

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Б.П.Довгий, Є.С.Вакал, Ю.Є.Вакал, А.В.Попов

**ВИКОРИСТАННЯ МАТЕМАТИЧНОГО ПАКЕТА
MATLAB ДЛЯ РОЗВ'ЯЗУВАННЯ ПРИКЛАДНИХ ЗАДАЧ**

НАВЧАЛЬНИЙ ПОСІБНИК

КИЇВ – 2012

УДК 681.3.06
ББК

Д

Рецензенти:
канд. фіз.-мат. наук, доц. Ю.В. Бондарчук
канд. фіз.-мат. наук, доц. О.С. Тригуб

*Рекомендовано до друку вченою радою механіко-математичного факультету
(протокол № 4 від 15 листопада 2012 року)*

Довгий Б.П.

Д

Використання математичного пакета MATLAB для розв'язування прикладних задач /
Б.П.Довгий, Є.С.Вакал, Ю.Є.Вакал, А.В.Попов. – К.:Фітосоціоцентр,2012. – 77 с.
ISBN

Призначено для ознайомлення з математичним пакетом MATLAB, який вивчається у відповідності з програмою дисципліни “Прикладні програми”, що викладається на механіко-математичному факультеті. За змістом видання поділено на теми, які містять основні відомості з операційного середовища MATLAB, елементів мови, матричних обчислень, графіки, чисельного аналізу та опис комп'ютерних технологій розв'язання прикладних задач математичної фізики за допомогою MATLAB.

Для студентів і викладачів механіко-математичного факультету і факультету кібернетики.

УДК 681.3.06

ББК

Навчальне видання

ДОВГИЙ Борис Павлович
ВАКАЛ Євген Сергійович
ВАКАЛ Юлія Євгеніївна
ПОПОВ Андрій Валерійович

**ВИКОРИСТАННЯ МАТЕМАТИЧНОГО ПАКЕТА MATLAB
ДЛЯ РОЗВ'ЯЗУВАННЯ ПРИКЛАДНИХ ЗАДАЧ**

Навчальний посібник

**Друкується в авторській редакції
Відповідальний за випуск – Соломаха І.В.**

**Видавництво Українського фітосоціологічного центру
тел/факс (044) 524-11-61**

**Підписано до друку 22.12.2012 р. Формат 60x84 1/16.
Друк різнографічний. Папір офсетний. Гарнітура Times.
Умовн. друк. арк. 4.1. Умов. вид. арк. 4.3.
Зам. № 691. Наклад 300 прим.**

ISBN

© Довгий Б.П., Вакал Є.С., Вакал Ю.Є., Попов А.В. 2012
© Український фітосоціологічний центр

ПЕРЕДМОВА

В останні роки наукове програмування зазнало серйозної трансформації: розвиваються інтегровані середовища, що ґрунтуються на спеціалізованих алгоритмічних мовах, зростає застосування універсальних математичних систем типу MATLAB, MAPLE, MATHEMATICA, MATCAD та ін. Ці системи мають дружній інтерфейс, потужні графічні засоби, власні мови програмування, реалізують ряд стандартних і спеціальних математичних операцій. Все це надає широкі можливості для ефективної роботи спеціалістів різних галузей, про що говорить активне застосування математичних пакетів у наукових дослідженнях і викладацькій діяльності. Математичні пакети – гарний вибір для проведення будь-якого дослідження, де вимагається математика, – від курсової роботи до наукового пошуку.

Навчальне видання призначено для ознайомлення студентів з математичним пакетом MATLAB, який вивчається у відповідності з програмою дисципліни “Прикладні програми”, що викладається на механіко-математичному факультеті. Серед інших пакетів обрано MATLAB з причини поступового розвитку університетського процесу навчання. MATLAB дозволяє студенту природно перейти у своєму навчанні від простих обчислень до можливостей програмування високого рівня через використання потужних обчислювальних і графічних модулів. Останнє означає, що студенти можуть займатися програмуванням в MATLAB навіть без попереднього вивчення його основ. Проста вбудована мова програмування дозволяє легко створювати власні алгоритми.

В MATLAB реалізовані класичні чисельні алгоритми розв’язання рівнянь, задач лінійної алгебри, знаходження значень визначених інтегралів, апроксимації, розв’язання окремих рівнянь і систем диференціальних рівнянь. Для застосування більшості можливостей пакета достатньо базових знань з чисельних методів. Розв’язання спеціальних задач, зрозуміло, вимагає відповідної теоретичної підготовки.

За змістом видання поділено на теми, які містять основні відомості з операційного середовища MATLAB, елементів мови, матричних обчислень, графіки, чисельного аналізу та конкретних прикладів опису комп’ютерних технологій розв’язання прикладних задач математичної фізики за допомогою MATLAB.

Навчальне видання може бути корисне студентам університетів при проведенні вправ на персональному комп’ютері, наприклад, з обчислювальної математики, інформатики та інших предметів, та контрольних і лабораторних робіт із спеціальних дисциплін, що вимагають елементів наукових досліджень. Викладений матеріал доцільно також використовувати для обробки результатів математичних досліджень при курсовому і дипломному проектуванні.

Тема №1

ВСТУП ДО СИСТЕМИ MATLAB

Призначення і сфера застосування математичного пакета MATLAB.

Для розв'язання прикладних задач важливим є застосування засобів автоматизації обчислень від простих до досить громіздких, а також графічна візуалізація отриманих результатів. Сучасне програмне забезпечення ПЕОМ має досить різноманітні, потужні математичні пакети, наприклад, MATLAB, MAPLE, MATHEMATICA, які використовуються у математичних обчисленнях і аналітичних перетвореннях. Пакет MATLAB є інтерактивною системою з розвиненою довідковою системою, містить велику кількість операторів та функцій, застосування яких дає змогу чисельно розв'язувати складні задачі математики, фізики, хімії, інженерії, фінансів [1-3,5-8,10]. Крім вбудованих засобів, пакет можна доповнювати власними розробками, використовуючи як власну мову (М-програми), так і зовнішні алгоритмічні мови, наприклад, C#, JAVA. Підсумовуючи, охарактеризуємо спектр задач, які розв'язуються за допомогою пакета MATLAB:

- проведення математичних досліджень, що вимагають обчислень і аналітичних розрахунків;
- розробка і аналіз алгоритмів;
- математичне моделювання, комп'ютерний експеримент;
- аналіз і обробка даних;
- візуалізація, наукова та інженерна графіка;
- розробка графічних і розрахункових додатків.

Перше знайомство з роботою в системі. Для практичної роботи будемо використовувати програмний комплекс MATLAB 7.6.0 (R2008a), робота в якому досить легка та комфортна.

При стандартному завантаженні (клацнувши мишею по ярлику програми MATLAB) на екрані відкривається система меню та три вікна: *робоча область*, *вікно історії*, *командне вікно*.

Робота з меню здійснюється за допомогою маніпулятора миша та набору певних клавіатурних комбінацій ("гарячих клавіш").

У вікні робочої області відображаються імена змінних, уведених користувачем, та їх поточні значення.

Вікно історії використовується для зберігання та відображення раніше виконаних команд користувача.

Інтерактивний сеанс роботи в командному вікні MATLAB прийнято іменувати сесією (session). Сесія, по суті, є поточним документом, що відображає роботу користувача із системою MATLAB. У ній є рядки введення даних,

результатів і повідомлень про помилки. За допомогою команд сесії користувач взаємодіє із середовищем пакета. Поіменовані дані поточної сесії, розташовані в робочій області пам'яті (але не саму сесію), можна записати на диск у вигляді файлів формату *.mat*, використовуючи команду *save* (зберегти). Командою *load* (завантажити) можна відновити з диска дані робочої області. Фрагменти сесії можна зберегти в певному документі за допомогою команди *diary*.

При роботі з MATLAB у командному режимі діє найпростіший *рядковий редактор*. Команди, які використовуються в редакторі, мають, як правило, відповідні комбінації клавіш для їх виклику. Ці комбінації наведені в табл. 1.1.

Таблиця 1.1. Комбінації клавіш для команд рядкового редактора

Клавіша	Призначення
<→> або <Ctrl+B>	на позицію вправо
<←> або <Ctrl+F>	на позицію вліво
<Ctrl + →> або <Ctrl+R>	на слово вправо
<Ctrl + ←> <Ctrl+L>	на слово вліво
<Home> або <Ctrl+A>	на початок рядка
<End> або <Ctrl+E>	на кінець рядка
<↑> або <Ctrl+P>	на попередню команду вгору
<↓> або <Ctrl+N>	на наступну команду вниз
 або <Ctrl+D>	видалити символ справа
<BkSp> або <Ctrl+H>	видалити символ зліва
<Ctrl+K>	видалити текст до кінця рядка
<Esc>	очистити рядок введення
<Ins>	вибрати режим вставки/заміни
<PgUp>	на сторінку сесії вгору
<PgDn>	на сторінку сесії вниз

Раніше введені команди автоматично утворюють список, який виводиться у вікні історії. Якщо з'явилася необхідність повторити раніше виконану команду, то треба відшукати її в цьому вікні і двічі клацнути по ній лівою кнопкою миші. Для активізації вікна історії необхідно вибрати вкладку з однойменною назвою і клацнути по ній лівою кнопкою миші. Якщо клацнути на команді вікна історії лівою кнопкою миші, то дана команда стає поточною (на синьому тлі). Можна виділити потрібну послідовність команд за допомогою комбінації клавіш <Shift+↑>, <Shift+↓>. Вставлений у командний рядок набір команд відправляється на виконання натисненням клавіші <Enter>. До цього вміст набору можна редагувати, використовуючи звичайні прийоми редагування, спільні для Windows-програм, у тому числі за допомогою миші. Можна вносити в команди необхідні зміни, видаляти зайві команди і додавати нові.

Тема № 2

ОПЕРАЦІЙНЕ СЕРЕДОВИЩЕ MATLAB І ОСНОВНІ КОМАНДИ

Теоретичні відомості

Основні пункти меню системи, інструментальна панель і опції MATLAB. Основне меню системи складається з 6 пунктів, кожний з яких у свою чергу містить декілька підпунктів різного ступеня вкладеності:

- *File* – робота з файлами;
- *Edit* – редагування сесії та документів М-файлів;
- *View* – управління вікнами та панеллю інструментів;
- *Web* – доступ до Інтернет-ресурсів;
- *Windows* – відображення відкритих додаткових вікон, перехід до потрібного вікна;
- *Help* — доступ до довідкових підсистем.

Розглянемо призначення деяких із них.

Серед команд меню *File*:

- *New* – відкриття підменю з пунктами:
 - *M-file* – відкриття вікна редагування М-файлів, аналог команди *edit*;
 - *Figure* – відкриття порожнього вікна графіки;
 - *Model* – відкриття порожнього вікна для створення Simulink-моделі;
 - *GUI* – відкриття вікна розробки елементів графічного інтерфейсу користувача;
- *Open* – відкриття вікна завантаження файла;
- *Close Command Windows* – закриття вікна командного режиму;
- *Import data* – відкриття вікна імпорту файлових даних;
- *Save Workspace As* – відкриття вікна запису робочої області у вигляді файла із заданим ім'ям;
- *Set Path* – відкриття вікна встановлення шляхів доступу файлової системи;
- *Preferences* – відкриття вікна налаштування елементів інтерфейсу;
- *Print* – відкриття вікна друку поточного документа;
- *Print Selection* – відкриття вікна друку виділеної частини документа.
- *Exit* – завершення роботи в системі.

Серед команд меню *Edit*:

- *Undo* – скасування результату попередньої операції;

- *Redo* – скасування дії останньої операції *Undo*;
- *Cut* – вирізання виділеного фрагмента і перенесення його в буфер обміну;
- *Copy* – копіювання виділеного фрагмента в буфер обміну;
- *Paste* – вставка фрагмента з буфера обміну в поточну позицію курсору;
- *Select All* – виділення всієї сесії;
- *Delete* – видалення виділеного об'єкта;
- *Clear Command Windows* – очистка тексту сесії зі збереженням створених об'єктів, аналог команди *clc*;
- *Clear Command History* – очистка вікна історії команд;
- *Clear Workspace* – очистка вікна робочої області.

Вікно управління редактора М-файлів. Для підготовки і редагування М-файлів (текстів програм на внутрішній алгоритмічній мові системи) використовується спеціальний системний редактор текстів або зовнішній текстовий редактор. Редактор можна викликати командою *edit* з командного рядка або через пункт меню *File*. Після закінчення роботи з текстом М-файла його необхідно записати на диск, використовуючи команду *File* → *Save as* у меню редактора.

При створенні тексту файла редактор виконує ретельну синтаксичну перевірку. Редактор має й інші важливі засоби – він дозволяє встановлювати в тексті М-файла спеціальні мітки переривання (breakpoints). При досягненні цих міток обчислення припиняються, і користувач може оцінити проміжні результати обчислень, наприклад, значення певних змінних, перевірити правильність виконання циклів і т.д. Для зручності роботи з редактором рядки програми в ньому нумеруються в послідовному порядку. Редактор є багатовіконним. Вікно кожної програми оформлене як вкладка.

Управління командним вікном. Розглянемо деякі команди управління вікном командного режиму:

- *clc* – очистка екрана і розміщення курсору в лівому верхньому куті порожнього вікна;
- *home* – повернення курсору в лівий верхній кут вікна;
- *echo <file_name> on* – увімкнення режиму збереження команд, які вводяться на екрані, в текстовий файл *<file_name>* (файл М-сценарію);
- *echo <file_name> off* – вимкнення попереднього режиму збереження команд;
- *echo <file_name>* – зміна режиму збереження на протилежний;
- *echo on all* – увімкнення режиму виведення на екран тексту для всіх М-файлів;
- *echo off all* – вимкнення режиму виведення на екран тексту М-файлів;

- *more on* – увімкнення режиму посторінкового виведення (відображення) (корисний при перегляді великих М-файлів);
- *more off* – вимкнення режиму посторінкового виведення, в цьому випадку для перегляду великих файлів треба користуватися смугою прокручування.

MATLAB у ролі калькулятора. Система MATLAB створена таким способом, що будь-які (часом досить складні) обчислення можна виконувати в режимі *прямих обчислень* (інтерактивному режимі), тобто без підготовки програми. Це перетворює MATLAB у надзвичайно потужний калькулятор, здатний здійснювати не тільки звичайні для калькуляторів обчислення (наприклад, виконувати арифметичні операції й обчислювати елементарні функції), але й операції з векторами й матрицями, комплексними числами, рядами й поліномами і т.д. Можна майже миттєво задати й вивести графіки різних функцій — від простої одновимірної до складної тривимірної фігури. Робота із системою в режимі прямих обчислень носить діалоговий характер і відбувається за правилом “поставив запитання, отримав відповідь”. Користувач набирає певний запит на клавіатурі, потім (якщо потрібно) редагує його в командному рядку і передає на виконання натисненням клавіші <Enter>.

Деякі правила роботи:

- ознакою готовності системи до введення запиту є наявність курсору після символу “>>” у командному вікні;
- дані вводяться і редагуються за допомогою найпростішого рядкового редактора;
- записи запитів розрізняються залежно від використання великих і малих літер;
- якщо в кінці запиту (команди) поставити знак “;”, то результат (відповідь) не відображається у командному вікні. Якщо цього не зробити (в тому числі і в М-файлі), то після успішного обчислення буде здійснене виведення результатів на екран;
- якщо не зазначена змінна для отримання результату обчислень, то MATLAB призначає таку змінну з ім'ям *ans* ;
- знаком присвоювання є “=” ;
- результат обчислень виводиться в рядках відображення (без знака “>>”);
- у загальному випадку для подання результату в командному вікні застосовується така форма: <ім'я_змінної> = <результат>.
- вбудовані функції записуються малими літерами, і їхні аргументи вказуються в круглих дужках, наприклад, $\sin(x)$.

Вихід із системи. Завершення роботи з системою MATLAB здійснюється за допомогою команд *quit*, *exit* або натисненням комбінації клавіш <Ctrl+0>.

Завдання для аудиторної роботи

1. Присвоїти змінній x значення 5 без виведення значення на екран.
2. Вивести значення змінної x на екран.
3. Вивести значення змінної X на екран.
4. Присвоїти змінній y значення 15 з виведенням значення на екран.
5. Знайти значення виразу $x+y$.
6. Знайти значення виразу $x+i*y$.
7. Знайти значення виразу $x-j*y$.
8. Знайти значення виразу $x+i*y$ і записати це значення в змінну u (використовуючи редагування попередньої команди).
9. Знайти значення виразу $u*ans$.
10. Ввести вектор V з елементами 1,2,3,4 (набрати у командному рядку текст $V=[1,2,3,4]$ і натиснути клавішу <Enter>).
11. Обчислити функції синуса та експоненти від вектора V .
12. Ввести матрицю A розмірності 1×4 (набрати у командному рядку текст $A=[1 \ 2 \ 3 \ 4]$ і натиснути клавішу <Enter>).
13. Ввести матрицю B розмірності 4×1 (набрати у командному рядку текст $B=[1;2;3;4]$ і натиснути клавішу <Enter>).
14. Знайти матрицю $C=A*B$.
15. Знайти вектор $Y=C*V$.
16. Проаналізувати стан вікон історії та робочої області після виконання завдань 1-15.
17. Відкрити вікно редактора М-файлів і перенести до нього інформацію з вікна історії.
18. Зберегти отриманий текст у М-файлі з ім'ям *prima2.m*.
19. Очистити командне вікно.
20. Очистити вікна історії та робочої області.
21. Виконати М-файл з ім'ям *prima2.m* (набираємо *prima2* після символу ">>" у командному вікні і натискаємо клавішу <Enter>).

У деяких випадках математичний вираз, що вводиться, може виявитись настільки довгим, що для нього не вистачить одного рядка. Тоді для його перенесення на новий рядок використовується символ "..." (три або більше крапки), наприклад:

$$s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 \dots \\ 1/8 + 1/9 - 1/10 + 1/11 - 1/12 .$$

Цей прийом може бути корисним для створення наочних документів (запобігання заходу рядків у невидиму область вікна або текстового документа, який друкується на принтері). Загалом кажучи, максимальне число символів в

одному рядку командного режиму – 4096, а в М-файлі – не обмежене, проте з такими довгими рядками працювати незручно. У ранніх версіях пакета рядок мав не більше 256 символів.

Завдання для самостійної роботи

1. Розглянути команди загального призначення: *help*, *demo/expo*, *info*, *ver*, *version*, *path*, *what*, *whatsnew*, *which*, *lasterr*.
 2. Розглянути (за допомогою *help*) команди роботи з шляхами доступу до файлів: *path*, *path(s)*, *addpath*, *rmpath*.
 3. Розглянути команди роботи з поточним каталогом: *cd*, *pwd*.
 4. Розглянути команди перегляду шляхів, роботи з файлами: *dir*, *ls*, *tempdir*, *tempname*, *edit*, *type*, *delete*, *!*, *diary*.
 5. Розглянути команди для роботи з робочою областю: *clear*, *disp*, *length*, *size*, *who*, *whos*, *save*, *load*, *pack*.
 6. Створити свою (індивідуальну) папку для роботи в системі MATLAB і додати її до системних директорій MATLAB.
 7. Використовуючи системний редактор текстів створити М-файл з ім'ям *primd1.m*, в якому ввести: вектор-рядок X розмірності 10 і вектор-стовпець Y розмірності 10. Виконати нормування векторів і знайти косинус кута між векторами X і Y . Зберегти цей файл і виконати його.
 8. Створити командою *edit* М-файл з ім'ям *primd2.m*, в якому ввести вектори a , b , c розмірності 3×1 . Знайти: векторний добуток векторів a і b – вектор x , і число $H = |abc|$ – змішаний добуток векторів a , b , c . Зберегти цей файл і виконати його.
 9. Створити М-файл з ім'ям *primd3.m*, в якому ввести вектори a , b , c , d , e , f розмірності 1×3 . Знайти число $H = |abc||def|$:
$$|abc||def| = \begin{vmatrix} (a,d) & (a,e) & (a,f) \\ (b,d) & (b,e) & (b,f) \\ (c,d) & (c,e) & (c,f) \end{vmatrix}$$
 10. Створити М-файл з ім'ям *primd4.m*, в якому ввести вектори a , b , c , d і числа p , q , r . Знайти розв'язок рівнянь (при умові $a \cdot (b \times c) \neq 0$):
 - a)
$$\begin{cases} x \cdot a = p, \\ x \times a = b. \end{cases}$$
 - b)
$$\begin{cases} x \cdot a = p, \\ x \cdot b = q, \\ x \cdot c = r. \end{cases}$$
 - c) $ax + by + cz + d = 0.$
 - d) $(b \times c)x + (c \times d)y + (a \times b)z + d = 0.$
- (Вказівка: див.[4], стор.166, формули (5.2–17) – (5.2–20)).

Тема №3 ТИПИ ДАНИХ ТА ПОБУДОВА ВИРАЗІВ

Теоретичні відомості

Типи даних, які використовуються в MATLAB. Усі типи даних, з структурної точки зору, є масивами, тому в системі MATLAB:

скалярна змінна	– це матриця розмірності 1×1 ;
вектор-рядок з N елементів	– це матриця розмірності $1 \times N$;
вектор-стовпець з N елементів	– це матриця розмірності $N \times 1$.

Надалі будемо розглядати роботу тільки з такими типами даних:

double – натуральні, цілі, дійсні і комплексні дані;

char – лінійний масив (рядок).

Звичайно в MATLAB є й інші типи даних: *sparse*, *uint8*, *cell*, *struct* та інші.

Константи, змінні, імена спеціальних змінних. Як і в інших алгоритмічних мовах *константи* використовуються для позначення незмінних величин. Вони не мають імен, а тип константи визначається при її запису, наприклад,

1, -25, 435.123, -2.345, 1e-12, 1.26E8, 'значення змінної X = '.

Змінна – це певна поіменована величина, яка здатна зберігати деякі, звичайно різні за значеннями, дані відповідних типів. Тип змінної заздалегідь не декларується, а визначається виразом, значення якого привласнюється змінній.

