

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
УКРАЇНИ**

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Методичні вказівки
для самостійної роботи студентів з дисципліни

**“МОДЕЛЮВАННЯ ФІЗИКО-ХІМІЧНИХ ПРОЦЕСІВ У
РАДІОЕКОЛОГІЇ”**

Одеса - 2012

Методичні вказівки до самостійної роботи студентів з дисципліни “Моделювання фізико-хімічних процесів у радіоекології”/ Худинцев М.М., Співак А.Я. – Одеса, ОДЕКУ, 2012. – 35с.

Методичні вказівки призначенні для студентів IV курсу денної форми навчання за напрямом “Екологія, охорона навколишнього середовища та збалансоване природокористування”, спеціалізація – “Радіоекологія”.

ЗМІСТ

Передмова.....	4
1 Місце і значення навчальної дисципліни	5
2 Структура навчальної дисципліни	7
3 Програма модулів курсу	13
3.1 Програма лекційного модуля	13
3.2 Програма практичного модуля.....	15
4 Повчання до послідовного вивчення теоретичного матеріалу	19
5 Організація поточного та підсумкового контролю знань	33

ПЕРЕДМОВА

Методичні вказівки для самостійної роботи студентів спрямовані на полегшення засвоєння студентами матеріалів курсу “Моделювання фізико-хімічних процесів у радіоекології” протягом 7(залік)-8(іспит) семестрів, на який відводиться 47 год. лекційних, 62 практичних та 32 лабораторних занять, а також 172 год. самостійної роботи студентів.

Вибрані питання повністю формують уявлення про предмет та головні напрямки розвитку цієї дисципліни. До методичних вказівок включені найбільш складні питання робочої програми відповідного курсу, які включені до перших шести частин першого розділу програми.

На думку авторів такий підхід до викладання курсу є найбільш спрямованим на ефективне засвоєння матеріалу та формування навиків їх практичного застосування. Список додаткової літератури досить повний.

Зазначені матеріали повністю перекривають теоретичні та практичні відомості, які є необхідними для засвоєння знань та умінь, що пов'язані з використанням програмних продуктів для моделювання фізико-хімічних процесів у радіоекології. Засвоєння саме цих відомостей є найбільш складною задачею для студентів під час навчання у рамках дисципліни “Моделювання фізико-хімічних процесів у радіоекології”.

Усі питання курсу, які не знайшли відображення у методичних вказівках, повністю та вичерпно висвітлюються у посиланні [1] переліку основної літератури (Худинцев М.М. Моделювання фізико-хімічних процесів у радіоекології: Конспект лекцій. – Одеса: Вид-во «ТЕС», 2009, 90 с.) Додаткове методичне доведення цих матеріалів буде зроблене у другій частині методичних вказівок.

1 МІСЦЕ І ЗНАЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна “Моделювання фізико-хімічних процесів в радіоекології” належить до професійно-практичного (професійно-орієнтованого) циклу підготовки спеціалістів.

Напрямок підготовки – «Екологія та охорона навколишнього середовища». Спеціалізація – “Радіоекологія”.

Дисципліна належить до професійно-практичного циклу ОПП, входить до вибіркової системи ОПП, посідає вибіркоче місце у підготовці спеціалістів.

Дисципліна базується на загальних фундаментальних дисциплінах («Фізика», «Вища математика», «Хімія», «Біологія»), а також на спеціальних курсах («Радіоекологія», «Теоретичні основи фізики іонізуючого випромінювання», «Фізичні основи радіометрії та дозиметрії»). а також відповідні спеціальні курси. У подальшому знання та вміння, отримані під час оволодіння цією дисципліною, будуть безпосередньо використані у науковій, науково-практичній роботі, а також протягом виконання курсових та дипломних проектів.

Завдання дисципліни - навчити студентів будувати математичні моделі фізичних та хімічних процесів з використанням наведених в курсі математичних методів.

Робочою програмою передбачені 47 год. лекцій, 62 год. Практичних та 32 лабораторних занять, а також 172 год. самостійної роботи студентів, передбачено виконання КП та складання заліку (7 семестр) та іспиту (8 семестр).

Основна мета дисципліни “Моделювання фізико-хімічних процесів в радіоекології” – підвищення рівня загальноосвітньої, математичної та фізико-хімічної освіти у напрямку використання отриманих раніше фундаментальних знань для вирішення конкретних наукових або практичних проблем.

Предмет дисципліни – природно-наукові моделі фізичних, хімічних, біологічних, екологічних систем, їхнє моделювання за допомогою сучасних теоретичних, експериментальних та розрахункових методів.

Практична значимість дисципліни у підготовці спеціалістів полягає у її відповідності до сучасного рівня підготовки фахівців вищого рівню, відповідає усім вимогам щодо зв'язків дисципліни з іншими (у тому числі попередніми) курсами.

В результаті вивчення курсу “Моделювання фізико-хімічних процесів у радіоекології” студент повинен отримати базові знання та вміння:

- знати основні прийоми і методи обчислень;

- уміти програмувати в інтегрованому пакеті MATLAB, як в операційній системі DOS, так і в WINDOWS;

- уміти практично здійснювати етапи вирішення конкретної фізичної задачі: аналіз фізичного процесу, розробка математичної моделі, вибір чисельного методу, складання і відлагодження програми, аналіз результатів розрахунків.

Контроль поточних знань виконується на базі модульної системи організації навчання.

2 СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс з дисципліни “Моделювання фізико-хімічних процесів у радіоекології” містить в собі:

- один змістовний теоретичний та один змістовний практичний модуль у 7-ому семестрі четвертого курсу;
- два змістовні теоретичні, один змістовний практичний модуль та КП у 8-ому семестрі четвертого курсу.

Крім того, передбачено модуль наукової роботи студентів. Загальна структура дисципліни в умовах кредитно-модульної системи показана на рис 1.

