МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Методичні вказівки
для самостійної роботи студентів з дисципліни

“ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ”

Одеса - 2012

Методичні вказівки до самостійної роботи студентів з дисципліни “Чисельне моделювання фізичних процесів”/ Худинцев М.М., Співак А.Я. – Одеса, ОДЕКУ, 2012. – 35с.

Методичні вказівки призначенні для студентів V курсу очної форми навчання за спеціальністю – “Радіоекологія”.

ЗМІСТ

ПЕРЕДМОВА

Методичні вказівки для самостійної роботи студентів спрямовані на полегшення засвоєння студентами матеріалів курсу ”Чисельне моделювання фізичних процесів” протягом 1 семестру, на який відводиться 30 год. лекційних, 30 год. лабораторних занять, а також 90 год. самостійної роботи студентів. Вибрані питання повністю формують уявлення про предмет та головні напрямки розвитку цієї дисципліни.

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.

1 МІСЦЕ І ЗНАЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна “Чисельне моделювання фізичних процесів” належить до професійно - орієнтованого циклу підготовки фахівців освітньо-кваліфікаційного рівня „магістр” за спеціальністю «Радіоекологія» шифр 8.04010605, але в той же час її зміст має безпосереднє відношення до фундаментального циклу підготовки.

Теоретичною основою курсу є курс вищої математики, знання з загальної теорії диференційних рівнянь, основ фізики та хімії, а також відповідні спеціальні курси.

Метою курсу є дати уявлення про фізичні концепції в моделюванні процесів у довкіллі, фізичні концепції у довкіллі в екології та елементи фізичного базису, що використовується в моделюванні навколишнього середовища, навчити студентів будувати математичні моделі тих або інших фізичних або хімічних процесів з використанням наведених в курсі математичних методів.

В результаті вивчення дисципліни студент повинен:

Знати: основні поняття складних систем, зв'язок між ієрархічними системами, елементів нелінійної динаміки.

Уміти: програмувати в інтегрованому пакеті MATLAB (або MATHEMATICA) як в операційній системі DOS, так і в WINDOWS.

Робочою програмою передбачені 30 год. лекцій, 30 год. лабораторних занять та 90 год. самостійної роботи студентів.

Контроль поточних знань виконується на базі кредитно-модульної системи організації навчання. Підсумковим контролем є іспит.

2 СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс з дисципліни “Чисельне моделювання фізичних процесів” містить в собі:

- два змістовних теоретичних;
- один змістовний практичний модуль.

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

Рис.1 Структурна схема дисципліни «Чисельне моделювання фізичних процесів»

3 ПРОГРАМА МОДУЛІВ КУРСУ

3.1 Програма лекційного модуля

Змістовні модулі	Розділи програми (назва)	Теми	Кіл-сть аудиторних годин	Кіл-сть годин СРС	Форми завдань на СРС	Форми поточного контролю СРС
ЗМ-Л1	1.1 Поняття складової системи	1. Модельні системи, складність, обчислювана складність.	2	4	Підготовка до лекційних занять Підготовка до КР1 Самостійне вивчення розділів	КР – 1
		2. Стохастична система.	2	4		
		3. Інформаційна ентропія, фізична ентропія, термодинамічна ентропія.	2	4		
	1.2 Дисипація	1. Дисипація, необратимість.	2	4		
		2. Прямування до рівноваги.	2	4		
		3. Релаксація системи, гідродинамічна стадія.	2	4		
		4. Стохастичні системи. Рівняння Ланжевєна. Рівняння Фоккера-Планка.	2	4		
		5. Зменшення складності, принцип підлеглості.	2	4		
ЗМ-Л2	1.1 Джерела інформації	1. Інформація, приріст інформації	2	4	Підготовка до лекційних занять Підготовка до КР2 Самостійне вивчення розділів	КР – 2
		2. Елементи теорії кодиювання та передачі інформації, роль електромагнітних хвиль.	2	4		
		3. Джерела інформації в ланцюгу Маркова.	2	2		
		4. Принцип максимума інформації.	2	2		
	1.2 Нелінійна динаміка	1. Точкові відображення.	2	4		
		2. Нелінійна динаміка.	2	4		
		3 Типи особливостей. Аттрактори.	1	4		
		4 Показники Ляпунова.	1	4		
		Разом:	30	60		

Після вивчення **ЗМ- Л1**, студенти повинні оволодіти наступними знаннями:

- поняття складової частини
- стохастична система
- дисипація, необратимість
- стохастичні системи

Після вивчення **ЗМ- Л2**, студенти повинні оволодіти наступними знаннями:

- інформація, приріст інформації
- елементи теорії кодиювання та передачі інформації, роль електромагнітних хвиль
- точкові відображення
- показники Ляпунова.

Наявне навчально-методичне забезпечення змістовних модулів ЗМ-Л1, ЗМ-Л2

Основна Література

1. Стратонович Р.Л. Нерівноважна нелінійна термодинаміка, М., Наука,1996р.
2. Хакен Г. Синергетика. М. Мир, 1980р.
3. Боголюбов Н.Н., Мітропольський Ю.А. Асимптотичні методи в теорії нелінійних коливань, М. Наука, 1974р.

Додаткова література

1. Хакен Г. Синергетика: ієрархії нестійкостей в самоорганізованих системах.
2. Куні Ф.М. Статистична фізика та термодинаміка, М., Наука, 1982р.

