

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи

**ДОСЛІДЖЕННЯ СПОСОБІВ ОБРОБКИ І МЕТОДІВ ВІЗУАЛІЗАЦІЇ
ІНФОРМАЦІЇ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ**

по дисципліні

СПЕКТРАЛЬНІ МЕТОДИ ОБРОБКИ ІНФОРМАЦІЇ

Одеса – 2013

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи

**ДОСЛІДЖЕННЯ СПОСОБІВ ОБРОБКИ І МЕТОДІВ ВІЗУАЛІЗАЦІЇ
ІНФОРМАЦІЇ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ**

по дисципліні

СПЕКТРАЛЬНІ МЕТОДИ ОБРОБКИ ІНФОРМАЦІЇ

Затверджено
методичною комісією факультету комп'ютерних наук

Одеса – 2013

Методичні вказівки до лабораторної роботи „Дослідження способів обробки і методів візуалізації інформації в системі комп'ютерної математики” по дисципліні „Спектральні методи обробки інформації” для студентів 3 і 4 курсів очної форми навчання і студентів 4 курсу заочної форми навчання за напрямом „Комп'ютерні науки”, спеціальністю „Інформаційні управляючі системи та технології” спеціалізацією „Геоінформаційні системи” і спеціальністю „Комп'ютерний еколого-економічний моніторинг”. / Перелигін Б.В. – Одеса, ОДЕКУ, 2013. – 42 с.

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи
„Дослідження способів обробки і методів візуалізації інформації в системі
комп'ютерної математики”
по дисципліні
„Спектральні методи обробки інформації”

Укладач: к.т.н., доц. Перелигін Б.В.

Підписано до друку
Умов. друк. арк.

Формат
Тираж

Папір
Зам №

Надруковано з готового оригінал-макету

Одеський державний екологічний університет
65016, Одеса, вул. Львівська, 15

ЗМІСТ

	стор.
ВСТУП.....	4
1 ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ЛАБОРАТОРНОЇ РОБОТИ...	8
1.1 Керуючі команди і функції.....	8
1.2 Типи даних.....	8
1.3 Об'єктно-орієнтоване програмування.....	14
1.4 Оператори, константи, службові символи, змінні.....	16
1.5 Програмування.....	20
1.6 Математичні функції.....	21
1.7 Аналіз і обробка даних.....	23
1.8 Графічні команди і функції.....	25
2 ПРАКТИЧНА ЧАСТИНА ЛАБОРАТОРНОЇ РОБОТИ.....	37
2.1 Мета лабораторної роботи.....	37
2.2 Завдання на лабораторну роботу.....	37
2.3 Технічне забезпечення лабораторної роботи.....	38
2.4 Перелік операцій, команд і функцій, рекомендований для застосування в лабораторній роботі.....	38
2.5 Порядок виконання лабораторної роботи.....	38
2.6 Зміст звіту про лабораторну роботу.....	39
КОНТРОЛЬНІ ПИТАННЯ.....	40
ЛІТЕРАТУРА.....	41
ДОДАТОК А. Вимоги до оформлення і форма титульного аркуша звіту про лабораторну роботу.....	42

ВСТУП

Дисципліна „Спектральні методи обробки інформації” є вибірковою дисципліною підготовки бакалаврів за напрямом „Комп'ютерні науки”, шифр 6.050101, відноситься до циклу професійної і практичної підготовки (цикл В) і знайомить майбутніх фахівців з сучасними методами обробки інформації.

Мета дисципліни – підготовка майбутніх фахівців в галузі спектральної обробки інформації, одержуваної від систем моніторингу навколишнього середовища.

Останнім часом задача обробки подібної інформації стала особливо актуальною. Існуючі численні системи метеорологічного, гідрологічного, океанологічного, екологічного моніторингу наземного і космічного базування, радіолокаційні системи дистанційного моніторингу надають велику кількість важливої інформації, яка потребує грамотної обробки, у тому числі і спектральної, для отримання її характеристик з метою подальшого прийняття рішень системами управління різного рівня. Фахівці з подібними знаннями необхідні і в державних структурах, і в наукових установах, і в комерційних фірмах.

Тому завдання дисципліни полягає у вивченні математичних основ спектральної обробки, а також способів і засобів спектральної обробки інформації, яка поступає від систем вимірювання параметрів стану навколишнього середовища, штучних супутників Землі, радіолокаційних станцій і інших систем моніторингу навколишнього середовища.

У дисципліні „Спектральні методи обробки інформації” розглядаються математичні основи спектральної обробки інформації на базі Фур'є-перетворення і вейвлет-перетворення і вивчаються ті програмні засоби, що її реалізують. Практична частина дисципліни включає лабораторні роботи по вивченню і дослідженню:

- а) способів обробки і методів візуалізації інформації в системі комп'ютерної математики;
- б) спектральних методів обробки за допомогою різних модельних сигналів;
- в) застосування засобів спектрального аналізу при обробці інформації.

Дана лабораторна робота „Дослідження способів обробки і методів візуалізації інформації в системі комп'ютерної математики” присвячена вивченню і дослідженню способів обробки і методів візуалізації інформації в системі комп'ютерної математики.

Системи комп'ютерної математики (СКМ) застосовуються в математиці, обчислювальному експерименті, імітаційному моделюванні і є наборами пакетів прикладних математичних програм, що об'єднані програмною оболонкою (MatLab, MathCad, Mathematica і ін.), тобто це

одночасно і операційне середовище і мова програмування. Операційне середовище СКМ – це множина інтерфейсів, які підтримують зв'язок системи із зовнішніми додатками. Робоча область СКМ – це область пам'яті, в якій розміщені змінні системи.

Наприклад, система MatLab (скорочено від: матрична лабораторія) є інтерактивною системою для виконання інженерних і наукових розрахунків, яка орієнтована на роботу з масивами даних. Ця система використовує математичний співпроцесор і допускає можливість звернення до програм, написаних на мовах Фортран, С, С++. Вона підтримує виконання операцій з векторами, матрицями і масивами даних, реалізує сингулярне і спектральне розкладання, обчислення рангу і чисел обумовленості матриць, підтримує роботу з алгебраїчними поліномами, вирішує нелінійні рівняння і задачі оптимізації, інтегрує в квадратурі, вирішує диференціальні і різницеві рівняння, будує різні види графіків, тривимірних поверхонь і ліній рівня. У системі реалізовано зручне операційне середовище, яке дозволяє формулювати проблеми і завдання і отримувати рішення в звичній математичній формі. Це – інтерактивна СКМ, основним об'єктом якої є масив, для якого не потрібно указувати розмірність явно. Це дозволяє вирішувати багато обчислювальних задач, пов'язаних з векторно-матричними даними, істотно скорочуючи час, необхідний для програмування на скалярних мовах програмування (С, Фортран і ін.).

Важливою і сильною стороною цієї СКМ є те, що в ній можуть бути написані програми для багатократного використання. Користувач самостійно може написати спеціалізовані функції і програми (М-файли) і оформити їх у вигляді пакету прикладних програм (ППП), що є колекцією подібних файлів, або у вигляді окремої програми (Р-файли).

В результаті підготовки і проведення лабораторної роботи „Дослідження способів обробки і методів візуалізації інформації в системі комп'ютерної математики” студенти повинні придбати:

знання:

- про систему комп'ютерної математики (оболонка і пакети прикладних програм),
- про представлення даних в системі комп'ютерної математики,
- про принципи програмування в системі комп'ютерної математики,
- про можливості графіки системи комп'ютерної математики;

уміння:

- працювати в середовищі комп'ютерної математики,
- формувати файли, масиви, вектори, матриці,
- виконувати різні математичні і логічні операції з даними,
- здійснювати графічні побудови для різних видів даних в системі

комп'ютерної математики.

У даних методичних вказівках приводяться теоретичні відомості, необхідні для виконання даної лабораторної роботи, а також мета, завдання і порядок виконання роботи. Приведені також вимоги до оформлення звіту про лабораторну роботу.

При виконанні лабораторної роботи кожен студент відповідає на теоретичні питання і, потім, після отримання допуску, практично виконує роботу.

Оцінюється лабораторна робота в рамках виділених на неї в робочій навчальній програмі балів, причому 50% цих балів доводиться на оцінку готовності студента до лабораторної роботи по теоретичних питаннях і 50% – на оцінку практичного виконання роботи. При отриманні студентом позитивної оцінки за відповідь на теоретичні питання він отримує допуск до виконання лабораторної роботи, після чого практично виконує лабораторну роботу. Якщо у студента немає допуску, то і роботу він не виконує.

Після демонстрації викладачеві результатів виконання лабораторної роботи і отримання його дозволу студент оформляє звіт, після чого захищає звіт у вигляді відповідей на питання викладача про хід виконання лабораторної роботи і про її результати.

Проводиться лабораторна робота в комп'ютерному класі на персональних електронно-обчислювальних машинах зі встановленою системою комп'ютерної математики.

Вимоги правил техніки безпеки при проведенні лабораторної роботи на персональних ЕОМ

- 1) Включити апаратуру комп'ютера вимикачами на корпусах в послідовності: стабілізатор напруги, відеодисплейний термінал, процесор.
- 2) Відрегулювати яскравість свічення екрану відеодисплейного терміналу, фокусування, контрастність. Не слід встановлювати велику яскравість свічення екрану щоб уникнути стомлення очей. Її слід встановити так, щоб відношення яскравості екрану до яскравості поверхонь, що оточують його в робочій зоні, було не більш, ніж 3:1.
- 3) Під час роботи за клавіатурою сидіти прямо, не напружуватися.
- 4) Для зменшення несприятливого впливу на користувача пристрою управління маркером „миша” слід зайняти велику поверхню столу для переміщення „миші” і зручного упору ліктявого суглоба.
- 5) Після закінчення роботи вимкнути апаратуру в порядку, зворотному включенню.

