

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА
ПОЛІТТЕХНІКА»

МЕТОДИЧНІ ВКАЗІВКИ

**до лабораторних та контрольних робіт з курсу
«Обчислювальна техніка та програмування »**

для студентів усіх форм навчання спеціальності
141 "Електроенергетика, електротехніка та електромеханіка"
Частина 2

2020

Методичні вказівки до лабораторних та контрольних робіт з курсу «Обчислювальна техніка та програмування» для студентів усіх форм навчання спеціальності 141 "Електроенергетика, електротехніка та електромеханіка. Частина 2 /Укл.: Скрупська Л.С., Каплієнко О.О., Шило С.І. - Запоріжжя: НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТТЕХНІКА», 2020.- 51 с.

Укладачі:

Л.С. Скрупська
О.О. Каплієнко
С.І. Шило

Рецензент:

В.В. Василевський

Відповідальний за випуск:

Л.С. Скрупська

Затверджено
на засіданні кафедри
ЕЕА
протокол № 5
від 27.01.2020р.
Рекомендовано до видання
НМК ЕТФ
Протокол № 7 від 20.02.2020

ЗМІСТ

Мета та зміст дисципліни	5
Лабораторна робота №1. Організація обчислень в системі	
Matlab	8
1.1 Мета роботи	8
1.2 Завдання до лабораторної роботи	8
1.3 Основні теоретичні відомості	8
1.3.1 Що таке система MatLab	8
1.3.2 Середовище MatLab	8
1.3.3 Система допомоги в Matlab	10
1.4 Відпрацювання головних правил та навиків роботи в інтерактивному режимі	10
1.5 Індивідуальні завдання до лабораторної роботи	14
1.6 Завдання на самоперевірку	16
1.7 Контрольні питання	16
Лабораторна робота №2. Оформлення результатів	
обчислень	17
2.1 Мета роботи	17
2.2 Завдання до лабораторної роботи	17
2.3 Основні теоретичні відомості	17
2.3.1. Побудова двовимірних графіків	15
2.3.2. Оформлення графіків	19
2.4 Приклади побудови двовимірних графіків	20
2.5 Індивідуальні завдання до лабораторної роботи	22
2.6 Контрольні питання	27
Лабораторна робота №3. Програмування в середо-	
вищі Matlab	28
3.1 Мета роботи	28
3.2 Завдання до лабораторної роботи	28
3.3 Основні теоретичні відомості	28
3.3.1 М-сценарії	28
3.3.2 М-функції	29
3.3.3 Введення і виведення інформації в діалоговому режимі	31
3.3.4 Організація повторення дій	32
3.4 Індивідуальні завдання до лабораторної роботи	33
3.5 Контрольні питання	34

Лабораторна робота №4. Рішення нелінійних рівнянь	
методами половинного розподілу та хорд.....	35
4.1 Мета роботи	35
4.2 Завдання до лабораторної роботи.....	35
4.3. Основні теоретичні відомості.....	35
4.3.1.Методи ітераційного уточнення коренів нелінійних	
рівнянь	35
4.3.2 Функції MatLab для вирішення нелінійних	
рівнянь	39
4.4 Індивідуальні завдання до лабораторної роботи.....	42
4.5 Контрольні питання.....	42
Лабораторна робота №5. Рішення нелінійних рівнянь методами	
Ньютона та простої ітерації	43
5.1 Мета роботи	43
5.2 Завдання до лабораторної роботи.....	43
5.3. Основні теоретичні відомості.....	43
5.4 Індивідуальні завдання до лабораторної роботи.....	49
5.5 Контрольні питання.....	49
Список літератури	50
Додаток А	51

МЕТА ТА ЗМІСТ ДИСЦИПЛІНИ

Мета викладання дисципліни:

- дисципліна повинна виховати у майбутнього фахівця математичну освіту, яка дозволить йому доводити розв'язування математичних моделей до числа, використовуючи обчислювальну техніку. Побічна користь від вивчення дисципліни в тому, що вона удосконалює загальну культуру мислення, виховує точність аргументації, що в свою чергу дає велику економію мислення при розв'язуванні конкретних прикладних задач;
- в процесі вивчення дисципліни студенти повинні зрозуміти і добре засвоїти ідеї чисельних методів, а не просто запам'ятати розрахункові формули;
- для вивчення даної дисципліни студентам необхідно знати математичний аналіз, лінійну алгебру, програмування в обсязі, передбаченому відповідними програмами для спеціальностей «Електричні апарати» та «Електромеханічне обладнання енергоємних виробництв»;
- освоєння дисципліни дасть змогу студенту використовувати ПК для виконання розрахункових робіт, курсових робіт, розробки дипломного проекту.

Задачі вивчення дисципліни

Студенти, які вивчали дисципліну **повинні вміти:**

- використовувати середовище MatLab для розв'язання інженерних задач;
- обирати потрібний чисельний метод для найкращого розв'язування прикладної задачі;
- реалізувати у вигляді програми обраний метод та алгоритм в середовищі MatLab;
- оцінити точність отриманого рішення;
- в процесі вивчення дисципліни студенти познайомляться з основними методами розв'язування систем лінійних та нелінійних рівнянь;

Зміст дисципліни

Дисципліна „Обчислювальна техніка та програмування за фахом” вивчається на 1 курсі у 2 семестрі.

Основні види занять: лекції, лабораторні заняття, консультації.

Навчальним планом передбачено вивчення слідуючих тем:

Тема 1. Програмні пакети моделювання та вирішення прикладних задач

Призначення, можливості, структура пакету MatLab. Обчислення в пакеті MatLab. Програмування в середовищі MatLab. Розширення пакету MatLab.

По даній темі студент на установчій сесії виконує під керівництвом викладача три лабораторні роботи.

Лабораторна робота №1

Організація обчислень в системі MatLab. Мета роботи: Навчитись виконувати прості обчислення функцій та матриць в системі MatLab.

Лабораторна робота №2

Оформлення результатів обчислень. Мета роботи: Навчитися будувати графіки функцій, що подають результати обчислень або експериментальних досліджень.

Лабораторна робота №3

Програмування в середовищі Matlab. Мета роботи: Придбання навичок програмування в середовищі Matlab на прикладі створення сценаріїв та функцій для розв'язання задачі відокремлювання коренів нелінійних рівнянь.

Література: [1] с. 31-69, [2] с. 20-80, [3], [4].

Тема 2. Методи розв'язування нелінійних рівнянь

Методи відокремлення дійсних коренів нелінійного рівняння. Загальна ідея ітераційних методів уточнення коренів. Метод поділу навпіл. Метод простої ітерації, метод хорд, Ньютона. Умови збіжності методів. Геометричне тлумачення.

По даній темі студент продовж семестру виконує самостійно дві лабораторні роботи (№№4, 5). Викладач консультує та перевіряє виконання цих робіт під час установлених розкладом консультацій.

Лабораторна робота № 4

Рішення нелінійних рівнянь методами половинного розподілу та хорд. Мета роботи: Вивчення чисельних методів розподілу навпіл та хорд рішення нелінійних рівнянь та придбання практичних навиків роботи в системі MatLab.

Лабораторна робота № 5

Рішення нелінійних рівнянь методами Ньютона та простих ітерацій. Мета роботи: Вивчення чисельних методів рішення нелінійних рівнянь та придбання практичних навиків роботи в системі MatLab.

Література: [1] с. 70-76, [2] с. 146-210, [3], [4].

Контрольні заходи: контрольна робота, іспит.

Студент виконує одну контрольну роботу, Яка передбачає виконання індивідуального завдання по темі №2 на комп'ютері та оформлення звіту по виконаній роботі.

ЛАБОРАТОРНА РОБОТА №1

ОРГАНІЗАЦІЯ ОБЧИСЛЕНЬ В СИСТЕМІ MATLAB

1.1 Мета роботи

Освоїти інтерфейс та основні правила роботи в інтерактивному режимі системи MatLab

1.2 Завдання до лабораторної роботи

1. Ознайомитись з основними теоретичними відомостями - розділ 1.3 методичних вказівок та розділи 1, 2, 4 електронного довідника по системі MatLab.
2. Виконати в MatLab всі приклади роботи, наведені в розділі 1.4. Протокол роботи зберегти в файлі *LaboratoryWork_1_1.txt*
3. Виконати в MatLab індивідуальне завдання наведене в розділі 1.5. Протокол роботи зберегти в файлі *LaboratoryWork_1_2.txt*
4. Для перевірки отриманих знань студент виконує завдання на самостійну роботу (розділ 1.6).

1.3 Основні теоретичні відомості

1.3.1 Що таке система MatLab

Система MatLab (Matrix Laboratory) - є інтерактивною системою для виконання інженерних і наукових розрахунків, орієнтована на роботу з масивами даних. Вона допускає написання на спеціальній мові програм, що оформляються у вигляді *м-файлів*, підтримує роботу в програмному і інтерактивному режимі з векторами і матрицями, дозволяє вирішувати системи рівнянь, будувати графіки та ін.

Система допускає використання пакетів прикладних програм (ППП) символічної математики, статистики, оптимізації, аналізу і синтезу систем управління, обробки сигналів і зображень, фінансів, картографії та ін.

Система дозволяє з легкістю обмінюватися інформацією з текстовим редактором Microsoft Word, зокрема переносити будь-які тексти і малюнки в буфер або читати текстові рядки з буфера як команди, що виконуються.

1.3.2 Середовище MatLab

Командне вікно (Command Window)

Система MatLab створена таким чином, що будь-які обчислення можна проводити в режимі прямих обчислень, тобто без підготовленої

програми. Командне вікно по суті працює як звичайний рядковий редактор, в командному рядку якого (після символів >>) в режимі діалогу можна набрати команду (оператор) або вираз і, натискуючи Enter, одержати відповідь.

Деякі команди:

clc - очищає командне вікно;

home - повертає курсор в лівий верхній кут вікна;

echo <file_name> on/off - включає/відключає режим виводу на екран тексту Script-файлу (файлу-сценарію);

more on/off - включає/відключає режим посторінкового виводу (корисний при перегляді великих m-файлів);

diary file_name.txt - веде запис на диск усіх команд у рядках введення й отриманих результатів, тобто протоколу роботи, у виді текстового файлу з зазначеним ім'ям;

diary off - призупинити запис у файл;

diary on - знову починати запис у файл;

Клавіши **↑ та ↓** - вибір раніше введених команд для повторного виконання або редагування.