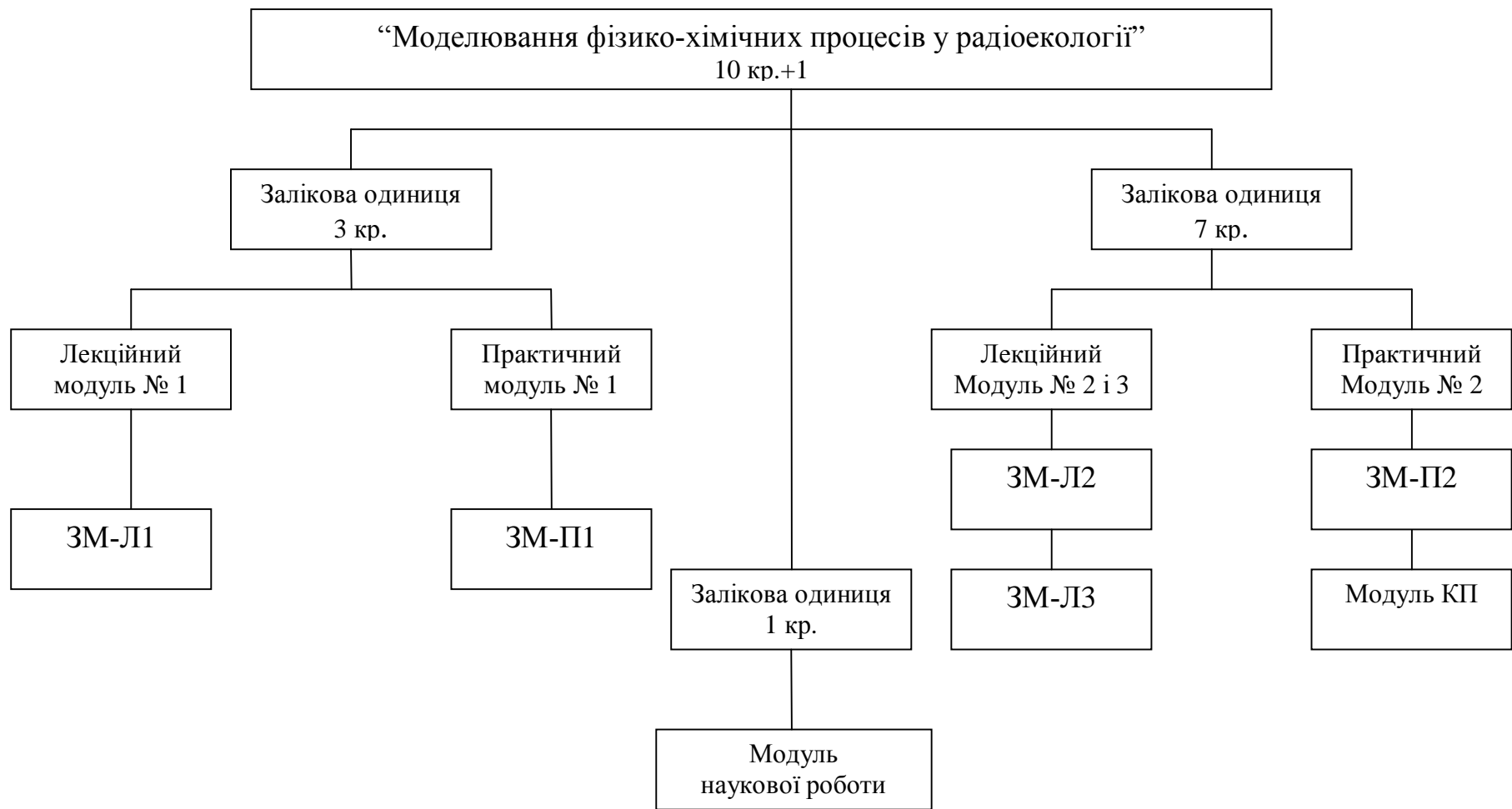


Рис.1 Структурна схема дисципліни “Моделювання фізико-хімічних процесів у радіоекології”

1) Структура лекційного курсу

1.1. Програмний пакет MATLAB

Основи синтаксису MATLAB. Команди і змінні. Константи. Арифметичні вирази. Числа. Комплексні числа і матриці. Формати видачі даних. Збереження і відновлення змінних. Введення матриць: загальні правила. Завантаження даних з зовнішнього М-файлу. Завантаження даних з MAT-файлу.

1.2. Операції над матрицями.

Транспонування. Додавання і віднімання матриць. Множення матриць. Ділення матриць. Піднесення до ступеня. Функції від матриць. Поелементні операції з матрицями. Операції порівняння. Логічні операції. Елементарні функції.

1.3. Операції з індексами.

Генерація векторів. Індукування. Порожні матриці.

1.4. Функції MATLAB.

1.4.1. Функції з декількома аргументами. Функції побудови матриць. Побудова великих матриць. Дискові і файлові функції. Виконання зовнішніх програм.

1.4.2. Функції аналізу даних. Пропущені значення. Вилучення викидів. Регресія і інтерполяція кривих.

1.4.3. Матричні функції. Трикутний розклад. Ортогональний розклад. Декомпозиція по сингулярним значенням. Власні значення. Ранг.

1.4.4. Функції роботи з поліномами і обробки. Поліноми. Обробка сигналів. Фільтрування. Швидке перетворення Фур'є.

1.4.5. Графічні функції. Двумірні графіки в координатах X-Y. Прості графіки. Множина кривих. Вигляди і кольори ліній. Вигляди ліній. Кольори. Фіктивні і комплексні дані. Логарифмічні, полярні і столбові діаграми. Тривимірні поверхні. Управління екраном. Ручне масштабування координатних осей. Отримання твердої копії.

1.5. Оператори управління процесом обчислень.

Цикл for. Цикл while. Оператори if і break.

1.6. М-файли:

Командні файли-програми і файли-функції. Командні (script) файли-програми. Файли-функції. Команди echo, input, pause, keyboard. Рядки і макровизначення. Зовнішні програми. Отримання графічної твердої копії. Імпорт-експорт і передача даних.

1.7. Чисельне інтегрування в системі MATLAB

Функції `quad` і `quad8`.

1.8. Чисельне вирішення звичайних диференціальних рівнянь та їх систем в системі MATLAB

Функції `ode23` і `ode45`.

1.9. Чисельний розв'язок диференціальних рівнянь у часткових похідних.

1.10. Інтегральні перетворення

Перетворення Лапласа. Фур'є-перетворення. Перетворення Бесселя. Застосування перетворень до конкретних фізичних задач.

1.11. Методи чисельного інтегрування

Прості засоби чисельного інтегрування. Чисельне інтегрування багатьох інтегралів. Обчислення інтегралів методом Монте-Карло. Погрешність методу Монте-Карло. Нерівномірні розподіли імовірностей.

1.12. Метод молекулярної динаміки

Потенціал міжмолекулярної взаємодії. Чисельні алгоритми та крайові умови. Програма молекулярної динаміки. Виміри макроскопічних величин.

1.13. Моделювання хаотичного руху динамічних систем

Просте одновимірне відображення. Дувимірне відображення. Універсальні властивості нелінійних відображень. Перехід до хаосу. Фазова діаграма.

1.14. Перколяція, фрактали систем та моделі кінетичного зросту і клітинні автомати

Перколяція. Критичні показники і кінечновимірне масштабування. Фрактали. Процеси зросту фракталів. Клітинні автомати.

1.15. Мікроканонічний ансамбль

Мікроканонічний ансамбль. Моделювання методом Монте-Карло. Одновимірний ідеальний газ. Температура і канонічний ансамбль. Модель Ізінга. Потік тепла.

2) Структура практичного розділу

2.1. Структура розділу практичних занять

- 2.1.1. Локальна рівновага та основне рівняння термодинаміки нерівноважних процесів. Рівняння балансу та закони збереження різних величин.
- 2.1.2. Термодинаміка лінійних необоротних процесів. Лінійний закон. Співвідношення Онзагера.
- 2.1.3. Термоелектричні явища.
- 2.1.4. Нелінійна термодинаміка. Універсальний критерій еволюції Гленсдорфа-Прігожина.
- 2.1.5. Фізична кінетика. Броунівський рух. Рівняння Ланжевена.
- 2.1.6. Формула Ейнштейна для середнього квадрата швидкості броунівської частинки.
- 2.1.7. Броунівський рух осцилятора.
- 2.1.8. Рівняння Фокера-Планка.
- 2.1.9. Часткові функції розподілу.
- 2.1.10. Вираз термодинамічних величин за допомогою часткових функцій розподілу.
- 2.1.11. Ланцюжок рівнянь для часткових функцій розподілу.
- 2.1.12. Кінетика фазових переходів.

Примітка. Програмою передбачається написання курсового проекту.

2.2. Структура розділу лабораторних занять

- 2.2.1. Програмний пакет MATLAB.
- 2.2.2. Команди і змінні. Константи. Арифметичні вирази. Числа. Комплексні числа і матриці. Формати видачі даних. Збереження і відновлення змінних. Введення матриць: загальні правила. Завантаження даних з зовнішнього M-файлу. Завантаження даних з MAT-файлу.
- 2.2.3. Операції над матрицями. Функції від матриць. Логічні операції. Елементарні функції. Операції з індексами. Генерація векторів. Індексуювання.
- 2.2.4. Функції MATLAB. Дискові і файлові функції. Виконання зовнішніх програм. Функції аналізу даних. Матричні функції. Функції роботи з поліномами і обробки. Поліноми. Обробка сигналів. Фільтрування. Швидке перетворення Фур'є.
- 2.2.5. Графічні функції. Отримання твердої копії.
- 2.2.6. M-файли: командні файли-програми і файли-функції. Командні (script) файли-програми. Файли-функції.
- 2.2.7. Чисельне інтегрування.
- 2.2.8. Чисельне розв'язання диференціальних рівнянь.
- 2.2.9. Динаміка систем багатьох частинок.

- 2.2.10. Хаотичний рух динамічних систем.
- 2.2.11. Клітинні автомати, фрактали і моделі кінетичного зросту.
- 2.2.12. Мікроканонічний ансамбль.