3.2 Програма практичного модуля

Змістовні модулі	Форма занять	Теми робіт (занять)	Кіл-сть аудиторних годин	Кіл-сть годин СРС	Форми завдань на СРС	Форми поточного контролю СРС
ЗМ-П1	Лабораторні роботи	1. Модель Лоренца.	8	8	Підготовка до лабораторних занять і усного опитування	Усне опитування
		2. Патерни поводження в біології.	8	8		
		3.Розпізнання образів.	6	6		
		4.Введення в квантову теорію інформації.	8	8		
		Разом:	30	30		

Після вивчення **ЗМ-П1**, студенти повинні оволодіти наступними вміннями:
- застосування моделі Лоренца
- розпізнавання образів
- приводити приклади складних систем.

Наявне навчально-методичне забезпечення змістовного модуля ЗП-П1:

Основна література

1 . Хакен Г. Синергетика: ієрархії нестійкостей в самоорганізованих системах.
2. Куні Ф.М. Статистична фізика та термодинаміка, М., Наука, 1982р.
3. Пригожин І.Р., Стенгерс І. Час, хаос,квант.,М., Мир, 1988р.

Додаткова література

1. Хакен Г. Синергетика: ієрархії нестійкостей в самоорганізованих системах.
2. Куні Ф.М. Статистична фізика та термодинаміка, М., Наука, 1982р.

3.3 Програма модуля наукової роботи

Для дисципліни «Сучасні методи моделювання систем навколишнього середовища» пропонуються наступні види науково-дослідної роботи студентів, що оцінюються за двома рівнями:

1 рівень НДР:

Е3 – університетські наукові конференції (Щорічна студентська наукова конференція, ОДЕКУ) – 0,25 кр.

2 рівень НДР:

Е3 – всеукраїнські і міжнародні конференції (Всеукраїнська наукова конференція студентів і аспірантів «Екологічні проблеми регіонів України», ОДЕКУ) – 0,5 кр.

Е4 – публікації (опублікування матеріалів тез доповідей виступів на конференціях) – 0,5 кр. Виконання наукового модуля оцінюється за представленими звітними документами – програма конференції або опубліковані матеріали тез доповідей.

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 та проводити наукові дослідження.

Нижче наведемо найбільш корисні, з точки зору чисельного моделювання фізичних процесів, структури коду.

Оператори цикла

Цикл for

Як і інші мови програмування MATLAB має свій цикл «for». Він дозволяє повторювати задане число разів вираз або групу виразів.

Наприклад:

```
for i=1: n
x(i) = 0
end
```

присвоює значення 0 першим n елементам x. Якщо n<1, конструкція також має сенс, але вирази, що входять до неї, виконуватись не будуть. Якщо x ще не визначенний або число його елементів менших n, необхідні елементи x будуть створенні автоматично. Цикли можуть бути вкладеними і звичайно для наочності записуються з отступом:

```
for i=1: m
  for j=1: n
     A (i, j)=1/(i+j-1);
   end
end
A
```

Крапка з комою «;» у кінці внутрішнього виразу подавляє багаторазове виведення результатів. A, що стоїть після усіх циклів виводить кінцевий результат. Кожному оператору for повинен відповідати оператор end. Якщо запровадити просто

```
for i = 1:n
x (i) =0
```

система буде терплячо очікувати, коли у тіло циклу будуть запровадженні інші оператори. Нічого не відбудеться, доки не буде запроваджений оператор end.

Для іншого прикладу присвоємо

```
t=
```

-1
0
1
3
5

і згенеруємо матрицю, стовпчики якої є ступенеми елементів вектору t.

A= 1 -1 1 -1 1
0 0 0 0 1
1 1 1 1 1
81 27 9 3 1
625 125 25 5 1

Найбільш очевидним є подвійний цикл:

```
n=max (size (t));
for j=1: n
    for i=1: n
       A (i, j)=t(i)^(n-j);
     end
end
```

Але наступний цикл з векторними операціями працює значно швидше і водночас ілюструє той факт, що змінна циклу for може змінюватись у зменшенному порядку:

```
A(:,n)=ones (n,1);
for j=n-1:-1:1
   A(:,j)=t.*A(:,j+1);
end
```

загальна форма циклу for має вигляд

```
for v=вираз
   оператори
end
```

Вираз є фактично матрицею, оскільки з ним працює MATLAB . Стовпчики матриці один за іншим присвоюються змінної v, а після цього виконуються оператори. Іншим словом, усе це виконується слідуючим чином:

```
E=вираз;
[m, n]=size (E);
for j=1:n
     v=E(:,j);
     оператори
end
```

Звичайно вираз має структурне m: n або m:i:n – матриця-рядок, стовпчики якої – скаляри. У цьому випадку цикл for є подібним циклам 'for' або 'do' інших мов програмування.

Цикл while

У MATLAB існує також власна версія циклів 'while', що здійснюють багаторазове виконання виразу або групи виразів наперед не визначене число разів, під управлінням логічного виразу. Наведемо простий приклад для ілюстрації циклу while: Визначити найменше ціле n, для якого n! є 100-значне число. Наведений нижче цикл while вирішує цю задачу:

```
n=1
while prod (1: n)<1.e100
      n=n+1;
end
n
```

Більш корисно ілюстрацією застосування є while є обчислення матричної експоненти, яка існує у MATLAB – expm (A). Один з можливих засобів її визначення – сума рядку:

expm(A)=1+A+A^2/2!+A^3/3+…

Його має сенс використовувати практично при не занадто великих значеннях елементів матриці А. Сенс у тому, щоб складати кілька членів цього ряду, необхідних для отримання результату, щоб складання наступних членів вже не змінювало результату завдяки кінцевій точності машинної арифметики. Нижче А – матриця, Е – бажана експонетна, F – конкретний член ряду, k – це індекс цього члена. Вирази, записані з отступом, будуть повторюватись до тих пір, доки F не стане настільки малим, що додавання його до Е не буде змінювати Е.

```
A= ;
E=zeros (A);
F=eye (A);
k=1;
while norm (E+F-E, 1)>0
        E=E+F;
        F=A*F/k;
        k=k+1;
end
E
```

Загальна форма циклу while

```
While вираз
   оператори
end
```

оператори виконуються до тих пір, доки всі елементи матричного виразу стануть ненулевими. Матричний вираз майже завжди має вимірність 1 на 1 (скляр), тому ненульове значення відповідає TRUE ('істина' і 'так').