Робочий простір (Workspace)

Робочий простір - це область пам'яті, що вміщує змінні, створені впродовж роботи системи MatLab. У вікні *Workspace* дозволяється переглядати, редагувати, видаляти ці змінні. Бінарні файли, у яких можуть зберігатися значення змінних робочий області, мають розширення *.mat. Для цієї ж мети можна використовувати наступні команди:

who - видає короткий список всіх змінних робочого простору;

whos - ім'я, розмір та тип масивів;

clear - очищає робочий простір;

clear змінна_1, змінна_2 [...] - для видалення вказаних змінних;

save name_file.mat- зберігає вміст робочого простору в *mat-файлі*;

save name_file.mat X Y Z - зберігає X, Y, Z в файлі.

load file.mat- завантажує змінні в робочий простір з *mat-файлу*.

Поточний каталог (Current Directory)

Пакет MatLab використовує поняття поточного каталогу при роботі з файлами під час сеансу роботи. Вікно *Current Directory* до-

зволяє встановлювати поточний каталог, переглядати, редагувати, видаляти та запускати файли цього каталогу на виконання.

Вікно Command History

Містить протокол роботи. Використовується для виконання раніше введених команд.

Launch Pad

Окрім середовища в якому можна проводити обчислення, MatLab включає цілий ряд спеціальних програм, пов'язаних з рішенням задач окремих класів - вони створюють інструментарій MatLab. Список тематичних підкаталогів цього інструментарію і відображається у вікні Launch Pads.

1.3.3 Система допомоги в Matlab

Всі команди та функції MatLab організовані в логічні групи і структура директорій MatLab базується на цьому групуванні. Команда *help* допомагає орієнтуватися в цій структурі та виводити на екран короткий/повний опис по кожній команді чи функції, та навіть відповідні тексти програмних кодів:

help - виводить список директорій системи;

help <директорія> - видає список команд директорії;

help <команда> - видає опис команди;

type <команда> - видає текст команди;

type <назва т.файлу> - видає текст програми користувача, якщо він складений в термінах MatLab.

1.4 Відпрацювання головних правил та навиків роботи в інтерактивному режимі

Змінні можуть бути числовими, текстовими і інших типів. Розглянемо числові і текстові змінні. Назва змінної починається з латинської букви, далі можуть бути букви і цифри (не більше 31 символу). Рядкові і прописні букви тут розрізняються. Не можна допускати збіги імені змінної з ім'ям якої-небудь команди.

Числові змінні - це числа, вектори, матриці і багатовимірні масиви. В комп'ютері всі числа представлені приблизно з 16 десятковими знаками, під кожне дійсне число відводиться 8 байтів, під комплексне - 16.

Цілі числа

В системі вони не виділяються явно. Наберемо і виконаємо окремо кожну команду:

$a=2$ $a=2.0$ $a=2;$ $a=1:6$ $b=1:20$ $c=10:-2:5$

Дійсні числа

Виконаємо окремо наступні команди:

$d=0.5:0.3:2.5$ $d=.5:.3:2.5$ $d=.5+1:.3-.1:2.5*2$ $\text{length}(d)$

$d(\text{end})$ $d(\text{end}-2)$ $d(1)$ $d(0)$ $d(2:7)$ $d(7:-1:2)$ $d(150)$

$f=\text{linspace}(1.5,30,143);$ $\text{length}(f)$

Діапазон дійсних чисел зберігається в константах: realmax
 realmin

Інші константи системи MatLab: π i j eps

Комплексні числа

$q=1+2*i$ $q=1+2i$ $\text{real}(q)$ $\text{imag}(q)$ $\text{abs}(q)$ $\text{conj}(q)$ $s=\text{angle}(q)$

тут $-p < s <= p$.

$q=1+2*i; r=3; fi=0:.01:\pi; z=q+r*\exp(i*fi); \text{plot}(z)$ - це верхнє пі-
вколо.

Введення векторів

Вектори-рядки: $a=1:6$ $\text{linspace}(1,6,10)$

Вектори-стовпці: $a=(1:6)'$ $\text{linspace}(1,6,10)'$

Оператори $.$ та $'$: $y1=\text{linspace}(1,6,4)'; y2=y1; y=y1+i*y2; y'$
 $y.'$

Введення матриць

$A(i,j)$ - елемент i -го рядка і j -го стовпця.

$A(k)$ - k -ий елемент таблиці, витягнутої в стовпець.

Індекси (i,j) завжди починаються із значення 1. Команди набираються на малому латинському регістрі.

Найпростішим способом введення матриці в середовищі пакета є набір всіх її елементів з клавіатури. Елементи матриці вводяться за рядками, які відділяють один від одного символом ";" або переходом на новий рядок. Окремі елементи в рядках відділяються пробілами або комами.

Деякі операції з матрицями з використанням оператора ":"

$A(:,j)$ виведення j -ї колонки матриці A

$A(:,j:k)$ виведення колонок $A(j), A(j+1), \dots, A(k)$ матриці A

$A(i,:)$ виведення i -го рядка матриці A

$A(i:k,:)$ виведення рядків $A(i), A(i+1), \dots, A(k)$ матриці A

$A(:)$ виведення всіх елементів матриці в колонку

$A(j:k)$ виведення в рядок елементів матриці A , почавши від елемента з індексом j до елемента з індексом k

Ще одним спеціальним типом матриці, що може існувати в робочій області є порожня матриця. Розмірність такої матриці - 0×0 і визначається вона так $A = []$.

Звичайно пустий масив (яким є така матриця) використовують для того, щоб видалити рядки або стовпці матриць.

Виконаємо окремо наступні команди:

$A = [1,2;3,4]$ $A = [1;2,3,4]$ $A(2,2)$ $A(3)$ $A(5)$ $size(A)$ $A(3,4)=10$
 $size(A)$

$A(5)=6$ $size(A)$ $A(22)=3$ $A=A(:)$ $A(22)=3$ $size(A)$
 $[m,n]=size(A)$

$A=reshape(1:24,4,6)$ $size(A)$ $A([1,end],:)=[]$ $A(:,[1,end])=[]$
 $size(A)$

Деякі спеціальні матриці

zeros - матриця, всі елементи якої дорівнюють 0;

ones - матриця, всі елементи якої дорівнюють 1;

eye - одинична матриця;

rand - матриця випадкових чисел.

Виконаємо наступні команди:

$m=3;n=4;$ $eye(m,n)$ $eye(m)$ $eye(n)$ $ones(m,n)$ $ones(m)$ $ones(n)$
 $zeros(m,n)$

$rand(m,n)$ $rand(m,n)$ $rand('state',0)$ $rand(m,n)$ $rand(m)$ - це рівномірний розподіл на інтервалі (0, 1).

$randn(m,n)$ $randn('state',100)$ - це нормальний розподіл, у нього математичне очікування=0, дисперсія=1.

Деякі елементарні матричні функції та операції трансформування і декомпозиції матриць

<i>expm</i>	- матрична експонента
<i>Logm</i>	- матричний логарифм
<i>sqrtm</i>	- матричний корінь
<i>norm</i>	- норма вектора і матриці
<i>rank</i>	- ранг матриці
<i>det</i>	- детермінант матриці
<i>trace</i>	- сума діагональних елементів матриці
<i>inv</i>	- знаходження оберненої матриці
<i>size</i>	- розмір матриці

reshape - зміна розміру матриці
length - довжина вектора
diag - діагональні елементи матриці
triu - верхня трикутна частина матриці
tril - нижня трикутна частина матриці

Виконаємо наступні команди:

```
A=reshape(1:24,4,6)   triu(A)   triu(A,0)   triu(A,2)   triu(A, -1)
tril(A)
```

```
v=1:5 diag(v) diag(v,2) diag(v, -1)
```

```
diag(A) diag(A,2) diag(A, -1)
```

```
A=reshape(1:24,4,6) rot90(A) rot90(A,2)
```

Деякі базові матричні операції

A'	- транспонування матриці
A+B	- додавання (віднімання) матриць
A*B	- добуток матриць
A/B	- праве ділення матриць
A\B	- ліве ділення матриць
A^p	- піднесення матриці до степеня
a^P	- піднесення скаляра до матричного степеня
A.*B	- поелементний додаток матриць
A./B	- поелементне ділення матриць
A.^p	- поелементне піднесення матриці до степеня

Виконаємо наступні команди:

```
m=3;n=4; a=rand(m,n) b=rand(m,n) c=rand(1,n)'
```

```
p1=a*b p2=a.*b p3=a*c p4=a.*c
```

Текстові змінні

Вони беруться в лапки (на букві “є” на латинському регістрі), символ займає 2 байти. Виконаємо наступні команди:

```
>>t='Аргумент'; f='Функція';
```

```
>>x=[pi:pi/180:pi];y=sin(x);
```

```
>>disp([t, ' ', f]),disp([x', y'])
```

Формати виводу чисел в пакеті MatLab

format short	- з фіксованою точкою та 4 знаками після точки;
format long	- з фіксованою точкою та 14 знаками після точки;

- format short e** - наукова нотація з 4 десятинними знаками;
format long e - наукова нотація з 15 десятинними знаками;
format short g - зручний для виводу цілих. Дійсні числа виводяться з фіксованою точкою та 5 знаками після точки;
format long g - зручний для цілих чисел. Дійсні числа виводяться в науковій нотації з 13 десятинними знаками.

Опція `format short e` дозволяє одержувати рівні стовпці.

1.5 Індивідуальні завдання до лабораторної роботи

Для заданої функції $F(a, b, c)$ і значень параметрів a і b в таблиці 1.1 виконати в системі MatLab:

- ввести значення c - вектору аргументів;
- зберегти значення a, b, c у `mat`-файлі;
- очистити вікна Command Window, Command History, Workspace, завантажити данні з `mat`-файлу, обчислити функцію $F(a, b, c)$, згідно завдання варіанта;
- результуючі значення вектору аргументів c та значень функції $F(a, b, c)$ вивести в вигляді вертикально розташованої таблиці;
- повторити вивід цієї таблиці задаючи різні формати даних;
- провести запис на диск фрагменту роботи. Створити текстові коментарі до роботи в отриманному файлі.

