Джівса і як його визначають?
7. Як метод Хука-Джівса оновлює поточне наближення до оптимуму на кожному кроці?
8. Які переваги та обмеження існують у методу Хука-Джівса порівняно з іншими методами багатовимірної оптимізації?

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скріншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

САМОСТІЙНА РОБОТА №7 МЕТОДИ БЕЗУМОВНОЇ БАГАТОМІРНОЇ ОПТИМІЗАЦІЇ. МЕТОД НЕЛДЕРА-МІДА (A. NELDER AND R. MEAD).

Мета роботи: Ознайомлення із загальними характеристиками чисельних методів вирішення багатовимірного оптимізаційного завдання без обмежень: загальна формула, збіжність, умови зупинки. Дослідження прямих методів пошуку, що використовують для формування рішення лише відомості про значення функції.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Метод Нелдера - Міда, також відомий як метод деформованого багатогранника і симплекс-метод, - метод безумовної оптимізації функції від кількох змінних, що не використовує похідної (точніше - градієнтів) функції, а тому легко застосовується до негладких та/або зашумлених функцій.

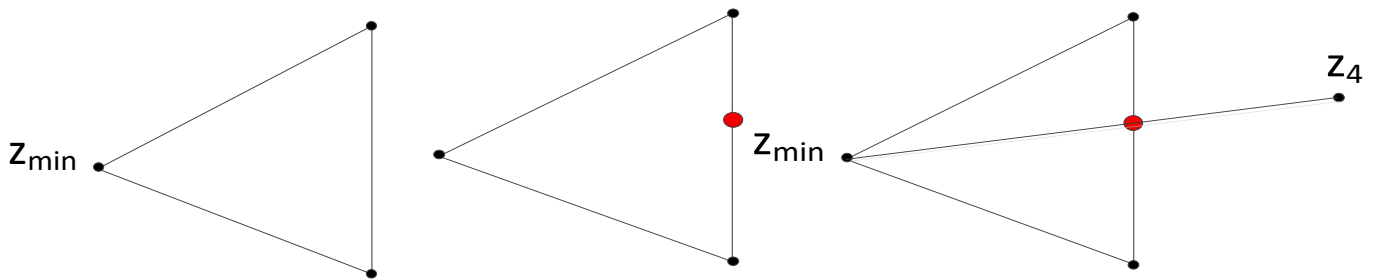
Суть методу полягає в послідовному переміщенні та деформуванні симплексу навколо точки екстремуму.

Метод знаходить локальний екстремум і може "застрягти" в одному з них. Якщо все ж таки потрібно знайти глобальний екстремум, можна намагатися вибирати інший початковий симплекс.

Опишемо ідею цього стосовно функції двох змінних для пошуку максимуму функції $z = f(x, y)$.

На початку розрахунку задається початковий симплекс, для 2-вимірної випадку, правильний трикутник і обчислюються значення функції у вершинах трикутника.

Виділяється вершина з найменшим значенням функції $z_{\min}$. Для двох вершин трикутника знаходимо центр тяжкості. Від знайденого центру тяжкості у знайденому напрямку робимо крок пошуку, знаходимо координати нової точки z_4 .



Якщо в точці z_4 знайдено нове мінімальне значення, то робимо черговий крок у цьому напрямку.

Якщо черговий крок призводить до збільшення функції, то в тій точці відбувається відображення симплекса, вибирається новий напрямок руху.

Метод Нелдера-Міда схожий на покоординатним спуском, проте тут рух проводиться у напрямку потенційно максимального градієнта.

7.1 Практичне завдання

З таблиці 7.1 вибрати двовимірну функцію, на яку потрібно знайти мінімум, задаючи початкову точку пошуку з тієї ж таблиці.

Таблиця 7.1 Двовимірні функції та початкова точка пошуку мінімуму методом Нелдера-Міда.

Варіант	Функція	Початкова точка пошуку
21.	$Y = (1 - X_1)^2 + (2 - X_2)^2$	[0, 0]
22.	$Y = 9 - 2 * (5 * X_1 + 2 * X_2) + X_1^2 + X_2^2$	[0.2, 0.1]
23.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - 0.01 * X_1 * X_2$	[-1, 0]
24.	$Y = (10 - 2 * X_1)^2 + (12 - 5 * X_2)^2$	[0.3, 0.5]
25.	$Y = (8 - X_1)^2 - (7 - X_2)^2 + 3X_2^4$	[-3, 5]
26.	$Y = 19 - 15 * X_1 - 8 * X_2 + 3 * X_1^2 + X_2^2$	[0, 0]
27.	$Y = X_1^6 - (1 - X_1)^2 + (2 - X_2)^2$	[10, 20]
28.	$Y = 2 * X_1^2 + 4 * X_1 * X_2^3 - 10 * X_1 * X_2 + X_2^3$	[0, 0]
29.	$Y = 9 - 25 * X_1 + X_1^2 - 22 * X_2 + X_2^2$	[0, 1]
30.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - 3 * X_1 * X_2$	[-6, 7]