Коли матричний вираз – не скляр, воно може бути скороченно за допомогою функцій any і all.

Оператори if і break

Наведемо два приклади, що ілюструють застосування у MATLAB вирази if. Перший показує, як обчислення можуть бути розбиті на 3 варіанти в залежності від знаку і парності числа n (виконання логічних умов).

```
If n < 0
   A=negative (n)
elseif mod (n,2) = = 0
   A=even (n)
else
   A=odd (n)
end
```

Другий приклад являє собою проблему з теорії чисел. Візьмемо будь – яке ціле позитивне число. Якщо воно парне, поділемо його на 2, якщо не парне, то помножимо його на 3 і прибавимо 1. Будемо повторювати цей процес, доки ціле не стане одиницею. Проблема теорії полягає в наступному: чи існує таке ціле число, для якого цей процес не має кінця? Програма ілюструє використання виразів if і break. Вона також демонструє функцію input, що робить запит на запровадження даних з клавіатури і оператор break, який забезпечує вихід з циклів.

```
%Класична ‘3n+1’ – проблема терії чисел
while 1
n=input (‘Запровадьте n, для виходу – негативне.’);
if n >0, break, end;
  while n >1
     if rem (n, 2) = = 0
        n=n/2
     elso
          n=n*3+1
       end;
   end
end
```

М-файли: командні файли-програми і файли-функції

Найпростіший режим роботи MATLAB – виконання команд у діалоговому режимі: команда подається на одному рядку, MATLAB негайно обробляє її і видає результат. Існує також можливість обробки послідовності команд, що оформленні у вигляді файлі. Разом ці два режими утворюють оточення MATLAB.

Дискові файли, що містять оператори MATLAB, називаються М-файлами, тому що як розширення імен цих файлів зазначається ".m". М – файл є послідовністю операторів MATLAB, до якої можуть бути включеними посиланнями на інші М-файли, а пр. необхідності – і рекурсивні посилання на самий даний М-файл.

Перша можливість використання М-файлів – автоматизація виконання довгих послідовностей команд. Такі М-файли мають назву командних (файли - програми).
Другий тип М-файлу забезпечує поширення MATLAB новими функціями, дає можливість додавати нові функції для розв»язування спеціальних задач. М-файли цього типу називаються функціями. Файли як одного, так і іншого типу записуються у коді ASCII і створюються будь-яким текстовим редактором.

Командні файли-програми

Як тільки активізується такий файл, MATLAB виконує усі команди, що містяться в ньому, не очікувати запровадження з клавіатури, – являє собою програму на мові MATLAB.
Оператори у командному файлі працюють з даними, що знаходяться у робочій області оперативної пам»яті. Командні файли використовуються звичайно в тих випадках, коли необхідне повторення довгих послідовностей операторів MATLAB. Наприклад:

```
% M – файл для обчислення чисел Фібоначчі
f=[1 1]; i=1;
while f (i)+f (i+1)<1000
  f(i+2)=f (i)+f(i+1); i=i+1
end
plot (t)
```

Ця послідовність з команд міститься в файлі з ім2ям fibno.m. Якщо набирати команду fibno з командного рядка MATLAB, то будуть обчислені перші 100 чисел Фібоначчі і побудований графік. Після завершення виконання цієї послідовності команд змінні f і I залишаються в робочій пам»яті.

Файли – функції

Якщо перший рядок М-файлу містить зазначення 'function', то цей файл є функцією. Функція відрізняється від програми засобом передачі аргументів – тим, що змінні, які визначаються та використовуються всередині файлу – функції, є локальними і не працюють поза цією функцією.
Файли – функції використовуються для поширення можливостей MATLAB за допомогою нових функцій, що написані на мові MATLAB. Наприклад, нехай файл mean.m містить оператори:

```
function y=mean(x)
%MEAN середнє значення. Для векторів MEAN (x) обчислює
% середнє значення. Для матриць MEAN (x) – це вектор – рядок,
% що містить середні значення відповідних стовпчиків.
[m, n]=size (x);
if  m= =1
      m=n;  % Виділення вектору – рядка.
```

```
end
y=sum (x)/m;
```

цей файл визначає нову функцію з ім.»ям mean, що використовується абсолютно таким же чином, як і всі інші функції MATLAB. Наприклад, якщо Z – вектор цілих чисел від 0 до 99.

```
Z =1:99
```

то середнє значення його елементів можна знайти, запровадив

```
mean (Z)
```

що дає

```
ans=50
```

Розглянемо деякі деталі файлу mean.m:

- Перший рядок оголошує ім.»я функції, вхідні та вихідні аргументи. Без цього рядка файл буде програмою, а не функцією.
- Символ % використаний для зазначення того, що кінець рядка – це коментар і при виконанні повинен ігноруватись.
- Декілька перших рядків документують М-файл і будуть виведені, якщо запровадити команду

```
help mean
```

- Змінні m, n і y – локальні для mean і не зберігаються у пам»яті після завершення виконання mean (або, якщо вони існували раніше, то їхнє значення не змінюються).
- Цілком не обов»язково містити цілі числа від 1 до 99 у змінні з ім.»ям x. Звернення до mean було здійснене за допомогою змінної Z. Вектор Z, що містить числа від 1 до 99, був використаний mean, де став локальною змінною x.