- 6) Під час лабораторної роботи не дозволяються сторонні розмови, дратівливі шуми.

При проведенні лабораторної роботи
ЗАБОРОНЯЄТЬСЯ:

- 1) Користуватися кабелями і проводами з пошкодженою ізоляцією.
- 2) Залишати під напругою кабелі і дроти з неізольованими провідниками.
- 3) Застосовувати саморобні подовжувачі, що не відповідають вимогам Правил побудови електроустановок.
- 4) Використовувати пошкоджені електричні розетки.
- 5) При необхідності перемикання мережевих кабелів робити це тільки при вимкненому електричному живленні комп'ютера.
- 6) Класти будь-які предмети на апаратуру комп'ютера.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ЛАБОРАТОРНОЇ РОБОТИ

„Дослідження способів обробки і методів візуалізації інформації в системі комп'ютерної математики”

1.1 Керуючі команди і функції

Управління робочою областю змінних

Звільнення робочої області пам'яті

Синтаксис:

- 1) clear
- 2) clear <ім'я>
- 3) clear <ім'я 1> <ім'я 2>.
- 4) clear global <ім'я>
- 5) clear <ключ>.

Перша команда видаляє всі змінні з робочої області.

Друга команда видаляє змінну або функцію з ім'ям <ім'я> з робочої області, але якщо змінна глобальна, то вона видаляється тільки з робочої області, але залишається доступною для будь-якої функції, де ця змінна оголошена глобальною.

Третя команда видаляє з робочої області змінні і функції з вказаними іменами.

Четверта команда видаляє глобальну змінну з вказаним ім'ям.

П'ята команда видаляє ті або інші змінні і функції залежно від значення ключа, наприклад по ключу all видаляються з робочої області всі типи змінних функцій, МЕХ-файлів, а по ключу variables видаляються з робочої області всі змінні і так далі.

Управління командним вікном

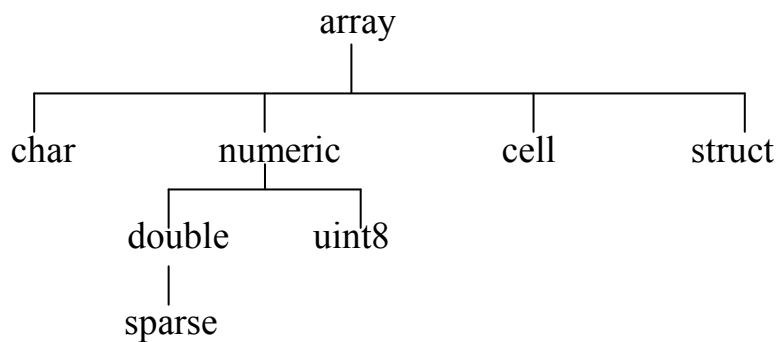
Очищення командного вікна

Синтаксис: clc.

Команда очищає командне вікно і повертає курсор в лівий верхній кут екрану.

1.2 Типи даних

У СКМ визначено 6 базових типів даних, кожен з яких є тим або іншим видом масиву (array):



де char – масив символів або строкових даних, numeric – числа, double – числа з подвоєною точністю, uint8 використовується при обробці зображень, sparse – при роботі з розрідженими матрицями, cell і struct – при роботі з масивами великої розмірності.

Докладніше:

double – числовий масив подвоєної точності (найбільш поширений тип змінної), наприклад, [12; 34] для дійсних чисел або 5+6i для комплексних чисел;

char – масив символів, кожен символ завдовжки 16 бітів, часто іменують рядком, наприклад, 'зенітний ракетний комплекс';

sparse – розріджена двовимірна матриця подвоєної точності. Розріджена структура застосовується для зберігання матриць з невеликою кількістю ненульових елементів, що дозволяє використовувати лише невелику частину пам'яті, необхідну для зберігання повної матриці. Розріджені матриці вимагають застосування спеціальних методів для вирішення завдань. Приклад: sparse(5);

cell – масив осередків. Елементи цього масиву містять інші масиви. Масиви осередків дозволяють об'єднати зв'язані дані, можливо різних розмірів, в єдину структуру. Приклад: {21 'привіт' eye(2)};

struct – масив записів. Він включає імена полів. Поля самі можуть містити масиви. Подібно до масивів осередків, масиви записів об'єднують зв'язані дані і інформацію про них. Приклад: A.day=12; A.color='Blue'; A.mat=magic(3);

uint8 – масив 8-розрядних цілих чисел без знаків. Він дозволяє зберігати цілі числа в діапазоні від 0 до 255 в 1/8 частині пам'яті, потрібної для масиву подвоєної точності. Ніякі математичні операції для цих масивів не визначені. Приклад: uint8(magic(3));

UserObject – тип даних, визначуваних користувачем, наприклад ('sin(x)').

Задати тип змінної array або numeric неможливо. Ці типи є віртуальними і служать тільки для того, щоб згрупувати змінні, які мають загальні атрибути.

Основні типи даних при обробці – double і char.

Тип `uint8` призначений для ефективного зберігання даних в пам'яті. До даних можна застосовувати тільки базові операції індексації і зміни розмірів, але не можна виконати ніякої математичної операції. Для цього такі масиви необхідно перетворити в тип `double`.

Окрім віртуальних решта типів даних називається вбудованими.

Числові масиви

Діставання доступу до підблоків одновимірних і двовимірних масивів.

- Синтаксис:
- 1) `m:k`
 - 2) `m:n:k`
 - 3) `A(m1:m2, k1:k2)`
 - 4) `A(n1:n2)`
 - 5) `A(m :)`
 - 6) `A(:, k)`
 - 7) `A(:)`

Оператор `:` застосовується для формування векторів і масивів або для виділення з них підвекторів і підблоків масиву.

Перша команда формує вектор. Якщо $m \geq k$, то це вектор вигляду $[m \ m+1 \ m+2 \dots k]$, якщо $m < k$, то це порожній вектор.

Друга команда також формує вектор. Якщо $m \geq k$, то це вектор вигляду $[m \ m+n \ m+2n \dots k]$, якщо $n < 0$ і $m < k$ або $n > 0$ і $m > k$, то це порожній вектор.

Третя і четверта команди виділяють підблоки з масиву, при цьому по третій команді виділяється підблок масиву A з рядками $m1:m2$ і стовпцями $k1:k2$, а по четвертій команді виділяються пронумеровані елементи від номера $n1$ до номера $n2$, оскільки в СКМ елементи масиву впорядковані по стовпцях.

П'ята і шоста команди виділяють, відповідно m -тий рядок масиву A і k -тий стовпець масиву A .

Сьома команда запише всі елементи масиву A у вигляді стовпця.

Об'єднання масивів

- Синтаксис:
- 1) `[A, B...]` або `[A B ...]`
 - 2) `[A; B; ...]`
 - 3) `[A B; C]`
 - 4) `[A B; [C D]]`

Перший оператор виконує горизонтальне об'єднання кінцевої кількості масивів, у яких кількість рядків має бути однаковою.

Другий оператор виконує вертикальне об'єднання кінцевої кількості масивів, у яких має бути однакова кількість стовпців.

Третій оператор є допустимим, якщо кількість рядків A і B однакова, а

кількість стовпців масиву D збігається з сумою стовпців масивів A і B.

Четвертий оператор допускається, якщо сума стовпців масивів C і D збігається з сумою стовпців масивів A і B.

Перетворення розмірів двовимірного масиву

Синтаксис: $B = \text{reshape}(A, m, n)$.

Ця формула повертає двовимірний масив розміру $m \times n$, сформований з елементів масиву A шляхом їх послідовної вибірки по стовпцях, при цьому кількість елементів масиву A повинна дорівнювати $m \times n$.

Визначення довжини вектора

Синтаксис: $\text{length}(x)$

Ця функція повертає кількість елементів вектора x.

Визначення розмірів масиву

Синтаксис: 1) $d = \text{size}(X)$

2) $[m, n] = \text{size}(X)$

3) $m = \text{size}(X, 1)$

4) $n = \text{size}(X, 2)$

Перша функція для масиву X розміром $m \times n$ повертає вектор-рядок $d = [m, n]$, який указує число рядків і стовпців.

Друга функція повертає число рядків і стовпців у вигляді двох змінних.

Третя функція повертає тільки число рядків.

Четверта функція повертає тільки число стовпців.

Формування вузлів двовимірної і тривимірної сіток

Синтаксис: 1) $[X, Y] = \text{meshgrid}(x, y)$

2) $[X, Y] = \text{meshgrid}(x)$

3) $[X, Y, Z] = \text{meshgrid}(x, y, z)$

Перша функція формує масиви X і Y, які визначають координати вузлів прямокутника, що задається векторами x і y. Цей прямокутник задає область визначення функції від двох змінних, яку можна побудувати у вигляді тривимірної поверхні.

Друга функція є скороченою формою запису функції $[X, Y] = \text{meshgrid}(x, x)$.

Третя функція формує масиви X, Y, Z, які визначають координати вузлів паралелепіпеда, що задається векторами x, y, z. Цей паралелепіпед задає область визначення для обчислення функції від трьох змінних і побудови тривимірних параметричних поверхонь.

Формування масивів нулів і одиниць

- Синтаксис: 1) $Y=zeros(n)$
2) $Y=zeros(m, n)$
3) $Y=zeros(size(A))$

`zeros` – формуються нулі, `ones` – формуються одиниці.