Таблиця 1.1 – Варіанти завдань

№	$F(a, b, c)$	a	b
1	$\frac{ab}{\sqrt[3]{c}}(a+b)\sin(3c)$	2456	0.00078
2	$\frac{a+b}{a-b}\arcsin(c)$	0.02456	0.007823
3	$\left[\frac{(a+b)c}{a-b}\right]^3 \ln(1+c)$	0.2456	0.20078
4	$\frac{c^3}{13}(a-b)^7 \cos(ac)$	0.02456	0.007823

5	$\frac{ab}{\sqrt[3]{c}}(a^3 + b) \sin^2(c)$	0.12456	0.0078
6	$\frac{a+b}{a-b} \arctg(c)$	0.02456	0.007823
7	$\left[\frac{(a+b)c}{a-b^2} \right]^3 \ln(1 + c^2)$	0.2456	0.20078
8	$\frac{c^2}{13}(a-b)^3 \cos(ac^2)$	0.02456	0.007823
9	$\frac{ab}{\sqrt[3]{c}}(a+b)^2 \sin(c)$	0.12456	0.078
10	$\frac{c^3}{13}(a-b)^3 \cos(a^2c)$	0.02456	0.007823
11	$\left[\frac{(a+b)c}{a-b} \right]^2 \ln(1 + c)$	0.2456	0.20078
12	$\frac{a+b^2}{a-b} \arccos(c)$	0.02456	0.007823
13	$\frac{a^2b}{\sqrt[3]{c}}(1+b) \sin(2c)$	2456	0.00078
14	$\frac{a+b}{a-b} \ln(1 + ac)$	0.02456	0.007823
15	$\frac{ab}{\sqrt[3]{1+c}}(a+b) \sin(c)$	0.2456	0.20078
16	$\frac{c^3}{13}(a-b)^7 \cos(ac)$	0.02456	0.007823
17	$\frac{ab}{\sqrt[3]{1+c}}(a+b) \sin(c)$	0.12456	0.0078
18	$\frac{a+b}{a-b} \arctg(ac)$	0.02456	0.007823
19	$\left[\frac{(a+b^2)c}{a-b^2} \right]^2 \ln(1 + ac^2)$	0.2556	0.50078

20	$\frac{c^2}{13}(a-b)\cos(1+ac^2)$	0.02456	0.007823
----	-----------------------------------	---------	----------

1.6 Завдання на самоперевірку

Виконайте наступні команди та прокоментуйте отриманні результати:

1. `>>a=[1 2 3; 4 5 6;7 8 9]`
2. `>> b1=[1; 2; 3]`
3. `>>c=[a b1]`
4. `>> d=zeros(3,4)`
5. `>> e=[1 2 3; 4 5 ]`
6. `>>b=[a(1,2:end) a(3,end-1:-1:1)]`
7. `>> b2=[a(1,2:end); a(3,end-1:-1:1)]`
8. `>> a[:,2]=[]`

1.7 Контрольні питання

1. Перелічити тематичні підкаталоги MatLab (toolbox) та їх призначення.
2. Робочий стіл пакету MatLab (меню, інструменти).
3. Типи та формати даних; системні змінні.
4. Елементарні математичні функції пакету.
5. Основні команди керування командного режиму.
6. Призначення та правила використання команди **help**.

ЛАБОРАТОРНА РОБОТА № 2 ОФОРМЛЕННЯ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ

2.1 Мета роботи

Навчитися будувати графіки функцій, що подають результати обчислень або експериментальних досліджень

2.2 Завдання до лабораторної роботи

1. Ознайомитись з основними теоретичними відомостями - розділ 2.3 методичних вказівок та розділи 1, 2, 4 електронного довідника.

2. Ознайомитись з основними теоретичними відомостями методичних вказівок - розділ 2.3 та розділ 9 електронного довідника по системі MatLab.. Виконати наведенні приклади.

3. Виконати в MatLab приклади побудови двовимірних графіків - розділ 2.4.

4. Виконати в MatLab індивідуальне завдання - розділ 2.5.

5. Створити текстовий файл - протокол виконання індивідуального завдання.

2.3 Основні теоретичні відомості

2.3.1 Побудова двовимірних графіків

Основні команди побудови двовимірних графіків:

plot(x,y)	- побудова графіка в декартовій системі координат;
loglog(x,y)	- побудова графіка в логарифмічних осях;
semilogx(x,y)	- побудова графіка з логарифмічною віссю абсцис;
semilogy(x,y)	- побудова графіка з логарифмічною віссю ординат;
polar(φ,r)	- побудова графіка в полярній системі координат;
plotyy(x,y,x,f)	- забезпечує вивід осей як ліворуч, так і праворуч, для кожної функції y та f окремо;
fplot(<nameFun>,limits)	- побудова графіка функції в інтервалі.

Функція **plot**

plot(x) - побудова графіка залежності значення елемента вектора від його порядкового номера.

plot(x,y) - будує графік залежності $y=f(x)$.

plot(x1, y1, x2, y2, x3, y3) - на одному графіку виводиться декілька залежностей.

plot(x, y, 'color_style_marker') - вказівкою рядка *color_style_marker* (до 3 символів), ідентифікують колір, тип лінії та маркер.

Таблиця 2.1 – Символи, що ідентифікують колір, тип лінії та маркер

<i>Колір</i>		<i>Тип лінії</i>		<i>Тип маркера</i>
c	cyan(голубий)	—	неперервна	+
k	black(чорний)	--	штрихова	o
y	yellow(жовтий)	:	пунктирна	*
r	red(червоний)	-.	штрих пунктирна	x
g	green(зелений)	none	відсутність лінії	.
b	blue(синій)			s -квадрат
w	white(білий)			<, >, ^ - трикутники

Наприклад:

plot(x, y, 'y:+') - виведе залежність **y(x)** жовтою пунктирною лінією, на якій кожна пара даних буде відмічена маркером +.

У випадку, якщо визначено тип маркера, але не визначено стиль лінії, на екран будуть виведені тільки точки залежності, задані у вхідних векторах.

Функція **fplot**

fplot(<ім'я функції>, limits) - будує графік функції (функцій) в інтервалі $limits=[xmin, xmax]$.

<ім'я функції> - це ім'я М-файлу в вигляді:

@ MyFun або 'MyFun'

або рядок типу:

```
'sin(x)',  
[sin(x) cos(x) ]',  
[sin(x), myfun1(x), myfun2(x)]'.
```

limits=[xmin xmax ymin ymax] - встановлює інтервали для
для значень аргументу та функції.

Наприклад:

```
fplot(@humps,[0 1]);  
fplot('humps',[0 1]);  
f = inline('abs(exp(-j*x*(0:9)))*ones(10,1))');  
fplot(f,[0 2*pi]);  
fplot('[tan(x),sin(x),cos(x)]',2*pi*[-1 1 -1 1]);  
fplot('sin(1 ./ x)', [0.01 0.1],1e-3).
```

Декілька допоміжних команд

figure – команда для відкриття нового вікна для виводу графіків.

figure(n) – для вибору поточного вікна серед відкритих графічних вікон, де *n* - номер вікна.

hold on - для виводу нових залежностей у поточному вікні графіка. При цьому MatLab не знищить створений вже графік, а тільки, якщо це необхідно, змінить параметри осей.

subplot (vha) - пакет дозволяє здійснювати поділ графічного вікна на декілька підграфіків зі своїми осями. Такий поділ з одночасним вибором активного вікна здійснюється за допомогою наведеної команди, в якій *v* та *h* – число під графіків по вертикалі та горизонталі, а параметр *a* визначає номер активного вікна.

Наприклад, команда

```
subplot (221), plot (x,y)
```

створить 4 вікна та виведе графік у перше вікно верхньої половини екрана.

2.3.2 Оформлення графіків

Для оформлення графіків служать команди:

axis ([xmin xmax ymin ymax]) - дає змогу користувачеві самому задавати діапазон координат;

axis equal - формує однаковий крок по *x*- та *y*-осям;

axis off - дозволяє не виводити градуїровку координатних осей;

grid on/off- дозволяє виводити/не виводити сітку на графік;

xlabel ('назва'), ylabel ('назва') - назви x-у-осей, відповідно;

title ('назва') – назва графіка;

text (x, y, 'текст') - зробити підпис в будь-якому місці графіка, тут x, y координати початку напису.

2.4 Приклади побудови двовимірних графіків

Приклад 1. Побудови характеристики неробочого ходу генератора.

Маємо результати експерименту:

Ток(I _m):	x	Напруга(U):	y
1.5		80	
2.13		100	
2.6		120	
3.09		140	
3.6		160	
4.15		180	
4.52		190	
4.78		200	
5.2		212	
5.59		220	
6.77		240	
7.74		252	
8.55		260	
9.13		264	
10.58		276	

Функція, що апроксимує ці експериментальні дані має вигляд:

$y1 = 239.984 * \text{atan}(0.222 * x)$;

Команди MatLab для побудови графіка характеристики неробочого ходу генератора (експериментальних даних та апроксимуючої функції):

```
>>y1=239.984*atan(0.222*x)
>>plot (x, y, 'ko', x, y1, 'k: ')
>>grid
>>axis ([0 11 0 286])
>>ylabel ('U, [B] ')
>>xlabel ('Im, [A] ')
>>text (3,60,'o - експериментальні дані')
```

```
>>text(3,40,'--- - U=239.984arctg(0.2221Im)')
>>title('Апроксимація характеристики неробочого ходу генератора')
```

Результат роботи поданої програми показано на рис.2.1.

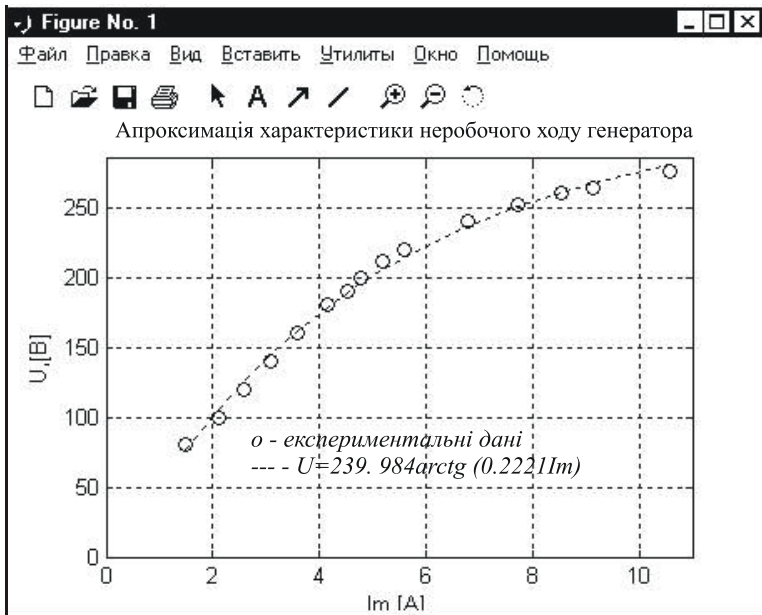


Рисунок 2.1- Вигляд графічного вікна після виконання команд програми

Використання розглянутих вище функцій для побудови та оформлення графіків є зручними при складанні m-файлів. Для оформлення простих графіків можна використовувати панель інструментів графічного вікна, яку рекомендується освоювати самостійно.

