Лабораторна робота 1\_1

Візуалізація в середовищі MatLab®

**Мета роботи:** Ознайомитися з основними елементами і складовими частинами  
системи комп’ютерної математики MatLab® та її можливостями візуалізації.  
**Теоретичні відомості**:  
Робоче середовище системи MATLAB  
При запуску даного програмного продукту на екрані відкривається робоче  
вікно програми, зображене на рис. 1.

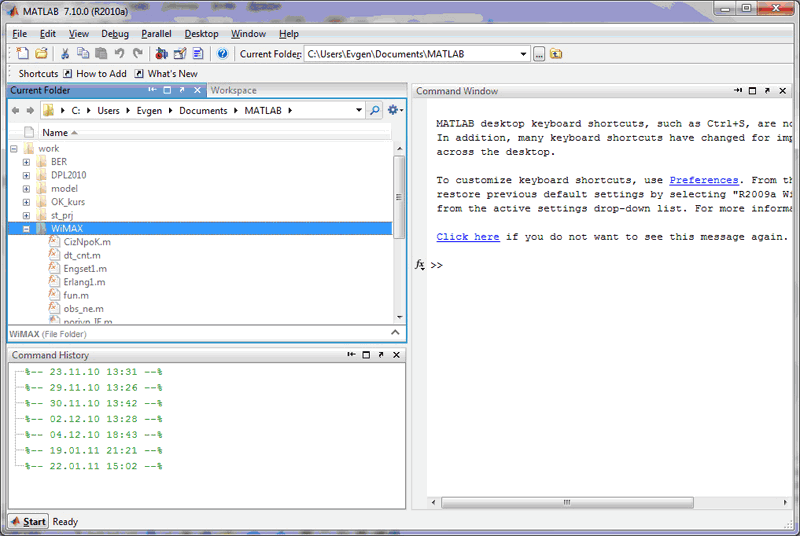
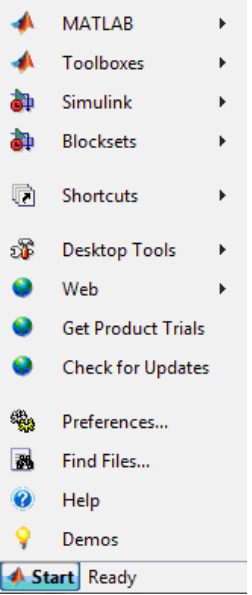


Рис. 1. Робоче середовище системи MATLAB

Основними елементами робочого середовища є:  
• меню;  
• панель інструментів з кнопками і випадаючим меню;  
• вікна з вкладками Workspace і Current Directory для перегляду змінних та  
для встановлення робочого каталогу з використовуваними файлами сценаріями і функціями;  
• вікно Command Window (аналог командної стрічки), призначене для  
введення команд і виведення результатів;  
• вікно Command Hisory - для перегляду і повторного виконання раніше  
введених команд;  
• рядок стану та кнопка Start.  
При натисненні на кнопку Start відкривається меню, приклад якого зображено  
на рис. 2, з його допомогою забезпечується доступ до усіх основних засобів  
системи.

  
Рис. 2. Меню, яке відкривається при натисненні на кнопку Start

Вікно Command Window складається з наступних елементів:  
• заголовка з назвою вікна і двома кнопками справа;  
• робочої області з командною стрічкою, в якій знаходиться мигаючий  
вертикальний курсор;  
• смуг прокрутки.  
В полі назви кожного вікна, поряд з кнопкою закривання, знаходиться кнопка  
«Undock...» для витягування вікна з робочого середовища (якщо воно  
вбудовано), або кнопка «Dock…» для вбудовування окремого вікна в робоче  
середовище MATLAB.  
При запуску пакету в робочій області вікна Command Window з’являються два  
посилання: MATLAB Help і Demos - для виклику довідкової системи або  
демонстраційних прикладів.

**Арифметичні обчислення**  
Вбудовані математичні функції дозволяють знаходити значення виразів будь-якої складності, причому для обчислень можна використовувати також мову  
сценаріїв системи, що є мовою програмування високого рівня.  
Також система MATLAB® надає можливість управління форматом виведення  
результатів.  
**Найпростіші обчислення**Виберіть вигляд робочого середовища „за замовчуванням”, наберіть в  
командному рядку 1+2 і натисніть <Enter>.  
В результаті в командному вікні MATLAB® відобразиться наступне:  
>> 1 + 2  
ans =  
3  
>> |  
Якщо обчислюваний вираз не присвоюється змінній, то створюється змінна ans  
(від слова answer - відповідь). Інформація про змінну ans відразу з’явилася у  
вікні Workspace (рис. 3.).  
В першому стовпці Name записано ім’я змінної.  
Наступний стовпець Value показує значення змінної у вибраному для її  
відображення форматі.  
Вміст стовпця Size, по суті, демонструє основний принцип роботи MATLAB®,  
яка всі дані представляє у вигляді масивів.  
Змінна ans є двовимірним масивом розміром один на один і займає 8 байт  
пам’яті, про що свідчить стовпець Bytes.  
І в останньому стовпці Class вказаний тип змінної – double array, тобто масив,  
що складається з чисел подвійної точності.  
Будь-який стовпець можна сховати або відобразити, якщо на назві вікна  
натиснути правою кнопкою мишки і викликати контекстне меню.