Серед поіменованих констант виділяють *системні змінні*, які мають фіксовані імена і значення. Основні системні змінні, що використовуються в системі MATLAB, наведені в табл. 2.1.

Таблиця 3.1. Системні змінні MATLAB

Позначення	Назва
<i>i, j</i>	уявна одиниця комплексного числа
<i>pi</i>	число π
<i>eps</i>	похибка операцій над числами з плаваючою крапкою
<i>realmin</i>	найменше число з плаваючою крапкою
<i>realmax</i>	найбільше число з плаваючою крапкою
<i>inf</i>	значення машинної нескінченності (∞)
<i>ans</i>	змінна, що зберігає результат останньої операції
<i>NaN</i>	вказівка на нечисловий характер даних (Not-a-Number), позначення невизначеного результату (наприклад, $0/0$; <i>inf/inf</i>)

Закінчення табл. 3.1

Позначення	Назва
<i>nargin</i>	кількість вхідних аргументів у функції
<i>nargout</i>	кількість вихідних аргументів у функції
<i>computer</i>	тип комп'ютера
<i>flops</i>	кількість операцій з плаваючою крапкою
<i>version</i>	номер версії MATLAB

Системні змінні можна перевизначити, але це можливо лише під час поточної сесії.

У системі MATLAB для іменування змінних використовуються ідентифікатори, які обов'язково починаються з букви і можуть містити букви, цифри й символ підкреслення “_”. Ідентифікатор може мати скільки завгодно символів, але значущими є тільки 31 перший (початковий) символ.

Побудова виразів. Як і в алгоритмічній мові, вирази використовуються для запису послідовності певних обчислень над даними. Зазвичай чітко визначається пріоритет їх виконання, як для групи операцій, так і всередині групи. До першої категорії (найвищий пріоритет в обчисленнях) відносяться арифметичні операції, до другої – відношення, до третьої – логічні операції.

Операції і спеціальні символи, які використовуються при записі виразів, наведені в табл. 3.2.

Таблиця 3.2. Операції і спеціальні символи у виразах MATLAB

Символ	Призначення
:	оператор утворення масиву даних
()	дужки для групування операцій або аргументу
[]	дужки для формування матриць (векторів)
{ }	фігурні дужки для формування масивів комірок
.	десятькова крапка
..	посилання на батьківський каталог
...	продовження рядка команди
,	розділовий знак
;	блокування виведення інформації на екран
%	коментар

В арифметичних операціях використовуються дані однакової розмірності. Якщо один із операндів є скаляром, а інший ні, то скаляр розширюється до розмірів другого операнда. Набір арифметичних операцій у MATLAB складається із стандартних операцій додавання, віднімання, множення, ділення, піднесення до степеня і доповнений спеціальними матричними операціями (табл. 3.3). У табл. 3.4 наведений порядок виконання арифметичних операцій.

Таблиця 3.3. Арифметичні операції MATLAB

Символ	Призначення
+	додавання
-	віднімання
*	добуток
.*	добуток (“скалярний”)
^	піднесення до степеня
.^	піднесення до степеня (“скалярне”)
\	“ліве” ділення
/	“праве” ділення
.\	“ліве” ділення (“скалярне”)
./	“праве” ділення (“скалярне”)
'	транспонування
.'	транспонування
kron	тензорний добуток (добуток Кронекера)

Таблиця 3.4. Ієрархія при виконанні арифметичних операцій

Рівень	Операція
1	“.”, “.^”, “.’”, “.^”
2	унарні “+”, “-”
3	“.*”, “./”, “.\”, “*”, “/”, “\”
4	бінарні “+”, “-”
5	оператор формування масиву “:”

Операції відношення (табл. 3.5) мають два операнди і використовуються для порівняння двох величин, векторів або матриць (в алгоритмічній мові – предикатів).

Таблиця 3.5. Операції відношення

Оператор	Функція	Призначення	Приклад
==	<i>eq</i>	дорівнює	$x=y$ або $eq(x,y)$
~=	<i>ne</i>	не дорівнює	$x \neq y$ або $ne(x,y)$
<	<i>lt</i>	менше	$x < y$ або $lt(x,y)$
>	<i>gt</i>	більше	$x > y$ або $gt(x,y)$
<=	<i>le</i>	менше або дорівнює	$x \leq y$ або $le(x,y)$
>=	<i>ge</i>	більше або дорівнює	$x \geq y$ або $ge(x,y)$

При обчисленні предиката ми отримуємо результат 1 – *Істина* або 0 – *Хибність*.

Примітка. Оператори “<”, “<=”, “>”, “>=” при порівнянні комплексних чисел порівнюють лише їх дійсні частини.

Логічні операції і функції (табл. 3.6) застосовуються поелементно до даних і використовуються для побудови булевих виразів.

Таблиця 3.6. Логічні операції і функції

Опе- ратор	Функція	Назва	Приклад
~	<i>not</i>	заперечення	$\sim x$ або <i>not(x)</i>
&	<i>and</i>	кон'юнкція	$x \& y$ або <i>and(x,y)</i>
	<i>or</i>	диз'юнкція	$x y$ або <i>or(x,y)</i>
	<i>xor</i>	виключаюче “або”	<i>xor(x)</i>
	<i>all</i>	Істина, якщо всі елементи вектора $\neq 0$	<i>all(x)</i>
	<i>any</i>	Істина, якщо не всі елементи вектора $= 0$	<i>any(x)</i>
	<i>exist</i>	перевірка існування змінної чи функції	<i>exist('x')</i>
	<i>find</i>	пошук індексу ненульового елемента (індексів, що задовольняють певні умові)	<i>find(x)</i>
	<i>isempty</i>	Істина, якщо масив порожній	<i>isempty(x)</i>
	<i>isinf</i>	Істина, якщо елемент $= \infty$	<i>isinf(Inf)</i>
	<i>isletter</i>	Істина, якщо символом є буква	<i>isleter('a')</i>
	<i>isnan</i>	Істина, якщо елемент не є числовим	<i>isnan(NaN)</i>
	<i>isreal</i>	Істина, якщо елемент дійсний	<i>isreal(1.2)</i>
	<i>isstr</i>	Істина, якщо елементом є текстовий рядок	<i>isstr('qwerty')</i>

Пакет MATLAB містить значну кількість елементарних і спеціальних функцій (табл. 3.7 – 3.14). Зі списком елементарних функцій можна ознайомитись за допомогою команди *help elfun*, а зі списком спеціальних функцій – за допомогою команди *help specfun*. Функції можуть бути вбудованими (внутрішніми) і зовнішніми, або М-функціями. Зовнішні функції містять свої визначення в М-файлах. Вбудовані функції зберігаються у відкомпільованому ядрі системи MATLAB, внаслідок чого вони виконуються гранично швидко.

Таблиця 3.7. Тригонометричні та гіперболічні функції

Тригонометрична функція	Обернена тригонометрична функція	Гіперболічна функція	Обернена гіперболічна функція
$\sin(x)$	$\arcsin(x)$	$\sinh(x)$	$\operatorname{arsinh}(x)$
$\cos(x)$	$\arccos(x)$	$\cosh(x)$	$\operatorname{acosh}(x)$
$\tan(x)$	$\arctan(x)$	$\tanh(x)$	$\operatorname{atanh}(x)$
$\cot(x)$	$\operatorname{acot}(x)$	$\coth(x)$	$\operatorname{acoth}(x)$

Таблиця 3.8. Трансцендентні функції

Назва	Призначення
$\exp(x)$	експонента x

Закінчення табл. 3.8

Назва	Призначення
$\log(x)$	натуральний логарифм x
$\log_2(x)$	логарифм x за основою 2
$\log_{10}(x)$	логарифм x за основою 10
$\text{pow}_2(x)$	експонента x за основою 2
$\text{sqrt}(x)$	корінь квадратний з x
$\text{nextpow}_2(x)$	найближча степінь експоненти за основою $2 \geq x$

Таблиця 3.9. Функції роботи з комплексними числами

Назва	Призначення
$\text{abs}(z)$	модуль z ($ z $)
$\text{angle}(z)$	аргумент z ($\arg(z)$)
$\text{conj}(z)$	спряжене до z (z^*)
$\text{imag}(z)$	уявна частина z ($\text{Im}(z)$)
$\text{real}(z)$	дійсна частина z ($\text{Re}(z)$)

Таблиця 3.10. Функції заокруглення і модульна арифметика

Назва	Призначення
$\text{abs}(x)$	абсолютна величина x
$\text{fix}(x)$	$[x]$ – ціла частина x
$\text{floor}(x)$	округлення x до найближчого меншого цілого (результат – ціле значення)
$\text{ceil}(x)$	округлення x до найближчого більшого цілого (результат – ціле значення)
$\text{round}(x)$	округлення x до цілого (результат – дійсне значення)
$\text{mod}(x,y)$	залишок від цілочисельного ділення x на y з урахуванням знака
$\text{rem}(x,y)$	залишок від цілочисельного ділення x на y . Порівняйте: $\text{mod}(5.345,2)$, $\text{rem}(5.345,2)$; $\text{mod}(-5.5,2)$, $\text{rem}(-5.5,2)$
$\text{sign}(x)$	знак числа

Таблиця 3.11. Теоретико-числові функції

Назва	Призначення
$\text{factor}(n)$	розклад числа n на прості множники
$\text{isprime}(n)$	Істина, якщо число n просте
$\text{primes}(n)$	формування списку простих чисел, які не перевищують n
$\text{gcd}(n,m)$	найбільший спільний дільник n,m
$\text{lcm}(n,m)$	найменше спільне кратне n,m
$\text{rat}(n)$	подання числа n у вигляді ланцюгового дробу
$\text{rats}(n)$	наближене число у вигляді раціонального дробу вигляду a/b
$\text{perms}(1:n)$	можливі перестановки елементів заданого числового вектора
$\text{nchoosek}(n,k)$	число сполучень C_n^k

Таблиця 3.12. Функції для роботи з рядками символів

Назва	Призначення
<i>abs('i')</i>	код символу <i>i</i>
<i>blanks(n)</i>	створення рядка з <i>n</i> пробілів
<i>deblank(s)</i>	видалення пробілів у кінці рядка
<i>double('xyz')</i>	перетворення символів рядка 'xyz' у числове значення
<i>strcmp(s,t)</i>	порівняння рядків <i>s</i> і <i>t</i>
<i>upper(s)</i>	переведення всіх символів рядка <i>s</i> у верхній регістр
<i>lower(s)</i>	переведення всіх символів рядка <i>s</i> у нижній регістр
<i>setstr(n)</i>	перетворення числа <i>n</i> у рядок
<i>Int2str(n)</i>	перетворення цілого числа <i>n</i> у рядок
<i>num2str(x)</i>	перетворення дійсного числа <i>x</i> у рядок
<i>str2num(s)</i>	перетворення рядка <i>s</i> у число
<i>findstr(s,p)</i>	пошук місця рядка <i>p</i> у рядку <i>s</i>
<i>strrep(s,c,p)</i>	заміна в рядку <i>s</i> рядка <i>c</i> на рядок <i>p</i>
<i>strcat(s,c)</i>	горизонтальне об'єднання рядків
<i>strvcat(s,c)</i>	вертикальне об'єднання рядків
<i>eval(s)</i>	обчислення рядкових виразів

Таблиця 3.13. Функції часу та дати

Назва	Призначення
<i>clock</i>	поточний час і дата у векторній формі [рік, місяць, день, година, хвилина, секунда]
<i>date</i>	поточна дата у формі рядка
<i>now</i>	поточний час і дата у формі числа
<i>datestr(d,n)</i>	перетворення числового формату дати <i>d</i> в один із 19 вихідних рядкових форматів дати і часу (<i>n</i> – номер формату від 0 до 18)
<i>etime(t2,t1)</i>	визначення тривалості проміжку часу (в сек), заданого векторами <i>t1</i> і <i>t2</i>

Таблиця 3.14. Функції для операцій з елементами матриць

Назва	Призначення
<i>diag(A)</i>	створення або видобування діагоналі матриці. Якщо <i>A</i> є вектором, то створюється квадратна матриця, в якій елементи вектора <i>A</i> розміщені на головній діагоналі. Якщо <i>A</i> є матрицею, створюється вектор-стовпець, що складається з елементів головної діагоналі матриці <i>A</i>
<i>fliplr(A)</i>	відбиття стовпців матриці <i>A</i> відносно вертикальної осі
<i>flipud(A)</i>	відбиття рядків матриці <i>A</i> відносно горизонтальної осі
<i>isreal(A)</i>	<i>Істина</i> , якщо всі елементи матриці <i>A</i> дійсні
<i>reshape(A,m,n)</i>	зміна розмірності матриці без зміни кількості її елементів. Утворюється матриця розмірності <i>m</i> × <i>n</i> . Нехай <i>A</i> – матриця розмірності 3×4, отримай <i>reshape(A,2,6)</i> ; <i>reshape(A,3,[])</i> ; <i>reshape(A,12,[])</i> ; (див. <i>help reshape</i>)

Назва	Призначення
<i>rot90(A)</i>	поворот матриці <i>A</i> на 90 градусів проти годинникової стрілки
<i>rot90(A,k)</i>	поворот матриці <i>A</i> на $90 \cdot k$ градусів, де $k = 1, -1, 2, -2, \dots$
<i>tril(A)</i>	створення лівої трикутної матриці
<i>tril(A,k)</i>	створення лівої трикутної матриці, починаючи з діагоналі з номером <i>k</i> . При $k > 0$ це номер <i>k</i> -тої верхньої діагоналі, при $k < 0$ це номер <i>k</i> -тої нижньої діагоналі
<i>triu(A)</i>	створення правої трикутної матриці
<i>triu(A,k)</i>	створення правої трикутної матриці, починаючи з діагоналі з номером <i>k</i> . При $k > 0$ це номер <i>k</i> -тої верхньої діагоналі, при $k < 0$ це номер <i>k</i> -тої нижньої діагоналі
:	виділення стовпця/рядка в матриці, генерація векторів
<i>size(A)</i>	визначення кількості рядків і стовпців матриці <i>A</i>
<i>length(A)</i>	визначення розмірності матриці <i>A</i> (для прямокутної матриці розмірності $m \times n$ $\text{length}(A) = \max(m, n)$)

Враховуючи значну специфіку спеціальних математичних функцій, наведемо тільки їх відповідні назви:

- функції Бесселя (*bessel*, *besseli*, *besselj*, *besselk*, *bessely*);
- бета-функція (*beta*, *betainc*, *betaln*);
- еліптичні функції Якобі (*ellipj*, *ellipke*);
- гама-функції (*gamma*, *gammainc*, *gammainc*);
- функції Лежандра (*legandre*).

Для детального ознайомлення з відповідною функцією можна використати команду *help <name_function>*, де *<name_function>* – ім'я функції.

Завдання для аудиторної роботи

1. Оформити за допомогою команд управління командним вікном усі наступні пункти вправи у вигляді М-сценарію з ім'ям свого прізвища.
2. Відобразити значення системних змінних *pi*, *eps*, *realmin*, *realmax*.
3. Записати декілька прикладів обчислення предикату

$$ab \neq 0; a > b; |a| \geq 0; \|a\| - |b| \leq |a + b|; |a + b| \leq |a| + |b|; b^2 - 4ac \geq 0$$

у двох варіантах – з використанням операції і функції. Звернути увагу на отримане значення.

4. Записати декілька прикладів обчислення булевого виразу

$$p \vee q; p \& q \& r; q \supset p; x < y \vee x = y; y \geq x \& y + x \geq 0 \& y \leq 1;$$

$$(x-1)^2 + (y-2)^2 \geq 9 \& (x-3)^2 + (y+1)^2 \leq 25; 0 < x \leq 1$$

у двох варіантах – з використанням операції і функції.

5. Отримати практичні навички роботи з рядковими функціями.
6. Отримати практичні навички роботи з функціями часу і дати.

7. Увести декілька матриць розмірності 4×4 , 5×5 , 4×5 , 5×4 . Отримати практичні навички роботи з функціями над елементами матриць.

Завдання для самостійної роботи

1. Отримати практичні навички роботи з функціями заокруглення і модульної арифметики.

2. По вектору $Z=[1-2i, 1+3i, 1+2i, 2, 5i, 0]$ створити матрицю: перший рядок – це модуль відповідних компонент вектора Z , другий – аргумент, третій – дійсна частина, четвертий – уявна частина. Визначити, скільки часу (в секундах) витрачається на виконання цієї вправи.

3. Для вектора Z з завдання 2 знайти скалярний добуток Z на Z^* .

4. Для натурального числа N і комплексного числа C ($C \neq 0$) знайти розв'язок рівняння $Z^N = C$.

5. Задати довільне ціле число N . Якщо це число не просте, то знайти його розклад на прості множники і список простих чисел, які не перевищують N .

6. Задати довільне дійсне число Y . Знайти його представлення у вигляді ланцюгового та раціонального дробу.

7. Отримати практичні навички роботи з функціями для генерації спеціальних матриць: *company*, *hadamard*, *hankel*, *hilb*, *invhilb*, *magic*, *pascal*, *rosser*, *toeplitz*, *wilkinson*.

8. Для матриці A (розмірності 6×6), згенерованої за допомогою функції *magic*, виділити головну діагональ і записати у вектор d . Розкласти матрицю A на суму піддіагональної ($A1$), діагональної (D) і наддіагональної ($A2$) матриць.

9. Увести тридіагональні неособливі матриці $A1, A2, A3, A4$ розмірності 4×4 . Знайти діагональні матриці $D1, D2, D3$, виділяючи головну діагональ матриць, відповідно $A1, A2, A3$, і поділити ці матриці на їхню норму (першу, другу, третю відповідно). Побудувати з цих матриць таку блочну матрицю A :

$$A = \begin{pmatrix} A1 & D1 & 0 & 0 \\ D1 & A2 & D2 & 0 \\ 0 & D2 & A3 & D3 \\ 0 & 0 & D3 & A4 \end{pmatrix}$$

10. Для матриці A з завдання 9, знайти матриці (без збереження результату):

- відбиттям стовпців матриці A відносно вертикальної (горизонтальної) осі;
- зміною розмірності матриці без зміни кількості її елементів, послідовно 8×32 , 32×8 , 16×16 ;
- поворотом матриці A на 90 градусів проти годинникової стрілки;
- створенням лівої трикутної матриці, починаючи з четвертої піддіагонали.

11. Виконати завдання для самостійної роботи з теми 2, використовуючи для знаходження визначника функцію $\det(A)$.

Тема № 4 ОСНОВНІ ДІЇ ЛІНІЙНОЇ АЛГЕБРИ

Теоретичні відомості

Робота з матрицями, матричний аналіз, власні числа. Масиви є основними об'єктами в системі MATLAB. Для створення масивів розмірності більше 2 можна її “нарощувати” в такий спосіб: $A=[1,2;3,4]$; $A(:,2)=[1,0;5,2]$; $A(:,3)=[0,20;50,-2]$; в результаті отримаємо матрицю A розмірності $2 \times 2 \times 3$.

У табл. 4.1 наведені функції формування масивів і матриць, операції над матрицями.

Таблиця 4.1. Функції для роботи з матрицями

Назва	Призначення
$eye(n)$ $eye(n,m)$ $eye(n,m,k,...)$	створення одиничної матриці розмірності $n \times n$ або матриці (матриць) розмірності $n \times m$ з одиницями на головній діагоналі. Порівняйте: $eye(5,1)$; $eye(1,5)$; $eye(3,5)$; $eye(5,5)$; $eye(5)$; $eye(3,5,4)$
$ones(n)$ $ones(n,m)$ $ones(n,m,k,...)$	створення матриці розмірності $n \times n$ або матриці (матриць) розмірності $n \times m$ з одиниць. Порівняйте: $ones(5,1)$; $ones(1,5)$; $ones(3,5)$; $ones(5,5)$; $ones(5)$; $ones(3,5,4)$
$zeros(n)$, $zeros(n,m)$ $zeros(n,m,k,...)$	створення матриці розмірності $n \times n$, що містить нулі. Якщо n не є скалярною величиною, то з'явиться повідомлення про помилку. Порівняйте: $zeros(5,1)$; $zeros(1,5)$; $zeros(3,5)$; $zeros(5,5)$; $zeros(5)$; $zeros(3,5,4)$
$[X,Y]=meshgrid(x,y)$	створення матриць розмірності $m \times n$, що містить координати прямокутної сітки $M(i,j)=(X(i,j),Y(i,j))$, де X – матриця, утворена копіюванням в рядки значень одновимірного вектора x (ординат) довжини n , а Y – матриця, утворена копіюванням в стовпці значень одновимірного вектора y (абсцис) довжини m .
$det(A)$	обчислення визначника (детермінанта) квадратної матриці A
$norm(A,n)$	обчислення норми матриці/вектора, $n = 1, 2, inf, 'fro'$. $norm(A) \equiv norm(A,2)$; $'fro'$ – Фробеніусова норма
$cond(A,n)$	обчислення числа обумовленості матриці в нормі $n = 1, 2, Inf, 'fro'$. $cond(A) \equiv cond(A,2)$
$rank(A)$	обчислення рангу матриці A
$trace(A)$	обчислення сліду матриці A
$orth(A)$	створення ортонормованого базису матриці A : $h=orth(A)$; $h'*h=E$
$null(A)$	нуль-простір матриці A
$inv(A)$	утворення матриці, оберненої до квадратної матриці A . Видається попередження, якщо A – погано обумовлена або близька до виродженої

Назва	Призначення
$\text{pinv}(A)$	утворення матриці, псевдооберненої до квадратної матриці A . Результатом є матриця B того ж розміру, що й A і задовольняє умовам $A*B*A=A$ та $B*A*B=B$. Обчислення ґрунтується на використанні функції $\text{svd}(A)$ і прирівнюванні до нуля всіх сингулярних чисел, менших величини tol
$[Q,R]=\text{qr}(A)$	QR -розклад матриці A : $A=Q*R$, де Q – унітарна матриця, R – верхня трикутна матриця
$[L,U]=\text{lu}(A)$	LU -розклад матриці A : $A=L*U$, де L – нижня трикутна матриця, U – верхня трикутна матриця
$\text{dot}(X,Y)$	обчислення скалярного добутку векторів X,Y
$\text{cross}(X,Y)$	обчислення векторного добутку векторів X,Y
$[H,L]=\text{eig}(A)$	обчислення власних значень і власних векторів матриці A . Тут L – діагональна матриця власних значень, H – матриця правих власних векторів, що задовольняють співвідношення $A*H = H*L$. Ці вектори $H(:,k)$ нормовані так, що норма кожного з них дорівнює одиниці
$\text{poly}(A)$	обчислення характеристичного полінома для матриці A
$[U,S,V]=\text{svd}(A)$	розклад $A=U*S*V'$, де U,V – унітарні матриці, S – діагональна матриця

Примітка. На практиці обчислення оберненої матриці в явному вигляді не так вже й необхідне. Цю операцію застосовують при розв'язанні системи лінійних рівнянь вигляду $Ax=b$. Дійсно, один із шляхів – обчислення $x=\text{inv}(A)*b$. Але кращим (з погляду мінімізації часу розрахунку) є використання оператора матричного ділення $x=A\b b$. Ця операція використовує метод виключення Гаусса без явного формування оберненої матриці.