2.3. Перелік тем курсових проектів

- 2.3.1. Моделювання фізико-хімічних процесів методом клітинних автоматів.
- 2.3.2. Моделювання кінетичних процесів методом Монте-Карло.
- 2.3.3. Моделювання канонічних ансамблей за допомогою моделі Ізінга.
- 2.3.4. Фрактали: моделювання росту фрактальних кластерів.
- 2.3.5. Моделювання квантових систем методом Монте-Карло та методом випадкових блукань.
- 2.3.6. Моделювання перколяційних процесів.
- 2.3.7. Моделювання гранульованих систем методом молекулярної динаміки.
- 2.3.8. Моделювання динамічних систем: одновимірні відображення.
- 2.3.9. Моделювання кінетичного росту кластерів методом Монте-Карло.
- 2.3.10. Моделювання динамічних систем: двовимірні відображення.
- 2.3.11. Моделювання кінетичних процесів методом випадкових блукань.

Перелік питань для самоперевірки

1. Створити алгоритм моделювання процесу альфа-розпаду методом клітинних автоматів.
2. Створити алгоритм моделювання дифузії радіоактивної домішки методом Монте-Карло.
3. Канонічний ансамбль. Одновимірна модель Ізінга.
4. Способи моделювання росту фрактальних кластерів.
5. Алгоритм моделювання перколяції у двовимірній системі.
6. Описати розсіяння електро-магнітної хвилі на одновимірній гранульованій системі.
7. Одновимірне відображення. Приклади
8. Алгоритми моделювання процесів методом випадкових блукань у двовимірному випадку.

3 ПРОГРАМА МОДУЛІВ КУРСУ

Перелік посилань до розділів

Змістовні модулі	Розділи програми (назва)	Джерело	стор.	Примітки
ЗМ-Л1	Вступ	O[1]	5	
	Прогр.пакет Matlab	O[1]	6	
	Основи синтаксису	O[1]	12, 25	
	Операцій з матрицями та індексами	O[1]	7,15,27	складний для багатьох студентів
	Функцій Matlab	O[1]	32	
ЗМ-Л2	Оператори управління	O[1]	29	
	М-файли	O[1]	32	
	Чисельне інтегрування	O[1]	41	
	Чисельне розв'язання ДР	O[1]	38	
	Інтегральні перетворення	O[1]	37	
ЗМ-Л3	Методи чисельних розрахунків	O[1]	38	
	Метод молекулярної динаміки. Метод Монте-Карло.	O[1]	38 41	
	Перколяція. Фрактали.	M[1]	3-27	
	Мікроканонічний ансамбль	O[1]	47	
	Моделювання хаотичного руху	O[1]	64,72	

Знання, якими повинен оволодіти студент після вивчення модулів

ЗМ-Л1 Поняття про синтаксис команд та структури з них, формати даних та операції з ними.

ЗМ-Л2 Поняття про основні методи та алгоритми чисельного інтегрування, розв'язку диференційних рівнянь та їх систем.

ЗМ-Л3 Поняття про методи Монте-Карло, молекулярної динаміки та хаотичного блукання.

ЗМ-П1 використовувати синтаксис команд та структури з них, формати даних та операції з ними для створення алгоритмів чисельного інтегрування, розв'язку диференціальних рівнянь та їх систем.

ЗМ-П2 використовувати методи Монте-Карло, молекулярної динаміки та хаотичного блукання для моделювання складних систем.

**Найважче навчально-методичне забезпечення змістовних модулів
ЗМ-Л1, ЗМ-Л2, ЗМ-П1, ЗМ-П2:** Основна і додаткова література.

Основна

1. Худинцев М.М. Моделювання фізико-хімічних процесів у радіоекології: Конспект лекцій. – Одеса: Вид-во «ТЕС», 2009, 90 с.
2. Потемкин В.Г. Система инженерных и научных расчетов МАТЛАБ 5.х: – В 2-х т. Том 1. 366 с., Том 2. 304 с. – М.: Диалог-МИФИ, 1999.
3. Тихонов А.Н., Костомаров Д.П. Вводные лекции по прикладной математике: Учеб. Пособие. М.:Наука, 1984.
4. Самарский А.А. Введение в численные методы: Учеб. Пособие. М.:Наука, 1982.
5. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. Часть 1 и 2. М.:Мир, 1990.
6. Жаблон К., Симон Ж.-К. Применение ЭВМ для численного моделирования в физике. М.:Наука, 1983.

Додаткова

1. Румер Ю.Б., Рывкин М.Ш. Термодинамика, статистическая физика и кинетика. М.: Наука, 1972.
2. Базаров И.П., Геворкян Э.В., Николаев П.Н. Неравновесная термодинамика и физическая кинетика. М.: Изд-во МГУ, 1989.
3. Ландау Л.Д., Лифшиц Е.М. Статистическая физика, часть 1. М.: Наука, 1976.
4. Хуанг К Статистическая механика. М.: Мир, 1966.
5. Задачи по термодинамике и статистической физике. Под ред. П. Ландсберга. М.: Мир, 1974.

Методичні вказівки

1. Герасимов О.І., Худинцев М.М. Методичні вказівки до лабораторних робіт з дисциплін «Методи нелінійного аналізу та динамічні процеси у радіоекології», «Моделювання фізико-хімічних процесів у радіоекології».-Одеса, ОДЕКУ, 2003, 29с.
2. Худинцев М.М. Методичні вказівки до виконання курсових та дипломних робіт (проектів).-Одеса: «ТЕС», ОДЕКУ, 2007, 20с.

4 ПОВЧАННЯ ДО ПОСЛІДОВНОГО ВИВЧЕННЯ ТЕОРЕТИЧНОГО МАТЕРІАЛУ

Загальні відомості MATLAB

MATLAB – це високопродуктивна мова для технічних розрахунків. Він містить у собі обчислення, візуалізацію і програмування в зручному середовищі, де задачі та розв'язки надаються у вигляді, який є наближеним до математичного. MATLAB дозволяє виконувати математичні розрахунки, створювати алгоритми, здійснювати моделювання, проводити аналіз даних, їх дослідження та візуалізацію з використанням наукової та інженерної графіки.

MATLAB – це інтерактивна система, у якій основним елементом даних є масив. Це дозволяє розв'язувати різні задачі, які пов'язані з розрахунками, насамперед у яких використовуються матриці й вектори, у кілька разів швидше, ніж при написанні програм з використанням "скалярних" мов програмування (зокрема, Сі або Фортран).

Слово MATLAB означає матрична лабораторія (matrix laboratory). MATLAB був спеціально написаний для забезпечення легкого доступу до LINPACK і EISPACK, які являють собою сучасні програмні засоби для матричних обчислень.

В MATLAB важлива роль приділяється спеціалізованим групам програм, які називаються *toolboxes*. Вони дуже важливі для більшості користувачів MATLAB, тому що дозволяють вивчати й застосовувати спеціалізовані методи. *Toolboxes* – це всебічна колекція функцій MATLAB (М-файлів), які дозволяють вирішувати значний клас задач. *Toolboxes* застосовуються для обробки сигналів, систем контролю, нейронних мереж, нечіткої логіки, вейвлетів, моделювання і т.д.

Система MATLAB складається з п'яти основних частин, які будуть розглянуті нижче:

- мова MATLAB ;
- середовище MATLAB ;
- графіка MATLAB ;
- бібліотека математичних функцій;
- програмний інтерфейс.

Щоб запустити MATLAB на PC, двічі клацніть на іконку MATLAB. Для запуску в системі UNIX напишіть *MATLAB* у рядку операційної системи. Для виходу з MATLAB необхідно набрати *quit* у рядку MATLAB.

Якщо вам необхідно одержати додаткову інформацію, наберіть *help* у рядку MATLAB або виберіть Help у меню на PC.

Додаткова інформація по розділах MATLAB одержується за допомогою команди *help help*.