Існує і трохи ускладнений варіант функції mean під назвою stat, що обчислює середнє квадратичне відхилення

```
function [mean, stdev]=stat (x)
[m, n]=size (x);
if m = = 1
     m=n;    %Handle isolated row vector.
end mean=sum(x)/m;
stdev=sqrt(sum(x.^2)/m – mean.^2);
```

Функція stat демонструє можливість використання декількох вихідних аргументів.

І, нарешті, функція, що обчислює ранг матриці, використовуючи декілька вхідних аргументів:

```
function r=ranc(x, tol)
% ranc of matrix
s=svd(x);
if(nargin = = 1)
    tol=max(size(x))*s(1)*eps;
end
```

r=sum (s>tol);

Цей приклад ілюструє використання змінної nargin, що змінюються, для визначення числа вхідних аргументів. Змінна nargout, що не використовується в данному прикладі, містить число вихідних змінніх.

Важливі зауваження

1) Коли М-файл використовується вперше під час сеансу, він компілюється і поміщується у пам»ять. Після цього файл доступний без повторної компіляції. Цей файл залишається в пам»яиі до пих пір, доки не проведена операція звільнення пам»яті.

2) Команда what виводить на екран перелік усіх М-файлів, що є доступними у поточному змісті диска; ture виводить перелік М-файлів; може «!» використовуватись для виклику текстового редактора, що дозволить створювати або змінювати М-файли.

3) Якщо запроваджується ім.»я або будь – яка послідовність символів, наприклад test, інтерпретатор MATLAB виконує наступні кроки:

- перевіряє, чи не є послідовність символів test змінною;
- перевіряє, чи не є вона ім.»ям вбудованої функції;
- перевіряє поточний зміст, чи не існує файл з ім.»ям test.mex;
- перевіряє змісти, які описані у показчикові робітничого оточення MATLABPATH в пошуках файлу з ім.»ям test.mex;
- виконує кроки 3) і 4) для поширення файлів test.m.

Команди echo, input, pause, keyboard

Звичайно під час виконання М-файлу команди на екрані не відображаються. Проте інколи буває корисно зробити М-файли видимими прямо під час їхнього виконання, що і дозволяє зробити команда echo.

Функція input організує отримання вихідних даних від користувача. Так,

input («Скільки в тебе яблук?»)

видає запит у вигляді текстового рядка, чекає, а після цього повертає число або вираз, що запроваджено з клавіатури. Аналогічно input, але більш потужним засобом є keyboard. Ця функція активізує клавіатуру вашого комп»ютера як М-файл – сценарій. Подібна інтерпретація клавіатури корисна під час відлагодження програми або при необхідності модифікації змінних під час виконання.

Команда pause викликає зупинку процедури і призводить систему у стан очікування, доки користувач не натисне будь-яку клавішу для продовження.

pause(n)

дає паузу тривалістю n секунд перед продовженням.

Графіка

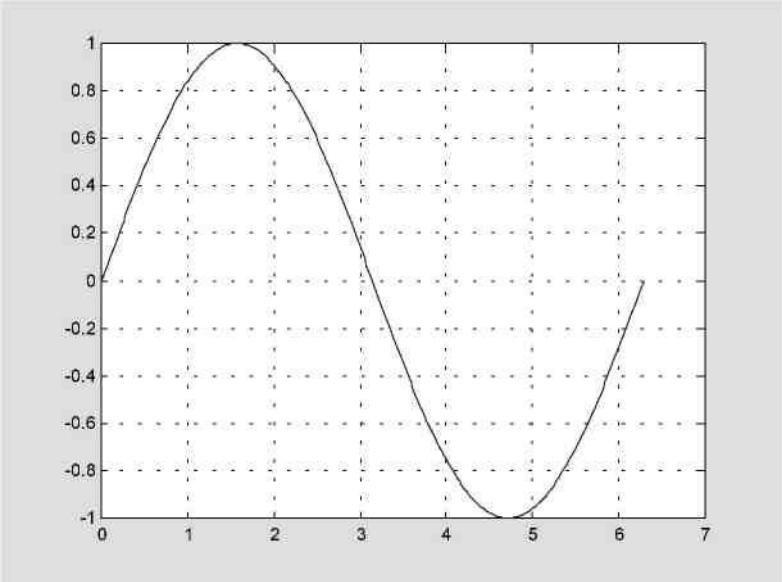
MATLAB має широкі можливості для графічного зображення векторів і матриць, а також для створення коментарів і друку графіки.

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 кожну 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 графіки

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

hold on

MATLAB не стирає існуючий графіки, 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 о на вершинах.

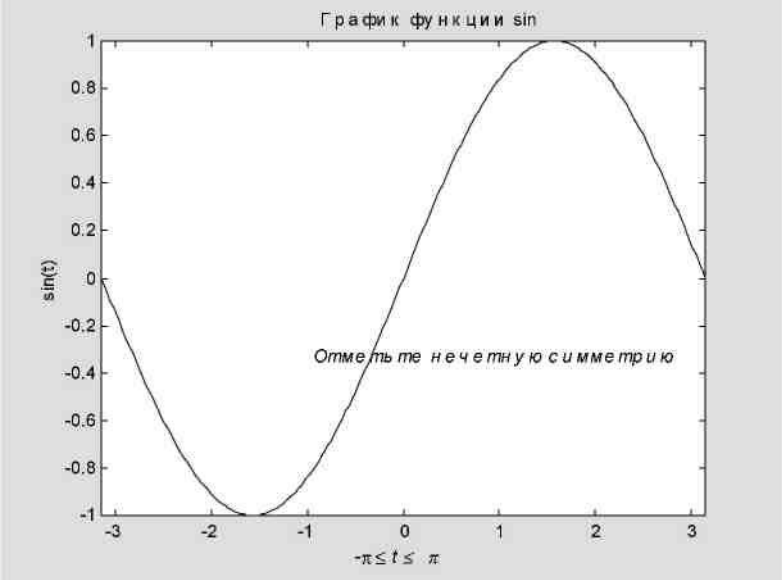
Управління вісями

Функція *axis* має кіілька можливостей для налагодження масштабу, орієниації 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 \itt \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 z 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 M-файлів. Результат може бути надісланий безпосереднью на принтер, обраний за замовчуванням, або збережений у заданому файлі. Можливо широке варіювання формату вихідних даних, включаючи використання PostScript.
Наприклад, команда
print -depsc2 magicsquare.eps
збереже поточне вікно зображення, як кольоровий PostScript Level 2 Encapsulated у файлі magicsquare.eps
Слід знати, що не всі Postscript принтери підтримують Level 2.