Нижче наведено приклад 1, в якому формуються матриця A і вектор xx ; знаходиться вектор b – добуток матриці A на вектор xx ; розв'язується СЛАР $Ax=b$ з використанням QR - і LU -розкладів матриці A .

```
% Приклад 1
clc
% Завдання вхідних даних
xx=[1;2;3];
A=[ 7, -1, 1;...
   -1, 10, 2;...
    1,  2,-4]
b=A*xx
pause

% 1-й варіант
x=A\b
pause

% 2-й варіант
[Q,R]=qr(A);
% Перевірте Q*Q' і Q'*Q
x=R\ (Q'*b)
pause
```

```
% 3-й варіант
[L,U]=lu(A);
x=U\ (L\b)
pause
```

Поліноми та дії над ними. В MATLAB поліном задається і зберігається у вигляді вектора, елементами якого є коефіцієнти полінома. В табл. 4.2 наведені функції, призначені для дій над поліномами.

Таблиця 4.2. Функції для роботи з поліномами

Назва	Значення
<i>polyval(P,x)</i>	обчислення значення полінома n -го степеня при заданому значенні змінної x : $y=P_1*x^n+\dots+P_n*x+P_{n+1}$
<i>roots(P)</i>	обчислення коренів полінома
<i>polysum(P,Q)</i>	обчислення суми поліномів
<i>conv(P,Q)</i>	обчислення добутку поліномів
<i>[Q,R]=deconv(B,A)</i>	ділення поліномів; частка від ділення повертається у вигляді вектора Q , залишок – у вигляді вектора R , так що виконується співвідношення $B=A*Q+R$
<i>polyder(P)</i>	обчислення похідної від полінома
<i>[Q,R]=polyder(B,A)</i>	$(B/A)' = Q/R$ – похідна від відношення поліномів повертається у вигляді відношення поліномів
<i>polyfit(x,Y,n)</i>	апроксимація функції $Y(x)$ поліномом степеня n
<i>[R,P,K]=residue(B,A)</i>	розклад $B(x)/A(x)$ на прості дробі: $B(x)/A(x) = R_1/(x-P_1) + R_2/(x-P_2) + \dots + R_n/(x-P_n) + K(x)$. Якщо корінь P_j має кратність t , будуть утворені вирази $R_j/(x-P_j) + R_{j+1}/(x-P_j)^2 + \dots + R_{j+t-1}/(x-P_j)^t$

Аналіз даних. У системі MATLAB визначені функції для аналізу та статистичної обробки даних, заданих у вигляді числових масивів (табл. 4.3).

Таблиця 4.3. Функції для аналізу і обробки числових масивів

Назва	Значення
<i>max(x)</i>	максимальна компонента масиву
<i>min(x)</i>	мінімальна компонента масиву
<i>mean(x)</i>	середнє значення елементів масиву
<i>median(x)</i>	серединне значення елементів масиву (медіана)
<i>std(x)</i>	стандартне відхилення елементів масиву
<i>sum(x)</i>	сума елементів масиву
<i>prod(x)</i>	добуток елементів масиву
<i>cumsum(x)</i>	сумування з накопиченням: $y_i = \sum_{j=1}^i x_j$

Назва	Значення
$\text{cumprod}(x)$	добуток з накопиченням: $y_i = \prod_{j=1}^i x_j$
$\text{sort}(x)$	сортування елементів масиву за зростанням

Завдання для аудиторної роботи

1. Отримати практичні навички роботи з функціями для роботи з матрицями.
2. Отримати практичні навички роботи з функціями для аналізу і обробки числових масивів.
3. Для матриці магічного квадрату розмірності 5×5 знайти характеристичний поліном і його корені, обчислити його значення в системі точок $[\text{min_корінь}, \text{max_корінь}]$ з кількістю точок не менше 50.
4. Знайти розв'язок системи

$$\begin{cases} Ax + By = f_1 \\ Cx + Dy = f_2 \end{cases},$$

де

$$A = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -3 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \quad C = \begin{pmatrix} -2 & 0 & 2 \\ 3 & 0 & 4 \end{pmatrix}, \quad D = \begin{pmatrix} 5 & 1 \\ 1 & -5 \end{pmatrix},$$

$$f_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \quad f_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

(Вказівка: створити СЛАР $Sz=f$, де S і f – блочна матриця і вектор відповідно).

Завдання для самостійної роботи

1. Отримати поліном, якщо задано коефіцієнт a при його старшому степені і корені полінома – вектор X .
2. Знайти корені полінома $x^5 + 3x^2 - 5$ і обчислити його значення в системі точок $x = -30:1:30$.
3. Отримати залишок від ділення полінома з завдання 1 на $(x - c)$, де c – відоме комплексне число. (Вказівка: використати теорему Безу).
4. Отримати поліном $(a + b * x)^n$ для заданих дійсних чисел a , b і натурального числа n . Використати: а) біном Ньютона; б) добуток поліномів.
5. Отримати поліном $a^n - (b * x)^n$ для заданих дійсних чисел a , b і натурального числа n .

6. Отримати поліном $(a * x)^n - b^n$ для заданих дійсних чисел a, b і парного натурального числа n .

7. Отримати поліном $a^n + (b * x)^n$ для заданих дійсних чисел a, b і непарного натурального числа n .

8. Визначити, чи має поліном $p(x)$ кратні корені. (Вказівка: якщо поліном $p(x)$ має кратні корені, то $p(x)$ і $p'(x)$ мають спільний дільник).

9. Нехай у дійсному векторі X записано довільним чином корені полінома $p(x)$. Знайти цей поліном у системі MATLAB і обчислити його значення на відрізку $[min(X), max(X)]$ з кількістю вузлів не менше 10.

10. Нехай у векторі P з комплексними коефіцієнтами записано поліном $p(x)$. Знайти його корені і отримати два вектори, впорядкувавши корені :

а) по зростанню модулів;

б) по спаданню аргументів.

11. Знайти явний вигляд (у системі MATLAB) полінома Чебишова $T_n(x)$ (для $n=7$), використовуючи рекурентні формули:

$$\begin{aligned}T_N(x) &= 2xT_{N-1}(x) - T_{N-2}(x), N = 2, \dots, n, \\T_0(x) &= 1, T_1(x) = x,\end{aligned}$$

а також його корені, і впорядкувати по зростанню (спаданню).

12. Знайти явний вигляд (у системі MATLAB) полінома Ерміта $H_n(x)$ (для $n=7$), використовуючи рекурентні формули:

$$\begin{aligned}H_N(x) &= 2xH_{N-1}(x) - 2(N-1)H_{N-2}(x), N = 2, \dots, n, \\H_0(x) &= 1, H_1(x) = 2x,\end{aligned}$$

а також значення $H_7(x)/7^3$ на відрізку $[0, 2]$ з кількістю вузлів не менше 10.

13. Знайти найбільший спільний дільник поліномів $F(x)$ і $f(x)$, де:

$$\begin{aligned}F(x) &= 3x^4 - 9x^3 - 9x^2 + 33x - 18, \\f(x) &= x^5 - 4x^3 + 2x^2 + 3x - 2.\end{aligned}$$

14. Знайти розклад на прості дроби відношення поліномів $F(x)$ і $f(x)$.

15. Виконати завдання 10 самостійної роботи теми 2, використовуючи функції: `dot()`, `cross()`, `det()`.

16. Для матриці A з завдання 9 самостійної роботи теми 3 знайти основні характеристики: визначник, норми (1, 2, 3-тю, Фробеніусову), число обумовленості, обернену матрицю, власні значення і власні вектори.

17. Розв'язати крайову задачу для ЗДР

$$\begin{cases} u'' - 5u = -x^2 + 1, \\ u|_{x=0} = 1, u|_{x=1} = 2, \end{cases}$$

використовуючи різницьевий метод. (Вказівка: побудувати СЛАР з квадратною матрицею і розв'язати її з використанням оператора матричного ділення).

Тема № 5

НАБЛИЖЕНІ ОБЧИСЛЕННЯ ЗАСОБАМИ MATLAB

Теоретичні відомості

Чисельне інтегрування, диференціювання, мінімізація функцій і знаходження коренів, апроксимація і інтерполяція даних. До бібліотеки MATLAB входять функції (табл. 4.4), що реалізують низку методів чисельного інтегрування та диференціювання, оптимізації, розв'язання нелінійних рівнянь, апроксимації та інтерполяції даних.

Таблиця 4.4. Функції чисельного інтегрування, диференціювання, мінімізації функцій і знаходження коренів, апроксимації та інтерполяції даних

Назва	Значення
<i>trapz(y)</i>	чисельне інтегрування методом трапецій для дискретної функції <i>y</i>
<i>quad('f',a,b)</i>	чисельне інтегрування методом Сімпсона для функції <i>f(x)</i> , описаної в М-функції <i>f.m</i> , де $x \in [a,b]$
<i>quadl('f',a,b)</i>	чисельне інтегрування (підвищеної точності) методом Сімпсона для функції <i>f(x)</i> , описаної в М- функції <i>f.m</i> , де $x \in [a,b]$
<i>dblquad('f',a,b,c,d)</i>	чисельне інтегрування методом Сімпсона для функції <i>f(x,y)</i> , описаної в М- функції <i>f.m</i> , де $x \in [a,b]$, $y \in [c,d]$
<i>diff(x)</i>	обчислення скінчених різниць, тобто значень вигляду $x_2 - x_1$, $x_3 - x_2$, ..., $x_n - x_{n-1}$
<i>[px, py] = gradient(z,hx,hy)</i>	обчислення дискретного аналога градієнта функції двох змінних <i>z(x,y)</i> . Тут <i>hx,hy</i> – кроки диференціювання по <i>x</i> і по <i>y</i> , <i>px, py</i> – матриці, що містять компоненти градієнта
<i>fmin('f',x_n,x_k)</i> <i>fminbnd('f',x_n,x_k) y V7.6</i>	знаходження мінімуму функції однієї змінної на інтервалі $[x_n, x_k]$
<i>fmins('f',x₀)</i> <i>fminsearch('f',x₀) y V7.6</i>	знаходження мінімуму функції декількох змінних в околі точки x_0
<i>fzero('f',x)</i>	знаходження нуля функції поблизу точки <i>x</i> , наприклад: <i>fzero('x^3-27',1)</i>
<i>w=interp1(x,y,t)</i>	одновимірна лінійна інтерполяція для функції <i>y</i> , заданій на сітці <i>x</i> ; вихідний масив <i>w</i> визначається на сітці <i>t</i> . Фрагмент М-сценарію: <i>x=0:10;</i> <i>y=sin(x);</i> <i>t=0:0.25:10;</i> <i>w=interp1(x,y,t);</i> <i>plot(x,y,'o',t,w)</i>
<i>w=spline(x,y,t)</i>	інтерполяція функції однієї змінної кубічним сплайном

Назва	Значення
<code>w=griddata(x,y,z,t,r,M)</code>	двовимірний інтерполяція функції $z(x,y)$ на нерівномірній сітці; t,r – точки сітки, на якій визначається вихідний масив w , M – метод: 'linear' (за замовчуванням), 'cubic'

Розв'язання задачі Коші для звичайних диференціальних рівнянь (ЗДР). Бібліотека MATLAB містить декілька функцій (табл. 4.5), що реалізують різні методи розв'язання задачі Коші для ЗДР (ODE, ordinary differential equations). Їх реалізації названі розв'язувачами ODE. Синтаксично ці функції розрізняються лише іменами (точніше кажучи, алфавітно-цифровими добавками до символів ode), спосіб звернення до них однаковий. Ці функції використовують методи різного порядку (іноді метод і/або порядок змінюється від кроку до кроку), існують й інші внутрішні відмінності.

Таблиця 4.5. Функції для розв'язання задачі Коші для звичайних диференціальних рівнянь

Назва	Значення
<code>[t,Y]=ode45('F',[t0 t1],Y0)</code>	розв'язання задачі Коші для нежорстких ЗДР $Y'=F(t,Y)$ методом Рунге-Кутта 4-го порядку. Тут F визначається М-функцією, $[t_0 t_1]$ – відрізок, на якому шукаємо розв'язок, Y_0 – початкові умови задачі Коші в точці t_0 , t – поточний час, Y – двовимірний масив результату
<code>[t,Y]=ode23('F',[t0 t1],Y0)</code>	розв'язання задачі Коші для нежорстких ЗДР методом Рунге-Кутта 2-го порядку
<code>[t,Y]=ode23s('F',[t0 t1],Y0)</code>	розв'язання задачі Коші для жорстких ЗДР методом Рунге-Кутта 2-го порядку

Приклад 2. Знайти наближений розв'язок задачі Коші для системи ЗДР на відрізку $[0, 10]$ (рівняння Лотки-Вольтерри для моделювання процесу “хижак-жертва”):

$$\begin{cases} u' = u - \alpha * u * v, \\ v' = v - \beta * u * v, \\ u(0) = 1, \\ v(0) = 1. \end{cases}$$

```
% Приклад 2
% М-функція p2_lotka.m
function yp=p1_lotka(t,y)
% p2_lotka - рівняння Лотки-Вольтерри для моделювання
% процесу "хижак-жертва"
global ALPFA BETA
yp=[y(1)-ALPFA.*y(1).*y(2); -y(2)+BETA.*y(1).*y(2)];
return
```

```
% М-файл p2_main.m - головна програма
clc
global ALPFA BETA
ALPFA=0.01;
BETA =0.02;
[t,y]=ode23(@p2_lotka,[0 10],[1;1]);
```

Завдання для аудиторної роботи

1. Знайти лінійну інтерполяцію для функції x^*x-1 на відрізку $[-2, 2]$ з кількістю вузлів не менше 20.
2. Знайти наближений розв'язок задачі Коші для рівняння Лотки-Вольтерри на відрізку $[0, 10]$, використавши метод Рунге-Кутта 4-го порядку точності.
3. Знайти мінімум функції x^5+3x^2-5 на інтервалі $[-20, 20]$.
4. Знайти інтеграл $\int_0^5 F(x)dx$ для підінтегральної функції $F(x) = x^3 + 3x^2 - 5$, використовуючи узагальнений метод а) трапецій; б) Сімпсона.

Завдання для самостійної роботи

1. Знайти інтеграл $\int_0^5 F(x)dx$, де $F(x) = e^{-x^2} \sin(x+5)$, використовуючи методи Сімпсона підвищеної точності.
2. Знайти інтеграл $\int_{-2}^2 \int_{-1}^2 F(x,y)dx dy$ для підінтегральної функції $F(x,y) = (x^2 + y^2 - 1)^2 + ((x-1)^2 + y^2 - 1)^2$.
3. Для полінома x^5+3x^2-5 на інтервалі $[-20, 20]$ з кроком 0,5 побудувати кубічний сплайн.
4. Розв'язати задачу Коші для системи ЗДР

$$\begin{cases} u_1' = \sin(u_1 + x), \\ u_2' = \cos(u_2 * x), \\ u_1|_{x=0} = 1, u_2|_{x=0} = 2. \end{cases}$$
 на відрізку $[0, 5]$.
5. Для функції $F(x,y) = (x^2 + y^2 - 1)^2 + (0.75x^3 - y + 0.9)^2$ знайти екстремум з початковим наближенням $M(x,y) = (0,5, -0,5)$.
6. Знайти нуль функції x^5+3x^2-5 в околі точки 1,5.
7. Для функції завдання 5 побудувати двовимірну інтерполяцію: а) лінійну з нерівномірними кроками; б) кубічну з рівномірними кроками по відповідних координатах.

Тема № 6 ОСНОВИ ПОБУДОВИ ГРАФІКІВ

Теоретичні відомості

Побудова графіків функції на площині. Одна з переваг системи MATLAB – широкий спектр засобів графіки: від команд побудови простих графіків функцій однієї змінної в декартовій системі координат і до комбінованих презентаційних графіків з елементами анімації.

Команда $\text{plot}(x,y,'s')$ використовується для побудови графіків функцій однієї змінної у декартовій системі координат на площині. Координати точок (x,y) беруться з векторів однакового розміру x, y ; s – рядок з трьох символів, значення яких наведені в табл. 5.1.

Таблиця 5.1. Символи управління кольором лінії, стилем лінії, типом маркера

1-й символ		2-й символ		3-й символ	
колір лінії		форма маркера		стиль лінії	
y	yellow (жовтий)	пробіл	відсутність маркера		відсутність лінії
m	magenta (малиновий)	.		–	суцільна
c	cyan (бірюзовий)	O		:	пунктирна
r	red (червоний)	X		–.	штрих-пунктирна
g	green (зелений)	+		--	штрихова
b	blue (синій)	*			
w	white (білий)	S			
k	black (чорний)	D			
		V			
		$\wedge$			
		$<$			
		$>$			
		P			
		H			

При побудові графіків використовуються додаткові команди:

- grid on/off – побудова/зняття ліній сітки;
- title('текст') – розміщення тексту над графіком (зверху);
- xlabel('текст') – розміщення тексту вздовж X -осі;
- ylabel('текст') – розміщення тексту вздовж Y -осі;
- legend() – опис (пояснення) елементів графіка;
- text() – розміщення тексту на графіку.

За допомогою функцій *bar*, *comet*, *hist*, *polar*, *compass*, *errorbar*, *feather*, *fplot*, *rose*, *stairs*, *stem* можуть також будуватись спеціальні X–Y графіки.

Побудова графіків функції в тривимірному просторі. Тривимірні (3–D) графіки будуються приблизно так само, як і X–Y графіки. Замість команди *plot* використовується її 3–D аналог *plot3(x,y,z)* – побудова ліній і точок в 3–D просторі.

Можуть також будуватись спеціальні 3–D графіки:

- *contour* – лінії рівня;
- *contour3* – тривимірні лінії рівня;
- *mesh*, *waterfall* – 3–D сітчаста поверхня;
- *meshc* – 3–D сітчаста поверхня з проекцією ліній постійного рівня;
- *meshz* – 3–D сітчаста поверхня з площиною відліку на нульовому рівні;
- *surf* – затінена 3–D сітчаста поверхня;
- *surfc* – затінена 3–D сітчаста поверхня з проекцією ліній постійного рівня;
- *surfl* – затінена 3–D сітчаста поверхня з підсвічуванням.

При побудові спеціальних 3–D графіків використовуються додаткові команди:

- *shading* – задає спосіб фарбування складових елементів сітчастої поверхні. Допускає використання одного з трьох параметрів – *faceted*, *flat* або *interp*. За умовчанням вважається встановленим режим *shading faceted*. Це призводить до зафарбовування кожного елемента поверхні, обмеженого чотирма точками, власним кольором і відображенням між ними чорних ребер. В режимі *flat* ребра не відображаються, а в режимі *interp* додатково здійснюється згладжування кольорів між сусідніми ділянками поверхні. Режим *interp* відповідає застосуванню алгоритму фарбування Гуро, при якому здійснюється білінійна інтерполяція кольорів всередині кожної грані і між сусідніми гранями. Ребра при цьому не вимальовуються, а створюється видимість суцільної поверхні. В режимі *flat* грані теж не відтворюються, але їх кольори зберігаються. За рахунок цього суміжні грані одного кольору об'єднуються у крупніші утворення;
- *colormap* – встановлює палітру фарбування за допомогою параметрів, що задають кольори (*gray*, *bone*, *copper*, *pink* та інші). Для перегляду повного набору можна скористатися командою *help colormap*;
- *alpha p* – задає ступінь прозорості сітчастої поверхні, де *p* приймає значення з відрізка $[0, 1]$. При $p=0$ поверхня повністю прозора, при $p=1$ – непрозора (по відношенню до невидимих елементів);
- *view(az,el)* – визначає точку знаходження камери спостерігача (надає можливість виконувати необхідні обертання поверхні). Має параметр

az , що визначає азимут – кут повороту камери навколо осі Oz в градусах. Відлік ведеться від від'ємного напрямку осі Oy . Параметр el визначає підвищення – кут в градусах, утворений вектором, який з'єднує початок координат з точкою огляду, і площиною Oxy . Команда $view(x,y,z)$ задає декартові координати точки зору. Виклик функції $view$ без параметрів дозволяє визначити параметри поточної точки зору. Скорочені варіанти виклику функцій $view(2)$ і $view(3)$ встановлюють значення азимута і кута підвищення за умовчанням для плоских ($az=0^\circ$, $el=90^\circ$) і об'ємних ($az=-37,5^\circ$, $el=30^\circ$) зображень.