31.	$Y = X_1^2 - X_1^3 * X_2^2 - 9 * X_1 * X_2 + X_2^3$	[20 , -10]
32.	$Y = 18 - 20 * X_1 - 8 * X_2 + 2 * X_1^2 + 2 * X_2^2$	[0 , 0]
33.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - X_1 * X_2$	[-1 , 6]
34.	$Y = 3 - 3.3 * X_1 - 1.1 * X_2 + 2 * X_1^2 + 2 * X_2^2$	[0 , 0]
35.	$Y = 4 * X_1^2 + 3 * X_2^2 - 4 * X_1 * X_2 + X_1$	[5 , 3]
36.	$Y = (1 - X_1)^2 + (2 - X_2)^2 - X_1 - X_2$	[2 , 8]
37.	$Y = 6 - 12 * X_1 + 12 * X_2 + 8 * X_1^2 + 6 * X_2^2$	[6 , 1]
38.	$Y = 0.5 * X_1^2 + 0.75 * X_1 * X_2^3 - 2 * X_1 * X_2$	[3, 6]
39.	$Y = 2.5 * X_1^2 + 1.5 * X_1 * X_2 + 2 * X_1^2$	[5, 2]
40.	$Y = (6 - 1.5 * X_1)^2 - (4 - 2.5 * X_2)^2 + 0.5 * X_2^3$	[7, 9]

Для пошуку екстремуму двовимірної функції методом Нелдера-Міда використана вбудована в OCTAVE та MATLAB функція `fminsearch`, виклик якої здійснюється наступним чином:

```
[endpt, zmin, exitflag] = fminsearch(func,x0, options);
```

Вхідні параметри

`func` - функція, для якої здійснюється пошук екстремуму, наприклад

```
func = @( x )(6-1.5* x (1)).^2 - (4-2.5* x (2)).^2 + 0.5* x (2).^3;
```

`x 0` – вектор координат початкової точки пошуку;

`options` – додаткові опції пошуку. Наприклад,

```
options = optimset('Display', 'iter', 'OutputFcn', @myoutput);
```

Вихідні параметри

`endpt` - вектор координат певної точки мінімуму

`zmin` - значення функції у певній точці мінімуму;

`exitflag` :

1 Функція збіглася до рішення `x` .

0 Кількість ітерацій перевищила `options . MaxIter` чи кількість обчислень функції перевищила `options . MaxFunEvals` .

-1 Алгоритм було завершено вихідною функцією.

7.2 Приклад сценарію вирішення самотійної роботи

```
% OCTAVE version
```

```
1;
```

```
function stop = myoutput(x,optimvalues,state);
```

```
stop = false;
```

```
if isequal(state,'iter')
```

```
SP = evalin('base', 'SearchPath');
```

```
SP = [SP; x];
```

```

assignin('base','SearchPath',SP);
end
end

clear;clc;close all;

SearchPath = [];

disp('Особиста інформація студента');
Student = 'Титюк В.К., група ЕПА-78-1'; %%=====
disp('ММУ самостійна робота №7. Пошук мінімуму методом Нелдера-Міда')
disp(Student)
disp(date())

disp('Завдання 1');
disp('Вихідні дані варіанта');
disp('--- Вихідні дані ---');
N = 5; %%=====
disp(['Номер варіанта за журналом = ', num2str(N)])
disp('-----')

% Minimize Rosenbrock's function, a notoriously difficult optimization problem для багатьох algorithms:
%func = @(x)100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;

func = @(x)(1-x(1)).^2 + (2-x(2)).^2;

options = optimset('Display', 'iter', 'OutputFcn', @myoutput);
x0 = [12, 5];
y0 = func(x0);
[endpt, zmin, exitflag] = fminsearch(func,x0, options);
% 1 Функція спрямована на рішення x.
% 0 Number of iterations exceeded options.MaxIter or number of function evaluations exceeded options.MaxFunEvals.
% -1 Цей algorithm був виконаний з виконання функції.

xmin = endpt(1);
ymin = endpt(2);

disp(['Результат пошуку мінімуму. Xmin = ', num2str(xmin), '. Ymin = ', num2str(ymin), '. Zmin = ', num2str(zmin)])

disp('Тривимірні графіки в декартовій системі координат');
x1 = -2; x2 = 14; hx = 0.1;
y1 = -3; y2 = 8; hy = 0.1;
[X,Y]=meshgrid(x1:hx:x2, y1:hy:y2);

for i = 1: size (X) (1)
for j = 1: size (X) (2)
Z(i, j) = func([X(i, j), Y(i, j)]);
end
end

figure(1); hold on; grid on;
colormap ("summer");
surf (X, Y, Z);

shading interp;
colorbar;
xlabel('X');
ylabel('Y');
zlabel('Z');
scatter3(x0(1), x0(2), y0, 80, 'filled', 'markerfacecolor', 'blue');
scatter3(xmin, ymin, zmin, 80, 'filled', 'markerfacecolor', 'red');

figure(2); hold on; grid on;
contour3(X,Y,Z);
xlabel('X');
ylabel('Y');
zlabel('Z');