Команди та змінні

Всі команди MATLAB інтерпретуються або транслуються системою, після чого виконуються, а змінні зберігаються для подальшого використання. Команди мають вигляд:

$$\text{Змінна}=\text{вираз}; \quad (1a)$$

$$\text{Змінна}=\text{ вираз}; \quad (1б)$$

або

$$\text{вираз}; \quad (2a)$$

$$\text{вираз}; \quad (2б)$$

Результат виконання (1a) і (2a) висвітлюється на екрані. (1б) та (2б) результат не висвітлюється, таким чином «;» давить його вивод на екран. Результат обчислень по командам (2) започинається лише в тимчасовій змінній з ім'ям `ans` (від `answer` – відповідь). Вирази укладаються з операторів і спецсимволів, функцій та імен змінних. Для переносу виразу на наступний рядок використовуються крапки (не менше двох крапок підряд). Наприклад:

$$z = 2 + \text{sqrt}(7 + a) - \sin(b - 1) \dots \\ + \cos(c) \hat{a} \hat{i} + z$$

Ім'я функції та змінної може складатись з довільного числа літер та цифр, розпочинаючи з літери, але започинаються лише перші 19. MATLAB розрізняє регістри – великі та маленькі літери сприймаються як різні, наприклад `a` і `A` – це різні змінні. Всі функції пишуться маленькими літерами: `cos(x)` – гіперболічний косинус, а `cosh(x)` – невідома функція. Щоб великі та маленькі літери не розрізнялись, потрібно скористуватись функцією `casesen`. Вона переключас режим чутливості до регістру лише латинських літер туди та назад при черговому зверненні. Первісно при завантаженні MATLAB цей режим включений. В MATLAB немає обмежень на використання виразів в якості будь-яких аргументів або індексів. Можливо звернення до функцій з різним числом параметрів в залежності від сенсу звернення. Немає описів вимірності та типів змінних та масивів (векторів та матриць) – вони визначаються та перевизначаються динамічно в процесі роботи. Це, з одного боку, надає величезну свободу, наближення до звичайного математичного запису, а з іншого – з'являється небезпека внесення логічних помилок, що потребує особливої уваги.

Константи

MATLAB автоматично створює під час запуску декілька змінних:

`eps` – допущення при визначення таких показників як ранг, виродження та т.п. Це число визначається мантисою найбільшого числа з плаваючою крапкою. Для стандарту IEEE – арифметики (що апаратно реалізована в сопроцесорах Intel 8x87 і процесорах 80486 та Pentium), величина $\text{eps}=2^{-(52)}$, що приблизно дорівнює $2.22 \cdot 10^{-16}$. Це значення має `eps` за умовчанням. Користувач може задати його будь-яким.

`pi` – `пi`, що вираховується за відомою формулою $4 * \text{atan}(1)$.

Inf – нескінченність (infinity), появлення якої звичайно викликає аварійне закінчення програми в інших мовах програмування. Inf визначається як 1/0 і видається на екран у вигляді 'б'(знак нескінченності для не русифікованих комп'ютерів) або ∞ з попередження Warning: Divide by Zero(ділення на нуль)

NaN – скорочення від Not a Number (не число, невизначеність). Може бути отримана як результат операцій типу ∞/∞ або $0/0$.

Значення всіх змінних, що існують в процесі роботи, можна діставати за допомогою команд who (хто?) whos (чій?).

who показує змінні, що визначені до даного моменту.

whos крім того, показує їхній розмір і тип.

Дійсна змінна може мати мантису довжиною до 16 значущих десяткових цифр і порядок від 10^{-308} до 10^{308} .

Арифметичні операції

В MATLAB і можна використовувати наступні арифметичні вирази:

+ додавання,

- віднімання,

* множення,

/ пряме ділення,

\ зворотне ділення,

^ піднесення до ступення.

Круглі дужки використовуються для підвищення наочності і визначення порядку дій. Роз'яснення вимагає лише зворотне ділення: 1/2 те ж саме, що й 2/1; система алгебраїчних рівнянь $Ax=b$ має рішення $x=A/b$, де A і b є матриці. Таким чином ці дії необхідно розрізняти коли виконується ділення матриць.

Дійсні числа

Числа запроваджуються звичайним шляхом, наприклад:

1 -23 0.4 .567

2.123e4 -56.789e-12 8.9876543

Комплексні числа і матриці.

Найпростішим способом отримання комплексного числа є обчислення уявної одиниці $j = \sqrt{-1}$ за допомогою команди

$$j = \text{sqrt}(-1)$$

Після того, як j обчислене, його можна використовувати як уявне число в усіх виразах, наприклад:

$$z = .23e - 4.56 * j$$

$$w = r \exp(j * \text{omega} * t)$$

Пробіли ставити поміж '+' (або '-') не можна, тому що це буде означати наступний елемент матриці.

Формати видачі даних

Всі обчислення виконуються з подвійною точністю. Команда `format` впливає лише на виведення результатів. За умовчанням (що відповідає команді `format short`)

- цілі виводяться без десяткової крапки,
- дійсні числа з п'ятьма знаками мантиси.

Інші можливості ілюструються прикладами вводу вектора `[4/3 1.2345e-6]` в різних форматах:

`format short`

1.3333 0.0000

`format short e`

1.3333e+000 1.2345e-006

`format long`

1.3333333333333333 0.000001234500000

`format long e`

1.3333333333333333e+000 1.2345000000000e-006

`format hex` (шістнадцятий формат)

33ff555555555555 3eb4b6231abfd271

`format +` (маємо + замість позитивного елемента; - – замість негативного елемента; пробіл – заміна нульового елемента)

Матриці

Кращий спосіб почати роботу з MATLAB – це навчитися роботі з матрицями. В MATLAB матриця – це прямокутний масив чисел. Особливе значення надається матрицям 1×1 , які є скалярами, і матрицям, що мають один стовбець або один рядок, – векторам. MATLAB використовує різні способи для зберігання чисельних і нечисельних даних, однак спочатку найкраще розглядати всі дані як матриці. MATLAB організований так, щоб всі операції були як можна більш природними. У той час як інші програмні мови працюють із числами як елементами мови, MATLAB дозволяє вам швидко й легко оперувати із цілими матрицями.

Побудова матриць

Ви можете задавати матриці в MATLAB декількома способами:

- записати повний перелік елементів
- завантажувати матриці із зовнішніх файлів
- генерувати матриці, використовуючи вбудовані функції
- створювати матриці за допомогою власних функцій у М-файлах

Запис елементів повинен задовольняти наступним умовам:

- відокремлення елементів рядка пробілами або комами
- викормстання крапки з комою для позначення закінчення кожного рядка
- оточення переліку елементів квадратними дужками [].

Наприклад: магічна матриця

`A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]`

MATLAB відобразить матрицю A так:

```
A =  
16      3      2     13  
5       10     11      8  
9       6       7     12  
4       15     14      1
```

Матриця автоматично запам'ятовується середовищем MATLAB . Ми можемо до неї звернутися так:

A (та натисніть клавішу Enter).

Операції підсумовування елементів, транспонування та діагоналізації матриці

У наведеному прикладі, якщо знайти суми елементів уздовж будь-якого рядка, стовпця, або головної діагоналі, ви завжди отримаєте однакові числа. Для знаходження сум стовпців потрібно записати:

```
sum(A)
```

MATLAB видасть відповідь

```
ans =  
34 34 34 34
```

Коли вихідна змінна не визначена, MATLAB використовує змінну *ans*, коротко від *answer* – відповідь, для зберігання результатів обчислення.

Для одержання суми у рядках потрібно спочатку транспонувати матрицю (тобто змінити стовпці та рядки місцями), підрахувати суму в стовпцях, а потім транспонувати результат. Операція транспонування позначається апострофом або одинарними лапками. Вона дзеркально відображає матрицю щодо головної діагоналі й міняє рядки на стовпці. У такий спосіб A'

викликає

```
ans =  
16      5      9      4  
3       10     6     15  
2       11     7     14  
13      8     12      1
```

A вираз

$\text{sum}(A)'$ викликає результат вектор-стовпець, який містить суми в рядках
 $\text{ans} = 34\ 34\ 34\ 34$

Суму елементів на головній діагоналі можна легко одержати за допомогою функції *diag*, що вибирає цю діагональ.

```
diag(A)
```

```
ans = 16 10
```

```
7 1
```

А функція $\text{sum}(\text{diag}(A))$ викликає

```
ans = 34
```

Для суми елементів іншої діагоналі, яка називається побічною, немає спеціальної функції.