Приклад 3. Доповнити текст програми прикладу 2 операторами виведення графіку розв'язку (t, y) :

```
plot(t,y);
title('Модель хижак-жертва');
ylabel('Модель хижак-жертва');
xlabel('t');
```

Приклад 4. Побудувати графіки алгебраїчних кривих: параболу Нейля, астроїду, кардіоїду, равлик Паскаля і трансцендентних кривих: спіраль Архімеда, просторову спіраль, епіциклоїду і гіпоциклоїду [4] (рис. 6.1–6.9).

```
% Приклад 4
% Деякі приклади побудови алгебраїчних і трансцендентних
кривих
clc
clear all

% параболо Нейля
a=1;
Y=[-15:0.2:15];
X=((Y./a).^2).^(1./3);
y=[-15:0.2:15];
x=y;
plot(X,Y,0,y,x,0)
title('парабола Нейля');
pause

% Астроїда
a=10;
t=[0:0.05:2.*pi];
X=a.*cos(t).^3;
Y=a.*sin(t).^3;
plot(X,Y)
title('Астроїда');
grid on
pause
```

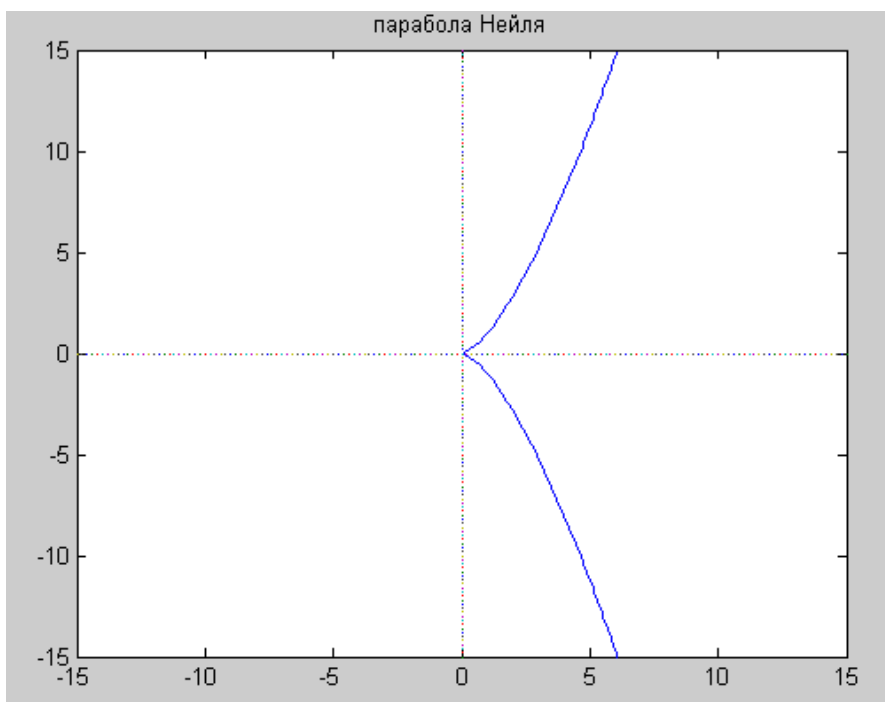


Рис.6.1. Графік параболы Нейля

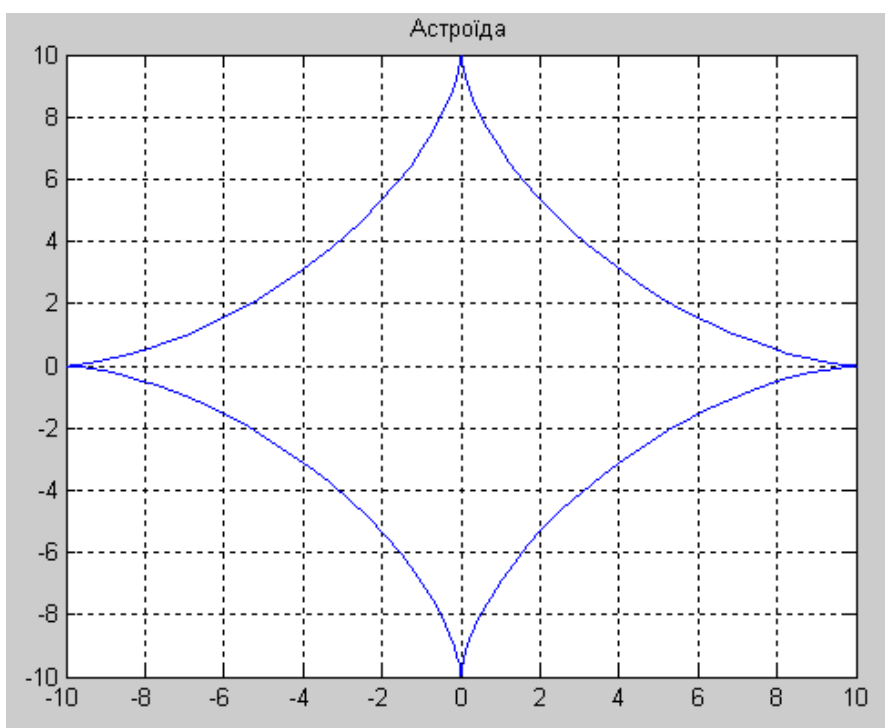


Рис. 6. 2. Графік астроїди

```
% Кардіоїда
polar(t,a.*(1+cos(t)))
title('Кардіоїда');
grid off
pause
```

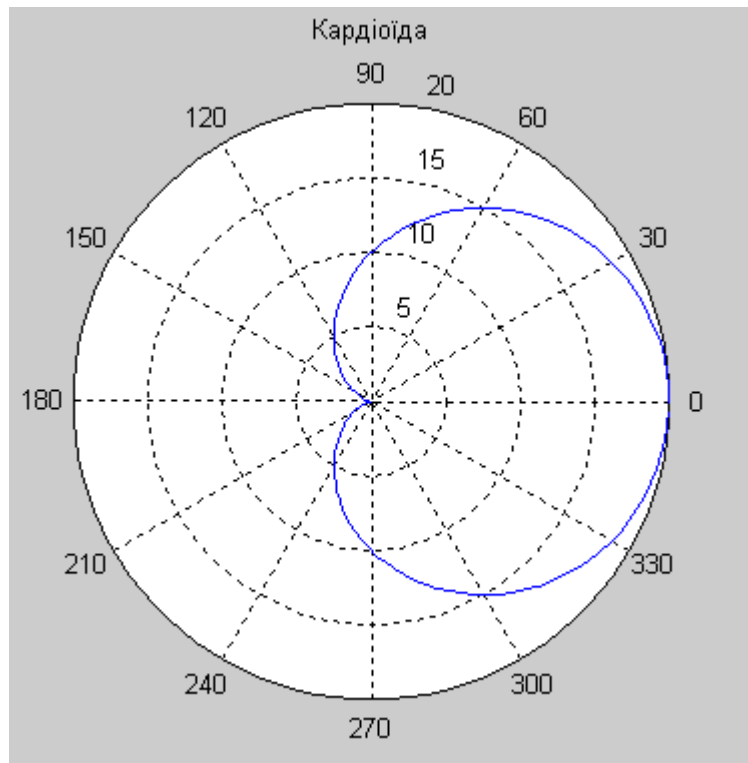


Рис. 6.3. Графік кардіоїди

```
% Равлик Паскаля
a=5;
b=1;
polar(t,b+a.*cos(t))
title('Равлик Паскаля (a>b)');
pause

a=1;
b=5;
polar(t,b+a.*cos(t))
title('Равлик Паскаля (a<b)');
pause

% Спіраль Архімеда
a=10;
polar(t,a.*t)
title('Спіраль Архімеда');
pause

% Просторова спіраль
z=1:0.25:40;
x=z.*cos(z);
y=z.*sin(z);
plot3(x,y,z);
grid on
title('Просторова спіраль');
pause
```

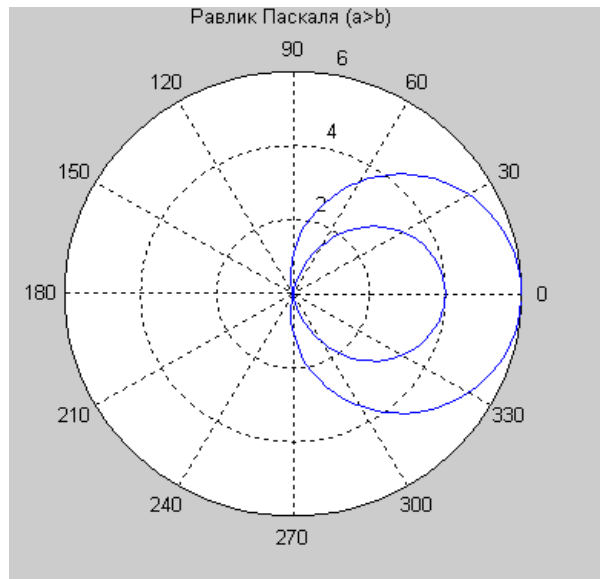


Рис. 6.4. Графік равлика Паскаля

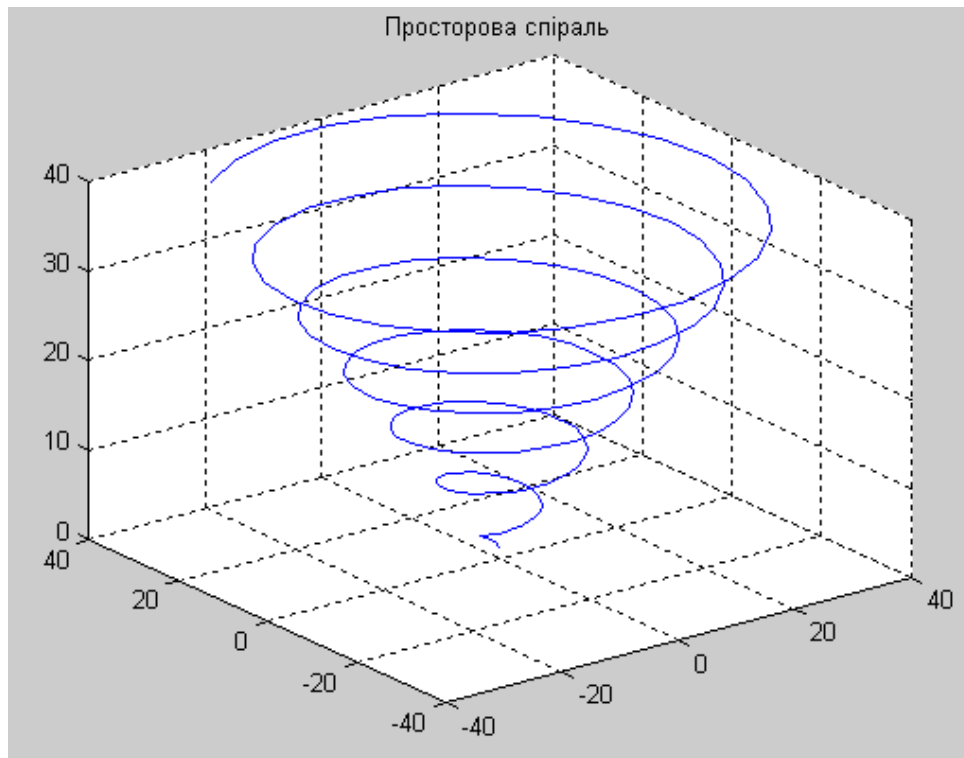


Рис. 6.5. Графік просторової спіралі

```
% Епіциклоїда і Гіпоциклоїда
a=2;
b=6;
a1=1;
t=[0:0.05:6.*pi];
X=(a+b).*sin(a./b.*t)-a1.*sin((a+b)./b.*t);
Y=(a+b).*cos(a./b.*t)-a1.*cos((a+b)./b.*t);
X1=(a-b).*sin(a./b.*t)-a1.*sin((a-b)./b.*t);
Y1=(a-b).*cos(a./b.*t)+a1.*cos((a-b)./b.*t);
```



```

plot(X,Y,X1,Y1,' :')
grid off;
legend('Епіциклоїда','Гіпоциклоїда','Location','Best');

```

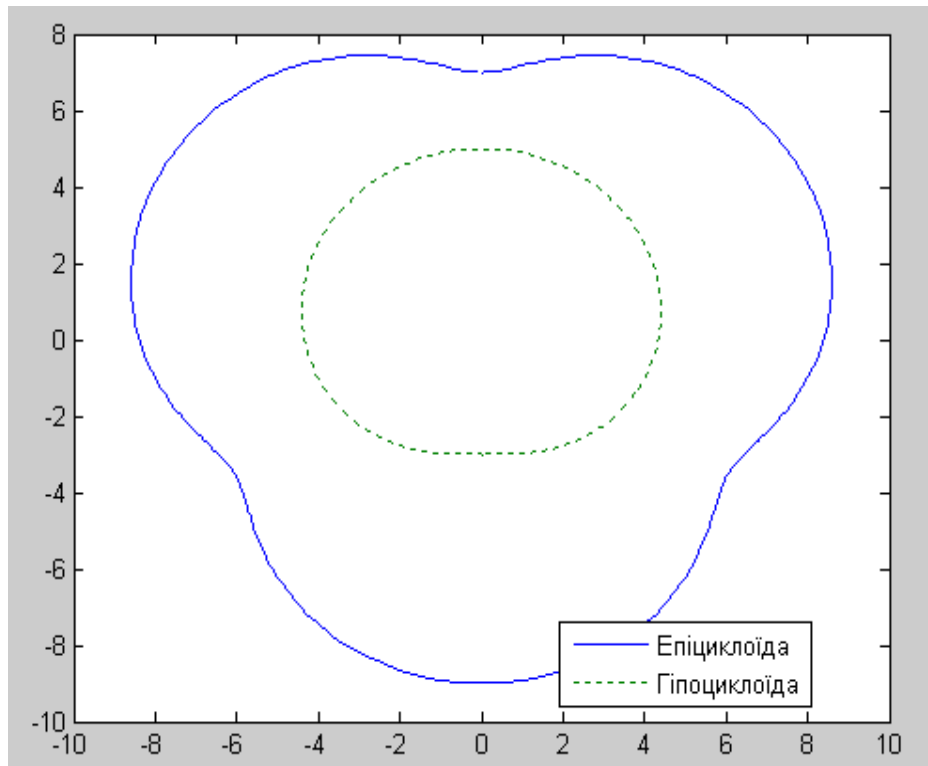


Рис. 6.6. Графік епіциклоїди і гіперциклоїди

Приклад 5. Побудувати графік поверхні гіперболічного параболоїда.

```

% Приклад 5
% Графік для гіперболічного параболоїда
p=7;
q=3;
pov=@(x,y) (0.5.*(x.^2./p-y.^2./q));
x=-10:0.05:10;
y=-10:0.05:10;
[X,Y]=meshgrid(x,y);
Z=pov(X,Y);

mesh(X,Y,Z)
colormap hot
pause
surf(X,Y,Z)
shading interp
colormap pink
pause(2)
contour(X,Y,Z,20)
colormap hsv

```

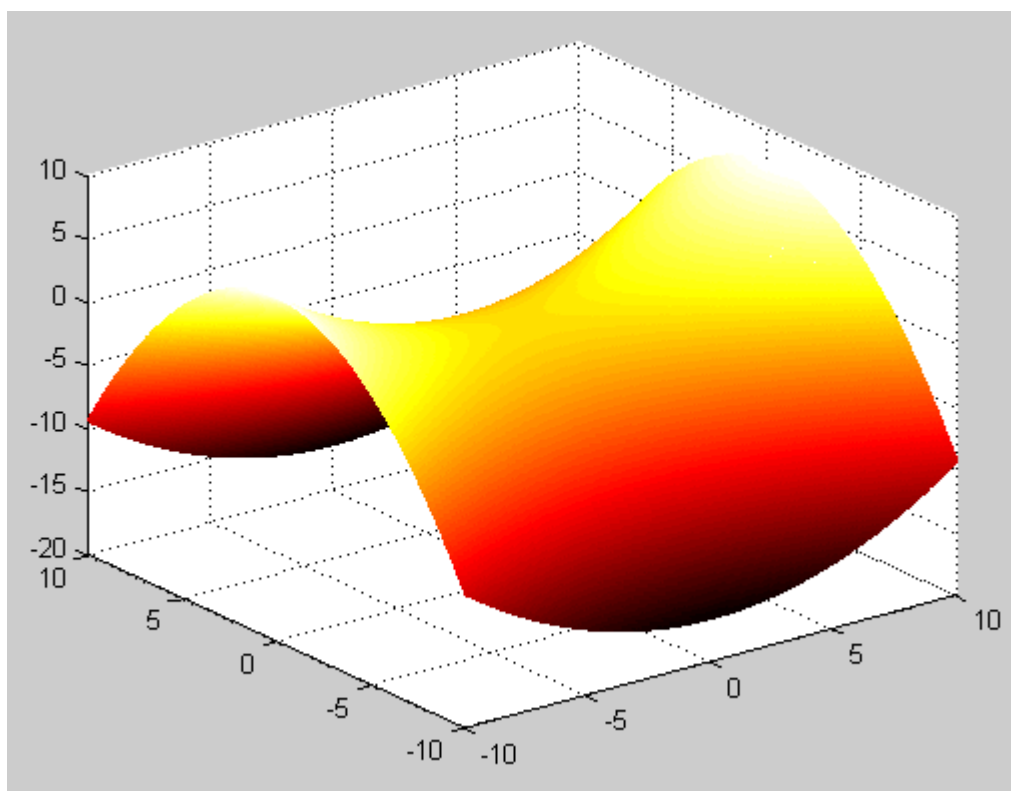


Рис. 6.7. Графік для гіперболічного параболоїда: `mesh()`

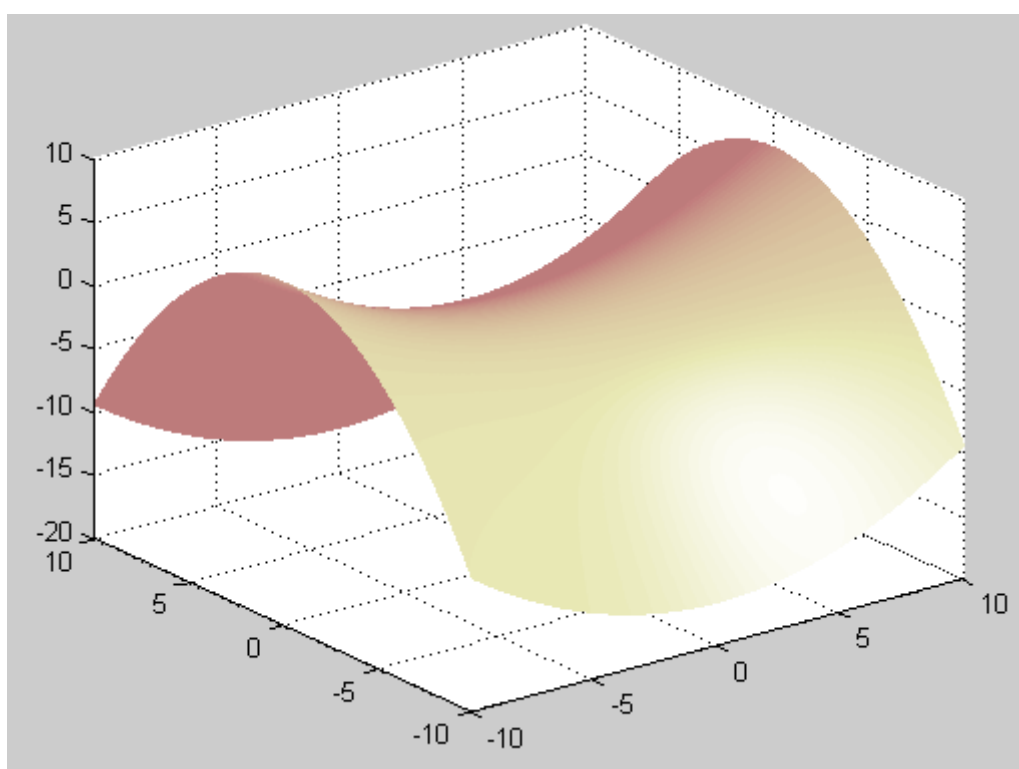


Рис. 6.8. Графік для гіперболічного параболоїда: `surf()`

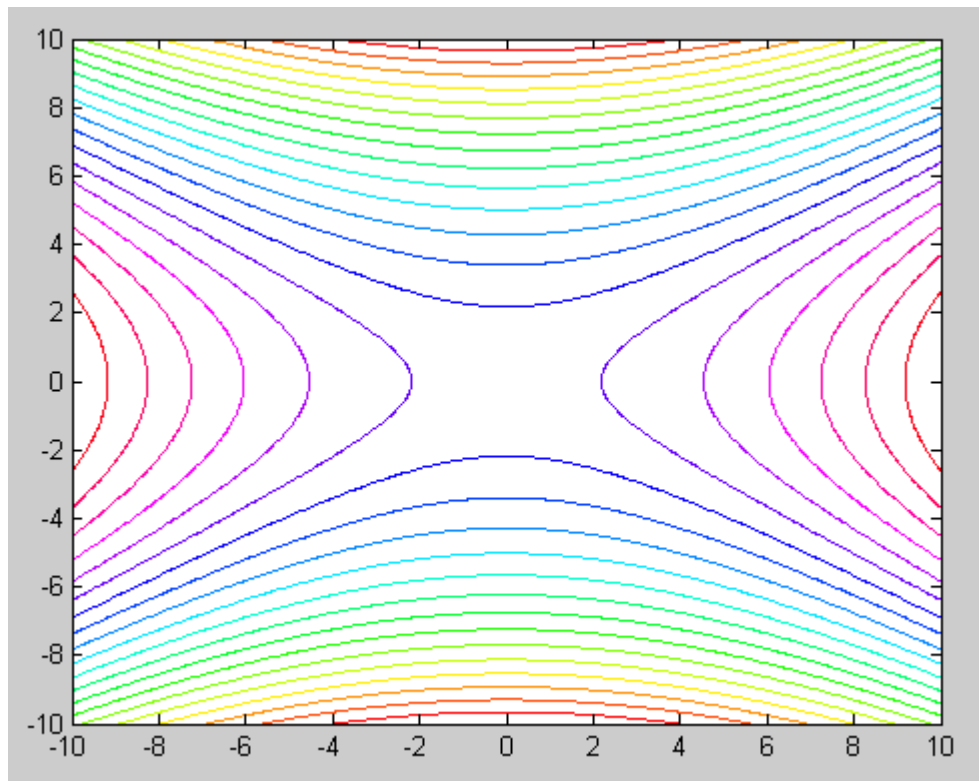


Рис. 6.9. Графік для гіперболічного параболоїда: contour()

Завдання для аудиторної роботи

1. Побудувати графіки підінтегральних функцій з завдання 1–2 теми 5 за допомогою функцій *plot* і *plot3* відповідно.
2. Побудувати графік полінома з завдання 3 теми 5.
3. Побудувати графік розв'язку задачі Коші для системи ЗДР з завдання 4 теми 5.