Довідка та поточна документація

Є кілька способів отримати поточну документацію по функціях MATLAB .

- Команда help
- Вікно довідки
- MATLAB Help Desk
- Поточні довідкові сторінки
- Зв'язок з The MathWorks, Inc.

Команда help

Команда *help* – основний спосіб визначення синтаксису й поводження окремих функцій. Інформація відображається у командному вікні. Наприклад
help magic

видасть
MAGIC Magic square.
MAGIC(N) is an N-by-N matrix constructed from the integers 1 through N^2 with equal row, column, and diagonal sums. Produces valid magic squares for N = 1,3,4,5,...
MATLAB у поточній довідці використовує великі літери для функцій і змінних. Однак імена функції складаються з малих літер.
Всі функції MATLAB організовані в логічні групи. Наприклад, всі функції лінійної алгебри перебувають у директорії matfun. Щоб вивести імена всіх функцій у цій директорії з коротким описом, треба набрати
help matfun
Matrix functions - numerical linear algebra.
Matrix analysis.
norm - Matrix or vector norm. normest - Estimate the matrix 2-norm.
Команда help
HELP topics:
MATLAB \general – General purpose commands.
MATLAB \ops - Operators and special characters

Вікно довідки

Вікно довідки MATLAB з'являється на PC або Mac після вибору опції Help Window у меню Help або натисканням кнопки питання на панелі інструментів. Та сама операція може бути виконана при наборі команди
helpwin
Для отримання вікна довідки по окремих розділах потрібна команда
helpwin *topic*
Вікно довідки надає таку саму інформацію, що і команда *help,* але віконний інтерфейс забезпечує більш зручний зв'язок з іншими розділами довідки.

Середовище MATLAB

Середовище MATLAB містить у собі як сукупність змінних, створених за час сеансу роботи MATLAB , так і набір файлів, що містять програми й дані, які продовжують існувати між сеансами роботи.

Робочий простір

Робочий простір – це область пам'яті, доступна з командного рядка MATLAB . Дві команди, *who* і *whos,* показують поточний зміст робочого простору. Команда *who* видає короткий список, а команда *whos* розмір і використовувану пам'ять.

Наприклад,

whos

Name	Size	Bytes	Class
A	4x4	128	double
D	5x3	120	double
M	10x1	3816	cell array
S	1x3	442	struct array
H	1x11	22	char array
N	1x1	8	double
S	1x5	10	char array
V	2x5	20	char array

Grand total is 471 elements using 4566 bytes.

Для видалення усіх існуючих змінних з робочого простору MATLAB , наберіть

clear

Команда save

Команда *save* зберігає зміст робочого простору у M-файлі, що може бути прочитаний командою *load* у наступних сеансах роботи MATLAB . Наприклад,

save August17th

зберігає зміст усього робочого простору у файлі August17th.mat. Якщо потрібно, ви можете зберегти тільки певні змінні, указуючи їхні імена після імені файлу.

-ascii	Використовує 8-знаковий текстовий формат
-ascii –double	Використовує 16-знаковий текстовий формат
-ascii -double –tabs	Розділяє елементи масиву табуляцією
-v4	Створює файл для MATLAB 4
-append	Додає дані в існуючий MAT-файл.

Команда **path**

указує маршрут пошуку по усіх платформах. На PC і Mac оберіть опцію Set Path з меню File для перегляду і зміни маршруту.

Операції над дисковими файлами

Команди *dir*, *type*, *delete* і *cd* здійснюють комплекс групових операційних системних команд для маніпуляцій над файлами.

Наприклад:

MATLAB	MS-DOS	UNIX	VAX/VMS
dir	Dir	Ls	dir
type	Type	cat	type
delete	del. erase	rm	delete
cd	Chdir	cd	set defalt

Команда diary

Команда *diary* створює щоденник сеансу MATLAB у дисковому файлі. Ви можете переглянути й відредагувати кінцевий текстовий файл, використовуючи будь-який текстовий процесор. Для створення файлу під ім'ям diary, який містить всі команди, які ви використовували, уведіть

```
diary
```

Для збереження сеансу MATLAB у файлі з певним ім'ям, використайте

```
diary filename
```

При зупинці запису сеансу роботи, наберіть

```
diary off
```

Операції з матрицями та масивами

Ви вже бачили операцію транспонування матриці, A'. Додавання матриці до її транспонованого дає симетричну матрицю.

```
A + A'
ans =
     32     8    11    17
 8   20    17    23
11   17    14    26
     17    23    26     2
```

Символ множення, *, позначає матричний добуток. Множення матриці на її транспоновану також дає симетричну матрицю.

```
A'*A
ans =
378         212    206    360
212         370    368    206
206         368    370    212
360         206    212    378
```

Визначник матриці обчислюється за допомогою

```
d = det(A)
d =
0
```

Оскільки задана матриця є сингулярною (det=0), вона не має зворотної. Якщо зробити спробу визначити зворотню матрицю

X = inv(A),

отримаємо попередження

Warning: Matrix is close to singular or badly scaled.