Індекси

Елемент у рядку i і столбці j матриці A позначається $A(i,j)$. Наприклад, $A(4,2)$ - це число в четвертому рядку й другому стовпці. Для нашого магічного квадрату $A(4,2) = 15$. Таким чином, можна обчислити суму елементів у четвертому стовпці матриці A , набравши

$$A(1,4) + A(2,4) + A(3,4) + A(4,4)$$

Одержимо

$$\text{ans} = 34$$

Також можливо звертатися до елементів матриці через один індекс, $A(k)$. Це звичайний спосіб посилання на рядки й стовпці матриці. Але його можна застосовувати тільки із двовимірними матрицями. У цьому випадку масив вважається вектором, який сформований із стовпців вихідної матриці.

Так, для нашого магічного квадрата, $A(8)$ – це інший спосіб посилатися на значення 15, що зберігається в $A(4,2)$.

Якщо ви намагаєтеся використати значення елемента поза матрицею, MATLAB ви[^]-дасть помилку:

```
t=A(4,5)
```

```
??? Index exceeds matrix dimensions.
```

З іншого боку, якщо ви зберігаєте значення поза матрицею, то розмір матри-ци збільшується.

```
X=A;
```

```
X(4,5) = 17
```

Відповідь має вигляд:

```
X =
```

```
16      3      2     13      0
 5     10     11      8      0
 9      6      7     12      0
 4     15     14      1     17
```

Оператор двокрапки

Двокрапка „:” , – це один з найбільш важливих операторів MATLAB .

Наприклад:

```
1:10
```

- це вектор-рядок, який містить цілі числа від 1 до 10

```
1  2  3  4  5  6  7  8  9  10
```

```
100:-7:50
```

дає

```
100  93  86  79  72  65  58  51
```

```
0:pi/4:pi
```

приводить до

```
0  0.7854  1.5708  2.3562  3.1416
```

Індексний вираз з двокрапкою описує частину матриці:

$A(1:k,j)$ – це перші k елементів j -го стовпця матриці A .

```
sum(A(1:4,4))
```

обчислює суму четвертого рядка.

Відокремлена двокрапка звертається до всіх елементів у рядку й стовпці матриці, а слово *end* – до останнього рядка або стовпця.

```
sum(A(:,end))
```

обчислює суму елементів в останньому стовпці матриці A

```
ans = 34
```

Функція *magic*

MATLAB має вбудовану функцію *magic*, яка створює магічний квадрат будь-якого розміру.

```
B=magic(4)
```

```
B =
```

```
16         2     3    13
 5         11    10     8
 9         7     6    12
 4        14    15     1
```

Ця матриця майже співпадає з матрицею A , яка розглядалася вище. Єдина відміна полягає у тому, що два середніх стовпця помінялися місцями. Для перетворення матриці B у матрицю A , переставимо їх місцями.

```
A=B(:,[1 3 2 4])
```

Це означає, що для кожного рядка матриці B елементи порядкуються за номерами рядків 1, 3, 2, 4

```
A =
```

```
16         3     2    13
 5        10    11     8
 9        67    12     6
 4        15    14     1
```

Вирази

Як і більшість інших мов програмування, MATLAB надає можливість використання математичних виразів, але на відміну від багатьох з них, ці вирази в MATLAB включають матриці. Основні складові вирази:

- змінні
- числа
- оператори
- функції

Змінні

В MATLAB немає необхідності у визначенні типу змінних або розмірності. Коли MATLAB зустрічає нове ім'я змінної, він автоматично утворює змінну й виділяє відповідний об'єм пам'яті. Якщо змінна вже існує, MATLAB змінює її склад і, якщо це необхідно, виділяє додаткову пам'ять. Наприклад,

`num_students = 25`

створює матрицю 1x1 з ім'ям `numstudents` і зберігає значення 25 у її елементі.

Імена змінних складаються з літер, цифр або символів підкреслення. MATLAB використовує тільки перші 31 символ імені змінної. MATLAB чутливий до регістрів, він розрізняє заголовні й малі літери. Тому `A` і `a` – різні змінні. Щоб побачити матрицю, яка пов'язана із змінною, введіть назву змінної.

Числа

MATLAB використає прийнятну десяткову систему числення, з необов'язковою десятковою крапкою й знаками плюс-мінус для чисел. Наукова система числення використовує літеру `e` для визначення множника ступеня десяти. Уявні числа використовують `i` або `j` як суфікс. Деякі приклади правильних чисел наведені нижче

<code>3</code>	<code>-99</code>	<code>0.0001</code>
<code>9.6397238</code>	<code>1.60210e-20</code>	<code>6.02252e23</code>
<code>1i</code>	<code>-3.14159j</code>	<code>3e5i</code>

Всі числа для зберігання використовують формат `long` з плаваючою крапкою. Числа із плаваючою крапкою мають обмежену точність – приблизно 16 значущих цифр і обмеженим діапазоном – приблизно від 10^{-3} до 10 .

Оператори

Вирази використовують звичайні арифметичні операції й правила старшинства.

- `+` додавання
- `-` вирахування
- `*` множення
- `/` ділення (для чисел та простих змінних)
- `\` ліве ділення (для векторів та матриць)
- `^` ступінь
- `'` комплексно сполучене транспонування
- `()` визначення порядку обчислення

Функції

MATLAB включає велику кількість елементарних математичних функцій, таких, як `abs`, `sqrt`, `exp`, `sin`. Обчислення квадратного кореня негативного числа не є помилкою: у цьому випадку результатом є відповідне комплексне число. MATLAB також включає спеціальні функції, наприклад, Гама-функцію або функції Бесселя. Більшість з них мають комплексні аргументи.

До переліку усіх елементарних математичних функцій звертається команда

```
help elfun
```

Для спеціальних математичних або матричних функцій – наберіть

help specfun

або

help elmat

відповідно.

Деякі функції, такі як *sqrt* і *sin*, є вбудованими. Але деякі інші функції, такі, як *gamma* і *sinh*, реалізовані в М-файлах. Тому ви можете легко побачити їхній код і, якщо буде потрібно, навіть модифікувати його.

Кілька спеціальних функцій надають значення часто використовуваних констант.

pi 3.14159265...

i мнима одиниця

j те ж саме, що й i

eps відносна точність числа із плаваючою крапкою

realmin найменше число з плаваючою крапкою

realmax найбільше число з плаваючою крапкою

Inf нескінченність

NaN не число

Нескінченність з'являється після ділення на нуль або при виконанні математичної операції, яка призводить до переповнення, тобто до перевищення *realmax*. Не число (*NaN*) генерується при обчисленні виразів типу $0/0$ або *Inf* – *Inf*, які не мають певного математичного значення.

Імена функцій не є зарезервованими, тому можливо змінювати їхні значення на нові, наприклад

eps = 1.e-6

і далі використовувати це значення в наступних обчисленнях. Початкове значення може бути відновлене в такий спосіб

clear eps

Вирази

Приклади з результатами:

rho = (1+sqrt(5))/2

rho =

1.6180

a = abs(3+4i)

a =

5

z = sqrt(besselk(4/3,rho-i))

z =

0.3730 + 0.3214i

huge = exp(log(realmax))

huge =

1.7977e+308

toobig = pi*huge

toobig =

Inf

Робота з матрицями

Цей розділ присвячено різним способам утворення матриць.