Завдання для самостійної роботи

1. Побудувати на одному рисунку графіки поліномів Чебишова для $\forall n \leq 5$.
2. Виконати аналогічну побудову для поліномів Ерміта $H_n(x)/n^3$ ($\forall n \leq 5$) на відрізьку $[0, 2.5]$.
3. Побудувати лінії рівня для функції, заданої в завданні 5 теми 5.
4. Виконати і проаналізувати роботу наступного М-файла:

```
clc
f=@(x,y) (x.*y.*(1-x).*(1-y).*(...
    exp(-5.*(x-0.35).^2+(y-0.35).^2)).*...
    -exp(-5.*(x-0.65).^2+(y-0.65).^2)));
x=0:0.05:1;
```

```

y=0:0.05:1;
[X,Y]=meshgrid(x,y);
Z=f(X,Y);
mesh(X,Y,Z)
colormap hot
pause(2)
contour(X,Y,Z,20)
colormap hsv
pause(2)
surfl(X,Y,Z)
shading interp
colormap pink
pause(2)
view(50,20)
pause(2)
view(-100,-20)
pause(2)
surfc(X,Y,Z)
shading faceted
colormap winter
view(50,20)
alpha 0
pause(2)
alpha 0.5
pause(2)
alpha 1

```

5. Розглянути за допомогою демонстраційної версії (команда *demo*) побудову спеціальних X–Y графіків.

6. Розглянути за допомогою демонстраційної версії побудову спеціальних 3–D графіків.

7. Побудувати графік підінтегральної функції з завдання 2 теми 5 за допомогою функцій *mesh*, *waterfall*, *meshc*, *meshz*, *surf*, *surfc*, *surfl*.

Тема № 7

МОВА ПРОГРАМУВАННЯ І ПОБУДОВА М-ФАЙЛІВ

Теоретичні відомості

Основні структури керування алгоритмічної мови MATLAB. Система MATLAB має вхідну мову, яка нагадує мову Бейсик. Запис програм у системі традиційний і тому звичний для більшості користувачів комп'ютерів.

Ланцюг команд. Команди мови, як і команди командного вікна, записуються по одній в рядок. Для продовження команди на наступний рядок використовуємо символ "...".

Розгалуження. У загальному випадку синтаксис оператора розгалуження має вигляд

```
if <умова>
    <оператори1>;
else
    <оператори2>;
end
```

Якщо <умова> справедлива, то виконується група команд <оператори1>, у протилежному випадку – група команд <оператори2>.

Можливий спрощений варіант оператора розгалуження

```
if <умова>
    <оператори>;
end
```

Якщо <умова> справедлива, то виконується група команд <оператори>, у протилежному випадку здійснюється перехід до команди, розташованої після умовного оператора, тобто після ключового слова *end*.

Зазначимо, що умовні оператори можуть бути вкладеними.

В деяких випадках зручно користуватись *оператором вибору*

Switch <вираз>

```
case <скалярний_вираз1>
    <оператори1>;
case <скалярний_вираз2>
    <оператори2>;
```

...

```

case    <скалярний_виразN>
        <операториN>;
otherwise
        <оператори_альтернативи>;
end

```

Оператори циклу. Для повторення виконання певної послідовності команд у MATLAB використовується структура керування, яка називається циклом або ітерацією. Існує декілька варіантів команди циклу, перший з них – *цикл з передумовою*

```

while <умова>
    <оператори>;
end

```

Поки <умова> є істинною, виконується група команд <оператори> всередині тіла циклу. Як тільки <умова> стає хибною, здійснюється вихід з циклу, і керування передається командам, розташованим після оператора циклу, тобто після ключового слова *end*.

Цикл з лічильником має вигляд

```

for k=e1:e2:e3
    <оператори>;
end

```

Тут *k* – змінна циклу, *e1* – її початкове значення, *e3* – кінцеве значення, *e2* – крок, на який збільшується *k* після кожного виконання групи команд <оператори>. Цикл закінчується, як тільки значення *k* стає більше *e3*.

При програмуванні циклічних програм буває корисно використати оператори:

continue – пропустити всі наступні оператори (до *end*) і передати керування на перший оператор після заголовка циклу;

break – завершити цикл і передати керування наступному оператору після *end*.

Оператори введення, виведення. Для інтерактивного введення даних (під час виконання програми на MATLAB) використовують оператор введення:

X = input('Текст');

Алгоритм виконання: а) виводиться на екран інформація *Текст*;

б) вводиться з клавіатури довільний вираз MATLAB;

в) значення цього виразу присвоюється змінній *X*.

Окрім запису в командному рядку змінної (як це використовувалось раніше), є можливість і виведення даних за допомогою операторів виведення:

disp(<вираз>); або *display(<вираз>);*

В результаті виконання цих операторів на екран виводиться значення *<вираз>*. (Порівняйте їх виконання!).

При оформленні діалогу з користувачем допоможуть оператори:

format – впливає на спосіб представлення інформації на екрані (див. детальнішу інформацію командою *help format*);

pause – призупиняє виконання М-файла до натиснення довільної клавіші на клавіатурі;

pause(n) – призупиняє виконання М-файла на *n* секунд.

Опис функції. Опис функції здійснюється в окремому М-файлі, ім'я якого повинно співпадати з ім'ям функції, яка в ньому розташована. Синтаксис цього файла (*<ім'я>.m*):

```
function [R1 R2 ... Rn]=ім'я(x1,x2,...,xn)
```

```
%xi – вхідні формальні параметри
```

```
%Ri – вихідні формальні параметри
```

```
% коментар
```

```
%
```

```
<оператори>;
```

```
return
```

```
end
```

Всі поіменовані дані, які можуть використовуватись у тілі функції, є локальними або формальними вхідними або вихідними параметрами.

Всі вихідні параметри повинні бути визначені в тілі функції.

Якщо деяка змінна *у* є вхідним і вихідним параметром, то вона повинна фігурувати в обох списках параметрів.

Інформацію, розміщену в тексті коментаря М-функції, можна відобразити командою *help <ім'я>*, а весь текст М-функції можна надрукувати командою *type <ім'я>*.

У випадку визначення “простої” функції, тобто такої, значення якої обчислюється у вигляді деякого виразу, опис можна подати так:

$$<ім'я>=@(x1,x2,...,xn)(<вираз>);$$

При програмуванні функцій буває корисно використовувати функції:

nargin – кількість вхідних параметрів поточної функції;

nargin(fun) – кількість вхідних параметрів функції з ім'ям *fun*;

nargout – кількість вихідних параметрів поточної функції;

nargout(fun) – кількість вихідних параметрів функції з ім'ям *fun*.

Локальні та глобальні змінні. Глобальними називаються змінні, оголошені за допомогою атрибуту *global*, наприклад, *global g1 g2 g3*.

Локальними називаються змінні, які використовуються лише у даній М-функції, даному М-файлі.

Для збереження інформації між викликами даної М-функції використовується атрибут *persistent*. Наприклад, *persistent t1 t2 t3*.

М-сценарії та їх виклик. Зазначимо що М-файли, які створені користувачем, умовно можна поділити на дві групи:

- *файли-сценарії*, які не мають вхідних параметрів;
- *файли-функції*, які мають вхідні параметри.

Файл-функція відрізняється від файла-сценарію тим, що в ній обмін інформації реалізується через апарат формальних-фактичних параметрів. Змінні всередині тіла функції, як правило, локальні. Зазначимо, що в М-файлі доступна вся інформація, яка є в робочій області.

Виклик М-файла відбувається аналогічно виклику команди користувача в командному вікні.

Приклад 6. Степеневим ітераційним методом знайти максимальне по модулю власне число для квадратної матриці порядку n і відповідний йому власний вектор (порівняйте з [11], стор. 69-72).

```
function [L x]=maxlam_h(A,e,x)
% знаходження з точністю(e) максимального власного значення (L)
% і відповідного власного вектора (x) для квадратної матриці (A)
% степеневим ітераційним методом
% Вхідні дані:
%     A - квадратна матриця;
%     e - точність;
%     x - вектор початкового наближення.
% Вихідні дані:
%     L - власне число;
%     x - власний вектор.
%
n=length(A);
B=0;
L=1;
while abs(B-L)>e
    B=L;
    L=0;
    for i=1:n
        yy=sum(A(i,:)*x(:));
        if abs(yy)>abs(L)
            L=yy;
        end
        y(i)=yy;
    end
    x=y'./L;
end
return
```



```

end

% Приклад 6 (головна програма p6_main.m )
clc
% формування вхідних даних
x=[1;1;1];
A=[7,   -1,   1;...
   -1,  10,   2;...
    1,   2,  -4];
e=1e-4;
[L x]=maxlam_h(A,e,x);
disp(L)
pause
display(x)

```

Виклик цього М-файла має вигляд: `>>p6_main`

Приклад 7. Побудувати формули Гаусса з фіксованою кількістю вузлів n ($n=3$) для наближеного знаходження визначеного інтеграла на інтервалі $[a, b]$.

```

function f = p7_ff(x)
% підінтегральна функція
global N M N1
f=(M+abs(x).^N).*cos(x);
return

function [b,A]= p7_forms()
% формування СЛАР  $\bar{A} \cdot a = b$ 
%          ВХІДНІ :
%
%          РЕЗУЛЬТАТ :
% b - вектор правої частини СЛАР
% A - матриця СЛАР
global N M N1
for l=1:3
    k=2+l;
    b(l)=-p7_fun(k);
    for t=1:3
        k=k-1;
        A(l,t)=p7_fun(k);
    end
end
b=b';
return

function s = p7_fun(k)
% точне обчислення інтеграла від вагової функції
% помноженої на  $x^k$ 
%          ВХІДНІ :
% k - степінь x

```

```

%          РЕЗУЛЬТАТ :
% s - значення інтеграла
global N M N1
k1=k+1;
s=2.*mod(k1,2).*(M./k1+1./(N+k1));
return

function [x,c] = p7_preob(a,b,z,w)
% лінійне перетворення для системи
% квадратурних вузлів і коефіцієнтів
%          ВХІДНІ :
% [a,b] - відрізок, на якому обчислені
%          квадратурні вузли і коефіцієнти
% z - квадратурні вузли
% w - квадратурні коефіцієнти
%          РЕЗУЛЬТАТ :
% x - квадратурні вузли
% c - квадратурні коефіцієнти
ab=0.5.*(a+b);
ba=0.5.*(b-a);
x=ab+ba.*z;
c=ba.*w;
return

function [w] = p7_vesa(a)
% обчислення квадратурних коефіцієнтів
%          ВХІДНІ :
% a - квадратурні вузли
%          РЕЗУЛЬТАТ :
% w - квадратурні коефіцієнти
global N M N1
for l=1:3
    v(l)=p7_fun(l-1);
end
w(1)=(v(3)-(a(2)+a(3)).*v(2)+a(2).*a(3).*v(1))./((a(1)-a(2)).*(a(1)-a(3))));
w(2)=(v(3)-(a(1)+a(3)).*v(2)+a(1).*a(3).*v(1))./((a(2)-a(1)).*(a(2)-a(3))));
w(3)=(v(3)-(a(1)+a(2)).*v(2)+a(1).*a(2).*v(1))./((a(3)-a(1)).*(a(3)-a(2))));
return

% Приклад 7 (головна програма p7_main.m)
global M N N1
clc
% тестова задача: формули Гаусса для n=3
M=0;
N=0;
N1=(-1).^N;
% формування СЛАР
[b,A]=p7_forms;

```

```

% розв'язок СЛАР : коефіцієнти ортогонального
%                      полінома  $W(x)=x^3+a(1)*x^2+a(2)*x+a(3)$ 
[L,U]=lu(A);
z=L\b;
a=U\z;
% побудова стандартного полінома (в системі MATLAB)
%                       $P(x)=P(1)*x^3+P(2)*x^2+P(3)*x+P(4)$ 
% і його графіку
P=[1;a];
h=-1:.1:1;
plot(h,polyval(P,h),'b.-',h,0.*h,'k.--');
title('P(x)=P(1)*x^3+P(2)*x^2+P(3)*x+P(4)');
xlabel('x');
ylabel('y');
% знаходження квадратурних вузлів (корені P(x))
z=sort(roots(P))
% знаходження квадратурних коефіцієнтів
w=p7_vesa(z)
% знаходження квадратурної суми для конкретної задачі
% лінійне перетворення для системи
%                      квадратурних вузлів і коефіцієнтів
aa=0;
bb=1;
[x,c]=p7_preob(aa,bb,z,w);
in=sum(c.*cos(x))
% знаходження квадратурної суми для конкретної задачі
% з використанням відповідної функції в системі MATLAB
inn=quadl(@p7_ff,aa,bb)

```

Виклик цього М-файла має вигляд: `>>p7_main`

Приклад 8. Написати програму для розв'язання лінійного диференціального рівняння

$$u''(x) - p(x) * u(x) = -f(x),$$

$$p(x) \geq 0, x \in (a, b)$$

з крайовими умовами, заданими на кінцях відрізка $[a, b]$

$$d_1 * u'(a) + d_2 * u(a) = -d_3,$$

$$d_4 * u'(b) + d_5 * u(b) = d_6,$$

наближеним методом скінчених різниць. Для розв'язку відповідної СЛАР використати метод монотонної правої прогонки, а при неможливості її застосування – метод немонотонної прогонки [10].

```

function [y alfa beta err]=m_progm(a,c,b,f)
% метод монотонної правої прогонки для розв'язання СЛАР
% з тридіагональною матрицею
%  $a(k)*y(k-1) - c(k)*y(k) + b(k)*y(k+1) = -f(k),$ 
%  $a(1) = b(n) = 0,$ 
% при виконанні умов :

```

```

%           |c(k)| >= |a(k)| + |b(k)|,
%           k= 1,2,...,n.
% Вхідні дані:
%           a - піддіагональ;
%           -c - діагональ матриці;
%           b - наддіагональ;
%           -f - права частина.
% Вихідні дані:
%           y - розв'язок;
%           alfa,beta - прогоночні коефіцієнти;
%           err = 1 - умови застосування методу не виконуються
%                   і розв'язок не знайдено,
%           err = 0 - розв'язок знайдено.
%
n=length(c);
err=0;
k=0;
while k<n & ~err
    k=k+1;
    err=err | abs(c(k))<abs(a(k))+abs(b(k));
end
if err
    y=zeros(1,n);
    alfa=y;
    beta=y;
    return
end
n1=n-1;
alfa(1)=0;
beta(1)=0;
% пряма прогонка:
for k=1:n1
    j=k+1;
    z=c(k)-a(k).*alfa(k);
    alfa(j)=b(k)./z;
    beta(j)=(f(k)+a(k).*beta(k))./z;
end
% зворотна прогонка:
y(n)=(f(n)+a(n).*beta(n))./(c(n)-a(n).*alfa(n));
for k=n1:-1:1
    j=k+1;
    y(k)=alfa(j).*y(j)+beta(j);
end
return
end

function [y alfa beta te ka]=m_progn(a,c,b,f)
% метод немонотонної прогонки для розв'язання СЛАР
% з тридіагональною матрицею
% (метод Гауса з вибором головного елемента по рядку)
% a(k)*y(k-1) - c(k)*y(k) + b(k)*y(k+1) = - f(k)

```

```

%          k= 1,2,...,n;  a(1)=b(n)=0
% Вхідні дані:
%          a - піддіагональ;
%          -c - діагональ матриці;
%          b - наддіагональ;
%          -f - права частина.
% Вихідні дані:
%          y - розв'язок;
%          alfa,beta - прогоночні коефіцієнти;
%          te, ka - цілочисельні коефіцієнти визначення y(k).
%
n=length(c);
n1=n-1;
alfa(1)=0;
beta(1)=0;
% пряма прогонка :
C=c(1);
A=a(2);
F=f(1);
Q=f(2);
ka(1)=1;
te(1)=0;
for k=1:n1
    j=k+1;
    if abs(C)>=abs(b(k))
        alfa(j)=b(k)./C;
        beta(j)=F./C;
        C=c(j)-A.*alfa(j);
        F=Q+A.*beta(j);
        te(j)=ka(k);
        ka(j)=j;
        if k~=n1
            j=j+1;
            A=a(j);
            Q=f(j);
        end
    else
        alfa(j)=C./b(k);
        beta(j)=-F./b(k);
        C=c(j).*alfa(j)-A;
        F=Q-c(j).*beta(j);
        te(j)=j;
        ka(j)=ka(k);
        if k~=n1
            j1=j;
            j=j+1;
            A=a(j).*alfa(j1);
            Q=f(j)+a(j).*beta(j1);
        end
    end
end
end
end

```

```

% зворотна прогонка:
y(n)=F./C;
for k=n1:-1:1
    j=k+1;
    y(te(j))=alfa(j).*y(ka(j))+beta(j);
end
return
end

function [X Y]=m_grid(pfun,ffun,a,b,D,n)
% Розв'язок лінійного диференціального рівняння
%  $u''(x) - p(x)u(x) = -f(x)$ ,
%  $p(x) \geq 0, \quad x \in (a,b)$ ,
% для якого виконуються на кінцях відрізка крайові умови
%  $D(1)u'(a) + D(2)u(a) = -D(3)$ ,
%  $D(4)u'(b) + D(5)u(b) = D(6)$ .
% Для параметрів D(1),D(2) і D(4),D(5) виконуються умови :
% значення D(1)=0 або 1 (при D(1)=0 повинно бути D(2)=1,
% а при D(1)=1 значення D(2)- довільне);
% значення D(4)=0 або 1 (при D(4)=0 повинно бути D(5)=1,
% а при D(4)=1 значення D(5)- довільне);
%
% Вхідні дані :
% pfun - функція користувача, з описом p(x);
% ffun - функція користувача, з описом правої частини f(x);
% a - нижня межа інтегрування;
% b - верхня межа інтегрування;
% D - масив зі значеннями коефіцієнтів крайових умов;
% n - число точок сітки різницевої схеми.
% Вихідні дані:
% X - масив зі значеннями вузлів сітки;
% Y - масив зі значеннями наближеного розв'язку.
%
if nargin<6
    error('M_GRID:NotEnoughInputs',...
        'Not enough input arguments. See M_GRID.');
```

```

    return
end
if length(D)<6
    error('M_GRID:NotEnoughInputs',...
        'Not enough input arguments D. See M_GRID.');
```

```

    return
end
switch D(1)
case 0
    if D(2)~=1
        error('M_GRID:Unknown paramet.',...
            'Unknown parameter D(2). See M_GRID.');
```

```

        return
    end
end

```

```

case 1

otherwise
    error('M_GRID:Unknown paramet.',...
        'Unknown parameter D(1). See M_GRID.');
```

return

```

end
switch D(4)
case 0
    if D(5)~=1
        error('M_GRID:Unknown paramet.',...
            'Unknown parameter D(5). See M_GRID.');
```

return

```

    end
case 1
otherwise
    error('M_GRID:Unknown paramet.',...
        'Unknown parameter D(4). See M_GRID.');
```

return

```

end
if a>=b
    error('M_GRID:error a>=b.',...
        'a>=b. See M_GRID.');
```

return

```

end
% формування вузлів
h=(b-a)/(n-1);
X=a:h:b;
% допоміжні змінні
hh=h.^2;
h2=2.*h;
n1=n-1;
% формування коефіцієнтів
% A(k), C(k), B(k), F(k), k=1,...,n для СЛАР
A(1)=0;
switch D(1)
case 0
    C(1)=1;
    B(1)=0;
    F(1)=-D(3);
case 1
    C(1)=2+hh.*pfun(a)-h2.*D(2);
    B(1)=2;
    F(1)=hh.*ffun(a)+h2.*D(3);
end
for k=2:1:n1
    A(k)=1;
    C(k)=2+hh.*pfun(X(k));
    B(k)=1;
    F(k)=hh.*ffun(X(k));
end
```

```

B(n)=0;
switch D(4)
case 0
    A(n)=0;
    C(n)=1;
    F(n)=D(6);
case 1
    A(n)=2;
    C(n)=2+hh.*pfun(b)-h2.*D(4);
    F(n)=hh.*ffun(b)+h2.*D(6);
end
% знаходження сіткової функції при розв'язанні СЛАР
% методом монотонної правої прогонки, а при
% неможливості її застосування – методом немонотонної прогонки
[Y AL BE er]=M_PROGM(A,C,B,F);
if er
    [Y AL BE T K]=M_PROGN(A,C,B,F);
end
return

```

Завдання для аудиторної роботи

1. Написати тестову програму для застосування методу монотонної прогонки для розв'язання СЛАР:

$$\begin{cases} 5y_1 - y_2 = 1, \\ -y_{k-1} + 5y_k - y_{k+1} = k, k = \overline{2, n-1}, \\ -y_{n-1} + 5y_n = n. \end{cases}$$

2. Написати тестову програму для застосування методу немонотонної прогонки для розв'язання СЛАР:

$$\begin{cases} -y_1 + y_2 = 0, \\ y_{k-1} - y_k + y_{k+1} = 0, k = \overline{2, n-1}, \\ y_{n-1} - y_n = 0. \end{cases}$$

Завдання для самостійної роботи

1. Написати М-функцію *minlam_h.m* для знаходження мінімального по модулю власного числа і відповідного йому власного вектора неособливої матриці, використовуючи М-функцію *maxlam_h.m*.

2. Написати М-файл для тестування М-функції *minlam_h.m*.

3. Написати М-файл для виконання завдання 11 самостійної роботи теми 4 з використанням програми *m_grid.m*.