Перша функція формує масив розміром $n \times n$.

Друга функція формує масив розміром $m \times n$.

Третя функція формує масив, що відповідає розміру масиву A .

Формування елементів, розподілених за рівномірним законом

- Синтаксис: 1) $X=rand(n)$
2) $X=rand(m, n)$
3) $X=rand(size(A))$
4) `rand`

Перша функція формує масив розміром $n \times n$, елементами якого є випадкові величини, розподілені за рівномірним законом в інтервалі $(0, 1)$.

Друга функція формує такий же масив як і перша, але розміром $m \times n$.

Третя функція формує такий же масив, як і перша, що відповідає розміру масиву A .

Четверта функція формує одне випадкове число, що підкоряється рівномірному закону розподілу в інтервалі $(0, 1)$, який змінюється при кожному подальшому виклику функції.

Формування масиву елементів, розподілених за нормальним законом

- Синтаксис: 1) $X=randn(n)$
2) $X=randn(m, n)$
3) $X=randn(size(A))$
4) `randn`

Перша функція формує масив розміром $n \times n$, елементами якого є випадкові величини, розподілені за нормальним законом з математичним очікуванням 0 і середньоквадратичним відхиленням 1.

Друга функція формує такий же масив як і перша, але розміром $m \times n$.

Третя функція формує такий же масив як і перша, але масив відповідний розміру масиву A .

Четверта функція формує одне випадкове число, розподілене за нормальним законом з математичним очікуванням 0 і середньоквадратичним відхиленням 1, яке змінюється при кожному подальшому виклику функції.

Багатовимірні масиви

Формування багатовимірного масиву

Синтаксис: `A(:, :, k)` : (двокрапка)

`A(:, :, k)` – k-а сторінка тривимірного масиву `A`.

`A(:, :, :, l)` – l-й тривимірний підмасив чотиривимірного масиву `A`.

`A(i, j, k, :)` – одновимірний підмасив чотиривимірного масиву `A`.

`A(n1:n2)` – послідовність елементів від `n1` до `n2`.

`A(:)` – всі елементи масиву `A`, згорнуті в стовпець.

Для формування багатовимірних масивів ці конструкції повинні використовуватися в лівій частині, а для виділення підмасивів – в правій частині.

Визначення розмірів масиву

Синтаксис: 1) `d=size(X)`

2) `m=size(X, dim)`

3) `[d1, d2, ..., dk]=size(X)`.

Перша функція для багатовимірного масиву `X` повертає вектор-рядок довжиною `ndims(X)`, кожен елемент якої вказує розмір масиву по відповідній розмірності.

Друга функція повертає розмір масиву `X` по розмірності `dim`.

Третя функція повертає розміри масиву `X` по розмірності у вигляді окремих змінних.

Визначення максимального розміру розмірності

Синтаксис: `n=length(X)`

Ця функція еквівалентна `max(size(X))` для непорожніх масивів і дорівнює нулю для порожніх масивів. Якщо `X` – вектор, то результатом є довжина вектора.

Перетворення розмірів багатовимірного масиву

Синтаксис: `B=reshape(A, m, n, p, ...)`.

Ця функція повертає багатовимірний масив розміру `m×n×p×...`, сформований з елементів масиву `A`. Добуток `m×n×p×...` має дорівнювати добутку (`size(A)`).

Функція `reshape (A [m n p ...])` рівносильна `reshape (A, m, n, p, ...)`.

Об'єднання масивів

Синтаксис: 1) `C=cat(dim, A, B)`

2) `C=cat(dim, A1, A2, A3, ...)`.

Перша функція об'єднує масиви `A` і `B` уздовж розмірності `dim`.

Друга функція об'єднує безліч початкових масивів уздовж розмірності

dim.

Якщо dim=1, то об'єднання йде уздовж рядків, якщо dim=2 – то об'єднання здійснюється уздовж стовпців.

Формування масиву нулів або одиниць

- Синтаксис: 1) $Y = \text{zeros}(d1, d2, d3, \dots)$
2) $Y = \text{zeros}([d1\ d2\ d3\ \dots])$
3) $Y = \text{zeros}(\text{size}(A))$.

zeros – формуються нулі, ones – формуються одиниці.

Перша і друга функції повертають масив нулів розмірності $d1 \times d2 \times d3 \times \dots$

Третя функція формує масив нулів, відповідний розміром масиву A.

Формування масиву елементів, розподілених по рівномірному закону.

- Синтаксис: 1) $Y = \text{rand}(d1, d2, d3, \dots)$
2) $Y = \text{rand}([d1\ d2\ d3\ \dots])$
3) $Y = \text{rand}(\text{size}(A))$.

Перша і друга функції повертають масив розміру $d1 \times d2 \times d3 \times \dots$, елементами якого є випадкові величини, розподілені по рівномірному закону в інтервалі (0, 1).

Третя функція робить те ж, але масив Y при цьому відповідний розміром масиву A.

1.3 Об'єктно-орієнтоване програмування

Перевизначення класів

Таблиця, приведена нижче, встановлює символні імена для вбудованих операторів СКМ.

Оператор	Ім'я М-файлу	Опис
a+b	plus (a, b)	двійкове складання
a-b	minus (a, b)	двійкове віднімання
-a	uminus (a)	унарне віднімання
+a	uplus (a)	унарне складання
a.*b	times (a, b)	поелементне множення
a*b	mtimes (a, b)	множення матриць
a./b	rdivide (s, b)	праве поелементне ділення
a.\b	ldivide (a, b)	ліве поелементне ділення
a/b	mrdivide (a, b)	праве ділення матриць
a\b	mldivide (a, b)	ліве ділення матриць
a.^b	power (a, b)	поелементне піднесення до ступеня

Оператор	Ім'я М-файлу	Опис
a^b	<code>mpower (a, b)</code>	піднесення матриці до ступеня
$a < b$	<code>lt (a, b)</code>	менше
$a > b$	<code>gt (a, b)</code>	більше
$a \leq b$	<code>le (a, b)</code>	менше або рівно
$a \geq b$	<code>ge (a, b)</code>	більше або рівно
$a \sim b$	<code>ne (a, b)</code>	не рівно
$a = b$	<code>eq (a, b)</code>	тотожно
$a \& b$	<code>and (a, b)</code>	логічне І
$a b$	<code>or (a, b)</code>	логічне АБО
$\sim a$	<code>not (a, b)</code>	логічне НІ
$a:d:b$	<code>colon (a, d, b)</code>	формування вектора
$a:b$	<code>colon (a, b)</code>	
a'	<code>ctranspose (a)</code>	транспонування матриці
$a.'$	<code>transpose (a)</code>	транспонування масиву
<code>command window output</code>	<code>display (a)</code>	вивід на термінал
$[a \ b]$	<code>horzcat(a, b, ...)</code>	об'єднання в рядок
$[a; b]$	<code>vertcat (a, b, ...)</code>	об'єднання в стовпець
$a(s_1, s_2 \dots, s_n)$	<code>subsref (a, s)</code>	індексне посилання
$a(s_1 \dots, s_n)=b$	<code>subsasgn (a, s, b)</code>	індексний вираз
$b(a)$	<code>subsindex (a, b)</code>	індекс підмасиву

1.4 Оператори, константи, службові символи, змінні

Арифметичні і логічні символи

Арифметичні оператори. Операції над масивами, матрицями і об'єктами

У СКМ рекомендується 2 типи арифметичних операцій. Операції над масивами виконуються поелементно, а операції над матрицями визначені відповідно до правил лінійної алгебри. Для розрізнення цих операцій, операції над масивами передуює крапка.

Синтаксис:

Операції над масивами		Операції над матрицями	
$+A$	<code>uplus (A)</code>	$+A$	<code>uplus (A)</code>
$- A$	<code>uminus (A)</code>	$- A$	<code>uminus (A)</code>
$A+B$	<code>plus (A, B)</code>	$A+B$	<code>plus (A, B)</code>
$A - B$	<code>minus (A, B)</code>	$A-B$	<code>minus (A, B)</code>
$A.*B$	<code>times (A, B)</code>	$A*B$	<code>mtimes (A, B)</code>
$A./B$	<code>rdivide (A, B)</code>	A/B	<code>mrdivide (A, B)</code>

$A \setminus B$	ldivide (A, B)	$A \setminus B$	mldivide (A, B)
$A.^B$	power (A, B)	$A.^B$	mpower (A, B)
$A.'$	transpose (A)	A'	ctranspose (A)

Опис:

	Унарний плюс: $+A$ uplus (A)	перетворення масиву в саме себе; для об'єкту – створення об'єкту $+A$
	Унарний мінус $-A$ uminus (A)	зміна знаку елементів масиву; для об'єкту – створення об'єкту $-A$
+	Складання: $A+B$ plus (A, B)	доданки мають бути однакового розміру за винятком випадку, коли одне з них скаляр; скаляр додається до всіх елементів іншого операнда; для об'єктів – створення об'єкту $A+B$
-	Віднімання: $A-B$ minus (A, B)	зменшуване і таке, що віднімається мають бути однакового розміру за винятком випадку, коли одне з них скаляр; скаляр віднімається зі всіх елементів іншого операнда; для об'єктів – створення об'єкту $A - B$
*	Множення матриць $A*B$ times (A, B)	при множенні матриць або матриць і векторів число стовпців першого співмножника має дорівнювати числу рядків другого; на скаляр множаться всі елементи співмножника
.*	Множення масивів: $A.*B$ mtimes (A, B)	поелементне перемножування двох масивів однакового розміру; на скаляр множаться всі елементи масиву
\	Вирішення систем лінійних рівнянь: $AX=B$ mldivide (A, B)	якщо A – квадратна матриця розміром $n \times n$ і B – матриця розміром $n \times k$, рівняння вирішується методом Гауса; якщо A – прямокутна матриця розміром $m \times n$ і B – матриця розміром $n \times k$, то система виявляється недовизначеною або перевизначеною і вирішується на основі мінімізації другої норми нев'язності. Ранг $r = \text{rank}(A)$ обчислюється з використанням QR розкладання з вибором провідного елементу. Таке рішення матиме найбільше r ненульових елементів на стовпець. Якщо $r < n$, то знайдене рішення, як правило, не збігатиметься з $\text{pinv}(A)*B$, яке має найменшу норму нев'язності