Приклад 2. Графік функцій, заданих параметрично
 $x(t) = 0.5 \sin(t)$, $y(t) = 0.7 \cos(t)$ для $t \in [0, 2\pi]$ (elinc)

будується при допомозі слідуючих команд:

```
>>t=[0:0.01:2*pi];
>>x=0.5*sin(t);
>>y=0.7*cos(t);
>>plot(x,y)
```

Приклад 3. Графік кусочно-заданої функції

$$y(x) = \begin{cases} \pi \cdot \sin(x), & -2\pi \leq x \leq -\pi, \\ \pi - |x|, & -\pi < x < \pi, \\ \pi \cdot \sin^3 x, & \pi \leq x \leq 2\pi. \end{cases}$$

будується при допомозі слідуючих команд:

```
>>x1=[-2*pi:0.01:-pi];
>>y1=pi*sin(x1);
>>x2=[-pi:0.01:pi];
>>y2=pi-abs(x2);
>>x3=[pi:0.01:2*pi];
>>y3=pi*sin(x3).^3;
>>x=[x1 x2 x3];
>>y=[y1 y2 y3];
>>plot(x,y)
```

Можливо побудувати й по іншому – як три різні функції:

```
>>plot(x1,y1,x2,y2,x3,y3)
```

2.5 Індивідуальні завдання до лабораторної роботи

За допомогою двох функцій Matab, **plot()** та **fplot()**, побудувати двовимірні графіки згідно індивідуального завдання представленого в табл.2.2 :

1. Графіки функцій **y(x)** та **g(x)**(наведені в 1 стовпчику таблиці) вивести:

- в одному графічному вікні;
- в різних графічних вікнах.
- в одному графічному вікні, в різних підграфіках зі своїми осями.

Побудову кусочно-заданої функції g(x) провести різними варіантами.

2. В 2 стовпчику таблиці наведені параметрично задані функції **y(x)** та **z(x)**, де параметр **x** приймає значення від **0** до **2 π**. Побудувати графік залежності функції **z(x)** від **y(x)** в новому графічному вікні.

3. Оформити графіки використовуючи:

- команди MatLab;
- можливості графічного редактора.

Таблиця 2.2 – Індивідуальні завдання

№	Для $x \in [-2, 2]$	Для $x \in [0, 2\pi]$
	1	2
1	$y = \sin(x)e^{-2x},$ $g = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, x \leq 0, \\ 2x + \frac{\sin^2(x)}{2+x}, x > 0 \end{cases}$	$y = 2\sin(x) \cos(x),$ $z = 3\cos^2(2x) \sin(x)$
2	$y = \frac{1+x^2}{1+2x^2},$ $g = \begin{cases} 3\sin(x) - \cos^2(x), x \leq 0, \\ 3\sqrt{1+x^2}, x > 0 \end{cases}$	$y = 2\sin(\pi x) - 3\cos(\pi x),$ $z = \cos^2(2\pi x) \sin(\pi x)$
3	$y = \frac{2+\sin^2(x)}{1+x^2},$ $g = \begin{cases} \frac{3x^2}{1+x^2}, x \leq 0, \\ \sqrt{1+\frac{2x}{1+x^2}}, x > 0 \end{cases}$	$y = 5\sin(\pi x) - \cos(3\pi x),$ $z = \cos(2\pi x) - 2\sin^3(\pi x)$
4	$y = \frac{2+\cos(x)}{1+e^{2x}},$ $g = \begin{cases} \frac{3+\sin^2(2x)}{1+\cos^2(x)}, x \leq 0, \\ 2\sqrt{1+2x}, x > 0 \end{cases}$	$y = 3\sin(2\pi x)\cos(\pi x) - \cos^2(3\pi x),$ $z = 2\cos^2(2\pi x) - 3\sin(3\pi x)$
5	$y = \sqrt[4]{1+e^{3x}},$ $g = \begin{cases} \frac{3+\sin(x)}{1+x^2}, x \leq 0, \\ 2x^2 \cos^2(x), x > 0 \end{cases}$	$y = 2\sin(\pi x)\cos(\pi x),$ $z = \cos^2(\pi x)\sin(3\pi x)$

6	$y = \frac{2+3x}{1+x+x^2},$ $g = \begin{cases} \sqrt{1+2x^2-\sin^2(x)}, x \leq 0, \\ \frac{2+x}{\sqrt[3]{2+e^{-0.1x}}}, x > 0 \end{cases}$	$y=3\sin(3\pi x)\cos(2\pi x),$ $z=\cos^3(4\pi x)\sin(\pi x)$
7	$y = \frac{1+x}{1+\sqrt{2+x+x^2}},$ $g = \begin{cases} \frac{1+x}{1+\cos^2(x)}, x > 0, \\ \sqrt{1+x^2}, x \leq 0 \end{cases}$	$y=2\sin(2\pi x)\cos(4\pi x),$ $z=\cos^2(3\pi x)-\cos(\pi x)\sin(\pi x)$
8	$y = \frac{1+xe^{-x}}{2+\sqrt{x^2+\sin^2(x)}},$ $g = \begin{cases} \frac{1+3x}{2+\sqrt[3]{1+x}}, x > 0, \\ \sqrt{1+ x }, x \leq 0 \end{cases}$	$y=\sin(3\pi x)+2\sin(2\pi x)\cos(3\pi x),$ $z=\cos(\pi x)-\cos(3\pi x)\sin^2(\pi x)$
9	$y = \frac{1+x}{1+\sqrt{2+x+x^2}},$ $g = \begin{cases} \frac{1+x}{1+\cos^2(x)}, x > 0, \\ \sqrt{1+x^2}, x \leq 0 \end{cases}$	$y=2\sin(2\pi x)\cos(4\pi x),$ $z=\cos^2(3\pi x)-\cos(\pi x)\sin(\pi x)$
10	$y = \frac{1+xe^{-x}}{2+x^2}\sin^2(x),$ $g = \begin{cases} \frac{1+x}{2+\cos^3(x)}, x > 0, \\ \frac{\sqrt{1+ x }}{2+ x }, x \leq 0 \end{cases}$	$y=\sin(\pi x)\cos(3\pi x)+$ $+2\sin(3\pi x)\cos(2\pi x),$ $z=\cos^2(\pi x)-\cos(3\pi x)$

11	$y = \sqrt[4]{1 + e^{3x}},$ $g = \begin{cases} \frac{3 + \sin(x)}{1 + x^2}, x \leq 0, \\ 2x^2 \cos^2(x), x > 0 \end{cases}$	$y = 2\sin(\pi x)\cos(\pi x),$ $z = \cos^2(\pi x)\sin(3\pi x)$
12	$y = \frac{2 + 3x}{1 + x + x^2},$ $g = \begin{cases} \sqrt{1 + 2x^2 - \sin^2(x)}, x \leq 0, \\ \frac{2 + x}{\sqrt[3]{2 + e^{-0.1x}}}, x > 0 \end{cases}$	$y = 3\sin(3\pi x)\cos(2\pi x),$ $z = \cos^3(4\pi x)\sin(\pi x)$
13	$y = \sin(x)e^{-2x},$ $g = \begin{cases} \frac{1 + x^2}{\sqrt{1 + x^4}}, x \leq 0, \\ 2x + \frac{\sin^2(x)}{2 + x}, x > 0 \end{cases}$	$y = 2\sin(x) \cos(x),$ $z = 3\cos^2(2x) \sin(x)$
14	$y = \frac{1 + x^2}{1 + 2x^2},$ $g = \begin{cases} 3\sin(x) - \cos^2(x), x \leq 0, \\ 3\sqrt{1 + x^2}, x > 0 \end{cases}$	$y = 2\sin(\pi x) - 3\cos(\pi x),$ $z = \cos^2(2\pi x) \sin(\pi x)$
15	$y = \frac{2 + \sin^2(x)}{1 + x^2},$ $g = \begin{cases} \frac{3x^2}{1 + x^2}, x \leq 0, \\ \sqrt{1 + \frac{2x}{1 + x^2}}, x > 0 \end{cases}$	$y = 5\sin(\pi x) - \cos(3\pi x),$ $z = \cos(2\pi x) - 2\sin^3(\pi x)$
16	$y = \frac{2 + \cos(x)}{1 + e^{2x}},$	$y = 3\sin(2\pi x)\cos(\pi x) - \cos^2(3\pi x),$ $z = 2\cos^2(2\pi x) - 3\sin(3\pi x)$

	$g = \begin{cases} \frac{3 + \sin^2(2x)}{1 + \cos^2(x)}, x \leq 0, \\ 2\sqrt{1 + 2x}, x > 0 \end{cases}$	
17	$y = \sqrt[4]{1 + e^{3x}},$ $g = \begin{cases} \frac{3 + \sin(x)}{1 + x^2}, x \leq 0, \\ 2x^2 \cos^2(x), x > 0 \end{cases}$	$y = 2\sin(\pi x)\cos(\pi x),$ $z = \cos^2(\pi x)\sin(3\pi x)$
18	$y = \frac{2 + 3x}{1 + x + x^2},$ $g = \begin{cases} \sqrt{1 + 2x^2 - \sin^2(x)}, x \leq 0, \\ \frac{2 + x}{\sqrt[3]{2 + e^{-0.1x}}}, x > 0 \end{cases}$	$y = 3\sin(3\pi x)\cos(2\pi x),$ $z = \cos^3(4\pi x)\sin(\pi x)$
19	$y = \frac{1 + x}{1 + \sqrt{2 + x + x^2}},$ $g = \begin{cases} \frac{1 + x}{1 + \cos^2(x)}, x > 0, \\ \sqrt{1 + x^2}, x \leq 0 \end{cases}$	$y = 2\sin(2\pi x)\cos(4\pi x),$ $z = \cos^2(3\pi x) - \cos(\pi x)\sin(\pi x)$
20	$y = \frac{1 + xe^{-x}}{2 + \sqrt{x^2 + \sin^2(x)}},$ $g = \begin{cases} \frac{1 + 3x}{2 + \sqrt[3]{1 + x}}, x > 0, \\ \sqrt{1 + x }, x \leq 0 \end{cases}$	$y = \sin(3\pi x) + 2\sin(2\pi x)\cos(3\pi x),$ $z = \cos(\pi x) - \cos(3\pi x)\sin^2(\pi x)$
21	$y = \frac{1 + x}{1 + \sqrt{2 + x + x^2}},$ $g = \begin{cases} \frac{1 + x}{1 + \cos^2(x)}, x > 0, \\ \sqrt{1 + x^2}, x \leq 0 \end{cases}$	$y = 2\sin(2\pi x)\cos(4\pi x),$ $z = \cos^2(3\pi x) - \cos(\pi x)\sin(\pi x)$