4. Написати М-функцію *m_progmm.m* матричної прогонки для розв'язування системи триточкових векторних рівнянь

$$\vec{Y}_{j-1} - C\vec{Y}_j + \vec{Y}_{j+1} = -\vec{F}_j, j = \overline{2, N-1}, \vec{Y}_1 = \vec{F}_1, \vec{Y}_N = \vec{F}_N.$$

Тема №8

ПРИКЛАДИ РОЗВ'ЯЗУВАННЯ ПРИКЛАДНИХ ЗАДАЧ

Розв'язання крайової задачі методом стрільби. Розглянемо крайову задачу (приклад 9)

$$\begin{cases} u'' = f(x, u, u'), \\ u|_{x=a} = u_0, u'|_{x=b} = u_1 \end{cases}$$

при $f(x, u, u') = 6x - 0,5$, $u_0 = 0$, $u_1 = 0,5$, $a = 0$, $b = 1$. Дана крайова задача еквівалентна двом задачам Коші [4]

$$\begin{cases} y_1' = y_2, \\ y_1|_{x=a} = u_0, \end{cases} \quad \begin{cases} y_2' = f(x, y_1, y_2), \\ y_2|_{x=b} = u_1. \end{cases}$$

Тут уведені позначення $y_1 = u(x)$, $y_2 = u'(x)$.

Нижче наведено текст відповідного М-файла для знаходження розв'язку та побудови його графіку.

```
function yp = p9_ff1(t,y)
global xx yy1 yy2
yp=spline(xx,yy2,t);
return
```

```
function yp = p9_ff2(t,y)
global xx yy1 yy2
yp=6.*(t-0.5);
return
```

```
% М-файл p9_main.m
global xx yy1 yy2
a=0;
b=1;
u0=0;
u1=0.5;
eps=1e-4;
kmax=1000;
n=101;
h=(b-a)/(n-1);
xx=[a:h:b];
for i=1:1:n
    yy2(i)=u1;
end
k=0;
```

```

F=1;
while F & k<kmax
    k=k+1;
    yy1(1)=u0;
    y1=u0;
    for i=2:1:n
        [x,y]=ode45(@p4_ff1,[xx(i-1) xx(i)],y1);
        y1=y(length(y));
        yy1(i)=y1;
    end
    yy2(n)=u1;
    y2=u1;
    for i=n:-1:2
        [x,y]=ode45(@p4_ff2,[xx(i) xx(i-1)],y2);
        y2=y(length(y));
        yy2(i-1)=y2;
    end
    if k>1
        F=norm(yy1-yy1p,2)>eps;
    end
    yy1p=yy1;
end
k
yy1;
% точний розв'язок
yy3=xx.*(xx-0.5).*(xx-1)
plot(xx,yy1,xx,yy3)
grid on

```

Виклик цього М-файла має вигляд: `>>p9_main`

Розв'язання лінійного неоднорідного інтегрального рівняння Фредгольма другого роду методом заміни інтеграла скінченою сумою. В загальному випадку відповідне інтегральне рівняння записується у вигляді

$$u(x) = f(x) + \lambda \int_a^b K(x,s)u(s)ds,$$

де $u(x)$ – невідома функція на відрізку $[a,b]$, $f(x)$, $K(x,s)$ – відомі функції. Права частина $f(x)$ визначена на відрізку $[a,b]$, ядро $K(x,s)$ визначене в області $[a,b] \times [a,b]$, λ – відоме число.

Нижче наведено текст відповідного М-файла для розв'язання поставленої задачі (приклад 10).

```

% М-функція p_fred.m
function [X Y n] = p_fred(lambda,a,b,nkf,n,Kfun,ffun)
% Розв'язок лінійного неоднорідного інтегрального рівняння
% другого роду типу Фредгольма методом заміни інтеграла
% скінченою сумою

```

```

%                                     b
%      y(x)=f(x)+lambda* ∫ K(x,s)*y(s) ds
%                                     a
% Вхідні дані :
% lambda - числовий параметр рівняння;
% a       - нижня межа інтегрування;
% b       - верхня межа інтегрування;
% nkf     - вибір методу
%          = 0 - квадратурна формула методу трапецій,
%          = 1 - квадратурна формула методу Сімпсона;
% n       - число точок сітки. Повинно задовольняти
%          умову:
%          n>1                при nkf=0,
%          n>2&mod(n,2)==1    при nkf=1;
% Kfun    - функція користувача, з описом ядра K(x,s);
% ffun    - функція користувача, з описом правої частини f(x);
% Вихідні дані:
% X       - масив зі значеннями вузлів сітки;
% Y       - масив зі значеннями наближеного розв'язку.
if nargin<7
    error('P_FRED:NotEnoughInputs',...
        'Not enough input arguments. See P_FRED. ');
    return
end
switch nkf
case 0
    if n<2
        n=2;
    end
case 1
    if n<3
        n=3;
    else
        if mod(n,2)==0
            n=n+1;
        end
    end
otherwise
    error('P_FRED:Unknown method.',...
        'Unknown method. See P_FRED. ');
    return
end
% формування вузлів
h=(b-a)/(n-1);
X=[a:h:b];
% формування коефіцієнтів A(i), i=1,...,n квадратурної
% формули
switch nkf
case 0
    A(1)=0.5.*h;
    for i=2:1:n-1

```

```

        A(i)=h;
    end
    A(n)=A(1);
case 1
    A(1)=h./3;
    c2=2.*A(1);
    c1=2.*c2;
    for i=2:2:n-1
        A(i)=c1;
        A(i+1)=c2;
    end
    A(n)=A(1);
end
% формування матриці B(i,k), вектора правої частини F(i) СЛАР
%
% 
$$Y(i) - \lambda \sum_{k=1}^n A(k) * K(X(i), X(k)) * Y(k) = f(X(i)),$$

%
% i=1,...,n
for i=1:1:n
    for k=1:1:n
        B(i,k)=-lambda*A(k)*Kfun(X(i),X(k));
    end
    B(i,i)=1+B(i,i);
    F(i)=ffun(X(i));
end
F=F';
% розв'язок СЛАР      B*Y=F
Y=B\F;
return

function yp=p10_f(x)
% опис правої частини f(x)
yp=x.^2.*(1-x)-1./(x+1);
return

function yp=p10_K(x,s)
% опис ядра K(x,s)
yp=1./(x+1);
return

% М-файл p10_main.m
clear all
lamb=12;
aa=0;
bb=1;
nn=101;
clc
[XX,YY,nn]=p_fred(lamb,aa,bb,1,nn,@p10_K,@p10_f);
plot(XX,YY,'m')
xlabel('x')
ylabel('U(x)')

```

Виклик цього М-файла має вигляд: `>>p10_main`. Результат його роботи наведено на рис. 8.1.

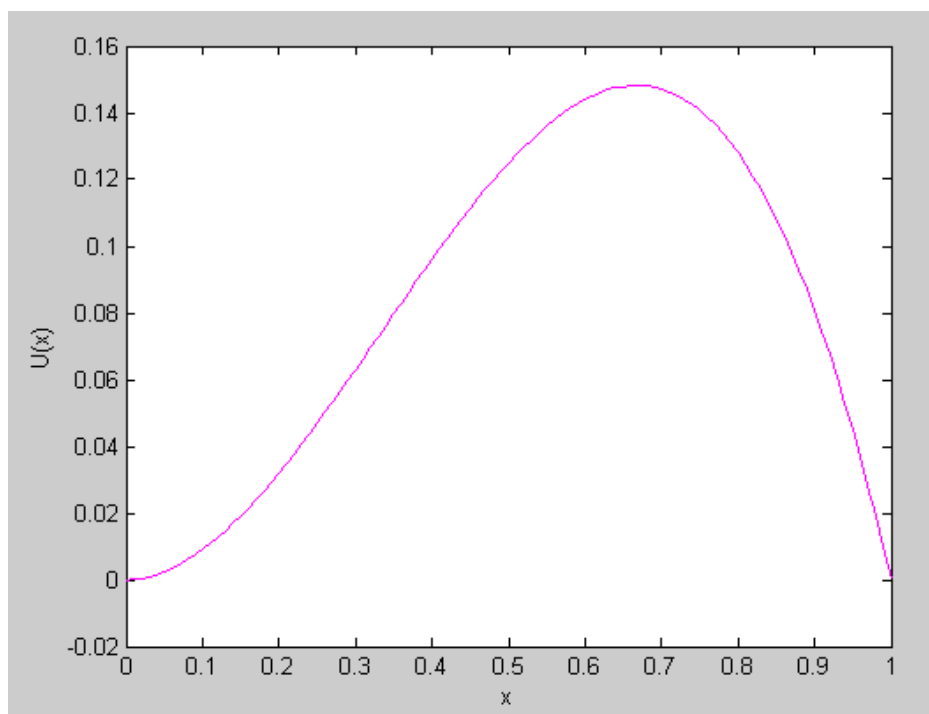


Рис. 8.1. Графік наближеного розв'язку $U(x)$

Чисельне розв'язання крайової задачі зі зв'язаними (нелокальними) крайовими умовами. Розглянемо для рівняння теплопровідності наступну крайову задачу (див. [12], стор. 93) :

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad x \in (0, 1), t > 0, \quad (8.1)$$

$$U(x, 0) = U_0(x), \quad (8.2)$$

$$U(0, t) - U(v(t), t) = \mu_1(t), \quad (8.3)$$

$$U(1, t) - U(v(t), t) = \mu_2(t), \quad (8.4)$$

де функція $x=v(t)$ приймає значення від 0 до 1.

Для чисельного розв'язання поставленої задачі:

- скористаємося методом скінчених різниць. Побудуємо різницеву задачу, а для розв'язання СЛАР розглянемо такі підходи:
 - не будемо використовувати спеціальні методи і враховувати структуру матриці (1-й варіант);
 - використаємо спеціальні методи з урахуванням структури матриці (2-й варіант);
- скористаємося методом Рунге (методом прямих) (3-й варіант).

Для перших двох варіантів потрібно побудувати різницеву задачу. Введемо сітку

$$\omega = \omega_h \times \omega_\tau, \quad \omega_h = \left\{ x_i = ih, h = \frac{1}{N}, i = \overline{0, N} \right\}, \quad \omega_\tau = \{ t_n = n\tau, n = 0, 1, 2, \dots \}.$$

Для рівняння (8.1) побудуємо схему з вагами ($0 < \sigma \leq 1$)

$$y_{t,i} = \sigma y_{xx,i}^{n+1} + (1 - \sigma) y_{xx,i}^n, \quad i = \overline{1, N-1}, \quad (8.5)$$

$$y_i^0 = U_0(x_i), \quad i = \overline{0, N}. \quad (8.6)$$

Апроксимуємо крайові умови (8.3) – (8.4)

$$y_0^n - y_k^n = \mu_1(t_n), \quad (8.7)$$

$$n = 0, 1, \dots; 1 \leq k \leq N-1,$$

$$y_N^n - y_k^n = \mu_2(t_n), \quad (8.8)$$

Апроксимуємо криву $x=v(t)$ ламаною

$$x_k = v(t_n) \quad (8.9)$$

так, щоб точки (x_k, t_n) були б вузловими точками нашої сітки ω .

Для реальних обчислень τ, h, σ будемо вибирати з умови

$$\left| \frac{2}{h^2} + \frac{1}{\sigma\tau} \right| \succ \frac{2}{h^2}, \quad \sigma > -\frac{h^2}{4\tau}.$$

Введемо позначення $\vec{Y} = \{y_i\}_{i=0}^N = \{y_i^{n+1}\}_{i=0}^N$, а також

$$\begin{cases} c_0 = 1, d_k = 1, (d_j = 0, j \neq k), f_0 = \mu_1(t_{n+1}), \Psi_0 = 0, \\ c_N = 1, g_k = 1, (g_j = 0, j \neq k), f_n = \mu_2(t_{n+1}), \Phi_N = 0, \\ \Phi_i = \Psi_i = 0, a_i = b_i = \frac{1}{h^2}, c_i = c_u + b_i + \frac{1}{\sigma\tau}, \\ f_i = \frac{1}{\sigma\tau} y_i^n + \left(\frac{1}{\sigma} - 1 \right) y_{xx,i}^n, i = 1, 2, \dots, N-1. \end{cases} \quad (8.10)$$

При позначеннях (8.10) ми отримуємо для знаходження y_i^{n+1} систему лінійних алгебраїчних рівнянь

$$\begin{cases} c_0 y_0 - \sum_{j=1}^{N-1} d_j y_j - \psi_0 y_N = f_0, & i = 0, \\ -\Phi_i y_0 - a_i y_{i-1} + c_i y_i - b_i y_{i+1} - \Psi_i y_N = f_i, & i = \overline{1, N-1}, \\ -\Phi_N y_0 - \sum_{j=1}^{N-1} g_j y_j + c_N y_N = f_N, & i = N. \end{cases} \quad (8.11)$$

Якщо позначити $\vec{F} = (f_0, f_1, \dots, f_N)$, то СЛАР (8.11) можна записати у матричному вигляді

$$A\vec{Y} = \vec{F},$$

де матриця A отримана об'єднанням тридіагональної матриці з чотирьох сторін відповідними векторами.

Для другого варіанту

Розв'язок (8.11) будемо шукати у вигляді лінійної комбінації

$$y_i = u_i + y_0 v_i + y_N w_i, \quad 0 \leq i \leq N, \quad (8.12)$$

де

u_i, v_i, w_i – розв'язки таких задач:

$$\begin{cases} -a_i u_{i-1} + c_i u_i - b_i u_{i+1} = f_i, & 1 \leq i \leq N-1, \\ u_0 = 0, u_N = 0, \end{cases} \quad (8.13)$$

$$\begin{cases} -a_i v_{i-1} + c_i v_i - b_i v_{i+1} = \varphi_i, & 1 \leq i \leq N-1, \\ v_0 = 1, v_N = 0, \end{cases} \quad (8.14)$$

$$\begin{cases} -a_i w_{i-1} + c_i w_i - b_i w_{i+1} = \psi_i, & 1 \leq i \leq N-1, \\ w_0 = 0, w_N = 1. \end{cases} \quad (8.15)$$

Знайдемо y_0, y_N із системи рівнянь (8.11) при $i=0, i=N$. Підставимо $y_0 = u_0 + y_0 v_0 + y_N w_0, y_N = u_N + y_0 v_N + y_N w_N$ у (8.11) і отримаємо систему для знаходження y_0, y_N :

$$\begin{cases} \left(c_0 - \sum_{j=1}^{N-1} d_j v_j \right) y_0 - \left(\psi_0 + \sum_{j=1}^{N-1} d_j w_j \right) y_N = f_0 + \sum_{j=1}^{N-1} d_j u_j, \\ \left(-\varphi_N + \sum_{j=1}^{N-1} g_j v_j \right) y_0 + \left(c_N - \sum_{j=1}^{N-1} g_j w_j \right) y_N = f_N + \sum_{j=1}^{N-1} g_j u_j. \end{cases} \quad (8.16)$$

Якщо визначник цієї матриці СЛАР відмінний від 0, то за формулами Крамера знаходимо

$$y_0 = \frac{\Delta_{y_0}}{\Delta}, \quad y_N = \frac{\Delta_{y_N}}{\Delta},$$

де $\Delta_{y_0}, \Delta_{y_N}, \Delta$ – відповідні визначники.

Для (8.13)–(8.15) маємо звичайну систему з тридіагональною матрицею, яку можна розв'язувати методом прогонки:

$$\begin{aligned} u_i &= \alpha_{i+1} u_{i+1} + \beta_{i+1}, \quad i=N-1, \dots, 0, \quad u_N = 0, \\ v_i &= \alpha_{i+1} u_{i+1} + \gamma_{i+1}, \quad i=N-1, \dots, 0, \quad v_N = 0, \\ w_i &= \alpha_{i+1} u_{i+1} + \delta_{i+1}, \quad i=N-1, \dots, 0, \quad u_N = 0, \end{aligned} \quad (8.17)$$

де

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i \alpha_i}, \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - a_i \alpha_i}, \quad i=1,2, \dots, N-1, \alpha_1=0, \beta_1=0,$$

$$\gamma_{i+1} = \frac{\varphi_i + a_i \gamma_i}{c_i - a_i \alpha_i}, \delta_{i+1} = \frac{\Psi_{i+1} + a_i \delta_i}{c_i - a_i \alpha_i}, \quad i=1,2, \dots, N-1, \gamma_1=1, \delta_1=0. \quad (8.18)$$

При виконанні умов:

- матриця СЛАР (8.16) неособлива;
- $|a_i| > 0, |b_i| > 0, |c_i| \geq |a_i| + |b_i|, 1 \leq i \leq N-1$

система (8.11) має єдиний розв'язок, який ми знаходимо стійким методом.

Для третього варіанту

Введемо сітку тільки по змінній x , отримаємо

$$\frac{dU_i(t)}{dt} = \frac{U_{i-1}(t) - 2U_i(t) + U_{i+1}(t)}{h^2}, \quad U_i(t) = U(x_i, t), \quad i = \overline{2, N-1}, \quad (8.19)$$

$$U_i(0) = U_0(x_i), i = \overline{1, N}, \quad (8.20)$$

$$U_1(t) = U_k(t) + \mu_1(t), \quad (8.21)$$

$$U_N(t) = U_k(t) + \mu_2(t), \quad (8.22)$$

$$\begin{cases} \frac{dU_2(t)}{dt} = \frac{U_1(t) - 2U_2(t) + U_3(t)}{h^2}, \\ \frac{dU_i(t)}{dt} = \frac{U_{i-1}(t) - 2U_i(t) + U_{i+1}(t)}{h^2}, \quad i = \overline{3, N-1}, \\ \frac{dU_{N-1}(t)}{dt} = \frac{U_{N-1}(t) - 2U_{N-1}(t) + U_N(t)}{h^2}. \end{cases}$$

Якщо в рівняння (8.19) для $i=2$ підставимо співвідношення (8.21), а в рівняння (8.19) для $i=N-1$ підставимо співвідношення (8.22) і введемо вектор-функцію $\vec{U} = \{U_2(t), \dots, U_{N-1}(t)\}$, то отримаємо систему звичайних диференціальних рівнянь у вигляді

$$\frac{d\vec{U}(t)}{dt} = \vec{F}(\vec{U}, t),$$

де

$$F_2(t) = \frac{U_k(t) + \mu_1(t) - 2U_2(t) + U_3(t)}{h^2},$$

$$F_i(t) = \frac{U_{i-1}(t) - 2U_i(t) + U_{i+1}(t)}{h^2}, \quad i = \overline{3, N-1},$$

$$F_{N-1}(t) = \frac{U_{N-1}(t) - 2U_{N-1}(t) + U_N(t)}{h^2}.$$

Нижче наведено текст програми для розв'язання поставленої задачі (приклади 11-13).

```
function f=ps1_ff(t,U)
% Опис правої частини системи ЗДР
global h HH N
K=round(ps1_ny(t)/h);
f(1)=HH*(U(K)+ps1_my1(t)-2*U(1)+U(2));
for j=2:N-3
    f(j)=HH*(U(j-1)-2*U(j)+U(j+1));
end
f(N-2)=HH*(U(N-3)-2*U(N-2)+U(K)+ps1_my2(t));
rf=size(f);
if rf(1)~=N-2
    f=f';
end
return

function y=ps1_my1(t)
% функція    мя1(t)
y=0;
return

function y=ps1_my2(t)
% функція    мя2(t)
y=1.0;
return

function x=ps1_ny(t)
% функція    ню(t)
global l
x=0.75*l;
return

function u=ps1_u0(g)
% функція    u0(x)
global l
u=5*sin(g*2*pi/l);
return

% М-файл ps11main.m
% Чисельне розв'язування крайової задачі з нелокальними
% крайовими умовами (Варіант 1)
clc
clear
global l
% параметри задачі:
l=1;
N=101;
N1=N-1;
```

```

h=1/N1;
tau=0.001;
M=100;
sigma=0.5;
x=[0:h:1];
save x x
HH=1/(h*h);
ST=1/(sigma*tau);
STH=(1/sigma-1)*HH;
ai=HH;
bi=HH;
ci=ai+bi+ST;
% формування початкових умов:
t=0;
for j=1:N
    Y(j)=ps1_u0(x(j));
end
Y0=Y;
Y=Y';
for n=1:M
    t=t+tau;
    % формування СЛАР  $A*Y=F$  для шару n :
    F(1)=ps1_my1(t);
    for j=2:N1
        F(j)=ST*Y(j)+STH*(Y(j-1)-2*Y(j)+Y(j+1));
    end
    F(N)=ps1_my2(t);
    rf=size(F);
    if rf(1)~=N
        F=F';
    end
    A=zeros(N);
    A(1,1)=1;
    for j=2:N1
        A(j,j-1)=-ai;
        A(j,j)=ci;
        A(j,j+1)=-bi;
    end
    K=round(ps1_ny(t)/h)+1;
    A(1,K)=-1;
    A(N,K)=-1;
    A(N,N)=1;
    % розв'язок СЛАР :
    [L,U]=lu(A);
    Y=L\F;
    Y=U\Y;
    % запам'ятовуємо розв'язок для  $M*tau/2$  :
    if n*2==M
        Y1=Y';
    end
end
end

```

```

Y=Y';
V1=Y;
save V1 V1
% побудова графіків розв'язку про  $t=0$ ,  $M\tau/2$ ,  $M\tau$  :
plot(x,Y0,'b.-',x,Y1,'k.--',x,Y,'r*-');
xlabel('x');
ylabel('u');
legend('u(x,0)', 'u(x,M\tau/2)', 'u(x,M\tau)');

% М-файл ps12main.m
% Чисельне розв'язування крайової задачі з нелокальними
% крайовими умовами (Варіант 2)
clc
clear
global l
% параметри задачі:
l=1;
N=101;
N1=N-1;
h=l/N1;
tau=0.001;
M=100;
sigma=0.5;
x=[0:h:l];
% додаткові обчислення:
HH=1/(h*h);
ST=1/(sigma*tau);
STH=(1/sigma-1)*HH;
ai=HH;
bi=HH;
ci=ST+ai+bi;
% формування початкових умов:
t=0;
for j=1:N
    Y(j)=ps1_u0(x(j));
end
Y0=Y;
% обчислення коефіцієнтів прогонки, які не змінюються з часом:
alfa(2)=0;
beta(2)=0;
gamma(2)=1;
delta(2)=0;
for j=2:N1
    k=j+1;
    s=ci-ai*alfa(j);
    alfa(k)=bi/s;
    gamma(k)=ai*gamma(j)/s;
    delta(k)=ai*delta(j)/s;
end
v(N)=0;
w(N)=1;