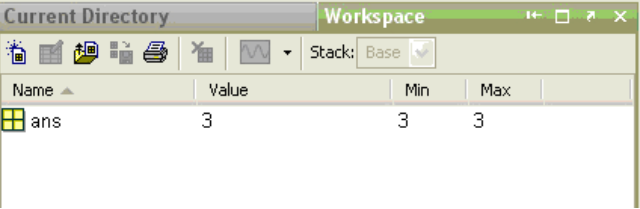


Рис. 3. Вікно Workspace з інформацією  
про використання змінних середовища

Якщо необхідно продовжити роботу з попереднім виразом, наприклад,  
обчислити (1+2)/4.5, то найпростіше скористатися існуючим результатом, який  
зберігається в змінній ans. Після набору в командній стрічці ans/4.5 і  
натиснення клавіші <Enter>, отримаємо:  
>> ans/4.5  
ans =  
0.6667  
>> |  
**Значення змінної ans автоматично замінюється результатом обчислень.**Формат виведення результатів обчислень  
Потрібний формат виведення результату визначається користувачем з меню  
робочого середовища MATLAB®.  
Виберіть в меню File пункт **Preferences**.  
На екрані з’явиться діалогове вікно Preferences, зображене на рис. 4. В  
пункті Command Window задається формат із списку, що розкривається,  
Numeric format панелі Text display.

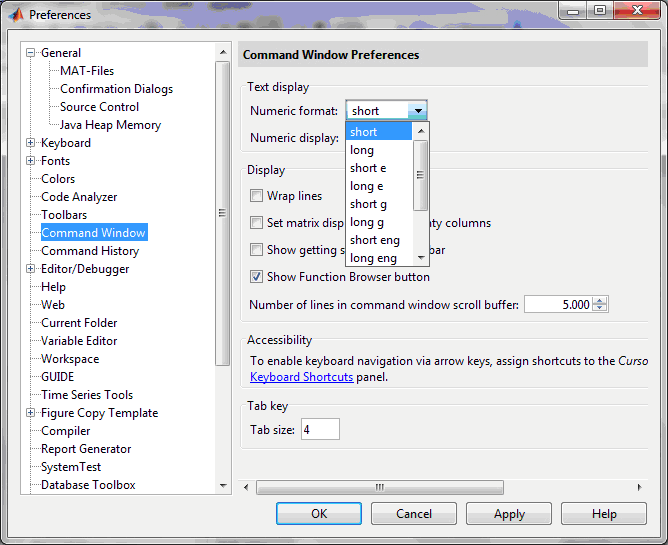
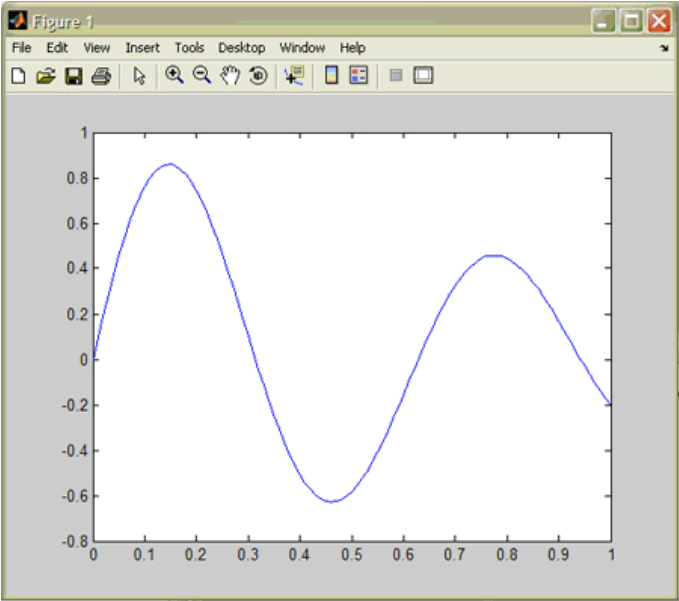