.\	Ліве ділення масивів: A.\B ldivide (A, B)	результатом є масив з елементами $B(i, j) / A(i, j)$; масиви мають бути однакових розмірів за винятком випадку, коли один з них вироджується в скаляр
/	Вирішення систем лінійних рівнянь: XA=B mrdivide (A, B)	операція B/A рівносильна операції $B * \text{inv}(A)$, якщо A має повний ранг; а якщо немає – то $B/A = (A \setminus B)'$
./	Праве ділення масивів: A./B rdivide (A, B)	результатом є масив з елементами $A(i, j) / B(i, j)$; масиви мають бути однакових розмірів за винятком випадку, коли один з них вироджується в скаляр
^	Ступінь матриці: A^p mpower (A, p)	якщо p – ціле позитивне число, то ступінь матриці обчислюється шляхом перемножування її на себе; якщо p – ціле негативне число, то те ж саме відноситься до зворотної матриці. Для інших значень p обчислюється власне значення і вектори, так що якщо $[R, D] = \text{eig}(A)$, то $A^p = R * D.^p / R$
.^	Ступінь масиву: A.^B power (A, B)	результатом є масив з елементами $A(i, j)^{B(i, j)}$; масиви мають бути однакових розмірів за винятком випадку, коли один з них вироджується в скаляр
'	Транспонування матриці: A ctranspose (A)	для дійсних – результатом є транспонована матриця; для комплексних – транспонування доповнюється комплексним сполученням
.'	Транспонування масиву: A.' transpose (A)	для дійсних і комплексних масивів рядки замінюються стовпцями; операція комплексного сполучення не виконується

Логічні оператори. Оператори відношення

- Синтаксис:
- 1) $A < B$ lt(A, B)
 - 2) $A \leq B$ le(A, B)
 - 3) $A > B$ gt(A, B)
 - 4) $A \geq B$ ge(A, B)
 - 5) $A = B$ eq(A, B)
 - 6) $A \sim B$ ne(A, B)
 - 7) strcmp

Операції відношення виконують поелементне порівняння двох масивів; вони повертають як результат масив того ж розміру і елементи якого дорівнюють одиниці, якщо ці елементи збігаються, і нулю, якщо

вони не збігаються.

Операції 1–4 використовують для порівняння тільки дійсної частини комплексних елементів, а операції 5–6 здійснюють порівняння як дійсних, так і уявних частин. Операція 7 використовується для порівняння рядків.

Функціональна форма операцій відношення застосовується в рамках об'єктно-орієнтованого програмування.

Логічні операції

- Синтаксис:
- 1) $A \& B$ $\text{and}(A, B)$
 - 2) $A | B$ $\text{or}(A, B)$
 - 3) $\sim A$ $\text{not}(A, B)$
 - 4) $\text{xor}(A, B)$

Опис:

Логічна операція	Символ операції	Оператор	Функція
логічне НІ (заперечення)	$\sim A$	$\sim A$	$\text{not}(A)$
логічне І (кон'юнкція)	$A \otimes B$	$A \& B$	$\text{and}(A, B)$
АБО, що не виключає (диз'юнкція)	$A + B$	$A B$	$\text{or}(A, B)$
АБО, що виключає (складання за модулем 2)	$A \oplus B$	-	$\text{xor}(A, B)$

Таблиця істинності для цих операторів має вигляд

A	B	$\text{not}(A)$	$\text{and}(A, B)$	$\text{or}(A, B)$	$\text{xor}(A, B)$
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

При обробці масивів ці операції застосовуються поелементно. У разі числового масиву значенню 0 відповідає булеве значення FALSE, а будь-якому іншому значенню – булеве значення TRUE.

Всі логічні оператори можуть бути використані при обробці об'єктів, якщо застосовується функціональна форма оператора.

Логічні операції мають нижчий пріоритет по відношенню до операції відношення і арифметичних операцій. Але серед логічних операцій найвищий пріоритет має операція NOT; операції AND і OR мають однаковий пріоритет і виконуються зліва направо.

Спеціальні символи, змінні і константи

Спеціальні символи

Синтаксис: $[] () = ' , ; : \% !$

Опис:

[]	Квадратні дужки	Формування векторів і матриць
()	Круглі дужки	Вказівка послідовності вихідних параметрів при зверненні до функцій, що повертають більш за один параметр Вказівка індексів елементу вектора, матриці, масиву Вказівка послідовності вхідних параметрів функції
=	Знак привласнення	Знак привласнення в арифметичних виразах
'	Транспонування матриць	Транспонування, що супроводжується комплексним сполученням для комплексних матриць
.	Роздільник – крапка	Десяткова крапка Поелементне виконання операцій над масивом даних
..	« »	Перехід по дереву каталогів на один рівень вгору
... і більш	« »	Ознака продовження рядка
, [,]	Роздільник – кома	Для відділення операторів Для вказівки індексів елементу вектора, матриці, масиву
; [:]	Роздільник – крапка з комою	Для придушення виводу на екран результату обчислень Для відділення рядків матриць або масивів
:[:]	Роздільник–двокрапка	Для формування векторів, виділення рядків, стовпців, підблоків масиву У заголовку циклу for
%	Показчик	Показчик логічного кінця рядка, наступний за ним текст ігнорується Показчик рядка коментаря
!	Показчик	Показчик введення команди DOS

Уявна одиниця

Синтаксис: I, i, J, j.

Цим константам спочатку привласнюється значення $\sqrt{-1}$. Використовуються для введення комплексних чисел. Одне і те ж число може виглядати по-різному:

$3+2i$ $3+2*i$ $3+2j$ $3+2*j$ $3+2*\sqrt{-1}$

Нескінченність

Синтаксис: `inf`

Це спеціальна змінна, що фіксує переповнювання розрядної сітки ЕОМ.

Нечислове значення

Синтаксис: `NaN`

Це спеціальна змінна, прийнята в стандарті IEEE арифметики, яка дозволяє фіксувати результат, що не представляється у вигляді числа. Наступні операції мають результат у вигляді змінної `NaN`:

- множення виду $0 * \text{inf}$;
- ділення вигляду $0/0$ і inf/inf ;
- складання і віднімання нескінченності;
- обчислення функції залишку $\text{rem}(x, y)$ для $y=0$ або $x=\text{inf}$;
- обчислення відношень з предикатами `<або>`, наприклад, функцій виду $\text{min}([\text{inf NaN}])$ і $\text{max}([\text{inf NaN}])$;
- особливі арифметичні операції із змінною `NaN`.

Функція обчислення часу

Запуск і зупинка таймера

Синтаксис: `tic`

оператор

`toc`

В цьому випадку виводяться відлік таймера від моменту запуску `tic` до моменту зупинки `toc`.

1.5 Програмування

Управління виконанням програм

Оператор циклу з певним числом операцій

Синтаксис: 1) `for <змінна циклу> = <початкове значення> :
<прирощення> : <кінцеве значення>
інструкції
end`
2) `for <змінна циклу>=A
інструкції
end.`

Перший оператор виконує інструкції зумовлене число разів.

Другий оператор визначає змінну циклу як вектор $A(:, k)$. Для першого кроку циклу k дорівнює 1, для другого k дорівнює 2 і так далі,

поки k не досягне значення n , тобто цикл виконується стільки разів, скільки стовпців в матриці A . Для кожного кроку змінна циклу – це вектор, що містить один із стовпців масиву A .

Допускаються вкладені цикли і цикли з векторною змінною.

Оператор циклу з невизначеним числом операцій

Синтаксис: `while <логічний вираз>`
`<оператори>`
`end`

В цьому випадку багато разів виконуються `<оператори>`, поки `<логічний вираз>` істинний.

Оператор умови

Синтаксис: 1) `if <логічний вираз>`
`<інструкції>`
`end`
2) `if <логічний вираз>`
`<інструкції>`
`else`
`<інструкції>`
`end`
3) `if <логічний вираз>`
`<інструкції>`
`elseif`
`<логічний вираз>`
`<інструкції>`
`else`
`<інструкції>`
`end`

1.6 Математичні функції

Абсолютне значення

Синтаксис: $Y = \text{abs}(X)$

Для дійсних чисел в Y повертаються абсолютні значення елементів X .

Для комплексних чисел в Y повертаються модулі комплексних елементів X .

Для строкових змінних в Y повертаються замість символів їх ASCII-коди.

Аргумент комплексного числа

Синтаксис: $P = \text{angle}(Z)$

Для комплексних чисел в P повертаються значення аргументів Z в радіанах від $-\pi$ до π .

Дійсна і уявна частина комплексного числа

Синтаксис: 1) $X=\text{real}(Z)$
2) $Y=\text{imag}(Z)$

Для комплексних чисел в X повертаються дійсні частини Z , а в Y – уявні частини Z .

Операція комплексного сполучення

Синтаксис: $V=\text{conj}(Z)$

Для комплексних чисел в V повертаються комплексно-зв'язані значення для Z .