Генерування матриць

MATLAB має чотири функції, які створюють основні матриці:

zeros всі нулі
ones всі одиниці
rand рівномірний розподіл випадкових елементів
randn нормальний розподіл випадкових елементів

Деякі приклади:

```
Z = zeros(2,4)
```

```
Z =
```

```
0            0    0   0  
0            0    0   0
```

```
F = 5*ones(3,3)
```

```
F =
```

```
5            5    5  
5            5    5  
5            5    5
```

```
N = fix(10*rand(1,10))
```

```
N =
```

```
9   2   6
```

```
R = randn(4,4)
```

```
R =
```

```
-0.4326        -1.1465    0.3273    -0.5883  
-1.6656        1.1909    0.1746    2.1832  
0.1253         1.1892    -0.1867   -0.1364  
0.2877         -0.0376   0.7258    0.1139
```

Завантаження матриць

Команда *load* зчитує файли, які містять матриці, що створені у MATLAB раніше, або текстові файли, які містять чисельні дані. Текстові файли повинні бути сформовані у вигляді прямокутної таблиці чисел, які виділятимуться пробілами, з рівною кількістю елементів у кожному рядку.

Наприклад, створимо поза MATLAB текстовий файл з чотирьох рядків:

```
16.0 3.0    2.0    13.0  
5.0 10.0   11.0   8.0  
9.0 6.0    7.0    12.0  
4.0 15.0   14.0   1.0
```

Збережемо цей файл з ім'ям *magik.dat*. Тоді команда *load magik.dat* прочитає цей файл і створить змінну *magik*, яка містить матрицю.

М-файли

Ви можете створювати свої власні матриці, використовуючи М-файли, які являють собою текстові файли, що містять код MATLAB . Достатньо

створити файл з командами у командному рядку MATLAB . Збережіть його з ім'ям *.m, де * – сукупність літер, цифр та символу „_”.

Наприклад, створимо файл, що включає наступних 5 рядків:

```
A = [16.0  3.0  2.0  13.0
      5.0 10.0  11.0    8.0
      9.0  6.0   7.0  12.0
      4.0 15.0  14.0   1.0];
```

Збережемо його з ім'ям `magic.m`. Тоді вираз `magic`

прочитає файл і створить змінну `A`, яка містить вихідну матрицю.

Командне вікно

Дотепер, ми використовували лише командний рядок MATLAB , записуючи команди й вирази та отримавши результати. Якщо ваша система дозволяє вам вибирати шрифт, рекомендується використовувати шрифти з фіксованою шириною, такі як `Fixedsys` або `Courier`, для забезпечення коректного інтервалу.

Команда `format`

Команда `format` управляє чисельним форматом значень. Ця операція впливає тільки на вигляд даних на екрані. Нижче наведені формати чисел, які використовуються для відображення вектора `x` з різними компонентами:

```
x = [4/3 1.2345e-6]
```

```
format short
```

```
1.3333 0.0000
```

```
format short e
```

```
1.3333e+000 1.2345e-006
```

```
format short g
```

```
1.3333 1.2345e-006
```

```
format long
```

```
1.33333333333333 0.00000123450000
```

```
format long e
```

```
1.33333333333333e+000 1.23450000000000e-006
```

```
format long g
```

```
1.33333333333333 1.2345e-006
```

```
format bank
```

```
1.33 0.00
```

Якщо найбільший елемент матриці більше 10 або сам маленький менше 10^{-3} , MATLAB застосовує загальний масштабний коефіцієнт для форматів *short* та *long*.

У додавання до команд *format* ще існує

```
format compact
```

який забирає порожні рядки, що з'являються на екрані. Це дозволяє бачити на екрані більше інформації.

Скорочення вихідних даних

Якщо наприкінці рядка ви поставите крапку з коми, MATLAB проведе обчислення, але не відобразить їх на екрані. Це дуже важливе зауваження для великих масивів даних. Наприклад,

```
A = magic(100);
```

Редактор командного рядка

Використовуйте кнопки з стрілками для звертання до окремих рядків програми без їхнього повторного набіру.

Операційні порівняння.

Визначенні шість дій порівняння, що справедливі для матриць однакової вимірності:

< – менше

<= – менше або дорівнює,

> – більше

>= – більше або дорівнює

== – дорівнює

~= – не дорівнює

Результатом операції порівняння є матриця із 1 (true – так, справедливо, якщо співвідношення виконується) або – із 0 (false – ні, несправедливо, якщо співвідношення не виконується), що стоять на відповідних місцях.

Наприклад:

```
2+2~=4
```

```
ans=
```

```
0
```

```
3+3==9
```

```
ans=
```

```
1
```

```
x==y
```

```
ans=
```

```
0 0 0
```

Для логічних операцій відноситься декілька стандартних функцій: find, isnan, finit (див. відповідний help).

Наприклад:

```
Find (x<2.5)
```

```
ans=
```

```
1 1 0
```

результатом є матриця з 1 і 0 на місцях відповідно, що задовільняють або не задовільняють умові.

Логічні операції

Визначенно три логічні операції:

& – і (and)

| – або (or)

~ – не (not)

Всі операції справедливі для логічних матриць, що складаються із (так, true) і 0 (ні, false).

Якщо матриця, що бере участь в операції, містить інші числа, то помилка не з'явиться – всі ненульові елементи будуть розглядатись як 1 (так, true). Результатом дії не буде матриця із 1 і 0, що стоять на відповідних місцях.

Операції & і | справедливі для двох скалярів або матриць однакової вимірності, що порівнюються поелементно.

Операція ~A поверне 0 там, де у A стоять не нульові елементи і, відповідно, 1 – там, де A містить нулі.

До логічних операцій примикають стандартні функції any і all. Для вектора x:

any(x) повертає 1, якщо хоча б один елемент x є ненульовим, інакше -0;

all(x) повертає 1, якщо всі елементи x є нульовими, інакше -0.

Наприклад

```
any(x<2.5)
```

```
ans=
```

```
1
```

```
all(x<2.5)
```

```
ans=
```

```
0.
```

Операції з індексами

Генерація векторів

Знак : (двокрапки) має сенс (ВІД: ДО). Наприклад, пропозиція

```
v=1:5
```

дає

```
v= [1 2 3 4 5]
```

Виробляється генерація елементів вектору ВІД 0 ДО 5 з кроком 1 за умовчанням.

```
z=3: -1: -2
```

дає

```
z=
```

```
3 2 1 0 -1 -2
```

Підкресливо, що від 0 до 10 з кроком 1 буде згенеровано 11 точок.

Наступний приклад:

```
t=(0: 0.1:1)';
```

```
y=sin(pi*t);
```

```
[t y]
```

```
ans=
```

```
0          0
```

```
0.1000    0.3090
```

```
0.2000    0.5878
```

```
0.3000    0.8090
```

```
0.4000    0.9511
```

0.5000	1.0000
0.6000	0.9511
0.7000	0.8090
0.8000	0.5878
0.9000	0.3090

Індесування

До елементів матриці можна звертатися, як звичайно, вказуючи їх індекси у круглих дужках: $A(3,1)$, $A(2,1)$ і т.д.

Індекс може бути і вектором. Якщо x і v – вектори, то $x(v)$ означає вектор вимірності v (рівної, наприклад, n) з елементами $[x(v(1)) \quad x(v(2)) \quad \dots \quad x(v(n))]$.

Наприклад:

A – матриця 10 на 10, $x=[1 \ 3 \ 5]$, тоді

$A(x,x)$ – підматриця матриці A розміру 3 на 3 з елементів, що лежать на перетинанні 1, 3 та 5 строк і стовчиків A ;

$A(1:5, 3)$ – вектор 5 на 1 з перших 5 елементів 3-го стовпчика A ;

$A(:, 5)$ – 5-й стовпчик A ; $A(1:3, 1:3)$ – під матриця A розміру 3 на 3 з її лівого верхнього куту;

$A([1 \ 2 \ 8 \ 10], x)$ – підматриця матриці A розміру 4 на 3 з елементів, що лежать на перетинні 1,2, 8 та 10 рядків і стовчиків, що визначаються елементами x .

Якщо елементи вектору x – не цілі, то вони заокруглюються відкиданням усіх знаків після десяткової крапки.

Перетворення матриці у вектор

Нехай $A=[1 \ 2; 3 \ 4; 5 \ 6]$

$A=$

1 2
3 4
5 6

$b=A(:)$

$b=$

1
2
3
4
5
6

Виразом $A(:)$ можна користуватись для отримання матриці в формі стовпчика. Наприклад: $A(:) = 15:17$ дає:

$A=$

15
16
17

Графіка

MATLAB має широкі можливості для графічного зображення векторів і матриць, а також для створення коментарів і друку графіки.