22	$y = \frac{1 + xe^{-x}}{2 + x^2} \sin^2(x),$ $g = \begin{cases} \frac{1 + x}{2 + \cos^3(x)}, x > 0, \\ \frac{\sqrt{1 + x }}{2 + x }, x \leq 0 \end{cases}$	$y = \sin(\pi x) \cos(3\pi x) + 2\sin(3\pi x) \cos(2\pi x),$ $z = \cos^2(\pi x) - \cos(3\pi x)$
23	$y = \sqrt[4]{1 + e^{3x}},$ $g = \begin{cases} \frac{3 + \sin(x)}{1 + x^2}, x \leq 0, \\ 2x^2 \cos^2(x), x > 0 \end{cases}$	$y = 2\sin(\pi x) \cos(\pi x),$ $z = \cos^2(\pi x) \sin(3\pi x)$
24	$y = \frac{2 + 3x}{1 + x + x^2},$ $g = \begin{cases} \sqrt{1 + 2x^2 - \sin^2(x)}, x \leq 0, \\ \frac{2 + x}{\sqrt[3]{2 + e^{-0.1x}}}, x > 0 \end{cases}$	$y = 3\sin(3\pi x) \cos(2\pi x),$ $z = \cos^3(4\pi x) \sin(\pi x)$
25	$y = \sin(x) e^{-2x},$ $g = \begin{cases} \frac{1 + x^2}{\sqrt{1 + x^4}}, x \leq 0, \\ 2x + \frac{\sin^2(x)}{2 + x}, x > 0 \end{cases}$	$y = 2\sin(x) \cos(x),$ $z = 3\cos^2(2x) \sin(x)$

2.5 Контрольні питання

1. Формат функції PLOT.
2. Формат функції FPLOT.
3. Загальні команди побудови графіків функцій однієї змінної.
4. Зміна властивостей ліній графіка, оформлення графіків.
5. Робота з декількома графіками

ЛАБОРАТОРНА РОБОТА №3 ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ MATLAB

3.1 Мета роботи

Придбання навичок програмування в середовищі Matlab

3.2 Завдання до лабораторної роботи

1. Для виконання лабораторної роботи необхідно пропрацювати теоретичний матеріал представлений в 3-ому розділі електронного довідника за системою MatLab.
2. Познайомитися із змістом пункту 3.3 методичних вказівок і виконати приклади цього пункту.
3. Виконати індивідуальне завдання, варіанти приведені в таблиці 3.2
4. У звіті привести тексти програм, створених вами для виконання індивідуального завдання а також протокол запуску на виконання, результат роботи програм у вигляді сценарію (файлу M-script).

3.3 Основні теоретичні відомості

Файли, які містять код язику Matlab, називаються m-файлами. М-файли є звичними текстовими файлами, які створюються за допомогою текстового редактора. Для операційного середовища персонального комп'ютера система Matlab підтримує спеціально вбудований редактор.

Відкрити редактор можна двома способами:

- з меню *File* обрати команду *New*, а потім *M-File*.
- використати команду редагування *edit*.

Існує два типи М-файлов: М-сценарії(M-Script) і М-функції(M-Function) з наступними характеристиками.

3.3.1 М-сценарії

Сценарії є найпростішим типом М-файлу - у них немає вхідних і вихідних аргументів. Вони використовуються для автоматизації обчислень, які виконуються багато раз. Сценарії оперують даними з робочої області і можуть генерувати нові дані для подальшої обробки в цьому ж файлі. Дані, які використовуються в сценарії, зберігаються в

робочій області після завершення сценарію і можуть бути використані для подальших обчислень.

Приклад 3.1

Наступні оператори обчислюють радіус-вектор ρ для різних тригонометричних функцій від кута θ і будують послідовність графіків в полярних координатах.

Рядки коментаря	% M-file petals - сценарій побудови пелюсткового графіку
Обчислення	theta = -pi:0.01:pi; rho(1, :) = 2*sin(5*theta).^2; rho(2, :) = cos(10*theta).^3; rho(3, :) = sin(theta).^2; rho(4, :) = 5*cos(3.5*theta).^3;
Команди графічного виводу	for i = 1:4 polar (theta, rho(i, :)) pause end

Створіть М-файл `petals.m`, вводячи вказані вище оператори. Цей файл є сценарієм. Введення команди `petals.m` в командному рядку системи MatLab викликає виконання операторів цього сценарію.

Після того, як сценарій відобразить перший графік, натисніть клавішу Return, щоб перейти до наступного графіка. В сценарії відсутні вхідні і вихідні аргументи; програма `petals.m` сама створює змінні, які зберігаються в робочій області системи MatLab. Коли виконання завершено, змінні (i , θ і ρ) залишаються в робочій області.

3.3.2 М-функції

М-функції є М-файлами, які допускають наявність вхідних і вихідних аргументів. Вони працюють із змінними в межах власної робочої області, відмінної від робочої області системи MATLAB. Як вхідні аргументи можуть бути задані імена функцій.

Приклад 3.2

Функція `average` обчислює середнє значення елементів вектора:

function y = average (x)

% AVERAGE Середнє значення елементів вектора.

% AVERAGE(X), де X - вектор. Обчислює середнє значення

*% елементів вектора. Якщо вхідний аргумент не є векто-
% ром, генерується помилка.*

```
[m,n] = size(x);
if ~(m == 1) | (n == 1) | (m == 1 & n == 1)
    error('Вхідний масив повинен бути вектором')
end
y = sum(x)/length(x);    % Власне обчислення
```

Створіть запропоновану функцію і збережіть її у файлі average.m. Функція average допускає єдиний вхідний і єдиний вихідний аргументи. Для того, щоб викликати функцію average, треба ввести наступні оператори:

```
>> z = 1:99;
>> average(z)
ans = 50
```

Приклад 3.3. Розглянемо можливість MatLab передавати ім'я функції як параметр на прикладі функції feval.

Функція feval('fun',c) або feval(@fun,c) служить для обчислення в точці x=c значення функції, опис якої знаходиться у відповідному файлі fun.m.

Якщо ж ім'я файлу fun.m зберігається в рядковій змінній, наприклад - f, то оператор обчислення значення функції записується таким чином: feval(f,c).

Наприклад потрібно знайти, використовуючи пакет MatLab, методом половинного розподілу корені рівняння $x^4 - 11x^3 + x^2 + x + 0.1 = 0$:

1. Створіть файл Fun1.m, що містить опис функції
 $y(x) = x^4 - 11x^3 + x^2 + x + 0.1$.

Лістинг файлу Fun1.m:

```
function z=Fun1(x)
z=x.^4-11*x^3+x+0.1;
```

2. Створіть файл CutAndTry.m, що містить опис функції, яка повертає значення коріння рівняння $f(x)=0$ методом половинного розподілу.

Лістинг файлу CutAndTry.m

```
function z=CutAndTry(f, a,b,eps)
%f -
% a -
%
```

```

% b -
%
% eps -
while abs(a-b)>eps
    c=(a+b) /2;
    if feval(f,c)*feval(f,a) <0
        b=c;
    else
        a=c;
    end;
z=c;

```

3. Побудуйте графік функції на інтервалі $[-1, 1]$, виконавши в командному вікні пакету MatLab наступну послідовність операторів:

```

>>x1=-1;
>>x2=1;
>>dx=0.1e-2;
>>x=x1:dx:x2;
>>plot(x,fun1(x)); grid on

```

4. Обчисліть значення коренів рівняння:

```

>>CutAndTry('Fun1',x1,x2,10^5)
ans=0.3942

```

5. Перевірте здобуте значення кореню рівняння:

```

>>Fun1(ans)
ans=7.4926e-006

```

3.3.3 Введення і виведення інформації в діалоговому режимі

Для забезпечення взаємодії з користувачем в процесі виконання М-файлу в системі MatLab використовуються такі функції:

disp(<змінна або текст в апострофах>) - виведення значень вказаної змінної або вказаного тексту в командне вікно(приклади приведені в таблиці 3.1);

x = input('<запрос>') - вихідний аргумент цієї функції є значенням, введеним користувачем з клавіатури в командний рядок у відповідь на запит. Наприклад, по оператору

myFun=input('Функція -')

у командному вікні з'явиться запит на ім'я функції, система переходить в стан очікування і користувач має нагоду ввести необхідне ім'я, наприклад відповідне m-файлу fun1.m, у вигляді:

Функція - 'fun1'

або саму функції з використанням команди inline

Функція - $\text{inline}('cos(x)*ln(x)+sin(x)')$

Таблиця 3.1 – Приклади функції disp

x='Аргумент'; disp(x);	Вивод значення текстової змінної в командне вікно.
x='Аргумент'; y='Функция' disp([x y]);	Вивод значення двох текстових змінних в командне вікно.
x=10.5; y=sin(x); f=cos(x); disp([x y f]);	Вивод значення декількох числових змінних в командне вікно.
x=[-pi:0.1:pi]; y=sin(x); f=cos(x); disp([x' y' f']);	Вивод значення елементів масивів у вигляді таблиці.
x=pi; n='Аргумент ='; disp([n, num2str(x)])	Вивод значення змінних змішанного типу (текстової та числової)
myFun=input('Имя функции-')	Ім'я функції-

3.3.4 Організація повторення дій

Однією з головних задач при самостійному створенні програми є забезпечення повернення до початку програми з метою продовження її виконання при нових значеннях початкових даних.

Для організації повторення дій дуже зручна функція menu:

$k = \text{menu}(\text{'Заголовок меню'}, \text{'Назва 1-ої кнопки'}, \text{'Назва 2-ої кнопки'}, \text{'Назва n-ої кнопки'})$ - функція виводить на екран меню. Виконання програми тимчасово припиняється, і система чекає вибору однієї з кнопок. Вихідному параметру k привласнюється номер натисненої кнопки. Нумерація кнопок починається з 1. Тепер, залежно від значення параметра k можна побудувати процес розгалуження обчислень.