```

```

for j=N1:-1:1
    k=j+1;
    v(j)=alfa(k)*v(k)+gamma(k);
    w(j)=alfa(k)*w(k)+delta(k);
end
for n=1:M
    t=t+tau;
    for j=2:N1
        k=j+1;
        fj=ST*Y(j)+STH*(Y(j-1)-2*Y(j)+Y(k));
        s=ci-ai*alfa(j);
        beta(k)=(fj+ai*beta(j))/s;
    end
    a11=1;
    a12=0;
    a21=0;
    a22=1;
    b1=ps1_my1(t);
    b2=ps1_my2(t);
    K=round(ps1_ny(t)/h)+1;
    u(N)=0;
    for j=N1:-1:1
        k=j+1;
        u(j)=alfa(k)*u(k)+beta(k);
        if j==K
            a11=a11-v(j);
            a12=a12+w(j);
            a21=a21+v(j);
            a22=a22-w(j);
            b1=b1+u(j);
            b2=b2+u(j);
        end
    end
    d=a11*a22-a12*a21;
    if d~=0
% розв'язок СЛАР для Y(1),Y(N):
        Y(1)=(b1*a22+b2*a12)/d;
        Y(N)=(b1*a21+b2*a11)/d;
% знаходження розв'язку на новому шарі:
        for j=2:N1
            Y(j)=u(j)+Y(1)*v(j)+Y(N)*w(j);
        end
% запам'ятовуємо розв'язок для M*tau/2 :
        if n*2==M
            Y1=Y;
        end
    end
end
V2=Y;
save V2 V2
% побудова графіків розв'язку про t=0, M*tau/2, M*tau :

```

```

plot(x,Y0,'b.-',x,Y1,'k.--',x,Y,'r*-');
xlabel('x');
ylabel('u');
legend('u(x,0)', 'u(x,M*tau/2)', 'u(x,M*tau)');

% М-файл ps13main.m
% Чисельне розв'язування крайової задачі з нелокальними
% крайовими умовами (Варіант 3)
clc
clear
global l h HH N
% параметри задачі:
l=1;
N=101;
N1=N-1;
h=l/N1;
tau=0.001;
M=100;
x=[0:h:l];
HH=1/(h*h);
% формування початкових умов:
for j=1:N
    Y(j)=ps1_u0(x(j));
end
Y0=Y;
tk=0;
for n=1:M
    t0=tk;
    tk=t0+tau;
    y0=(Y(2:N1))';
    [t,y]=ode45(@ps1_ff,[t0 tk],y0);
    rf=size(y);
    m=rf(1);
    K=round(ps1_ny(tk)/h);
    Y(1)=y(m,K)+ps1_my1(tk);
    for j=2:N1
        Y(j)=y(m,j-1);
    end
    Y(N)=y(m,K)+ps1_my2(tk);
% запам'ятовуємо розв'язок для M*tau/2 :
    if n*2==M
        Y1=Y;
    end
end
V3=Y;
save V3 V3
% побудова графіків розв'язку про t=0, M*tau/2, M*tau :
plot(x,Y0,'b.-',x,Y1,'k.--',x,Y,'r*-');
xlabel('x');
ylabel('u');
legend('u(x,0)', 'u(x,M*tau/2)', 'u(x,M*tau)');

```

Послідовно виконати М-файли *ps11main*, *ps12main*, *ps13main*. Результати їх роботи наведено на рис. 8.2.

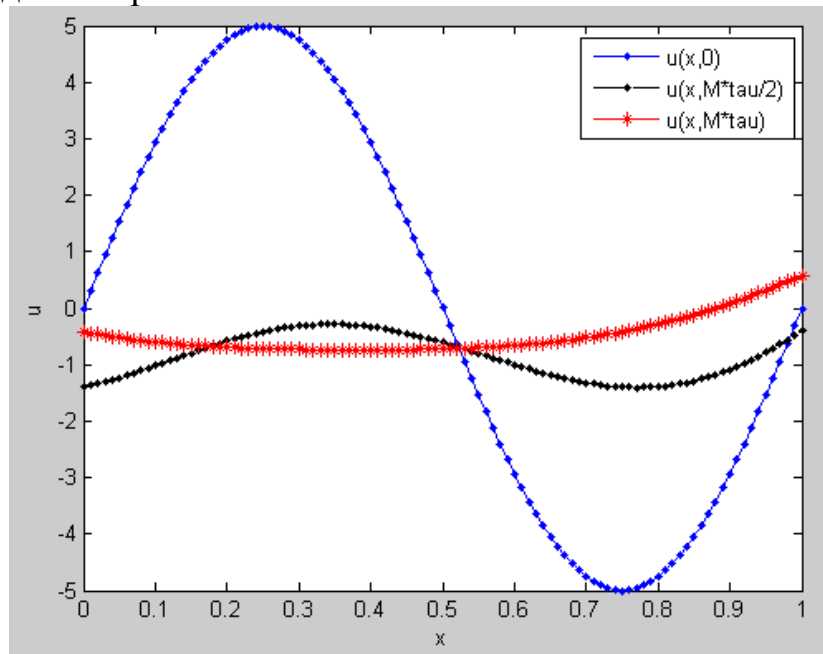


Рис. 8.2. Графік наближеного розв'язку $U(x,t)$ для різних значень t (варіант 1)

Для порівняння розв'язків виконаємо М-файл *psdiff*. Результат його роботи наведено на рис. 12.

```
% М-файл psdiff.m
% побудова графіка розв'язків варіантів 1,2,3
load x x
load V1 V1
load V2 V2
load V3 V3
plot(x,V1,'b.-',x,V2,'k.--',x,V3,'r*.-');
legend('v1','v2','v3');
```

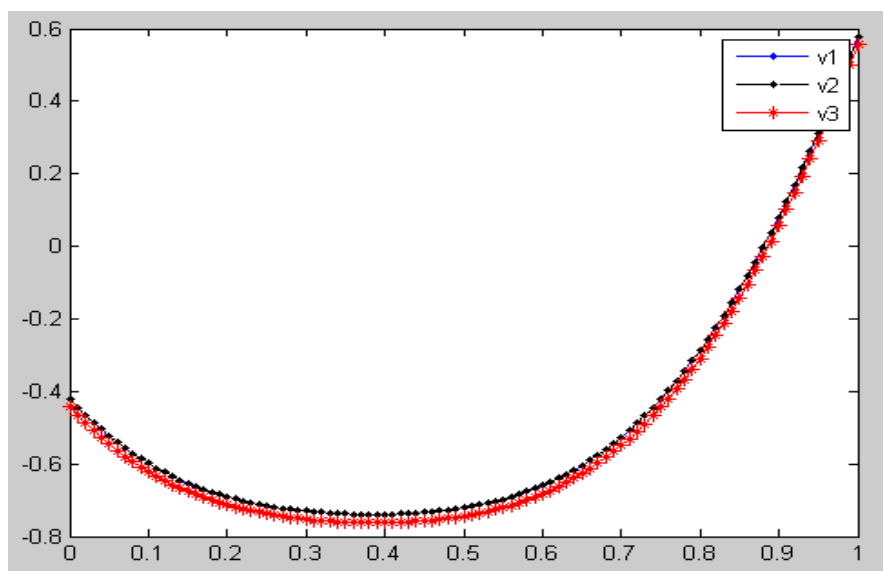


Рис. 8.3. Графік наближеного розв'язку $U(x,M*tau)$ для варіантів 1-3

Чисельне розв'язання крайової задачі для рівняння Пуассона зі змішаними крайовими умовами в трикутній області. Розглянемо для рівняння Пуассона:

$$\Delta U(x,y) = -f(x,y), \quad (x,y) \in D;$$

крайову задачу зі змішаними крайовими умовами:

$$\frac{\partial U}{\partial n} = \mu_1(x,y), \quad (x,y) \in \partial D1;$$

$$U(x,y) = \mu_2(x,y), \quad (x,y) \in \partial D2;$$

$$U(x,y) = \mu(x,y), \quad (x,y) \in \partial D3;$$

в трикутній області $\bar{D} = D \cup \partial D = \{(0 \leq x \leq a) \& (0 \leq y \leq b) \& (y \leq 1-x)\}$, де

$$\partial D1 = \{((0 \leq x < 0,5a) \& (y = 0)) \cup ((0 \leq y < 0,5b) \& (x = 0))\};$$

$$\partial D2 = \{((0,5a \leq x \leq a) \& (y = 0)) \cup ((0,5b \leq y \leq b) \& (x = 0))\};$$

$$\partial D3 = \{(0 \leq x \leq a) \& (y = 1-x)\}.$$

Для наближеного чисельного розв'язання задачі застосуємо метод скінчених різниць [12], а для отримання розв'язку СЛАР різницевої задачі – ітераційний метод типу Зейделя.

Нижче наведено текст відповідного М-файла для розв'язання поставленої задачі (приклад 14).

```
function f=ps2_f(x,y)
% функція f(x,y)
f=-2*(x+y);
return

function u=ps2_mx11(x)
% функція mx11(x) для (0<=x<0.5a) & (y=0)
u=x*x;
return

function u=ps2_mx12(x)
% функція mx12(x) для (0.5a<=x<=a) & (y=0)
u=0;
return

function u=ps2_mxy(x)
% функція mxy(x) для (0<=x<=a) & (y=1-x)
u=x*(1-x);
return

function u=ps2_my11(y)
% функція my11(y) для (0<=y<0.5a) & (x=0)
u=y*y;
return

function u=ps2_my12(y)
% функція my12(y) для (0.5b<=y<=b) & (x=0)
u=0;
```

```

return

% М-файл ps14main.m
% Чисельне розв'язування крайової задачі
% для рівняння Пуассона
% зі змішаними крайовими умовами в трикутній області D:
%  $D = \{ (0 \leq x \leq a) \ \& \ (0 \leq y \leq b) \ \& \ (y \leq 1-x) \}$ 
clc
clear
% параметри задачі:
a=1;
b=1;
Nx=101;
Ny=101;
e=1e-4;
kmax=10000;
Nx1=Nx-1;
Ny1=Ny-1;
Hx=a/Nx1;
Hy=b/Ny1;
X=[0:Hx:a];
Y=[0:Hy:b];
NxS=round(0.5*a/Hx);
NyS=round(0.5*b/Hy);
NxS1=NxS-1;
NyS1=NyS-1;
HxH=1/(Hx*Hx);
HyH=1/(Hy*Hy);
C=0.5/(HxH+HyH);
A=HxH*C;
B=HyH*C;
% формування крайових умов та початкового наближення:
U=zeros(Ny,Nx);
%  $(0.5a \leq X < a) \ \& \ (0 = Y)$ 
for i=NxS:Nx1
    U(1,i)=ps2_mx12(X(i));
end
%  $(0 = X) \ \& \ (0.5b \leq Y < b)$ 
for j=NyS:Ny1
    U(j,1)=ps2_my12(Y(j));
end
%  $(0 \leq X \leq a) \ \& \ (Y = 1-X)$ 
for j=1:Ny
    n=Nx-j+1;
    U(j,n)=ps2_mxy(X(n));
end
k=0;
Fla=0;
while ~Fla & k<kmax
    Fla=1;
    k=k+1;
end

```



```

u=U(1,1);
UU=C*ps2_f(X(1),Y(1))+A*2*(U(1,2)-Hx*ps2_my11(Y(1)))+...
    B*2*(U(2,1)-Hy*ps2_mx11(X(1)));
if Fla
    Fla=abs(UU-u)<e;
end
U(1,1)=UU;
for i=2:NxS1
    u=U(1,i);
    UU=C*ps2_f(X(i),Y(1))+A*(U(1,i-1)+U(1,i+1))+...
        B*2*(U(2,i)-Hy*ps2_mx11(X(i)));
    if Fla
        Fla=abs(UU-u)<e;
    end
    U(1,i)=UU;
end
for j=2:NyS1
    u=U(j,1);
    UU=C*ps2_f(X(1),Y(j))+A*2*(U(j,2)-Hx*ps2_my11(Y(j)))+...
        B*(U(j-1,1)+U(j+1,1));
    if Fla
        Fla=abs(UU-u)<e;
    end
    U(j,1)=UU;
    for i=2:Nx-j
        u=U(j,i);
        UU=C*ps2_f(X(i),Y(j))+A*(U(j,i-1)+U(j,i+1))+...
            B*(U(j-1,i)+U(j+1,i));
        if Fla
            Fla=abs(UU-u)<e;
        end
        U(j,i)=UU;
    end
end
for j=NyS:Ny1-1
    for i=2:Nx-j
        u=U(j,i);
        UU=C*ps2_f(X(i),Y(j))+A*(U(j,i-1)+U(j,i+1))+...
            B*(U(j-1,i)+U(j+1,i));
        if Fla
            Fla=abs(UU-u)<e;
        end
        U(j,i)=UU;
    end
end
end
% побудова графіків розв'язку:
k
surfl(U);
shading interp;
colormap(pink);

```

Виклик цього М-файла має вигляд: `>>ps14main`. Результат його роботи наведено на рис. 8.4.

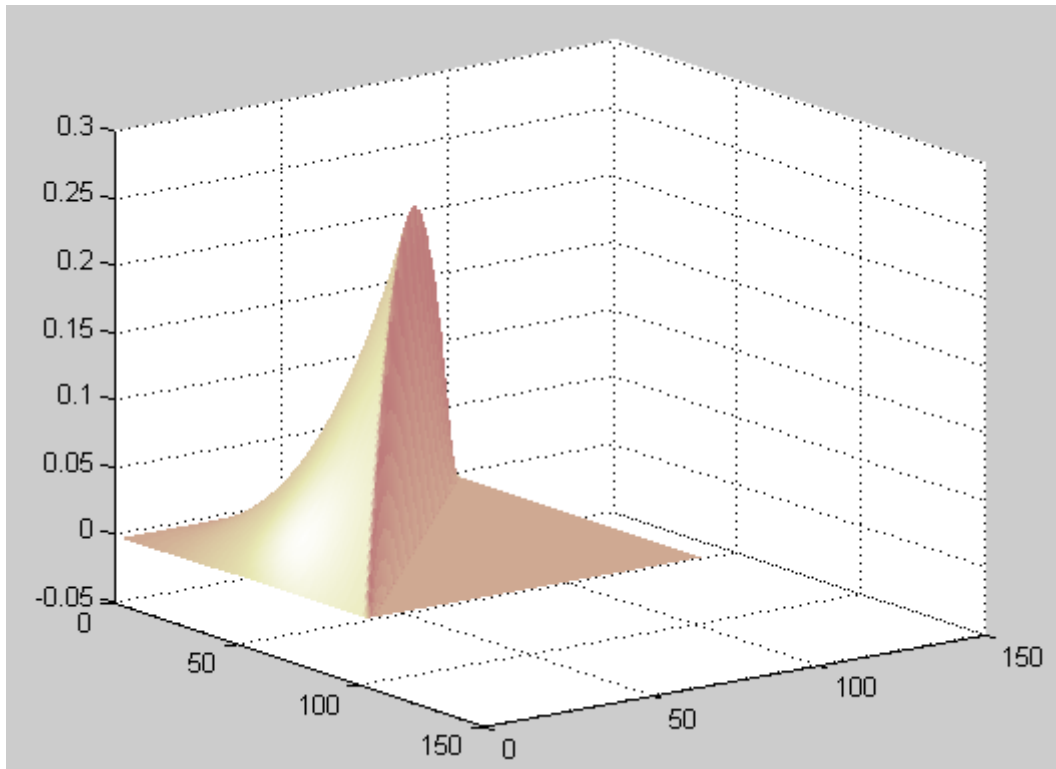


Рис. 8.4. Графік наближеного розв'язку $U(x,y)$

Розв'язання крайових задач математичної фізики, рівнянь та систем параболічного типу за допомогою *pdepe*. Для розв'язання одновимірних по координаті x крайових задач для систем параболічних диференціальних рівнянь з частинними похідними (PDE) використовується М-функція *pdepe*.

Постановка задачі. Розглядається рівняння

$$c\left(x,t,U,\frac{\partial U}{\partial x}\right)\frac{\partial U}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,U,\frac{\partial U}{\partial x}\right)\right)+s\left(x,t,U,\frac{\partial U}{\partial x}\right). \quad (8.23)$$

PDE або система PDE розв'язується для просторово-часового інтервалу $t_0 \leq t \leq t_f$ і $a \leq x \leq b$. Параметр m може приймати значення 0, 1 або 2, що відповідає плоскій, циліндричній, або сферичній симетрії.

Для $t = t_0$ і всіх значень x розв'язок задовольняє початкові умови вигляду

$$U(x,t_0)=U_0(x). \quad (8.24)$$

Для всіх t , а також $x = a$, $x = b$ компоненти розв'язку задовольняють граничні умови вигляду

$$p(x,t,U)+q(x,t)\cdot f\left(x,t,U,\frac{\partial U}{\partial x}\right)=0. \quad (8.25)$$

Синтаксис функції *pdepe*:

```
sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan)
sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan,options)
sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan,options,p1,p2...)
```

Аргументи функції *pdepe*:

- *m* – параметр, що описує симетрію задачі:
 - *m* = 0 – “плоска” симетрія (плоскопаралельна крайова задача),
 - *m* = 1 – циліндрична симетрія,
 - *m* = 2 – сферична симетрія;
- *pdefun* – функція користувача, яка визначає коефіцієнти PDE;
- *icfun* – функція користувача, яка визначає початкові умови;
- *bcfun* – функція користувача, яка визначає граничні умови;
- *xmesh* – вектор [*x*₀, *x*₁, ..., *x*_{*n*}], що задає координати точок *x* (*xmesh* – масив координат вузлів одномірної сітки по просторовій координаті *x*). Елементи *xmesh* повинні задовольнити умову *x*₀ < *x*₁ < ... < *x*_{*n*}, тобто розташовуватися в порядку зростання. Кількість елементів *xmesh* повинна бути не менше 3;
- *tspan* – вектор [*t*₀, *t*₁, ..., *t*_{*f*}] моментів часу (*tspan* – масив координат вузлів одномірної сітки по часовій координаті *t*). Елементи *tspan* повинні задовольнити умову *t*₀ < *t*₁ < ... < *t*_{*f*}, тобто розташовуватися в порядку зростання. Кількість елементів *tspan* повинна бути не менше 3;
- *options* – деякі додаткові параметри розв’язувача ODE, що викликається з функції *pdepe*: *RelTol*, *AbsTol*, *NormControl*, *InitialStep*, *i* *MaxStep*. У більшості випадків значення за замовчуванням для цих параметрів породжують задовільні за точністю розв’язки. Для детальнішого ознайомлення див. функцію *odeset*;
- *p1*, *p2*, ... – необов’язкові параметри, які потрібно передати до функцій *pdefun*, *icfun*, і *bcfun*.

При виконанні функції

```
sol = pdepe (m, pdefun, icfun, bcfun, xmesh, tspan):
```

- *m* відповідає *m*;
- *xmesh* (1) і *xmesh* (end) відповідають *a* і *b*;
- *tspan* (1) і *tspan* (end) відповідають *t*₀ і *t*_{*f*};
- функція *pdefun* обчислює значення коефіцієнтів *c*, *f*, і *s* з PDE (8.23). Ця функція викликається за формою [*c*, *f*, *s*] = *pdefun* (*x*, *t*, *u*, *dudx*). Її вхідними параметрами є скалярні величини *x* і *t* і вектори *u* і *dudx*, які наближено відповідають розв’язку *U* та його частинній похідній по *x*;
- функція *icfun* обчислює початкові умови (8.24). Ця функція викликається за формою *u* = *icfun* (*x*);
- функція *bcfun* обчислює значення параметрів *p* та *q* граничних умов (8.25). Ця функція викликається за формою [*pl*, *ql*, *pr*, *qr*] = *bcfun* (*xl*, *ul*, *xr*, *ur*, *t*). Тут *ul* – наближений розв’язок на лівому кінці відрізка

$x_l = a$, u_r – наближений розв'язок на правому кінці відрізка $x_r = b$, p_l і q_l – відповідні масиви функцій p і q для лівої граничної точки x_l , аналогічно p_r і q_r відповідні масиви функцій p і q для правої граничної точки x_r .

Функція *pdepe* повертає розв'язок у багатовимірний масив *sol*.

$U_i = u_i = sol(:, :, i)$ є наближенням до i -тої компоненти вектора розв'язку U . Елемент $u_i(j, k) = sol(j, k, i)$ є наближенням U_i при $(t, x) = (tspan(j), xmesh(k))$.

При використанні функції *pdepe* у вигляді

sol = *pdepe* (*m*, *pdefun*, *icfun*, *bcfun*, *xmesh*, *tspan*, *options*, *p1*, *p2*, ...)

функція *pdepe* передає додатково параметри *p1*, *p2*, ... функціям *pdefun*, *icfun*, *bcfun*. У цьому випадку вказані функції можуть бути оформлені у вигляді

$[c, f, s] = pdefun(x, t, u, dudx, p1, p2, \dots)$

$u = icfun(x, p1, p2, \dots)$

$[pl, ql, pr, qr] = bcfun(xl, ul, xr, ur, t, p1, p2, \dots)$

Можна задати параметр *options* = [], якщо параметри режиму встановлювати не потрібно (див. функцію *odeset* для детальнішого ознайомлення).

Розглянемо деякі приклади використання М-функції *pdepe*.