Створення графіка

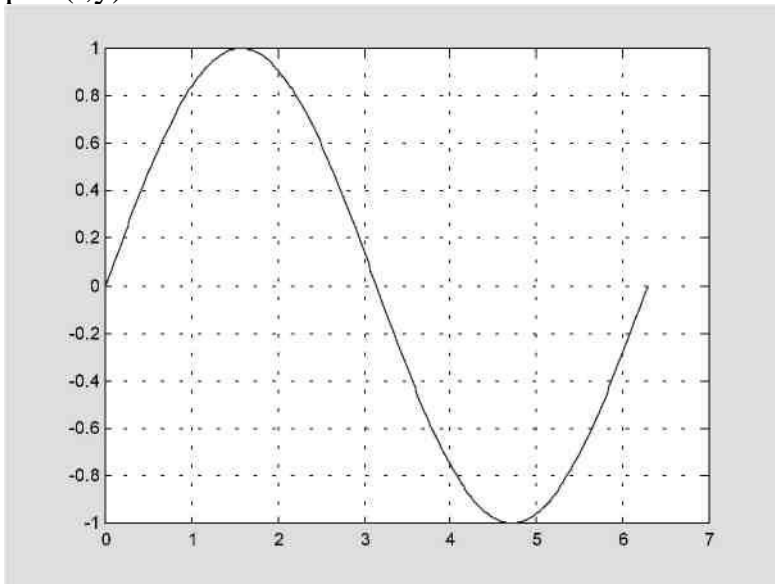
Функція `plot` має різні форми, що пов'язані із вхідними параметрами, наприклад `plot(y)` створює кусочно-лінійний графік залежності елементів від їхніх індексів. Якщо задати два вектори, як аргументи, `plot(x,y)` створить графік залежності y від x .

Наприклад, для побудови графіка значень функції \sin від нуля до 2π , потрібно зробити наступне

```
t = 0:pi/100:2*pi;
```

```
y = sin(t);
```

```
plot(t,y)
```



Можливо створити водночас декілька різнокольорових графіків

```
y2 = sin(t-.25);
```

```
y3 = sin(t-.5);
```

```
plot( t, y, t, y2, t, y3)
```

Можливі зміна кольорів, стилю ліній й маркерів. Наприклад, вираз

```
plot(x,y,'y:+')
```

будує жовтий пунктирний графік і поміщає маркери '+' у кожному крапку даних. Якщо ви визначаєте тільки тип маркера, але не визначаєте тип стилю ліній, то MATLAB виведе тільки маркери.

Детальніше про кольори та маркери дивіться за допомогою команди `help plot`.

Вікна зображень

Функція *plot* автоматично відкриває нове вікно зображення (далі вікно), якщо його не було на екрані. Якщо воно вже існує, *plot* використовуватиме його за замовчуванням. Для відкриття та вибору нового вікна наберіть

```
figure
```

або

```
figure(n)
```

де *n* – це номер у заголовку вікна. У цьому випадку результати всіх наступних команд будуть виводитися в це вікно.

Додавання кривих на існуючий графік

Команда *hold* дозволяє додавати криві на існуючий графік. Коли ви набираєте

```
hold on
```

MATLAB не стирає існуючий графік, а додає в нього нові дані, змінюючи осі, якщо це необхідно. Наприклад, наведений елемент програми спочатку створює контурні лінії функції *peaks*, а потім накладає графік тієї ж функції:

```
[x,y,z] = peaks; contour(x,y,z,20, k') hold on pcolor(x,y,z) shading interp
```

Команда *hold on* спричиняє комбінацію графіку *pcolor* з графіком *contour* в одному вікні.

Підграфіки

Функція *subplot* дозволяє виводити множину графіків в одному вікні або роздруковувати їх на одному аркуші паперу.

```
subplot(m,n,p)
```

розбиває вікно зображень на матрицю *m* на *n* підграфіків та обирає *p*-ий підграфік поточним. Графіки нумеруються уздовж першого у верхньому рядку, потім у другий і т.д. Наприклад, для того, щоб навести графічні дані в чотирьох різних підвікнах необхідно виконати наступне:

```
t = 0:pi/10:2*pi;
```

```
[X,Y,Z] = cylinder(4*cos(t));
```

```
subplot(2,2,1)
```

```
mesh(X)
```

```
subplot(2,2,2); mesh(Y)
```

```
subplot(2,2,3); mesh(Z)
```

```
subplot(2,2,4); mesh(X,Y,Z)
```

Уявні та комплексні дані

Якщо аргумент функції *plot* комплексне число, уявна частина ігнорується, за винятком випадків, коли комплексний аргумент один. Тому

```
plot(Z),
```

де Z – комплексний вектор або матриця, еквівалентно
`plot(real(Z),imag(Z))`

Наприклад,
`t = 0:pi/10:2*pi; plot(exp(i*t),'-o')`
відобразить двадцятибічний багатокутник з символами o на вершинах.

Управління вісями

Функція *axis* має кілька можливостей для налагодження масштабу, орієнтації й коефіцієнта стиснення.

Звичайно MATLAB обирає максимальне й мінімальне значення й вибирає відповідний масштаб і маркіровку вісей. Функція *axis* змінює значення за замовчуванням граничними значення, які вводяться користувачем.

`axis([xmin xmax ymin ymax])`

У функції *axis* можна також використовувати ключові слова для управління зовнішнім виглядом вісей. Наприклад

`axis square`

створює x і y осі рівної довжини,

`plot(exp(i*t))`

наступна або за *axis square*, або за *axis equal* перетворює овал у правильне коло,

`axis auto`

повертає значення за замовчуванням і переходить в автоматичний режим,

`grid off`

виключає сітку координат, а

`grid on`

включає її заново.

Підписи до вісей і заголовки

Функції *xlabel*, *ylabel*, *zlabel* додають підписи до відповідних вісей, функція *title* додає заголовок у верхню частину вікна, а функція *text* додає текст у будь-яке місце графіка. Використання TEX подання дозволяє застосовувати грецькі літери, математичні символи й різні шрифти. Наступний приклад демонструє цю можливість.

`t = -pi:pi/100:pi;`

`y = sin(t);`

`plot(t,y)`

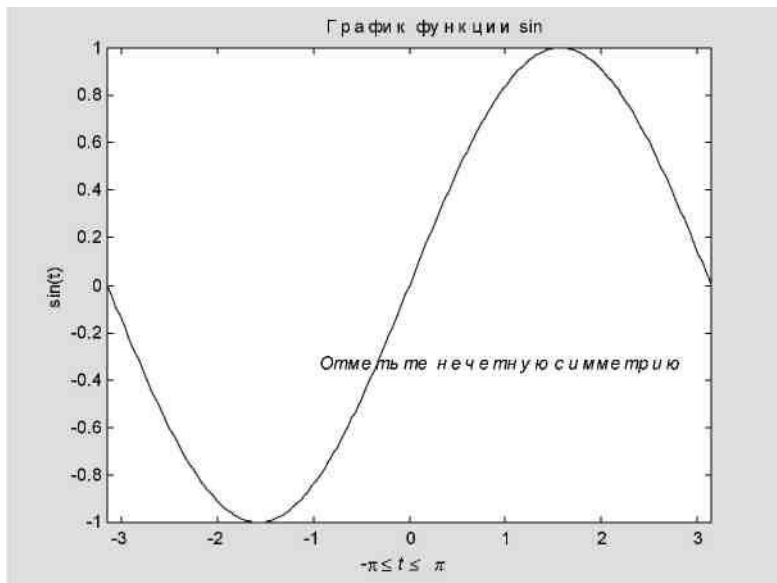
`axis([-pi pi -1 1])`

`xlabel('-\pi \leq t \leq \pi ')`

`ylabel(' sin(t) ')`

`title(' Графік функції sin ')`

`text(-1, -1/3, ' \it{Отметьте нечетную симметрию} ')`



Функції *mesh* і *surface*

MATLAB визначає поверхню, як з координати точок над координатною сіткою площини x - y , використовуючи прямі лінії для з'єднання сусідніх точок. Функції *mesh* і *surface* відображають поверхню у трьох вимірах. При цьому *mesh* створює каркасну поверхню, де кольорові лінії з'єднують тільки задані точки, а функція *surface* разом з лініями відображає кольором й саму поверхню.