Функції округлення

Синтаксис: 1) $Y=\text{ceil}(X)$
2) $Y=\text{fix}(X)$
3) $Y=\text{floor}(X)$
4) $Y=\text{round}(X)$

Для дійсних чисел:

перша функція в Y повертає значення, закруглені до найближчого цілого $\geq X$,

друга функція в Y повертає значення з усіканням дробової частини числа,

третя функція в Y повертає значення, закруглені до найближчого цілого $\leq X$,

четверта функція в Y повертає значення, закруглені до найближчого цілого.

Для комплексних чисел ці функції застосовуються одночасно до дійсної і уявної частини.

Квадратний корінь

Синтаксис: $V=\text{sqrt}(Z)$

Функція обчислює квадратне коріння елементів масиву Z . Для негативних і позитивних значень результат є комплексним числом.

Експоненціальна функція

Синтаксис: $V=\text{exp}(Z)$

Функція обчислює експоненти значень елементів масиву Z . Для комплексних значень $Z=x+iy$ справедлива формула Ейлера $e^z = e^x(\cos(y) + j \sin(y))$.

Функції логарифмів

Синтаксис: $V=\log(Z)$

Обчислюється натуральний логарифм значень елементів масиву Z .

Для комплексних значень $z=x+jy$ справедлива формула $\ln(z) = \ln(abs(z)) + j \arctan 2(y, x)$.

Логарифм по основі 2 обчислюється по функції $\log 2$, по основі 10 – по функції $\log 10$.

Функція синуса, косинуса, тангенса

Синтаксис: $V=\sin(Z)$

Функція обчислює синус від значень елементів масиву Z , при цьому для Z допускаються комплексні значення, кути вимірюються в радіанах.

Для обчислення косинуса використовується функція \cos , тангенса – \tan , котангенса – \cot .

1.7 Аналіз і обробка даних

Підсумовування (добуток) елементів масиву

Синтаксис: 1) $sx=\text{sum}(X)$;
2) $sx=\text{sum}(X, \text{dim})$;
3) $csx=\text{cumsum}(X)$;
4) $csx=\text{cumsum}(X, \text{dim})$.

Перша функція при одновимірному масиві повертає суму елементів масиву, при двовимірному масиві – вектор-рядок, що містить суми елементів кожного стовпця, при багатовимірному масиві – суми елементів в стовпцях уздовж першої неодиначної розмірності.

Друга функція повертає вектор-рядок, що містить суми елементів по стовпцях уздовж розмірності dim .

Третя і четверта функції, крім того, повертають всі проміжні результати підсумовування.

При добутку функція sum замінюється на prod , а функція cumsum – на cumprod .

Сортування елементів масиву

Синтаксис: 1) $Y=\text{sort}(X)$;
2) $Y=\text{sort}(X, \text{dim})$;
3) $[Y, I]=\text{sort}(X)$;
4) $Y=\text{sortrows}(X)$;
5) $Y=\text{sortrows}(X, \text{column})$;
6) $[Y, I]=\text{sortrows}(X)$.

Перша функція при одновимірному масиві упорядковує елементи за збільшенням, при двовимірному масиві відбувається впорядкування

елементів кожного стовпця, при багатовимірному масиві сортування виконується уздовж першої неединичної розмірності. Якщо X – масив строкових осередків, то проводиться лексикографічне впорядкування елементів.

Друга функція виконує сортування уздовж вказаної розмірності.

Третя функція окрім масиву впорядкованих елементів повертає масив індексів, що дозволяють відновити структуру початкового масиву.

Якщо аналізований масив містить комплексні елементи, то спочатку елементи сортуються по модулю, а у разі рівності модулів – по значеннях фазового кута в діапазоні $[-\pi, \pi]$. Якщо масив містить елементи NaN, вони розміщуються в кінці сортованого масиву.

Четверта функція сортує рядки матриці в порядку зростання елементів першого стовпця. X може бути матрицею і вектором-стовпцем. Якщо X матриця строкових даних, то вони сортуються лексикографічно.

П'ята функція виконує сортування елементів вказаного стовпця.

Шоста функція аналогічна третій.

Визначення максимальних (мінімальних) елементів масиву

- Синтаксис:
- 1) $C = \max(A)$;
 - 2) $C = \max(A, B)$;
 - 3) $C = \max(A, [], \text{dim})$;
 - 4) $[C, I] = \max(\dots)$.

Перша функція у разі одновимірного масиву повертає найбільший елемент, у разі двовимірного масиву повертає вектор-рядок, що містить максимальні елементи кожного стовпця, у разі багатовимірного масиву – це вектор максимальних елементів уздовж першої неединичної розмірності.

Друга функція повертає масив C тих же розмірів, які мають масиви A і B , кожен елемент якого є максимальний з відповідних елементів цих масивів.

Третя функція повертає вектор максимальних елементів уздовж розмірності dim .

Четверта функція окрім самих максимальних елементів повертає вектор-рядок індексів цих елементів в даному стовпці.

Якщо масив містить комплексні елементи, то максимальні елементи визначаються з умови $\max(\text{abs}(A))$, а якщо містить елементи типу NaN, то вони до уваги не беруться.

При знаходженні мінімальних елементів масиву функція \max замінюється на \min .

Визначення середніх значень елементів масиву

- Синтаксис:
- 1) $m_x = \text{mean}(X)$;

2) $m_x = \text{mean}(X, \text{dim})$.

Перша функція при одновимірному масиві повертає середнє арифметичне елементів масиву, при двовимірному масиві – це вектор-рядок, що містить середні арифметичні елементів кожного стовпця, при багатовимірному масиві – це вектор середніх арифметичних значень уздовж першої неодиначної розмірності.

Друга функція повертає вектор середніх арифметичних значень уздовж розмірності dim .

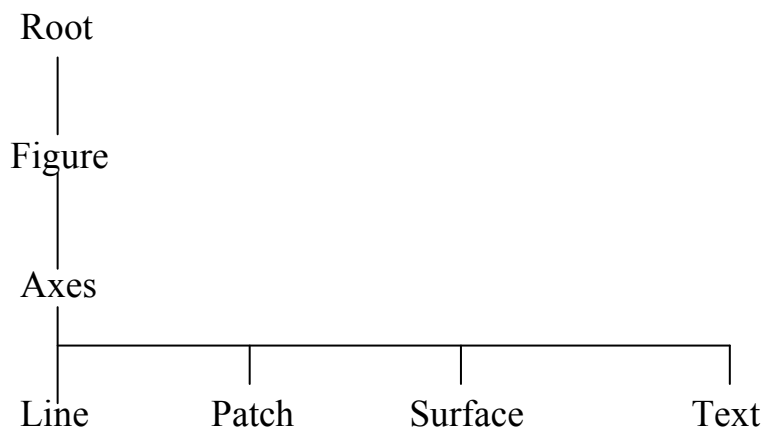
1.8 Графічні команди і функції

Дескрипторна графіка

Дескрипторна графіка – об'єктно-орієнтована графічна система, що підтримує компоненти, необхідні для створення комп'ютерних графіків.

Об'єктами в цьому випадку є базові графічні елементи, які використовуються для візуалізації даних. Кожному об'єкту привласнюється ідентифікатор, званий дескриптором, використання якого дозволяє управляти властивостями графічного об'єкту.

Ієрархія графічних об'єктів пов'язана з візуалізацією даних і представлена нижче.



Графічний об'єкт Root відповідає екрану комп'ютера. Існує єдиний такий об'єкт, який не має батьків, а всі інші графічні об'єкти є його нащадками. Він створюється у момент запуску системи і не може бути видалений.

Графічні об'єкти Figure відповідають окремим графічним вікнам на екрані терміналу, де відображаються дані, що візуалізуються. Всі високорівневі команди і функції, що виводять графіки, автоматично створюють об'єкт Figure, якщо він раніше не був створений.

Графічний об'єкт Axes задає область розміщення графіка у вікні

об'єкту Figure. Всі команди і функції, що виводять графіки, автоматично створюють об'єкт Axes, якщо він не існує.

Графічний об'єкт Line є графічним примітивом, використовуваним для створення двовимірних і тривимірних графіків. Графічні команди і функції створюють цей об'єкт.

Графічний об'єкт Patch – це один або декілька закрашених багатокутників з виділеними межами. Деякі високорівневі графічні команди і функції створюють графічні об'єкти Patch.

Графічний об'єкт Surface – це тривимірна візуалізація масиву даних, коли елемент масиву визначає висоту крапки над площиною x-y. Так формується тривимірна поверхня, що складається з чотирикутників, вершини яких визначаються масивом початкових даних. Поверхня може бути суцільною або інтерпольованою кольором або бути тільки сіткою ліній, що сполучають вершини. Високорівневі графічні команди і функції створюють графічний об'єкт Surface.

Графічні об'єкти Text є рядками символів і створюються високорівневими командами і функціями.

Елементарна графіка

- Синтаксис:
- 1) plot(y);
 - 2) plot(x,y);
 - 3) plot(x, y, lineSpec);
 - 4) plot(x1, y1, LineSpec1, x2, y2, LineSpec2, ...);
 - 5) plot(..., 'PropertyName', PropertyValue, ...);
 - 6) h=plot(...).

Перша команда буде графік елементів одновимірного масиву у залежно від номера елемента; якщо елементи масиву у комплексні, то будується графік plot (real (y), imag(y)). Якщо у – двовимірний дійсний масив, то будуються графіки для стовпців; у разі комплексних елементів їх уявні частини ігноруються.