Нехай основні оператори створеної програми розташовані у вище приведений функції average.m. Тоді схема забезпечення повернення до початку виконання цієї функції може бути такою:

% Script-файл, що служить прикладом організації

% повторення дій.

k=1;

while k==1

```

x=input('x= ');
z=average(x);
disp(['середнє значення = ', num2str(z)]);
k=menu('Що робити?'.
        'Продовжити роботу'.
        'Закінчити роботу');
end

```

3.4 Індивідуальні завдання до лабораторної роботи

Для заданих нелінійних рівнянь (таблиця 1, колонки 1 і 2) знайти відрізки локалізації коріння:

1. Скласти програму-функцію для знаходження відрізків локалізації коріння будь-якого нелінійного рівняння з виводом:

- графіка функції, що входить в нелінійне рівняння;
- вертикально розташованої таблиці табуляції функції з наступною шапкою

№	Аргумент	Функція
---	----------	---------

- списку знайдених відрізків локалізації.

2. Скласти програму-сценарій для організації повторного виклику функції, створеної в п.2. Організувати зміну початкових даних в діалоговому режимі.

Таблиця 3.1 - Варіанти індивідуальних завдань

№	1	2
1	$3x^4 + 4x^3 - 12x^2 - 5 = 0$	$\ln(x) + (x+1)^3 = 0$
2	$2x^3 - 9x^2 - 60x + 1 = 0$	$X \cdot 2^x = 1$
3	$x^4 - x - 1 = 0$	$x + \cos(x) = 1$
4	$2x^4 - x^2 - 10 = 0$	$x + \lg(1+x) = 1.5$
5	$3x^4 + 8x^3 + 6x^2 - 10 = 0$	$\lg(2+x) + 2x = 3$
6	$x^4 - 18x^2 + 5x - 8 = 0$	$2^x + 5x - 3 = 0$
7	$x^4 + 4x^3 - 12x^2 + 1 = 0$	$5^x + 3x = 0$
8	$x^4 - x^3 - 2x^2 + 3x - 3 = 0$	$3e^x = 5x + 2$
9	$3x^4 + 4x^3 - 12x^2 + 1 = 0$	$5^x = 6x + 3$
10	$3x^4 - 8x^3 - 18x^2 + 2 = 0$	$2e^x + 5x - 6 = 0$
11	$2x^4 - 8x^3 + 8x^2 - 1 = 0$	$2\arctg(x) - x + 3 = 0$
12	$2x^4 + 8x^3 + 8x^2 - 1 = 0$	$(x-3) \cos(x) = 1$

13	$x^4-4x^3-8x^2+1=0$	$x^x=20-9x$
14	$2x^4-9x^3-60x^2+1=0$	$x \lg(x)=1$
15	$x^5+x^2-5=0$	$\operatorname{tg}^3 x=x-1$
16	$3x^4+4x^3-12x^2-7=0$	$5^x=1+e^{-x}$
17	$3x^4+8x^3+6x^2-11=0$	$5^x=3-e^x$
18	$x^4-18x^3-10=0$	$\arctg(x^2+1/x)=x$
19	$3x^4-8x^3-18x^2+2=0$	$\operatorname{tg}(0.55x+0.1)=x^2$
20	$x^4-18x-10=0$	$5^x-6x=7$
21	$x^4+18x-10=0$	$5^x-6x=3$
22	$x^4+18x^3-6x^2+x-10=0$	$5^x=1+e^{-2x}$
23	$x^5+12x^3-6x^2+x-10=0$	$7^x-6x=2$
24	$3x^5-8x^3-18x^2+2=0$	$5^x=2+e^{-2x}$
25	$x^3-18x-10=0$	$x \cdot 2^x=3$

3.5 Контрольні питання

1. Чим відрізняються два види М-файлів: *М-сценарії* и *М-функції*..
2. Форма операторів:
 - циклу із заданим числом повторень;
 - циклу з передумовою;
 - умовного;
 - перемикання.
3. Уведення даних із клавіатури.
4. Оператори виводу значень на екран.
5. Оператори припинення виконання програми, створення меню.
6. Призначення та формат функції **feval**.

ЛАБОРАТОРНА РОБОТА № 4 РІШЕННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДАМИ ПОЛОВИННОГО РОЗПОДІЛУ ТА ХОРД

4.1 Мета роботи

Вивчення чисельних методів рішення нелінійних рівнянь та придбання практичних навиків роботи в системі MatLab.

4.2 Завдання до лабораторної роботи

1. Вивчити алгоритми ітераційних методів (*половинного розподілу* та *хорд*).
1. Скласти програми-функції для знаходження коренів будь-якого нелінійного рівняння методами *половинного розподілу* та *хорд* з заданою точністю.
3. Виконати в MatLab індивідуальні завдання наведені в розділі 4.4 та завдання на самостійну роботу – розділ 4.5.
4. Зберегти результати роботи (програми, протоколи обчислень, графіки).
5. Скласти звіт.

4.3 Основні теоретичні відомості

Найбільш загальний вигляд нелінійного рівняння:

$$f(x) = 0, \quad (4.1)$$

де функція $f(x)$ визначена і безперервна на кінцевому або безкінцевому відрізку $[a, b]$.

Коренем рівняння називається таке значення $\bar{x}$, при якому $f(\bar{x}) = 0$. Корінь $\bar{x}$ називається простим, якщо $f'(\bar{x}) \neq 0$, інакше корінь називається кратним. Ціле число k називається кратністю кореня, якщо при $x = \bar{x}$ разом з функцією дорівнюють нулю її похідні до $(k-1)$ порядку включно:

$$f(\bar{x}) = f'(\bar{x}) = \dots = f^{(k-1)}(\bar{x}) = 0$$

Постановка обчислення наближеного значення кореня точністю ε : знайти таке значення x , що $\left| x - \bar{x} \right| < \varepsilon$.

Рішення задачі розбивається на два етапи: на першому етапі здійснюють локалізацію кореня, на другому етапі проводять ітераційне уточнення кореня. На етапі локалізації кореня знаходять достатньо вузькі відрізки (або відрізок, якщо корінь єдиний), які містять один і лише один корінь рівняння $f(x) = 0$. На другому етапі наближене значення кореня із заданою точністю. Часто замість відрізка локалізації досить вказати початкове наближення до кореня.

4.3.1 Методи ітераційного уточнення коренів нелінійних рівнянь

Метод бісекції (розподілу навпіл)

Нехай $[a, b]$ – відрізок локалізації. Припустимо, що функція $f(x)$ безперервна на $[a, b]$ і на кінцях набуває значень різних знаків $f(a) \cdot f(b) < 0$.

Алгоритм методу розподілу навпіл полягає в побудові послідовності вкладених відрізків, на кінцях яких функція набуває значень різних знаків. Кожен наступний відрізок отримують діленням навпіл попереднього. Опишемо один крок ітераційного методу. Нехай на k -ом кроці знайдений відрізок $[a^{(k)}, b^{(k)}]$ такий, що $f(a^{(k)}) \cdot f(b^{(k)}) < 0$.

Знайдемо середину відрізка $c^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}$. Якщо $f(c^{(k)}) = 0$, то

$c^{(k)}$ – корінь і задача вирішена. Якщо ні, то із двох половин відрізка вибираємо той, на кінцях якого функція має протилежні знаки:

$$a^{(k+1)} = a^{(k)}, \quad b^{(k+1)} = c^{(k)}, \quad \text{якщо} \quad f(a^{(k)}) \cdot f(c^{(k)}) < 0,$$

$$a^{(k+1)} = c^{(k)}, \quad b^{(k+1)} = b^{(k)}, \quad \text{якщо} \quad f(a^{(k)}) \cdot f(c^{(k)}) > 0,$$

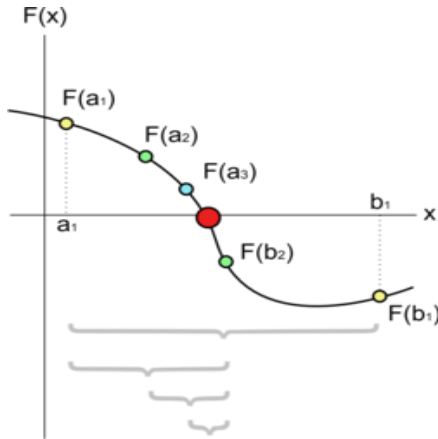


Рисунок 4.1 – Метод розподілу навпіл

Критерій закінчення ітераційного процесу: якщо довжина відрізка локалізації менша 2ε , то ітерації припиняють і як значення кореня із заданою точністю приймають середину відрізка.

Теорема про збіжність методу бісекцій. Нехай функція $f(x)$ безперервна на $[a, b]$ і на кінцях набуває значень різних знаків $f(a) \cdot f(b) < 0$. Тоді метод сходиться і справедлива оцінка погрешності: $|x^{(k)} - \bar{x}| \leq (b^{(k)} - a^{(k)}) / 2 = (b - a) / 2^{k+1}$.

З останнього рівняння легко отримати апріорну оцінку кількості кроків, необхідних для досягнення заданої точності:

$$\frac{b - a}{2^{k+1}} \leq \varepsilon, \text{ звідки отримуємо } k \geq 1 + \frac{\ln((b - a) / \varepsilon)}{\ln 2}.$$

Приклад програми-функції, яка вирішує рівняння методом розподілу навпіл, наведений в Лістингу 4.1.

Лістинг 4.1. Файл Bisection.m

```
function [x,k]=Bisection(f,a,b,eps)
%Data Input f – ім'я функції;
%           a і b – ліва і права границі інтервалу локалізації;
%           eps – точність;
%Data Output x – корінь;
%           k – кількість ітерацій.
```

```

ya= feval(f,a); yb= feval(f,b);
if ya*yb>0
    error('[a, b] не є інтервалом локалізації');
end;
k=0;
while abs(a-b)>eps
    k=k+1; x=(a+b)/2; fx=feval(f,x);
    if fx==0
        break;
    elseif fa*fx<0
        b=x; fb=fx;
    else
        a=x; fa=fx;
    end;
end;
end;

```

Метод хорд

В деяких випадках декілька більшою швидкістю збіжності володіє метод хорд, якого на другому етапі при виборі чергового наближення усередині відрізка, що містить корінь, враховується величина нев'язки на кінцях відрізка: вибирається ближче до того кінця, де нев'язка менша.

Геометричний сенс полягає в заміні кривої $y = f(x)$ хордою. Чергове наближення знаходиться як точка пересічення хорди віссю абсцис.