Друк графіки

Опція Print у меню File і команда *print* друкують графікові MATLAB. Меню Print викликає діалогове вікно, яке дозволяє обирати загальні стандартні варіанти друку. Команда *print* забезпечує значну гнучкість надання вихідних даних і дозволяє контролювати друк з М-файлів. Результат може бути надісланий безпосередньо на принтер, обраний за замовчуванням, або збережений у заданому файлі. Можливо широке варіювання формату вихідних даних, включаючи використання PostScript.

Наприклад, команда

```
print -depsc2 magicssquare.eps
```

збереже поточне вікно зображення, як кольоровий PostScript Level 2 Encapsulated у файлі magicssquare.eps

Слід знати, що не всі Postscript принтери підтримують Level 2.

5 ОРГАНІЗАЦІЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

Контроль поточних знань виконується на базі модульної системи організації навчання. В залежності від семестру підсумковим контролем є:

- залік (7 семестр);
- іспит (8 семестр).

7 семестр

дисципліна „Моделювання фізико-хімічних процесів у радіоекології” містить по 1 змістовному модулю з теоретичної та практичної частин. В якості форми поточного контролю лекційних модулів дисципліни використовується проведення контрольної роботи, практичного модуля дисципліни – усного опитування під час практичних занять.

Критерії оцінки:

Максимальна сума балів з ЗМ-Л1 – 50 балів

Максимальна сума балів з ЗМ-П2 – 50 балів

Загальна кількість балів складає 100 балів.

Пропуски: мінус 1 бал за кожний пропуск заняття (2 години)

Залік отримують студенти які після проходження усіх заходів семестрового контролю отримали не менш як 60% балів від максимальної кількості за практичну та теоретичну частину.

8 семестр

дисципліна „Моделювання фізико-хімічних процесів у радіоекології” містить 2 змістовних модуля з теоретичної та 1 з практичної частини. В якості форми поточного контролю лекційних модулів дисципліни використовується проведення контрольної роботи, практичного модуля дисципліни – усного опитування під час практичних занять.

Курсові проекти виконуються протягом 8-го семестру, захист передбачено протягом 14-16 тижня семестру. Звіт про виконання передбачає наявність твердої та електронної копій роботи, тексту працездатної програми чисельних розрахунків, графічних або анімаційних матеріалів.

Критерії оцінки:

Максимальна сума балів з ЗМ-Л2 – 25 балів

Максимальна сума балів з ЗМ-Л2 – 25 балів

Максимальна сума балів з ЗМ-П2 – 50 балів

Загальна кількість балів складає 100 балів.

Пропуски: мінус 1 бал за кожний пропуск заняття (2 години)

До іспиту допускаються студенти, у яких фактична сума накопичених балів за практичну частину складає не менше 50%. В іншому

випадку студент вважається таким, що не виконав навчального плану дисципліни, і не допускається до іспиту.

Підсумковий семестровий контроль передбачає дві форми оцінювання успішності засвоєння студентом навчального матеріалу дисципліни:

- кількісна оцінка (бал успішності);
- якісна оцінка.

Кількісна оцінка (бал успішності) – це відсоток, який становить інтегральна сума балів, отриманих студентом на контролюючих заходах, по відношенню до максимально можливої суми балів, яка встановлена робочою програмою дисципліни.

Якісна оцінка – це оцінка, яка виставляється на підставі кількісної оцінки (бал успішності) за будь-якою якісною шкалою. На цей час в університеті використовуються такі шкали якісних оцінок:

- чотирьохбальна (відмінно, добре, задовільно, незадовільно) – для форми семестрового контролю у вигляді семестрового екзамену;
- семибальна шкала оцінювання ECTS – використовується при кредитно-модульній системі організації навчального процесу як для семестрового екзамену, так й для семестрового заліку.

Студент, який не має на початок заліково-екзаменаційної сесії заборгованості

а) по дисципліні, яка завершується заліком:

- має на останній день семестру інтегральну суму балів поточного контролю, достатню (60% та більше) для отримання позитивної оцінки, викладач виставляє якісну оцінку у заліково-екзаменаційній відомості, згідно п. 2.5 цього Положення;

- має на останній день семестру інтегральну суму балів, недостатню для отримання позитивної оцінки (менше 60%) – складає письмовий залік по тестових завданнях, що розроблені на кафедрах за процедурою, яка визначена у п. 2.15-2.19 цього Положення.

б) по дисципліні, яка завершується іспитом, складає письмовий іспит за затвердженим розкладом та процедурою, яка виписана у п. 2.7-2.10 цього Положення.

Згідно з Інструкцією про «Порядок проведення та критерії оцінювання відповідей студентів під час письмових іспитів» (п.2.3) до робочої програми дисципліни включається:

«Таблиця для визначення кількісних показників успішності за підсумками іспиту»

Кількість балів за іспит	За системою університету (у відсотках)	Визначення
1	2	3
50-45	90 - 100	відмінне виконання лише з незначною кількістю помилок
42.5-44.5	85 - 89	вище середнього рівня з кількома помилками
37.5-42	75 - 84	в загальному правильна робота з певною кількістю грубих помилок
34-37	68 - 74	непогано, але зі значною кількістю помилок
30-33.5	60 - 67	виконання задовольняє мінімальні критерії
17.5-29.5	35 - 59	з можливістю перескласти
0.5-17	1 - 34	з обов'язковим повторним курсом навчання

Бали успішності (у відсотках), які студент отримав за підсумками іспитів переносяться до графі 4 заліково-екзаменаційній відомості.

Згідно з п 1.3 «Положення про критерії оцінки знань студентів в ОДЕКУ» процедура проведення іспиту, максимальна кількість балів за кожне питання та по білету в цілому, доводиться до відома студентів на початку семестру.

Згідно з п.2.10.1. „Положення про проведення підсумкового контролю знань студентів в ОДЕКУ” загальна кількісна оцінка є усередненою між кількісною оцінкою поточних контролюючих заходів (графа 3 відомості) та кількісною оцінкою семестрового контролюючого заходу (графа 4 відомості) і виставляється у графі 5 заліково-екзаменаційні відомості (загальний бал успішності).

Якщо студент за підсумками іспиту отримав загальну кількісну оцінку менше 50% (від максимально можливої на екзамені), то викладачем виставляється у графі 5 загальний бал успішності, який дорівнює балу успішності на іспиті (графа 4).

Шкала відповідності оцінок в системах університету, національній системі та системі ЄКТАС

За шкалою ECTS	За національною системою	За системою університету (в процентах)
A	5 (відмінно)	90 - 100
B	4 (добре)	85 - 89
C	4 (добре)	75 - 84
D	3 (задовільно)	68 - 74
E	3 (задовільно)	60 - 67
FX	2 (незадовільно)	35 - 59
F	2 (незадовільно)	1 - 34

Методичні вказівки до самостійної роботи студентів IV курсу денної форми з дисципліни “Моделювання фізико-хімічних процесів у радіоекології”. Напрямок “Екологія, охорона навколишнього середовища та збалансоване природокористування”, спеціалізація – “Радіоекологія”.

Укладачі: к.ф.-м.н., доц. Худинцев М.М.; ст. викл. Співак А.Я.
Одеса, ОДЕКУ, 35 с., укр. мова.

Відповідальний за випуск – Герасимов О.І., доктор фіз.-мат. наук,
професор, зав. каф. загальної і теоретичної фізики

Підп. до друку
Умовн. друк. арк..

Формат
Тираж

Папір. друк.
Зам.№

Одеський Державний Екологічний Університет
65016, м.Одеса, вул. Львівська, 15
Надруковано з готового оригінал-макета