Друга команда відповідає побудові графіка функції, коли одновимірний масив x відповідає значенням аргументу, а одновимірний масив у – значенням функції. Коли один з масивів або обидва є двовимірними, то проводяться наступні побудови:

- якщо x – одновимірний, а у – двовимірний, то будуються графіки для стовпців масиву у залежно від елементів вектора x;
- якщо x – двовимірний, а у – одновимірний, то будуються графіки для стовпців масиву x залежно від елементів вектора у;
- якщо обидва масиви двовимірні, то будуються залежності стовпців масиву у від стовпців масиву x.

Третя команда дозволяє виділити графік функції, вказавши спосіб відображення лінії, спосіб відображення маркера крапок, колір ліній і

маркера за допомогою строкової змінної LineSpec, яка може включати до трьох символів з наступного переліку:

Тип лінії	Тип маркера	Колір
безперервна -	точка ·	жовтий y
штрихова --	плюс +	фіолетовий m
подвійний пунктир :	зірочка *	блакитний c
штрихпунктирна -.	кружок o	червоний r
	хрест ×	зелений g
	квадрат s	синій b
	ромб d	білий w
	п'ятигранник p	чорний k
	шестигранник h	
	стрілка вниз v	
	стрілка вгору ^	
	стрілка вліво <	
	стрілка управо >	

Якщо колір лінії не вказаний, він вибирається за умовчанням з жовтого до синього, повторюючись циклічно.

Четверта команда будує на одному графіку декілька графічних об'єктів Line для функцій $y_1(x_1)$, $y_2(x_2)$..., визначивши для кожної з них свій спосіб відображення.

П'ята команда дозволяє задати значення властивостей графічного об'єкта Line, відповідного побудованому графіку.

Шоста функція повертає вектор дескрипторів для всіх графічних об'єктів Line поточного об'єкта Axes.

Тривимірна графіка

У подібній графіці значення елементів числового масиву розглядаються як Z-координати точок над площиною, визначуваною координатами x і y. Можливі декілька способів з'єднання цих точок.

Перший – побудова однієї лінії в тривимірному просторі або побудова ліній в перетинах (функція plot3).

Другий – побудова поверхонь (функції mesh і surf). Для функції mesh – це сітчаста поверхня, осередки якої мають колір фону, а їх межі можуть міняти колір, визначуваний властивостями графічного об'єкта Surface. Для функції surf – це суцільна поверхня, у якої може бути заданий колір не тільки межі, але і самого осередку, також визначуваний властивостями графічного об'єкта Surface.

При побудові тривимірних графіків необхідно дотримуватися певної послідовності:

Крок	Дія	Функції (приклад)
1	Підготувати початкові дані (сітку x, y, функцію Z)	<code>[X, Y]=meshgrid([-2:0.1:2]) Z=X.*exp(-X.^2-Y.^2)</code>
2	Виділити графічне вікно і вказати положення графіка усередині вікна	<code>figure(1) subplot 212</code>
3	Викликати функцію побудови тривимірного графіка	<code>h=surf(z)</code>
4	Встановити палітру і спосіб зафарбовування і затінювання поверхні	<code>colormap autumn shading interp set (h, 'EdgeColor', 'k')</code>
5	Встановити точку перегляду	<code>view ([30, 25]) set(gca, 'CameraViewAngleMode', 'Manual')</code>
6	Встановити межі і розмітку осей	<code>axis ([5 15 5 15 -8 8]) set (gca, 'Zticklabel', 'Negative Positive')</code>
7	Встановити масштаби по осях	<code>set (gca, 'PlotBoxAspectRatio' [2 2 1])</code>
8	Зробити підписи і пояснення до графіка	<code>xlabel ('x'), ylabel ('y'), zlabel('z'). title ('Графік функції') legend (...)</code>
9	При необхідності вивести графік на друк	<code>set (gcf, 'PaperPositionMode', 'auto') print -dps2</code>

Значну частину параметрів можна використовувати за умовчанням.

Побудова ліній і точок в тривимірному просторі

- Синтаксис: 1) `plot3(X, Y, Z)`
 2) `plot3(X1, Y1, Z1, LineSpec1, X2, Y2, Z2, LineSpec2 ...)`
 3) `plot3(..., 'Property Name', Property Value, ...)`
 4) `h=plot3(...)`

Ці команди є тривимірним аналогом команди `plot(...)`.

Перша команда при X, Y, Z – двовимірних масивах однакового розміру будує точки з координатами X(i, :), Y(i, :), Z(i, :) для кожного стовпця і сполучає їх прямими лініями.

Друга команда дозволяє виділити графік функції, вказавши спосіб відображення лінії, маркера точок і їх колір аналогічно команді `plot(...)`.

Третя команда дозволяє задати значення властивостей графічного об'єкта Line, відповідного побудованому графіку.

Четверта команда повертає вектор дескрипторів для всіх графічних

об'єктів Line поточного об'єкта Axes.

Формування прямокутної сітки

- Синтаксис: 1) $[X, Y]=\text{meshgrid}(x, y)$
2) $[X, Y]=\text{meshgrid}(x)$
3) $[X, Y, Z]=\text{meshgrid}(x, y, z)$

Перша функція задає сітку на площині x-y у вигляді двовимірних масивів X, Y, які визначаються одновимірними масивами x і y. Рядки масиву X є копіями вектора x, а стовпці – копіями вектора y. Формування таких масивів спрощує обчислення функцій двох змінних, дозволяючи застосовувати операції над масивами.

Друга функція – спрощена форма запису для функції $[X, Y]=\text{meshgrid}(x, x)$.

Третя функція формує тривимірну сітку для обчислення функцій від трьох змінних.

Функція meshgrid аналогічна функції ndgrid, за винятком того, що порядок проходження перших двох аргументів у них різний, тобто $[X, Y, Z]=\text{meshgrid}(x, y, z)$, це те ж саме, що $[X, Y, Z]=\text{ndgrid}(y, x, z)$. Тому функція meshgrid краще пристосована для вирішення завдань в декартових координатах, а функція ndgrid – для багатовимірних завдань.

Застосування функції meshgrid обмежується двовимірними і тривимірними сітками.

Тривимірна сітчаста поверхня і її перетини

- Синтаксис: 1) $\text{mesh}(X, Y, Z, C)$
2) $\text{mesh}(Z, C)$
3) $\text{mesh}(Z)$
4) $h=\text{mesh}(\dots)$

Перша команда виводить на екран сітчасту поверхню для значень масиву Z, визначених на множині значень масивів X і Y. Кольори вузлів задаються масивом C. Кольори ребер визначаються властивістю EdgeColor об'єкту Surface.

Можна задати однаковий колір для всіх ребер, визначивши його у вигляді вектора $[r\ g\ b]$ інтенсивності основних кольорів. Якщо визначити специфікацію none, то ребра не промальовуватимуться. Якщо визначити специфікацію flat, то колір ребер осередку визначатиметься кольором того вузла, який був першим при обході цього осередку. Оскільки одні і ті ж ребра обходяться кілька разів, то кольори замічатимуться. Якщо визначити специфікацію interp, то буде реалізована лінійна інтерполяція кольору між вершинами ребра.

Застосування функції затінювання сітчастої поверхні shading після звернення до функції mesh змінює специфікації властивостей EdgeColor і

FaceColor.

Друга команда виконує ту ж операцію, але при цьому по осях x , y використовуються відліки $X=1 : n$, $Y=1 : m$, де $[m, n]=\text{size}(Z)$.

Третя команда виводить поверхню, як і в попередньому випадку, але як масив кольору використовується масив $C=z$, тобто колір в цьому випадку пропорційний висоті поверхні.

Четверта команда повертає дескриптор графічного об'єкту Surface.

Якщо в цій групі з чотирьох команд замінити команду mesh на команду meshc(...), то на додаток до тривимірних поверхонь будуть побудовані проекції ліній постійного рівня на площині x - y , а якщо замінити на команду meshz(...), то на додаток до тривимірних поверхонь буде побудована площина відліку на нульовому рівні, що закриває поверхні, лежачі нижче за цей рівень.

Тривимірна суцільна поверхня

- Синтаксис:
- 1) surf(X, Y, Z, C)
 - 2) surf(Z, C)
 - 3) surf(Z)
 - 4) h=surf(...)

Перша команда виводить на екран поверхню з ребрами для значень масиву Z , визначеного на множині значень масивів X і Y . Колір осередку задається масивом C . Колір ребер – чорний, визначається властивістю EdgeColor, специфікованому як $[0\ 0\ 0]$. Можна задати однаковий колір для всіх ребер, визначивши його у вигляді вектора $[r\ g\ b]$ основних кольорів. Якщо визначити специфікацію none, то ребра не промальовуватимуться.

Застосування функції затінювання суцільної поверхні shading після звернення до функції surf змінює специфікації властивостей EdgeColor і FaceColor графічного об'єкта Surface.

Друга команда виконує ту ж операцію, але при цьому по осях x , y використовуються відліки $X=1 : n$, $Y=1 : m$, де $[m, n]=\text{size}(Z)$.

Третя команда виводить поверхню, використовуючи як колір масив $C=z$, що означає завдання кольору пропорційно висоті поверхні.

Четверта команда повертає дескриптор h для графічного об'єкта Surface.

Якщо застосувати замість surf команду surfc(...), то на додаток до тривимірних поверхонь будуть побудовані проекції на площину x - y лінії постійного рівня.

Створення підвікон

Синтаксис:

- 1) subplot(m, n, p)
- 2) subplot(m n p) або subplot mnp

- 3) subplot(h)
- 4) subplot('position' [left bottom width height])
- 5) h=subplot(...)