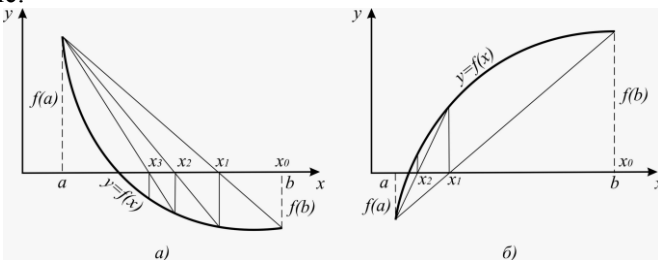


Рисунок 4.2 – Метод хорд

Якщо $[a, b]$ – відрізок, що містить корінь, то рівняння хорди

$$\frac{x-a}{b-a} = \frac{y-f(a)}{f(b)-f(a)}$$

Для точки пересічення хорди віссю абсцис $x = x_1$, $y = 0$ маємо

$$x_1 = a - \frac{b-a}{f(b)-f(a)} \cdot f(a),$$

$x = x_1$ береться за чергове наближення до кореню. Далі вибирається той з проміжків, на кінцях якого функція має значення різних знаків, і т.д. При цьому, якщо функція $f'(x)$ і її друга похідна $f''(x)$ зберігають знак на даному проміжку, то отримані наближення сходяться до кореню монотонно. В цьому випадку у всіх отриманих проміжків один кінець буде, саме той, на якому збігаються знаки функції і другої похідної. Цей кінець називають нерухомим, а протилежний кінець відрізка приймають за початкове наближення до кореню. Н c – той з кінців відрізка $[a, b]$, для якого виконується умова $f(c) \cdot f''(c) > 0$, а x_0 - початкове значення (протилежний кінець відрізка), тоді ітерації проводимо по формулі:

$$x_{k+1} = x_k - \frac{c - x_k}{f(c) - f(x_k)} \cdot f(x_k), \quad k=0, 1, 2, \dots$$

4.3.2 Функції MatLab для рішення нелінійних рівнянь

Функція fzero

Функція

[x, fval, exitflag, output] = fzero(fun, x0, Options, p1, p2,...)

Пробує знайти нуль функції **fun** поблизу **x0**, якщо **x0** є скаляром.

Вхідні аргументи:

fun – покажчик на функцію;

x0 – початкове наближення або інтервал локалізації кореня;

Options - параметри, що задають точність і спосіб представлення результатів обчислень (докладніше читайте в *help*);

p1, p2, ... - додаткові аргументи функції **fun**.

Вихідні аргументи:

x – знайдене значення кореня;

Fval – значення функції для знайденого значення кореня;

Exitflag = 1, означає, що корінь знайдений;

output – структура з значеннями:

output.algorithm – алгоритми, які використовувалися для знаходження кореня;

output.func - кількість обчислень значень функції;

output.iterations - кількість ітерацій.

Приклад 4.1. Обчислити число π , шляхом знаходження нуля функції $\sin$ поблизу 3.

```
>>x = fzero(@sin,3)
x = 3.1416
```

Приклад 4.2. Знайти нулі функції $\cos$ між 1 і 2:

```
>>x = fzero(@cos,[1 2])
x = 1.5708
```

Відмітимо, що $\cos(1)$ и $\cos(2)$ мають різні знаки.

Приклад 4.3. Вирішимо нелінійне рівняння $x^3 - 5x^2 - x + 2 = 0$ поблизу $x = 4$. Для задавання функції, що входить в нелінійне рівняння, використовуємо команду **inline**.

```
>>f=inline('x.^3-5*x.^2-x+2');
>> fzero(f,4)
ans = 5.1190
```

Змінимо початкове значення на $x = 2$

```
>> fzero(f,2)
```

отримаємо

```
ans = 0.5684
```

Задамо інтервал локалізації корня $x \in [3 \ 6]$:

```
>>z = fzero(f; [3 6])
ans = 5.1190
```

Приклад 4.4. Для функції, що входить в нелінійне рівняння $x - \cos(x) = 0$, створимо файл **fun.m** (лістинг 1.1).

Лістинг 1.1. Файл **fun.m**

```
function y=fun(x)
y=x-cos(x);
```

Рішення:

```
>> x=fzero('fun', 4)
x = 5.1190
```

Функція *roots*

Функція

$$\mathbf{r}=\text{roots}(\mathbf{c})$$

служить для знаходження всіх коренів нелінійного рівняння виду:

$$c_1x^n + c_2x^{n-1} + \dots + c_{n+1}x^0 = 0$$

Вхідний аргумент $\mathbf{c}$ – вектор-рядок, вектор-рядок, що містить коефіцієнти полінома.

Приклад 4.5. Знайти нулі функції $f(x) = x^3 - 2x - 5$

Визначимо анонімну функцію f , і знайдемо її нулі поблизу 2:

```
>>f = @(x)x.^3-2*x-5;
```

```
>>z = fzero(f,2)
```

```
z = 2.0946
```

Оскільки ця функція є поліномом, виклик функції

```
>>roots([1 0 -2 -5])
```

знайде той же дійсний корінь, а також комплексну пару коренів:

```
2.0946
```

```
-1.0473 + 1.1359i
```

```
-1.0473 - 1.1359i
```

4.4 Індивідуальні завдання до лабораторної роботи

Для заданого нелінійного рівняння (варіанти індивідуальних завдань наведені в першій колонці Додатка А, стор. 58):

1. визначити кількість коренів нелінійного рівняння в заданій вам області, для кожного кореня знайти відрізки локалізації;
2. знайдіть один з коренів методами:
 - а) розподілу навпіл,
 - б) хорд;
3. разом з "ручним" рішенням представте рішення, що отримуються стандартними засобами MatLab;

4.5 Контрольні питання

1. Сформулюйте постановку задачі наближеного розв'язання нелінійного рівняння і основні етапи її розв'язання.
2. Опишіть алгоритм методу розподілу навіпл.
3. Опишіть алгоритм методу хорд.
4. Перерахуйте вбудовані функції MatLab (і їх формати) для вирішення нелінійних рівнянь. На яких методах засновані алгоритми реалізовані в цих функціях? Що можна сказати про точність отриманих рішень?

ЛАБОРАТОРНА РОБОТА № 5 РІШЕННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДАМИ НЬЮТОНА ТА ПРОСТИХ ІТЕРАЦІЙ

5.1 Мета роботи.

Вивчення чисельних методів рішення нелінійних рівнянь та придбання практичних навиків роботи в системі MatLab.

5.2 Завдання до лабораторної роботи

1. Вивчити алгоритми ітераційних методів (*Ньютона та Простих ітерацій*).
2. Скласти програми-функції для знаходження коренів будь-якого нелінійного рівняння методами *Ньютона та Простих ітерацій* з заданою точністю.
3. Виконати в MatLab індивідуальні завдання наведені в розділі 5.4.
4. Зберегти результати роботи (програми, протоколи обчислень, графіки) в відповідних файлах.
5. Скласти звіт.

5.3 Основні теоретичні відомості

Метод Ньютона (метод дотичних). Розрахункова формула методу Ньютона має :

$$x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}.$$

Геометрично метод Ньютона означає, що наступне наближення до кореня $x^{(n+1)}$ є точка пересічення з віссю ОХ дотичної, що проведена до графіка функції $y=f(x)$ в точці $(x^{(n)}, f(x^{(n)}))$.

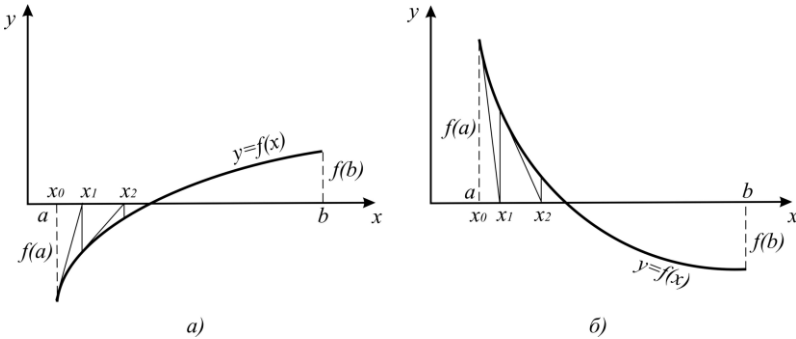


Рисунок 5.1 – Метод Ньютона

Ефективність методу залежить від початкового наближення x^0 . Наприклад, воно може збігатися з одним із кінців інтервалу ізоляції кореня $[a, b]$, для якого виконується умова $f(x) * f''(x) > 0$. На практиці ж вибір значення x^0 набагато складніший. Умови збіжності методу Ньютона, можна визначити у такий спосіб:

- функція $f(x)$ має бути принаймі двічі диференційована на інтервалі $[a, b]$;
- перша похідна функції не повинна перетворюватися на нуль $f'(x) \neq 0$;
- виконується умова $|f(x) * f''(x)| < f'(x)^2$ на інтервалі $[a, b]$.

Таким чином, збіжність обчислень методу гарантується лише для монотонних і обмежених за крутістю функцій.

Критерій закінчення ітераційного процесу. При заданій точності $\varepsilon > 0$ обчислення слід вести до тих пір, поки не виявиться виконаним нерівенство $|x^{(n)} - x^{(n-1)}| < \varepsilon$ або $|f(x^{(n)})| < \varepsilon$.

Програма, що реалізовує метод Ньютона, наведена в лістингу 5.1.

Лістинг 5.1. Файл newton.m.

```
function [x,k]=newton(fname,fname1,x0,eps)
% корінь методом Ньютона,
% повертає корінь і кількість ітерацій.
% fname - ім'я функції, fname1 – похідна,
% x0 – початкове наближення до кореня,
```

```
% eps – точність.
k=0;
f=feval(fname,x0); f1= feval(fname1,x0);
x1=x0-f/f1;
while abs(x0-x1)>eps & k < 100
    x0=x1;
    f=feval(fname,x0);
    f1= feval(fname1,x0);
    x1=x0-f/f1;
    k=k+1;
end;
x=x1;
```

Приклад 5.1. Вирішити рівняння $f(x)=0$, де

$$f(x) = 3x^4 + 4x^3 - 12x^2 - 5$$

методами Ньютона и розподілу навіпіл. Порівняти методи за швидкістю збіжності і обчислень.

1. Створимо функцію $f(x)$, ім'я функції var1

```
function y=var1(x)
    y=3*x.^4+4*x.^3-12*x.^2-5;
```
2. Її похідна

```
function y=var11(x)
    y=12*x^3+12*x^2-24*x;
```
3. Програма-сценарій (файл *test.m*) для порівняння швидкості збіжності методів Ньютона і розподілу навіпіл:

```
x1=zeros(1,6); k1=zeros(1,6); x2=zeros(1,6); k2=zeros(1,6);
eps=[0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001];
for i=1:6
    [x1(i),k1(i)]=bisec(@var1,-3,-2,eps(i));
    [x2(i),k2(i)]=newton(@var1,@var11,-2.5,eps(i));
end;
disp x1,x2
plot(log(1./eps),k1,'k-.',log(1./eps),k2,'k:')
grid on
```

Результат роботи скрипта `test.m` наведений на малюнку 5.2. По осі Ox відкладені натуральні логарифми від зворотної величини точності (із зростанням точності зростає і аргумент по осі Ox). По осі Oy – число ітерацій, необхідне для обчислення кореня із заданою точністю.