Results may be inaccurate. RCOND = 1.175530e-017.

Власні значення визначаються

e = eig(A)

e =

34.0000

8.0000

-0.0000

-8.0000

Одне із власних значень дорівнює нулю, що є наслідком сінгулярності. Найбільше власне значення дорівнює 34, магічній сумі. Це відбувається тому, що вектор, який складається з одиниць, є власним вектором.

Ступінь матриці позначається

A^2

Коефіцієнти характеристичного поліному отримуються за допомогою

poly(A)

ans =

1.0e+003 *

0.0010 -0.0340 -0.0640 2.1760 0.0000

Масиви

Коли ми виходимо з миру лінійної алгебри, матриці стають двовимірними чисельними масивами. Арифметичні операції з масивами здійснюються поелементно. Це означає, що підсумовування й вирахування є однаковими операціями для матриць і масивів, а множення – по-різному.

Список операторів містить у собі:

+	підсумовування
–	вирахування
.*	множення
./	ділення
.\	ліве ділення
.^	піднесення в ступінь
.'	матричне транспонування

Елементарні математичні функції працюють із масивами поелементно.

Так

format short g x = (1.5:0.1:2)';

```
logs = [x log10(x)]
```

створює таблицю логарифмів

```
logs =
    1.5    0.17609
    1.6    0.20412
    1.7    0.23045
    1.8    0.25527
    1.9    0.27875
    2      0.30103
```

Управління потоками

MATLAB має п'ять видів структур керування потоками:

- оператор *if*
- оператор *switch*
- цикли *for*
- цикли *while*
- оператор *break*

if

Оператор *if* обчислює логічний вираз й виконує групу операторів, якщо значення виразу „істина". Необов'язкові ключові слова *elseif* і *else* слугують для виконання альтернативних груп операторів. Ключове слово *end*, що погоджується з *if*, завершує останню групу операторів без залучення фігурних або звичайних дужок.

Алгоритм MATLAB для створення магічного квадрата порядку n поділяється на три різних випадки : n непарне, n парне, але не ділиться на 4, і n парне й ділиться на 4. Нижче наведенj приклад відповідного коду

```
if rem(n,2) ~= 0
   M = oddmagic(n) elseif rem(n,4) ~= 0
   M = single_even_magic(n) else
   M = doubleevenmagic(n) end
```

switch и case

Оператор *switch* виконує групу операторів, базуючись на значенні змінний або вираження. Ключові слова *case* і *otherwise* розділяють ці групи. Виконується тільки перший відповідний випадок. Необхідно використовувати *end* для узгодження з *switch.*

Логіка алгоритму магічних квадратів може бути описана на наступному прикладі

```
switch (rem(n,4)==0) + (rem(n,2)==0) case 0
   M = oddmagic(n) case 1
   M = single_even_magic(n) case 2
```

```
    M = doubleevenmagic(n)
    Otherwise error( ' Це неможливо' ) end
```
for

Цикл *for* повторює групу операторів фіксоване, визначене число раз. Ключове слово *end* окреслює тіло циклу.

```
for n = 3:32
   r(n) = rank(magic(n));
   end r
```

Крапка з комою після виразув тілі циклу запобігає повторення виводу результатів на екран, а r після циклу виводить остаточний результат.

while

Цикл *while* повторює групу операторів певне число раз, поки виконується логічна умова. Ключове слово *end* окреслює тіло циклу.

Нижче наведена повна програма, яка ілюструє роботу операторів *while, if, else* і *end,* що використає метод ділення відрізка навпіл для знаходження нулів поліному.

```
a = 0;
fa = -Inf;
b = 3;
fb = Inf;
while b-a > eps*b
  x = (a+b)/2;
  fx = x^3-2*x-5;
if sign(fx) == sign(fa)
 a = x; fa = fx;
  else
  b = x; fb = fx;
    end end x
```

Результатом буде корінь полінома

```
x =
2.09455148154233
```

Для оператора while вірні ті ж застереження щодо матричного порівняння, що й для оператора if, які обговорювалися раніше.

break

Оператор *break* дозволяє достроково виходити із циклів *for* або *while.* У вкладених циклах *break* здійснює вихід тільки із самого внутрішнього циклу.

Нижче представлений варіант попереднього прикладу. Чому використання оператора *break* у цьому випадку вигідно?

```
a = 0; fa = -Inf; b = 3; fb = Inf; while b-a > eps*b x = (a+b)/2; fx = x^3-2*x-5; if fx
==0
   break elseif sign(fx) == sign(fa)
   a = x; fa = fx; else
   b = x; fb = fx; end end x
```

Сценарії та функції

MATLAB – це потужна мова програмування, також як і інтерактивне обчислювальне середовище. Файли, які містять код мовою MATLAB , називаються М-файлами. Ви створюєте М-файли, який є водночас функцію або командою MATLAB .

Існує два види М-файлів

- Сценарії, які не мають вхідних і вихідних аргументів. Вони оперують з даними з робочого простору.
- Функції, які мають вхідні й вихідні аргументи. Вони оперують з локальними змінними.

Якщо ви є новачком в MATLAB програмуванні, створюйте М-файли у поточній директорії. Якщо ж ви розробили багато М-файлів, для їх группування у різні директорії й персональні пакети програм (toolboxes) необхідно додати їхній маршрут пошуку MATLAB .

Якщо ви повторюєте ім'я функції, то MATLAB викликають лише першу з них.

Зміст М-файлу редактуватиметься звичайним способом за допомогою будь-якого текстового редактору.