```

```
scatter3(x0(1), x0(2), y0, 80, 'filled', 'markerfacecolor', 'blue');
scatter3(xmin, ymin, zmin, 80, 'filled', 'markerfacecolor', 'red');
```

7.3 Приклад результатів самостійної роботи

Особиста інформація студента

ММУ Самостійна робота №7. Пошук мінімуму методом Нелдера-Міда

Титюк В.К., група ЕПА-78-1

04-Nov-2023

Завдання 1

Вихідні дані варіанта

--- Вихідні дані ---

Номер варіанта за журналом = 5

$f(x_0) = 1.3000e+02$

Iter. 1, how = initial , nf = 3, f = 1.3000e+02 (0.0%)

Iter. 2, how = reflect, nf = 4, f = 1.3000e+02 (0.0%)

Iter. 3, how = expand, nf = 6, f = 1.1860e+02 (8.8%)

Iter. 4, how = contract, nf = 8, f = 2.5519e+01 (78.5%)

Iter. 5, how = contract, nf = 10, f = 1.6796e+01 (34.2%)

Iter. 6, how = contract, nf = 12, f = 1.1670e+01 (30.5%)

Iter. 7, how = contract, nf = 14, f = 5.6514e+00 (51.6%)

Iter. 8, how = contract, nf = 16, f = 3.1293e+00 (44.6%)

Iter. 9, how = contract, nf = 18, f = 5.9872e-01 (80.9%)

Iter. 10, how = contract, nf = 20, f = 5.9872e-01 (0.0%)

Iter. 11, how = contract, nf = 22, f = 5.9872e-01 (0.0%)

Iter. 12, how = reflect, nf = 24, f = 8.3311e-02 (86.1%)

Iter. 13, how = contract, nf = 26, f = 8.3311e-02 (0.0%)

Iter. 14, how = contract, nf = 28, f = 8.3311e-02 (0.0%)

Iter. 15, how = contract, nf = 30, f = 3.5370e-02 (57.5%)

Iter. 16, how = contract, nf = 32, f = 9.2669e-03 (73.8%)

Iter. 17, how = contract, nf = 34, f = 9.2669e-03 (0.0%)

Iter. 18, how = reflect, nf = 35, f = 9.2669e-03 (0.0%)

Iter. 19, how = contract, nf = 37, f = 2.2689e-03 (75.5%)

Iter. 20, how = contract, nf = 39, f = 1.2315e-03 (45.7%)

Iter. 21, how = contract, nf = 41, f = 1.2315e-03 (0.0%)

Iter. 22, how = contract, nf = 43, f = 1.9930e-04 (83.8%)

Iter. 23, how = contract, nf = 45, f = 1.9122e-04 (4.1%)

Iter. 24, how = shrink, nf = 49, f = 3.3849e-07 (99.8%)

Iter. 25, how = contract, nf = 51, f = 3.3849e-07 (0.0%)

Iter. 26, how = contract, nf = 53, f = 3.3849e-07 (0.0%)

Iter. 27, how = contract, nf = 55, f = 3.3849e-07 (0.0%)

Iter. 28, how = contract, nf = 57, f = 3.3849e-07 (0.0%)

Iter. 29, how = reflect, nf = 58, f = 3.3849e-07 (0.0%)

Iter. 30, how = contract, nf = 60, f = 3.3849e-07 (0.0%)

Iter. 31, how = contract, nf = 62, f = 3.3849e-07 (0.0%)

Iter. 32, how = contract, nf = 64, f = 3.3849e-07 (0.0%)

Iter. 33, how = contract, nf = 66, f = 3.3849e-07 (0.0%)

Iter. 34, how = reflect, nf = 68, f = 2.7592e-07 (18.5%)

Iter. 35, how = contract, nf = 70, f = 4.5894e-08 (83.4%)

Iter. 36, how = contract, nf = 72, f = 4.5894e-08 (0.0%)

Iter. 37, how = shrink, nf = 76, f = 3.0557e-10 (99.3%)