Ці команди проводять розбиття графічного вікна на прямокутні підвікна, причому кожному підвікну відповідає свій графічний об'єкт Axes. Графік виводиться в те підвікно, яке є в даний момент поточним.

Перша і друга команди проводять розбиття графічного вікна на декілька підвікон, при цьому значення *m* указує на кількість підвікон по горизонталі, *n* – по вертикалі, а *p* – номер підвікна, куди виводитиметься черговий графік. Ці ж команди можуть використовуватися для переходу від одного підвікна до іншого.

Третя команда робить осі з дескриптором *h* поточними.

Четверта команда створює координатні осі в позиції, визначуваній вектором [left bottom width height] в нормалізованих координатах, що змінюються в діапазоні від 0.0 до 1.0, де left=xmin/xmax, bottom=ymin/ymax, width=1-left, height=1-bottom.

П'ята функція повертає дескриптор графічного об'єкта Axes.

Слід зазначити, що з перших двох команд слід використовувати першу, оскільки друга команда залишена для забезпечення сумісності з передуючими версіями СКМ.

Нанесення сітки

- Синтаксис:
- 1) grid on
 - 2) grid off
 - 3) grid

Перша команда наносить на поточні осі координатну сітку.

Друга команда видаляє координатну сітку.

Третя команда виконує роль перемикача з однієї функції на іншу.

Управління режимом збереження поточного графічного вікна

- Синтаксис:
- 1) hold on
 - 2) hold off
 - 3) hold

Перша команда включає режим збереження поточного графіка і властивостей об'єкта Axes, так що подальші команди приведуть до додавання нових графіків в графічному вікні.

Друга команда вимикає режим збереження графіка.

Третя команда реалізує перемикач з одного режиму на іншій.

Палітра кольорів

- Синтаксис:
- 1) colormap(C)
 - 2) colormap('default') або colormap(hsv)

3) C=colormap

Палітра кольорів C – матриця розміру $m \times 3$ дійсних чисел з діапазону [0.0 1.0]. Рядок k палітри сформований з трьох чисел, які указують інтенсивність червоного, зеленого і синього кольорів, тобто $C(k, :) = [r(k) \ g(k) \ b(k)]$.

Перша команда встановлює палітру згідно матриці C.

Друга команда встановлює штатну палітру, відповідну моделі відтінок – насиченість – значення (hue–saturation–value). Послідовність кольорів цієї палітри відповідає барвам веселки: червоний – жовтий – зелений – блакитний – синій – фіолетовий.

Третя функція привласнює матриці C значення поточної палітри.

У СКМ реалізуються багато палітр, наприклад:

autumn	від червоного до жовтого через оранжевий
bone	сіра палітра з високою інтенсивністю сірого
colorcube	веселка з великою кількістю сірого, червоного, зеленого і синього
cool	палітра з відтінком блакитного і фіолетового
copper	від чорного до яскравої міді (лінійна)
flag	послідовна зміна червоного, білого, синього і чорного
gray	відтінки сірого (лінійні)
hot	послідовна зміна чорного, червоного, жовтого, білого
hsv	веселка (лінійна)
jet	варіант веселки від синього до червоного через блакитний, жовтий і оранжевий
lines	відтінки сірого, визначувані властивістю Axes ColorOrder
pink	відтінки пастелі
prism	послідовна зміна червоного, оранжевого, жовтого, зеленого, синього і фіолетового
spring	відтінки фіолетового і жовтого
summer	відтінки зеленого і жовтого
white	біла палітра
winter	відтінки синього і зеленого

Зображення ліній рівня для тривимірної поверхні

- Синтаксис:
- 1) contour(Z)
 - 2) contour(x, y, Z)
 - 3) contour(Z, n)
 - 4) contour(x, y, Z, n)
 - 5) contour(Z, v)
 - 6) contour(x, y, Z, v)
 - 7) contour(..., LineSpec)

8) [C, h]=contour(...)

Перша команда малює двовимірні лінії рівня для масиву даних Z , що визначають поверхню в тривимірному просторі без урахування діапазону зміни координат x і y .

Друга команда виконує те ж, що і перша, але з урахуванням діапазону зміни координат x і y .

Третя і четверта команда малюють n ліній рівня для масиву даних Z , за умовчанням $n=10$.

П'ята і шоста команда малюють лінії рівня для заданих значень, які вказані у векторі v .

Сьома команда малює лінії рівня, тип і колір яких визначається параметром `LineStyle` команди `plot`.

Восьма команда повертає масив C і вектор-стовпець дескрипторів h графічних об'єктів для кожної лінії рівня. Якщо вказаний параметр `LineStyle`, то функція `contour` сформує графічні об'єкти `Line`, якщо немає, то графічні об'єкти `Patch`.

Якщо необхідно відобразити тривимірні лінії рівня, то команда `contour` замінюється на команду `contour3`, при цьому x і y мають бути двовимірними масивами, обчисленими за допомогою функції `meshgrid`.

Заголовки для графіків

- Синтаксис:
- 1) `title('<текст>')`
 - 2) `title(<им'я_функції>)`
 - 3) `title(..., 'PropertyName', PropertyValue, ...)`
 - 4) `h=title(...)`

Кожен графічний об'єкт `Axes` може супроводитися своїм заголовком, який розміщується над графіком.

Перша команда розміщує заголовок у вигляді вказаного тексту над графіком.

Друга команда дозволяє застосувати функцію, яка повертає результат у вигляді рядка, використовуюваного як заголовок.

Третя команда дозволяє вказати пару властивість/значення для графічного об'єкта `Text`, який створюється командою `title`.

Четверта команда повертає дескриптор графічного об'єкта `Text`.

Позначення осей

- Синтаксис:
- 1) `xlabel('<текст>')`
 - 2) `xlabel(<им'я_функції>)`
 - 3) `xlabel(..., 'PropertyName', PropertyValue, ...)`
 - 4) `h=xlabel(...)`

Графічний об'єкт `Axes` може мати тільки один підпис для кожної з координатних осей x , y і Z .

Перша команда поміщає напис уздовж осі x.

Друга команда дозволяє застосувати функцію, яка повертає результат у вигляді рядка, і використовувати її як підпис осі.

Третя команда дозволяє вказати пару властивість/значення для графічного об'єкта Text, який створюється командою xlabel.

Четверта функція повертає дескриптор графічного об'єкта Text.

Уздовж осі y команда виглядає як ylabel, а уздовж осі z команда виглядає як zlabel.

Маркірування ліній рівня

- Синтаксис:
- 1) xlabel(C, h)
 - 2) xlabel(C, h, v)
 - 3) xlabel(C, h, 'manual')
 - 4) xlabel(C)
 - 5) xlabel(C, v)
 - 6) xlabel(C, 'manual')

Команди цієї групи маркірують лінії рівня, сформовані функціями contour, contour3 і ін.

Перша команда маркірує лінії рівня, розміщуючи мітки в розривах ліній рівня і повертаючи їх для зручності читання.

Друга команда маркірує тільки лінії рівня, задані вектором v.

Третя команда маркірує лінії рівня в позиціях, що указуються переміщенням курсору. Натиснення лівої клавіші миші або клавіші пропуску маркірує вказану лінію рівня, а натиснення правої кнопки миші або клавіші Return завершує процес маркірування.

Четверта команда додає мітки до ліній рівня у випадково вибраних позиціях.

П'ята команда маркірує лінії рівня контуру, які задані у векторі v.

Шоста команда маркірує лінії рівня в позиціях, що указуються за допомогою миші.

Пояснення до графіка

- Синтаксис:
- 1) legend('<текст1>' '<текст2>' .)
 - 2) legend(M)
 - 3) legend(h, M)
 - 4) legend('off') або legend off
 - 5) legend(h, ...)
 - 6) legend(..., pos)
 - 7) h=legend(...)

Ця група команд дозволяє супроводжувати графік поясненням або легендою, в якій відображається тип кожної лінії або маркера з текстом пояснення.

Перша команда додає до поточного графіка пояснення у вигляді вказаних текстових рядків.

Друга команда формує пояснення з рядків масиву M , при цьому рядки масиву M повинні мати однакову довжину.

Третя команда асоціює кожен рядок масиву M з відповідним графічним об'єктом, визначуваним вектором дескрипторів h .

Четверта команда видаляє пояснення з поточного графіка.

П'ята команда виводить пояснення до графічного об'єкта $Axes$ з дескриптором h .

Шоста команда використовує наступні значення параметра pos , щоб задати місцеположення легенди: -1 – поза графіком справа; 0 – в одному з чотирьох кутів області графіка так, щоб закрити найменшу кількість точок графіка; 1 – в правому верхньому кутку графіка (за умовчанням); 2 – в лівому верхньому кутку графіка; 3 – в лівому нижньому кутку графіка; 4 – в правому нижньому кутку графіка; $[x, y]$ - координати лівого нижнього кута області легенди.

Сьома функція повертає дескриптор h графічного об'єкта $Axes$.

Для переміщення області легенди слід натиснути ліву кнопку миші, знаходячись в цій області, а потім перемістити пояснення в потрібну позицію.

Шкала палітри

- Синтаксис:
- 1) `colorbar('vert')`
 - 2) `colorbar('horiz')`
 - 3) `colorbar(h)`
 - 4) `colorbar`
 - 5) `h=colorbar(...)`

Перша команда додає до поточного графіка вертикальну шкалу палітри.

Друга команда додає до поточного графіка горизонтальну шкалу палітри.

Третя команда додає до графіка з дескриптором h шкалу палітри. Розміщення шкали здійснюється автоматично залежно від співвідношення ширини і висоти графіка.