Приклад 15. Розв'язати крайову задачу [9]:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + f(x, t), x \in (0, 1), t \in (0, T),$$

$$U(x, 0) = g(x), x \in [0, 1],$$

$$a_0(t)U(0, t) + b_0(t)\frac{\partial U(0, t)}{\partial x} = g_0(t), t \in [0, T],$$

$$a_1(t)U(1, t) + b_1(t)\frac{\partial U(1, t)}{\partial x} = g_1(t), t \in [0, T].$$

для наступних даних:

$$T = 1, f(x, t) = \alpha(2 + \beta x^2) \exp(-t\beta),$$

$$a_0(t) = 1, b_0(t) = 1, g_0(t) = 0, a_1(t) = 1, b_1(t) = -1, g_1(t) = \alpha \exp(-t\beta), \alpha = 4, \beta = 4.$$

Нижче наведено текст відповідних М-функцій та М-файла задачі.

```
function [c, f, s] = pdex1pde(x, t, u, DuDx)
c = 1;
f = DuDx;
s = 4*(2+4*x^2)*exp(-4*t);
```

```
function u0 = pdex1ic(x)
u0 = -4*x^2;
```

```
function [pl, ql, pr, qr] = pdex1bc(xl, ul, xr, ur, t)
pl = ul;
```

```

ql = 1;
pr = ur-4*exp(-4*t);
qr = -1;

% М-файл p15_main.m
m = 0;
x = linspace(0,1,100);
t = linspace(0,1,10);
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
% Extract the first solution component as u.
u = sol(:,:,1)

% A surface plot is often a good way to study a solution.
surf(x,t,u)
shading flat
title('Numerical solution computed with 100 mesh points.')
xlabel('Distance x')
ylabel('Time t')

% A solution profile can also be illuminating.
figure
plot(x,u(end,:))
title('Solution at t = 1')
xlabel('Distance x')
ylabel('u(x,1)')

```

На рис. 8.5, 8.6 зображено результати роботи М-файла *p15_main.m*

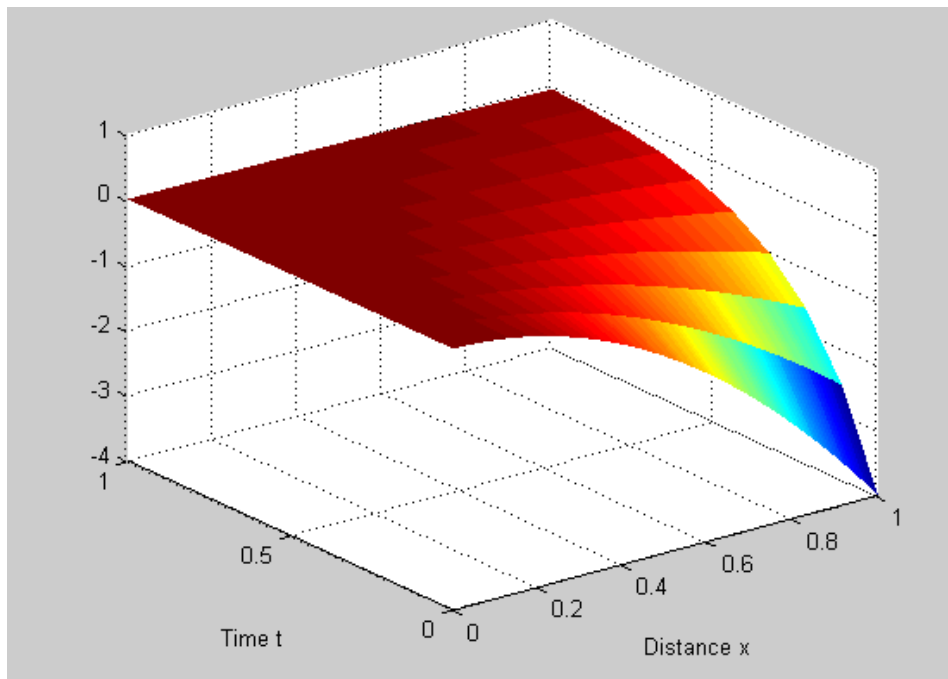


Рис. 8.5. Графік наближеного розв'язку $U(x,t)$

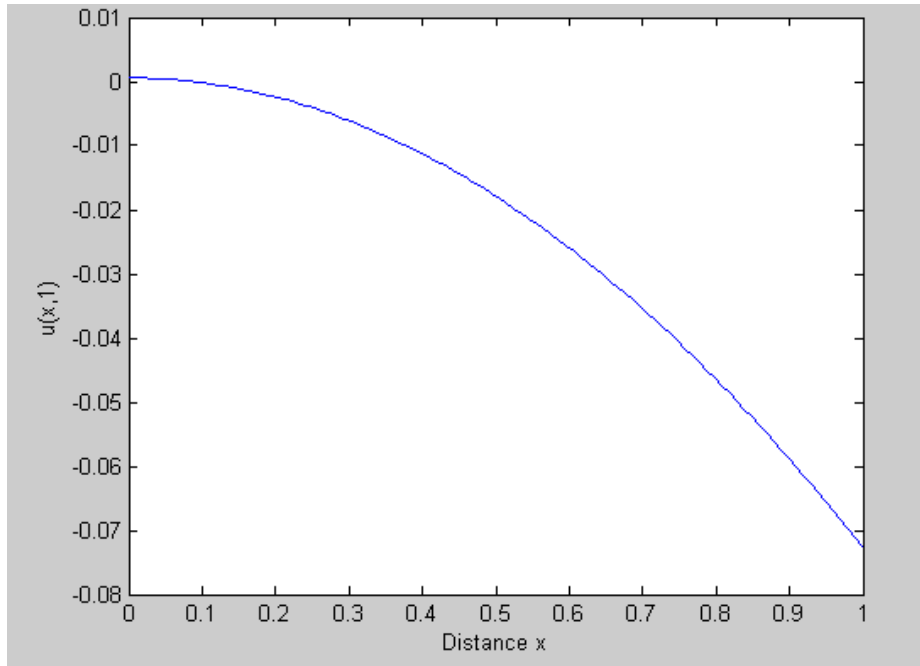


Рис. 8.6. Графік наближеного розв'язку $U(x, T)$

Приклад 16. Розв'язати крайову задачу для наступної системи квазілінійних параболічних рівнянь:

$$\begin{aligned} \frac{\partial u_1}{\partial t} &= 0.024 \frac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2), \\ \frac{\partial u_2}{\partial t} &= 0.170 \frac{\partial^2 u_2}{\partial x^2} + F(u_1 - u_2), \end{aligned} \quad x \in (0, 1),$$

де

$$F(y) = \exp(5.73y) - \exp(-11.46y).$$

Початкові умови при $t=0$ мають вигляд

$$u_1(x, 0) = 1, \quad u_2(x, 0) = 0, \quad x \in [0, 1].$$

Граничні умови на лівому ($x=0$) і правому ($x=1$) кінці відрізка задані у вигляді

$$\frac{\partial u_1(0, t)}{\partial x} = 0, \quad u_2(0, t) = 0, \quad u_1(1, t) = 1, \quad \frac{\partial u_2(1, t)}{\partial x} = 0, \quad t \geq 0.$$

Переформулюємо задачу, щоб скористатися функцією *pdepe*:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \frac{\partial}{\partial t} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \frac{\partial}{\partial x} \begin{bmatrix} 0.024(\partial u_1 / \partial x) \\ 0.170(\partial u_2 / \partial x) \end{bmatrix} + \begin{bmatrix} -F(u_1 - u_2) \\ F(u_1 - u_2) \end{bmatrix}$$

Граничні умови на лівому кінці

$$\begin{bmatrix} 0 \\ u_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0.024(\partial u_1 / \partial x) \\ 0.170(\partial u_2 / \partial x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Граничні умови на правому кінці

$$\begin{bmatrix} u_1 - 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0.024(\partial u_1 / \partial x) \\ 0.170(\partial u_2 / \partial x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Нижче наведено текст відповідних М-функцій та М-файла задачі.

```
function [c,f,s] = pdex2pde(x,t,u,DuDx)
c=[1;1];
f = [0.024; 0.17] .* DuDx;
y = u(1) - u(2);
F = exp(5.73*y)-exp(-11.47*y);
s = [-F; F];

function u0 = pdex2ic(x)
u0 = [1;0];

function [pl,ql,pr,qr] = pdex2bc(xl,ul,xr,ur,t)
pl = [0; ul(2)];
ql = [1; 0];
pr = [ur(1)-1; 0];
qr = [0; 1];

% М-файл p16_main.m
m = 0;
x = linspace(0,1,100);
t = linspace(0,1,10);

sol = pdepe(m,@pdex2pde,@pdex2ic,@pdex2bc,x,t);
u1 = sol(:, :, 1)
u2 = sol(:, :, 2)

figure
surf(x,t,u1)
title('u1(x,t)')
xlabel('Distance x')
ylabel('Time t')
shading flat
figure
surf(x,t,u2)
title('u2(x,t)')
xlabel('Distance x')
ylabel('Time t')
shading flat
```

На рис. 8.7, 8.8 зображено результати роботи М-файла *p16_main.m*

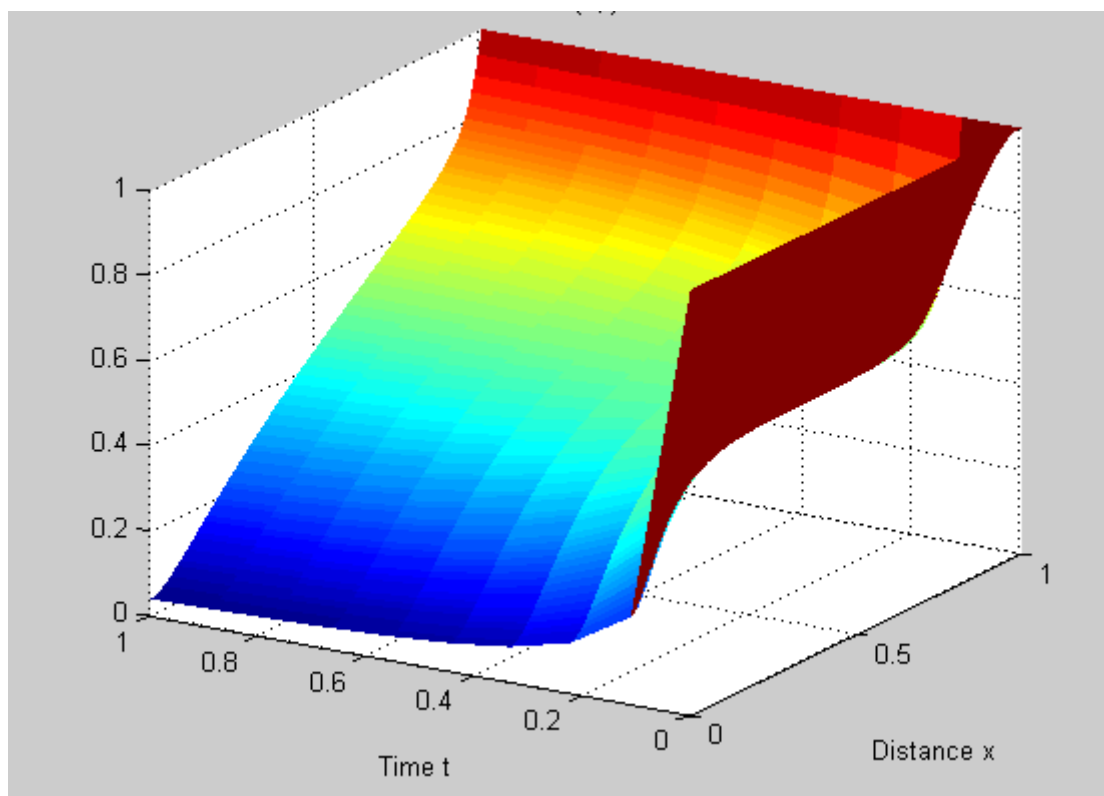


Рис. 8.7. Графік наближеного розв'язку $u_1(x, t)$

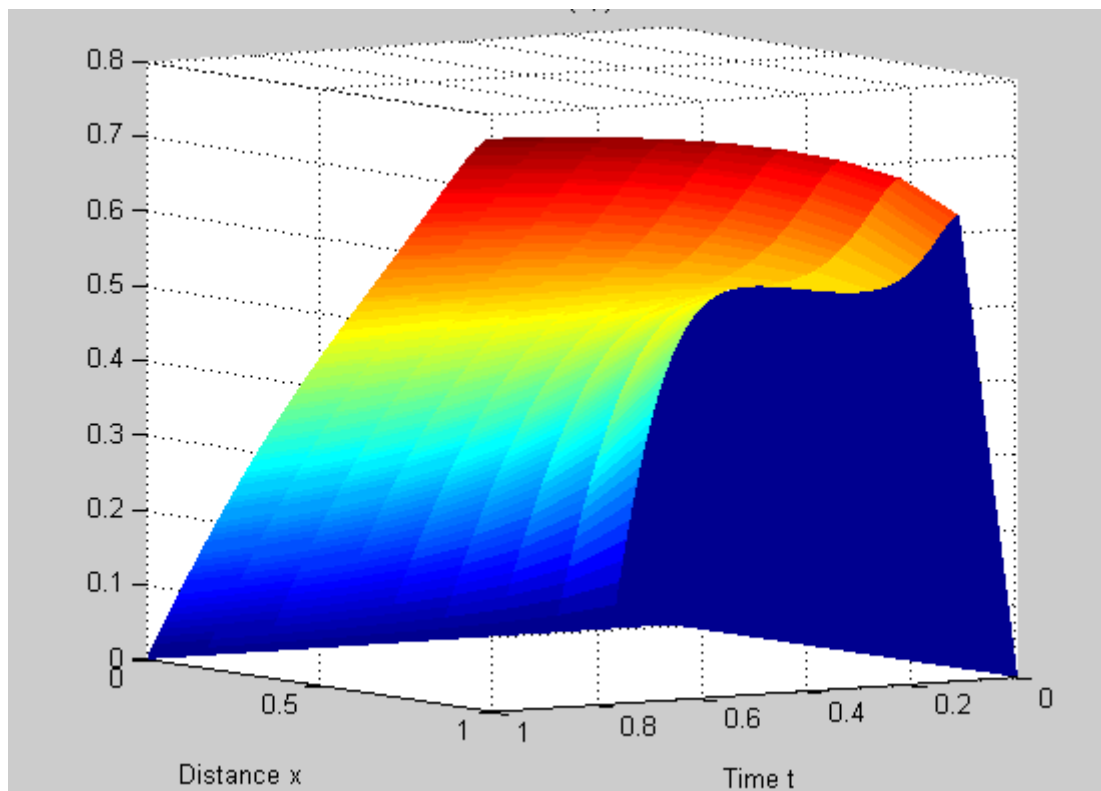


Рис. 8.8. Графік наближеного розв'язку $u_2(x, t)$

Завдання для аудиторної роботи

Розв'язати інтегральне рівняння

$$y(x) = \frac{13}{16}x - \frac{1}{6} + \frac{1}{4} \int_0^1 (3xs + 2)s ds$$

методом заміни інтеграла скінченою сумою при $n=6$, використовуючи узагальнену формулу трапецій.

Завдання для самостійної роботи

1. В прямокутній області $\bar{D} = D \cup \partial D = \{(0 \leq x \leq 1, 2) \& (0 \leq y \leq 0, 6)\}$ знайти наближений розв'язок наступної крайової задачі

$$\frac{\partial^2 U(x, y)}{\partial x^2} + \frac{\partial^2 U(x, y)}{\partial y^2} = -\phi(x, y), (x, y) \in D;$$

$$U(x, y) = g(x, y), (x, y) \in \partial D,$$

де ∂D – межа області D , а $\phi(x, y) = 0, g(x, y) = x + y$, та побудувати його графік. Для чисельного розв'язання використати різницекий метод (див. [10], стор. 104–109) і М-функцію *m_progmm.m* (матрична прогонка для розв'язування системи триточкових векторних рівнянь).

2. Розв'язати попередню задачу, використовуючи М-функцію *pdere*. (Вказівка: використати той факт, що $u(x, y, t) \rightarrow U(x, y)$, при $t \rightarrow \infty$, розглянути відповідну крайову задачу для параболічного рівняння).

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАЛІКУ

1. Призначення пакета MATLAB.
2. Сфера застосування пакета MATLAB.
3. Основні вікна середовища MATLAB.
4. Сеанс роботи в командному вікні MATLAB.
5. Основні команди рядкового редактора.
6. Основні пункти меню системи MATLAB та їх призначення.
7. Вікно управління редактора М-файлів.
8. Команди управління вікном командного режиму.
9. Основні правила роботи в режимі прямих обчислень.
10. Формат та призначення загальних команд (*help, demo/expo, info, ver, version, path, what, whatsnew, which, lasterr*).
11. Формат та призначення команд роботи з шляхами доступу до файлів (*path, path(s), addpath, rmpath*).
12. Формат та призначення команд роботи з поточним каталогом (*cd, pwd*).
13. Формат та призначення команд перегляду шляхів, роботи з файлами (*dir, ls, tempdir, tempname, edit, type, delete, !, diary*).
14. Формат та призначення команд для роботи з робочою областю (*clear, disp, length, size, who, whos, save, load, pack*).
15. Основні типи даних в MATLAB.
16. Константи, змінні, імена спеціальних змінних в MATLAB.
17. Побудова виразів в MATLAB.
18. Основні операції і спеціальні символи у виразах MATLAB.
19. Арифметичні операції в MATLAB.
20. Ієрархія при виконанні арифметичних операцій в MATLAB.
21. Операції відношення в MATLAB.
22. Логічні операції і функції в MATLAB.
23. Основні математичні функції MATLAB та їх призначення.
24. Теоретико-числові функції MATLAB та їх призначення.
25. Функції для роботи з рядками символів у MATLAB.
26. Функції часу та дати в MATLAB.
27. Функції для операцій з елементами матриць у MATLAB.
28. Функції для роботи з матрицями в MATLAB.
29. Функції для роботи з поліномами в MATLAB.
30. Функції для аналізу і обробки числових масивів у MATLAB.
31. Функції чисельного інтегрування і диференціювання в MATLAB.
32. Функції мінімізації і знаходження коренів, апроксимації та інтерполяції даних у MATLAB.

33. Функції для розв'язання задачі Коші для звичайних диференціальних рівнянь у MATLAB.
34. Основи побудови графіків у MATLAB на площині.
35. Символи управління лініями графіків у MATLAB.
36. Додаткові команди, що використовуються при побудові 2-D графіків у MATLAB.
37. Особливості побудови 3-D (тривимірних) графіків.
38. Основні структури керування алгоритмічної мови MATLAB.
39. Особливості використання функцій у MATLAB (опис, глобальні та локальні змінні, формальні параметри). Виклик М-функцій.
40. М-сценарії та їх виклик.
41. Особливості застосування функцій MATLAB при розв'язуванні крайової задачі методом стрільби.
42. Особливості розв'язання лінійного неоднорідного інтегрального рівняння Фредгольма другого роду засобами MATLAB.
43. Особливості розв'язання крайової задачі зі зв'язаними (нелокальними) крайовими умовами засобами MATLAB.
44. Особливості розв'язання крайової задачі для рівняння Пуассона засобами MATLAB.
45. Особливості розв'язання крайових задач математичної фізики, рівнянь та систем рівнянь параболічного та еліптичного типу засобами MATLAB.

ЛІТЕРАТУРА

1. *Ануфриев И.Е.* MATLAB 7 / И.Е.Ануфриев, А.Б.Смирнов, Е.Н.Смирнова. – СПб.: БХВ-Петербург, 2005. – 1104 с.
2. *Дьяконов В.П.* Математические пакеты расширения MATLAB: Специальный справочник / В.П. Дьяконов, В.В.Круглов. – СПб.: Питер, 2001. – 480 с.
3. *Кетков Ю.Л.* MATLAB 7: программирование, численные методы / Ю.Л.Кетков, А.Ю.Кетков, М.М.Шульц. – СПб.: БХВ-Петербург, 2005. – 752 с.
4. *Корн Г.* Справочник по математике. Для научных работников и инженеров / Г.Корн, Т.Корн. – М.: Наука, 1973. – 832 с.
5. *Кривилев А.В.* Основы компьютерной математики с использованием системы MATLAB / А.В.Кривилев. – М.: Лекс-Книга, 2005. – 496 с.
6. *Лазарев Ю.Ф.* Моделирование процессов и систем в MATLAB / Ю.Ф. Лазарев. – СПб.: Питер; Киев: Издательская группа BHV, 2005. – 512 с.
7. *Мартынов Н.Н.* Введение в MATLAB 6 / Н.Н.Мартынов. – М.: КУДИЦ-ОБРАЗ, 2002. – 352 с.
8. *Половко А. М.* MATLAB для студента / А.М.Половко, П.Н.Бутусов. – СПб.: БХВ-Петербург, 2005. – 320 с.
9. *Попов В.В.* Лабораторні роботи та домашні завдання для самостійної роботи з дисципліни “Методи обчислень” для студентів механіко-математичного факультету / В.В. Попов, Б.П. Довгий, Є.С. Вакал. – Київ: Український фітосоціологічний центр, 2006. – 32 с.
10. *Потемкин В.Г.* Справочник по MATLAB / В.Г. Потемкин [Електронний ресурс]. – Режим доступу : http://www.nsu.ru/matlab/MatLab_RU/ml/book2/index.asp.htm (15.01.2003).
11. Программное обеспечение ЭВМ Мир-1 и Мир-2: В 3 т. – Киев: Наукова думка, 1976. – Т.2: Программы. – 372 с.
12. *Самарский А.А.* Методы решения сеточных уравнений / А.А. Самарский, Е.С. Николаев. – М.: Наука, 1978. – 592 с.

ЗМІСТ

ПЕРЕДМОВА	3
Тема №1	
ВСТУП ДО СИСТЕМИ MATLAB	4
Тема №2	
ОПЕРАЦІЙНЕ СЕРЕДОВИЩЕ MATLAB І ОСНОВНІ КОМАНДИ	6
Тема №3	
ТИПИ ДАНИХ ТА ПОБУДОВА ВИРАЗІВ	11
Тема №4	
ОСНОВНІ ДІЇ ЛІНІЙНОЇ АЛГЕБРИ	19
Тема №5	
НАБЛИЖЕНІ ОБЧИСЛЕННЯ ЗАСОБАМИ MATLAB	24
Тема №6	
ОСНОВИ ПОБУДОВИ ГРАФІКІВ.....	27
Тема №7	
МОВА ПРОГРАМУВАННЯ І ПОБУДОВА М-ФАЙЛІВ	37
Тема №8	
ПРИКЛАДИ РОЗВ’ЯЗУВАННЯ ПРИКЛАДНИХ ЗАДАЧ	49
КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАЛІКУ	74
ЛІТЕРАТУРА	76