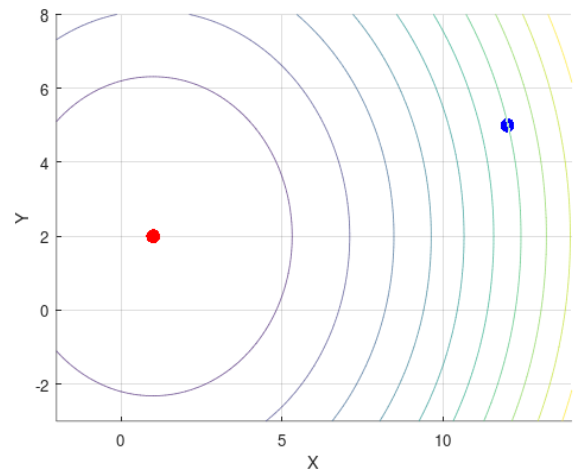
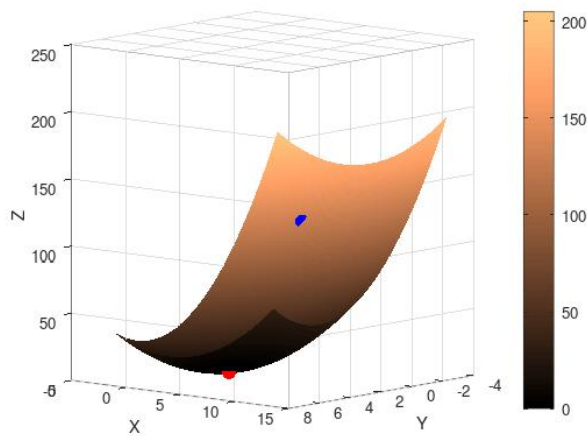
Iter. 38, how = contract, nf = 78, f = 3.0557e-10 (0.0%)

Iter. 39, how = contract, nf = 80, f = 3.0557e-10 (0.0%)

Algorithm converged. Simplex size 6.8676e-05 <= 1.0000e-04 and step in function value 1.8854e-08 <= 1.0000e-04

Результат пошуку мінімуму. Xmin = 1. Ymin = 2. Zmin = 3.0557e-10

Тривимірні графіки у декартовій системі координат



Контрольні запитання.

1. Що таке метод Нелдера-Міда і в чому полягає його основна ідея?
2. Які основні призначення методу Нелдера-Міда у чисельних методах оптимізації?
3. Які етапи становлять процес оптимізації з допомогою методу Нелдера-Міда?
4. Які параметри впливають на ефективність методу Нелдера-Міда і як їх налаштовувати?
5. Як визначається точність результатів, одержаних за допомогою методу Нелдера-Міда?
6. Як метод Нелдера-Міда використовується для вирішення задач оптимізації з обмеженнями?
7. Які переваги та недоліки методу Нелдера-Міда порівняно з іншими чисельними методами оптимізації?

Зміст звіту

1. Титульний лист
2. Цілі, завдання роботи
3. Текст завдання
4. Результати роботи (скриншоти даних, формул, діаграми)
5. Відповіді на контрольні питання

Список рекомендованих джерел

1. Dingyu X., Feng P. MatLAB and Simulink in Action: Programming, Scientific Computing and Simulation. Springer, 2024. 472 p. ISBN 978-981-99-1175-2.
2. Palm III William J. MatLAB for Engineering Applications. 5th Edition. McGraw-Hill LLC, 2023. 656 p. ISBN 978-1-265-13919-3.
3. Eshkabilov S. Beginning MatLAB and Simulink: From Beginner to Pro. 2nd Edition. Apress, 2022. 627 p.
4. MathWorks. MatLAB Coder User's Guide R2021a. The MathWorks, Inc., 2021. 1298 p.
5. Lopez C. MatLAB Matrix Algebra. N.-Y.: Springer, 2014. 240p.
6. Asadi Farzin. Applied Numerical Analysis with MatLAB/Simulink: For Engineers and Scientists. Springer, 2023. 327 p. ISBN 978-3-031-19365-1.
7. Yao Ray. MatLAB Programming in 8 Hours, For Beginners, Learn Coding Fast: MatLAB Quick start Guide & Exercises. Independently published, 2021. 121 p. ASIN B08ZSRXTYC.

Методичні вказівки до самостійної роботи з дисципліни «Математичні моделі та програмні засоби розв'язання комплексних практичних задач електротехніки та електромеханіки» першого (бакалаврського) рівня вищої освіти галузі знань: G «Інженерія, виробництво та будівництво», спеціальність: G3 «Електрична інженерія».

Укладачі:

Валерій Титюк

Володимир Власенко

Реєстраційний № ____ від _____ 2025 р.

Підписано до друку _____ 2025 р.

Формат А5

Обсяг 46 стор.

Тираж _____ прим.

м. Кривий Ріг,
вул. Віталія Матусевича, 11