Четверта команда або використовує вже існуючу вертикальну шкалу палітри, або розміщує на поточному графіку нову вертикальну шкалу палітри.

П'ята функція повертає дескриптор h графічного об'єкта $Axes$.

Побудова гістограми

- Синтаксис:
- 1) `hist(y)`
 - 2) `hist(y, n)`

3) `hist(y, x)`

4) `[p, x]=hist(y, ...)`

Ці команди підраховують кількості елементів масиву y , значення яких потрапляють в задані інтервали і відображають їх на графіці. Для цього весь діапазон значень y ділиться на n інтервалів (за умовчанням 10) і підраховується кількість елементів в кожному з них.

Перша команда виводить гістограму для 10 інтервалів.

Друга команда виводить гістограму для n інтервалів.

Третя команда виводить гістограму з урахуванням діапазону зміни змінної x .

Четверта функція формує такі масиви p і x , що функція побудови стовпчикових діаграм `bar(x, p)` є гістограмою.

Якщо масив Y – двовимірний, то гістограма виводиться як стовпчикова діаграма з групами стовпців.

2 ПРАКТИЧНА ЧАСТИНА ЛАБОРАТОРНОЇ РОБОТИ

„Дослідження способів обробки і методів візуалізації інформації в системі комп'ютерної математики”

2.1 Мета лабораторної роботи

Вивчити і дослідити математичні операції, способи формування масивів даних і методи візуалізації, які застосовуються в процесі обробки інформації.

2.2 Завдання на лабораторну роботу

- 1) сформувати вектор-рядок,
- 2) створити з вектора-рядка зв'язаний з ним вектор-стовпець (транспонування з комплексним сполученням),
- 3) створити з вектора-рядка транспонований вектор-стовпець,
- 4) помножити і розділити отриманий вектор-рядок і вектор-стовпець на скаляр,
- 5) здійснити функціональне перетворення вектора-рядка і вектора-стовпця,
- 6) розділити вектор-рядок на вектор-рядок,
- 7) показати некоректність множення вектора-рядка на вектор-рядок,
- 8) здійснити поелементне множення вектора-рядка на вектор-рядок,
- 9) здійснити узгоджене множення вектора-рядка на вектор-стовпець,
- 10) здійснити узгоджене множення вектора-стовпця на вектор-рядок,
- 11) сформувати і ввести матрицю-рядок,
- 12) сформувати і ввести матрицю-стовпець,
- 13) сформувати і ввести дві узгоджених для множення матриці розмірністю більше, ніж 2×2 і перемножити їх,
- 14) визначити розмірність останнього сформованого масиву і визначити його довжину як стовпця і як рядка,
- 15) побудувати масив двовимірної сітки з одновимірному масиву,
- 16) побудувати масив двовимірної сітки з двох одновимірних масивів різної розмірності і кроку,
- 17) побудувати в двовимірній сітці функції $(\sin x) / x$ і експоненту,
- 18) відобразити побудовану тривимірну функцію $(\sin x) / x$ в першому підвікні у вигляді сполучених значень функції над сіткою з відповідним підписом всіх осей і підвікна зображення,
- 19) відобразити побудовану тривимірну функцію експоненти в другому підвікні у вигляді згладженої поверхні над сіткою з відповідним підписом всіх осей і підвікна зображення,

- 20) побудувати масив сітки для перетинів функцій,
- 21) у вікні відображення побудувати контури перетинів функції $(\sin x) / x$ на площині з підписом рівня перетинів,
- 22) у вікні відображення побудувати контури перетинів функції $(\sin x) / x$ в тривимірному просторі з підписом рівня перетинів,
- 23) у вікні відображення побудувати функцію $(\sin x) / x$ з перетинами під нею в горизонтальній площині,
- 24) сформуванати одновимірний масив випадкових величин з рівномірним (нормальним) законом розподілу,
- 25) побудувати гістограму сформованого одновимірного масиву випадкових величин з рівномірним (нормальним) законом розподілу,
- 26) сформуванати двовимірний масив випадкових величин з рівномірним (нормальним) законом розподілу,
- 27) побудувати гістограму сформованого двовимірного масиву випадкових величин з рівномірним (нормальним) законом розподілу,
- 28) відобразити у вікні відображення сформований двовимірний масив випадкових величин з рівномірним (нормальним) законом розподілу у вигляді тривимірного зображення з відповідним підписом всіх осей і вікна зображення,
- 29) відобразити у вікні відображення сформований двовимірний масив випадкових величин з рівномірним (нормальним) законом розподілу у вигляді яскравісного малюнка,
- 30) по кожному з пунктів завдання проаналізувати отриманий результат.

2.3 Технічне забезпечення лабораторної роботи

- 1) персональний комп'ютер,
- 2) програмне забезпечення – система комп'ютерної математики.

2.4 Перелік операцій, команд і функцій, рекомендований для застосування в лабораторній роботі

clear all, clc, close all, tic, toc *, ., /, ^, [], size, sort, length, sinc, exp ;, :, meshgrid, sqrt, mesh, xyzlabel, legend, title, meshz, subplot, surf, contour, clabel, contour3, surfc, rand, randn, hist, normrnd, unifrnd, imshow.

2.5 Порядок виконання лабораторної роботи

Напередодні лабораторної роботи:

- 1) вивчити завдання і порядок виконання лабораторної роботи,
- 2) вивчити теоретичні відомості по лабораторній роботі.

Під час лабораторної роботи:

- 1) отримати допуск до проведення лабораторної роботи, відповівши на питання викладача по теоретичній частині досліджень при проведенні лабораторної роботи,
- 2) написати програмний код по кожному пункту завдання і проаналізувати результати, отримані при виконанні програмних кодів,
- 3) зробити висновки по лабораторній роботі.

Після лабораторної роботи:

- 1) підготувати звіт про лабораторну роботу відповідно до приведених в даному методичному посібнику вимог,
- 2) захистити звіт перед викладачем, відповівши на його питання по практичному проведенню досліджень при виконанні лабораторної роботи.

Час, що відводиться на проведення лабораторної роботи в комп'ютерному класі, – 4 години, на самостійну роботу з підготовки до проведення лабораторної роботи і до захисту звіту – 5 годин.

2.6 Зміст звіту про лабораторну роботу

- 1) Титульний лист,
- 2) Розділ: Мета лабораторної роботи і Завдання на лабораторну роботу,
- 3) Розділ: Хід лабораторної роботи – програмні коди до всіх пунктів завдання з додаванням результатів виконання всіх операцій і команд,
- 4) Розділ: Висновки по лабораторній роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. Яким чином можна сформувати вектор-рядок ?
2. Як виконуються арифметичні операції з масивами ?
3. Як виконуються арифметичні операції з матрицями ?
4. Які команди дозволяють визначити розмір і довжину масиву ?
5. Який порядок введення матриці в СКМ ?
6. Принципи формування двовимірної сітки
7. Порядок побудови двовимірної функції на сітці
8. Команди побудови вікон і підвікон зображення
9. Функції підпису графіків, підвікон і вікон зображення
10. Створення сітки для побудови перетинів функції
11. Порядок побудови одновимірних і тривимірних перетинів функції
12. Підпис рівнів перетинів функції
13. Порядок формування одновимірних і тривимірних масивів з різними видами розподілу випадкових величин
14. Сортування масивів
15. Побудова гістограм масивів випадкових величин
16. Побудова тривимірних зображень масивів
17. Побудова яскравісних зображень масивів

ЛІТЕРАТУРА

Основна:

1. Смоленцев Н.К. Основы теории вейвлетов. Вейвлеты в MATLAB. – М.: ДМК Пресс, 2005. – 304 с.
2. Дьяконов В.П. Вейвлеты. От теории к практике. Изд. 2-е, перераб. и доп. – М.: Солон-Пресс, 2004. – 400 с.
3. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. – М.: Наука, Гл. ред. физ.-мат. лит., 1989. – 496 с.

Додаткова:

1. Потёмкин В.Г. Система инженерных и научных расчётов MATLAB 5.X. В 2-х томах. – М: Логос, 2000. – 364 и 303 с.

ДОДАТОК А

Вимоги до оформлення і форма титульного аркуша звіту про лабораторну роботу

- 1) Звіт виконується на аркушах формату 11 (А4) машинописним або комп'ютерним способом у будь-якому текстовому редакторі,
- 2) Колір шрифту – чорний, гарнітура – Таймс, кегль – 14, поля з усіх боків – 20 мм,
- 3) Мета лабораторної роботи і Завдання на лабораторну роботу оформляються в одному розділі з нового аркуша,
- 4) Хід лабораторної роботи оформляється в одному розділі з нового аркуша,
- 5) Висновки щодо лабораторної роботи оформляються в одному розділі з нового аркуша,
- 6) Допускається для розділу Хід лабораторної роботи, з метою економії паперу, виконання тексту і програмних кодів кеглем – 12, при необхідності малюнки можуть виконуватися в кольорі.

Форма титульного аркушу звіту

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК		
Кафедра АСМНС		
ЗВІТ про лабораторну роботу ДОСЛІДЖЕННЯ СПОСОБІВ ОБРОБКИ І МЕТОДІВ ВІЗУАЛІЗАЦІЇ ІНФОРМАЦІЇ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ по дисципліні СПЕКТРАЛЬНІ МЕТОДИ ОБРОБКИ ІНФОРМАЦІЇ		
Виконав(ла) студент(ка) групи К-41 Іванов Петро Сидорович <hr/> (підпис студента)		
Перевірив Перелигін Б.В.		
Оцінка за підготовку до лабораторної роботи	Оцінка за виконання лабораторної роботи	Загальна оцінка
Одеса – 2013		