Видно, що із зростанням точності метод Ньютона вимагає значно менше ітерацій, правда, на кожній ітерації значення функції і значення похідної, тоді як в методі розподілу навпіл на кожній ітерації обчислюється лише одне значення функції.

Як вказано вище, метод Ньютона володіє локальною збіжністю, тобто областю його збіжності є мала околиця кореня. Невдалий вибір може дати ітераційну послідовність, що розходиться.

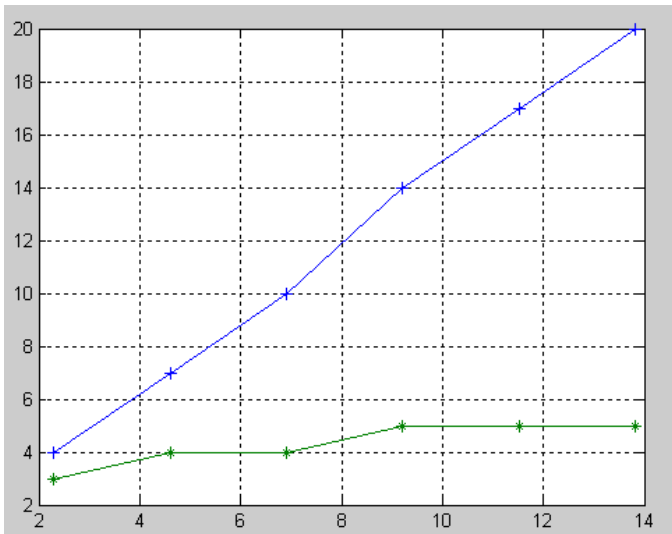


Рисунок 5.2 – Залежність кількості ітерацій від точності для методів розподілу навпіл (верхня лінія) і методу Ньютона (нижня лінія)

Приклад 5.2. Чутливість методу Ньютона до вибору початкового наближення.

1. Лістинг функції – файл `f24.m`

```
function y=f24(x)
y=(x-1.3)/((x-1.3).^2+1);
```

2. Лістинг похідної функції – файл *f241.m*

```
function y=f241(x)
y=(1-(x-1.3).^2)./((x-1.3).^2+1).^2;
```

3. Лістинг скрипта – файл *test1.m*

% Програма тестування методу Ньютона на чутливість

% до вибору початкового наближення

```
[root1, iter1] = newton('f24','f241', 1.87,1e-8);
```

```
% Перевіримо корені
```

```
y=f24(root1)
```

```
[root2, iter2] = newton('f24','f241', 1.88,1e-8);
```

```
y1=f24(root2)
```

```
% Порівняємо кількість ітерацій
```

```
iter1
```

```
iter2
```

4. Протокол запуску програми і результати її роботи:

```
>> test1
```

```
y = 0
```

```
y1 =4.1016e-030
```

```
iter1 = 7
```

```
iter2 = 100
```

Бачимо, що число ітерацій в першому випадку набагато менше, ніж в другому.

Метод простої ітерації (метод послідовних повторень)

Для методу простої ітерації потрібно початкове рівняння $f(x)=0$ перетворити до виду, зручному для ітерацій $x = \varphi(x)$. Це перетворення можна виконати різними способами. Функція $\varphi(x)$ називається ітераційною функцією. Розрахункова формула методу простої ітерації має $x^{(n+1)} = \varphi(x^{(n)})$.

Теорема про збіжність методу простої ітерації.

Хай рівняння $x = \varphi(x)$ має єдиний корінь на відрізку $[a, b]$ і виконані умови:

1. $\varphi(x)$ визначена і диференційована на $[a, b]$;

2. Існує таке дійсне q , що $|\varphi'(x)| \leq q < 1$ для всіх $x \in [a, b]$.

Тоді ітераційна послідовність $x^{(n+1)} = \varphi(x^{(n)})$ ($n = 1, 2, \dots$) сходиться при будь-якому початковому наближенні $x^0 \in [a, b]$.

Критерій закінчення ітераційного процесу. При заданій точності $\varepsilon > 0$ обчислення слід вести до тих пір, поки не виявиться виконаною нерівність $|x^{(n)} - x^{(n-1)}| < \varepsilon$, ($n = 1, 2, \dots$).

Приклад 5.3. Знайти методом простих ітерацій розв'язок рівняння

$$x^4 - 11x^3 + x^2 + x + 0.1 = 0$$

а також точність досягнуту за 10 ітерацій.

1. Лістинг програми-функції, що обчислює точність досягнуту за k ітерацій при розв'язуванні рівнянь методом простих ітерацій:

Function [x,eps]=Iter(fi,fi1,x0,k)

% fi – ім'я функції;

% fi1 – похідна функції;

% x0 – початкове наближення;

% k – кількість ітерацій;

% x – корінь;

% eps – точність.

2. Задамо початкову функцію $f(x) = x^4 - 11x^3 + x^2 + x + 0.1$, функцію $\varphi(x) = -x^4 + 11x^3 - x^2 - 0.1$ и її першу похідну використовуючи команду *inline*:

```
>>f=inline('x.^4-11*x.^3+x.^2+x+0.1');
```

```
>>fi=inline('-x.^4+11*x.^3-x.^2-0.1');
```

```
>>fi1=inline('-4*x.^3+33*x.^2-2*x');
```

3. Для коренів нелінійного рівняння використовуємо графічний метод:

```
>>x1= -1;
```

```
>>x2=1;
```

```
>>dx=10^-3;
```

```
>>x=x1:dx:x2;
```

```
>>plot(x,f(x)); grid on
```

4. Виберемо інтервал локалізації – $[-1,1]$ і за 10 ітерацій:

```
>> [x,eps]=Iter('fi','fi1',-1,10)
```

```
x=0.3942
```

```
eps=10^-5
```

5.4 Індивідуальні завдання до лабораторної роботи

Для заданого нелінійного рівняння (варіанти індивідуальних завдань наведені в другій колонці Додатка А, стор.58):

1. Виконати відокремлення коренів нелінійного рівняння та провести уточнення цих коренів з заданою точністю методами:
 - а) Ньютона;
 - б) простої ітерації.
2. Порівняти обчислень при розрахунку указаними методами із різними значеннями точності.
3. Разом з "ручним" рішенням представте рішення, що отримуються стандартними засобами MatLab.

5.5 Контрольні питання

1. Запишіть розрахункову формулу методу Ньютона і дайте геометричну інтерпретацію методу.
2. Опишіть алгоритм методу простої ітерації і дайте геометричну інтерпретацію методу.
3. Вкажіть вади і недоліки методу простої ітерації в порівнянні з методом Ньютона.
4. Умова збіжності методу Ньютона.
5. Умова збіжності методу простої ітерації.

Список літератури

1. Балашова С.Д. Чисельні методи: Ч.1. Методи розв'язування задач аналізу та алгебри: Навч. посібник.- К.: НМК ВО, 1992.-280 с.
2. Балашова С.Д. Чисельні методи: Ч.2. Методи розв'язування диференціальних та інтегральних рівнянь: Навч. посібник. - К.: НМК ВО, 1992.-328 с.
3. Гаврилук І.П., Макаров В.Л. Методи обчислень: - К.: Вища шк., 1995.- Ч.1.- 368 с., Ч.2. 431 с..
4. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MATLAB 7: Программирование, численные методы. – СПб.: БХВ-Петербург, 2005. – 752 с.
5. Турчак Л.И. Основы численных методов: Учеб.пособие.-М.: Наука, 1987. -320 с.
6. Крылов В.И., Бобков В.В., Монастырный П.И. Вычислительные методы: Т.1. - М.: Наука, 1976. - 304 с.
7. Крылов В.И., Бобков В.В., Монастырный П.И. Вычислительные методы: Т.2. - М.: Наука, 1977. - 400 с.
8. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы: Учеб. пособие. - М.: Лаборатория базовых знаний, 2000 г.. - 624 с.

Додаток А

Індивідуальні завдання до лабораторних робіт № 4, 5

№	1	2
1	$3x^4+4x^3-12x^2-5=0$	$\ln(x)+(x+1)^3=0$
2	$2x^3-9x^2-60x+1=0$	$x \cdot 2^x=1$
3	$x^4-x-1=0$	$x+\cos(x)=1$
4	$2x^4-x^2-10=0$	$x+\lg(1+x)=1.5$
5	$3x^4+8x^3+6x^2-10=0$	$\lg(2+x)+2x=3$
6	$x^4-18x^2+5x-8=0$	$2^x+5x-3=0$
7	$x^4+4x^3-12x^2+1=0$	$5^x+3x=0$
8	$x^4-x^3-2x^2+3x-3=0$	$3e^x=5x+2$
9	$3x^4+4x^3-12x^2+1=0$	$5^x=6x+3$
10	$3x^4-8x^3-18x^2+2=0$	$2e^x+5x-6=0$
11	$2x^4-8x^3+8x^2-1=0$	$2\arctg(x)-x+3=0$
12	$2x^4+8x^3+8x^2-1=0$	$(x-3) \cdot \cos(x)=1$
13	$x^4-4x^3-8x^2+1=0$	$x^x=20-9x$
14	$2x^4-9x^3-60x^2+1=0$	$x \cdot \lg(x)=1$
15	$x^5+x^2-5=0$	$\operatorname{tg}^3 x=x-1$
16	$3x^4+4x^3-12x^2-7=0$	$5^x=1+e^{-x}$
17	$3x^4+8x^3+6x^2-11=0$	$5^x=3-e^x$
18	$x^4-18x^3-10=0$	$\arctg(x^2+1/x)=x$
19	$3x^4-8x^3-18x^2+2=0$	$\operatorname{tg}(0.55x+0.1)=x^2$
20	$x^4-18x-10=0$	$5^x-6x=7$
21	$x^4+18x-10=0$	$5^x-6x=3$
22	$x^4+18x^3-6x^2+x-10=0$	$5^x=1+e^{-2x}$
23	$x^5+12x^3-6x^2+x-10=0$	$7^x-6x=2$
24	$3x^5-8x^3-18x^2+2=0$	$5^x=2+e^{-2x}$
25	$x^3-18x-10=0$	$x \cdot 2^x=3$