Сценарії

Коду ви викликаєте сценарій, MATLAB викликає команди з файлу. Сценарії можуть оперувати існуючими даними в робочому просторі або вони можуть створювати ці дані. Хоча сценарії не повертають значень, всі змінні, які вони створюють, залишаються у робочому просторі для використання в наступних обчисленнях.

Функції

Функції – це М-файли, які матимуть вхідні й вихідні дані. Ім'я М-файлу й функції повинне бути тим самим. Функції працюють із змінними в межах їх власного робочого простору.

Приклад функції *rank.* М-файл *rank.m* перебуває у директорії toolbox/MATLAB /matfun

Ви можете переглянути його зміст за допомогою

```
type rank
```

```
function r = rank(A,tol)
 %RANK  Matrix rank.
 %  RANK(A) provides an estimate of the number of linearly
 %   independent rows or columns of a matrix A.
 %  RANK(A,tol) is the number of singular values of A
 %  that are larger than tol.
 %  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
 %    Copyright (c) 1984-98 by The MathWorks, Inc.  %    $Revision: 5.7 $
$Date: 1997/11/21 23:38:49 $
s = svd(A);
if nargin==1
    tol = max(size(A)') * max(s) * eps; end r = sum(s > tol);
```

Перший рядок функції М-файлу починається з слову *function.* Тут відбувається визначення імені та переліку аргументів. У нашому випадку, використовується до двох вхідних аргументів і один вихідний.

Функція *rank* може бути використана у інших сценаріях у різних формах:

```
rank(A)
r = rank(A)
r = rank(A, 1.e-6)
```

Глобальні змінні

Якщо ви хочете, щоб більше однієї функції використали окрему копію змінної, просто оголосите її як *global* у всіх функціях:

```
global d1 p s
```

Функція від функцій

Клас функцій з умовною назвою "функція від функцій", працює з нелінійними функціями скалярних змінних. Тобто одна функція працює з іншою. Функції від функцій містять у собі

- Знаходження нуля
- Оптимізація
- Інтегрування
- Звичайні диференціальні рівняння

MATLAB представляє нелінійні функції через М-файли. Наприклад, нижче наведена спрощена версія функції *humps* з директорії *MATLAB /demos*

```
function y = humps(x)
y = 1./((x-.3).^2+.01)+1./((x-.9)^2+.04)-6;
```

Наприклад, функція

```
p = fmins( 'humps', .5)
p =
```

0.6370

визначає положення мінімуму функції y , a humps(p)

ans =

11.2528

значення сааї функції у точці мінімуму.

Q = quad8('humps',0,1)

обчислює площу під графіком функції на інтервалі [0,1] за допомогою визначеного інтегралу

Q =29.8583

Керована графіка

MATLAB надає комплекс функцій низького рівня, які дозволяють створювати й обробляти лінії, поверхні й інші графічні об'єкти. Ця система називається керована графіка (Handle Graphics).

Графічні об'єкти

Графічні об'єкти – це базисні елементи системи керованої графіки в MATLAB . Вони сформовані в дерево структурної ієрархії.

Типи об'єктів керованої графіки:

- Об'єкти *Root* є вершиною ієрархії. Вони відповідають екрану комп'ютера. MATLAB автоматично їх створює спочатку сеансу роботи.
- Об'єкти *Figure* – це вікна на екрані, крім командного вікна.
- Об'єкти *Uicontrol* – це користувальницьке керування інтерфейсом. Коли користувач активує об'єкт, викликається відповідна функція. Вони містять у собі *pushbutton, radio button* і *slider.*
- Об'єкти *Axes* визначають область у вікні *Figure* і орієнтацію дочірніх об'єктів у цій області.
- Об'єкти *Uimenu* являють собою меню интерфейсу користувача, що розташовано у верхній частині вікна *Figure.*
- Об'єкти *Image* – це двовимірні об'єкти, які виводить MATLAB , використовуючи елементи прямокутного масиву як індекси в палітрі.
- Об'єкти *Line* є основними графічними базисними елементами для більшості двовимірних графіків.
- Об'єкти *Surface* – це тривимірне подання матриці, створене шляхом графічного відображення даних як висот над площиною x-y.
- Об'єкти *Text* – це рядки символів.
- Об'єкти *Light* визначають джерело світла, що діє на всі об'єкти в межах *Axes.*

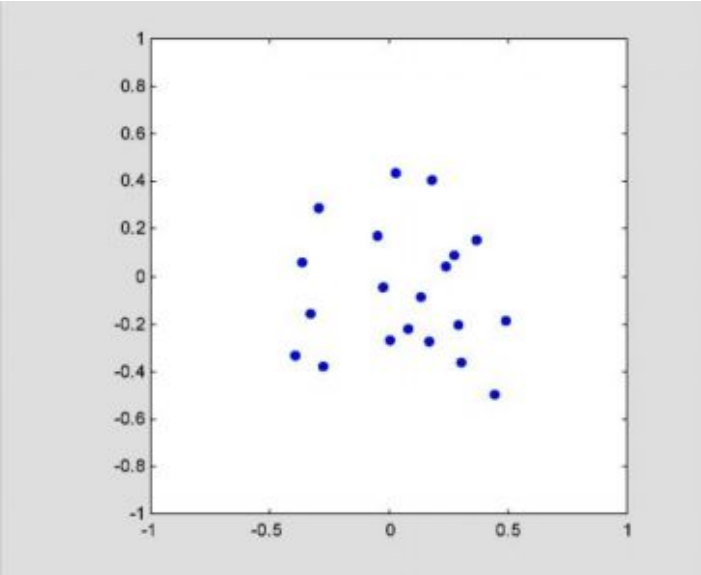
Анімація

MATLAB надає кілька способів для створення анімаційної графіки. Використання властивості *EraseMode* призначено для довгої послідовності простих графіків, де зміна від кадру до кадру мінімально.

Нижче наведено приклад, що моделює броунівський рух.
Визначимо кількість частинок n = 20;
Температуру або швидкість як s = .02;
Згенеруємо n випадкових частинок з координатами (x,y) у межах від -0.5 до 0.5

```
x = rand(n,1)-0.5; y = rand(n,1)-0.5;
h = plot(x ,y,'.');
axis([-1 1 -1 1])
axis square
grid off
set(h,'EraseMode','xor','MarkerSize',18);
while 1
x = x + s*randn(n,1);
y = y + s*randn(n,1);
   set(h,'Xdata',x,'Ydata',y)
   end
```

Одна з можливих конфігурацій стану системи броунівськи частинок, наведено у рисунку



Демонстрація
Щоб побачити додаткові приклади MATLAB, виберіть Examples and Demos з меню, або наберіть
demo
у командному рядку MATLA . Виберіть з меню приклади і додержуйтеся інструкцій на екрані.

5 ОРГАНІЗАЦІЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

Контроль поточних знань виконується на базі кредитно-модульної системи організації навчання. Підсумковим контролем є іспит.

В дисципліні „чисельне моделювання фізичних процесів" використовуються 3 змістовних модуля: 2 з теоретичної частини і 1 з практичної частини. Крім того існує окремий змістовний модуль наукової роботи.

В якості форми поточного контролю **лекційних модулів** дисципліни „ чисельне моделювання фізичних процесів " використовується проведення 1 контрольної роботи з кожного змістовного модуля та усного опитування під час практичних занять, **наукового модулю (**у разі його виконування студентом) – виступ на університетських, всеукраїнських студентських конференціях та публікація матеріалів тез доповідей цих виступів.

Критерії оцінки

Максимальна сума балів з **ЗМ-Л1** – 40 балів
Максимальна сума балів з **ЗМ-Л2** – 40 балів
Максимальна сума балів з **ЗМ-П1** – 20 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.

Оцінювання письмових відповідей проводиться у відповідності з Положенням «Про критерії оцінки знань студентів в ОДЕКУ », 2002р. та наступними роз'ясненнями:

- а)У разі використання екзаменаційних білетів у вигляді тестових завдань *відкритого типу*, відповідь на кожне питання оцінюється викладачем у відсотках до максимально можливої за критеріями, які наведенні у таблиці 1

Білет складається з 2-х питань відкритого типу

Таблиця 1

Кількісні та якісні критерії оцінки письмової відповіді на тестове завдання відкритого типу

Діапазон оцінки	Якісні критерії оцінки відповідей
90 - 100	відмінне виконання лише з незначною кількістю помилок
85 - 89	вище середнього рівня з кількома помилок
75 - 84	в загальному правильна робота з певною кількістю грубих помилок
68 - 74	непогано, але зі значною кількістю помилок
60 - 67	Відповідь в цілому достатня, що свідчить про певні знання студента з поставленого питання, але у відповіді є суттєві помилки або виявляються прогалини у знаннях з поставленного питання
35 - 59	Є окремі вірні думки, аде в цілому відповідь недостатня, або багато помилок, які формують в цілому невірну відповідь
1 - 34	Студент зовсім не відповів на питання або відповідь у більшій частині невірна.

Загальна екзаменаційна оцінка (бал успішності) у цьому випадку є арифметичною(або зваженою - за вагою питань у білеті – за розсудом викладача, але у всіх білетах методика та вагомі коефіцієнти повинні бути однаковими!) середньою з оцінок кожного питання

Шкала переходу від оцінки поточного контролю та екзамену до підсумкової оцінки

% від максимальної кількості балів	Оцінка за п’ятибальною шкалою
90 – 100 75-89 60-74 1-59	Відмінно Добре Задовільно Незадовільно

Шкала оцінювання за системою ECTS та системою університету

За шкалою ECTS	За національною системою	Визначення	За системою університету (у відсотках)
A	5 (відмінно)	відмінне виконання лише з незначною кількістю помилок	90 - 100
B	4 (добре)	вище середнього рівня з кількома помилок	85 - 89
C	4 (добре)	в загальному правильна робота з певною кількістю грубих помилок	75 - 84
D	3 (задовільно)	непогано, але зі значною кількістю помилок	68 - 74
E	3 (задовільно)	Відповідь в цілому достатня, що свідчить про певні знання студента з поставленого питання, але у відповіді є суттєві помилки або виявляються прогалини у знаннях з поставленного питання	60 - 67
FX	2 (незадовільно)	Є окремі вірні думки, але в цілому відповідь недостатня, або багато помилок, які формують в цілому невірну відповідь	35 - 59
F	2 (незадовільно)	Студент зовсім не відповів на питання або відповідь у більшій частині невірна.	1 - 34

Основна література

1. Стратонович Р.Л. Нерівноважна нелінійна термодинаміка, М., Наука,1996р.460ст
2. Хакен Г. Синергетика. М. Мир, 1980р. 482ст
3. Боголюбов Н.Н., Мітропольський Ю.А. Асимптотичні методи в теорії нелінійних коливань, М. Наука, 1974р. 336ст

Додаткова література

1. Хакен Г. Синергетика: ієрархії нестійкостей в самоорганізованих системах.280ст
2. Куні Ф.М. Статистична фізика та термодинаміка, М., Наука, 1982р. 365ст
3. Пригожин І.Р., Стенгерс І. Час, хаос,квант., М., Мир, 1983р 356ст

Методичні вказівки до самостійної роботи студентів V курсу денної форми з дисципліни “Чисельне моделювання фізичних процесів”. Спеціальність – “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	Формат	Папір. друк.
Умовн. друк. арк..	Тираж	Зам.№

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