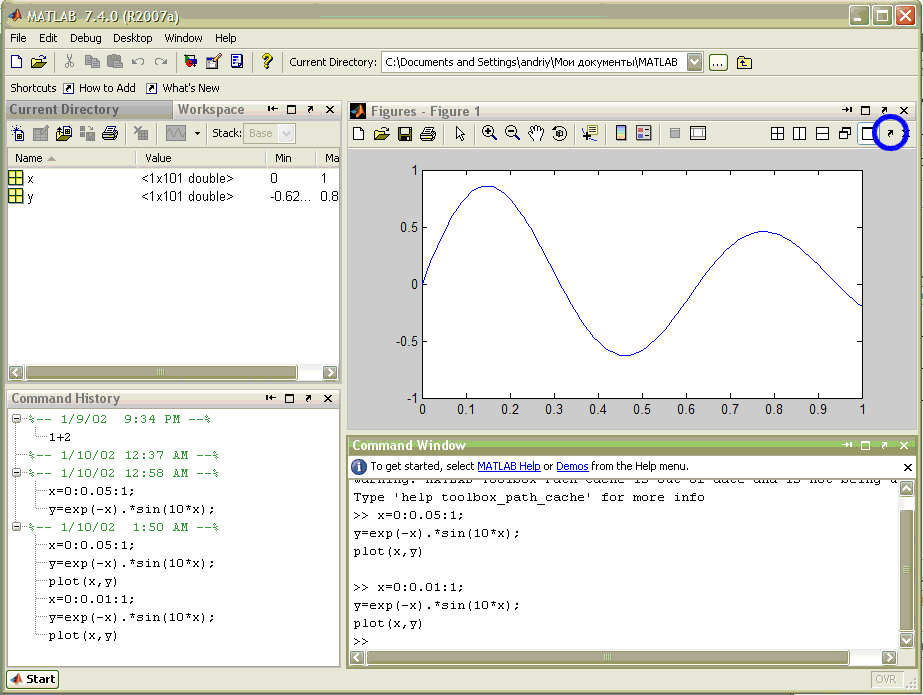
Рис. 4. Діалогове вікно Preferences. Використання елементарних функцій

Наприклад, необхідно обчислити наступний вираз:  
Для цього в командній стрічці потрібно ввести даний вираз і натиснути  
<Enter>:  
>> exp(-2.5)\*log(11.3)^0.3-sqrt((sin(2.45\*pi)+cos(3.78\*pi))/tan(3.3))  
Результат виводиться в командне вікно:  
ans =  
-3.2105  
Арифметичні операції в MATLAB® виконуються в звичайному порядку,  
властивому для більшості мов програмування.  
При проведенні обчислень з комплексними числами в командній стрічці  
MATLAB® можна використовувати i або j, а самі числа при множенні, діленні  
або піднесенні до степеня необхідно брати у круглі дужки:  
>> (2.1 + 3.2i)\*2 + (4.2 + 1.7i)^2  
ans =  
18.9500 + 20.6800i

**Побудова графіків**  
Система MATLAB володіє широкими можливостями для графічного  
представлення результатів обчислень і візуалізації даних.  
Виведення відображення функції  
у вигляді графіка складається з наступних етапів:  
завдання вектора значень аргументу x;  
обчислення вектора y значень функції y(x);  
виклик команди plot для побудови графіка.  
Команди для завдання вектора x і обчислення функції краще завершувати  
крапкою з комою, що дозволяє уникнути виведення в командне вікно їх  
значень:  
>> x=0:0.01:1;  
>> y=exp(-x).\*sin(10\*x);  
>> plot(x,y)  
Для побудови графіка функції в робочому середовищі MATLAB® повинні бути  
визначені два вектори однакових розмірів, наприклад x та y. Вектор x містить  
значення аргументів, а y – значення функції цих аргументів.  
Команда plot з’єднує точки з координатами (x(i), y(i)) прямими лініями, автоматично масштабуючи осі для найкращого візуального розміщення графіка по осях.

Рис. 5. Графік функції .

Якщо ж відображення графічного результату в системі за замовчуванням  
користувачу не підходить, можна параметри візуалізації задати індивідуально  
для кожного графіка, або налаштувати їх вже після відображення результату.  
Використовуючи кнопку Dock Figure (справа в стрічці меню вікна) можна  
вмонтувати графічне вікно в робоче середовище так, як це показано на рис. 6.  
Порівняння декількох функцій зручно виконувати, якщо їх графіки відобразити  
в одній системі координат.  
Для побудови графіків функцій і потрібно  
набрати наступну послідовність команд:  
>> x = -1:0.005:-0.3;  
>> f = sin(x.^-2);  
>> g = sin(1.2\*x.^-2);  
>> plot(x, f, x, g)  
Також з допомогою команди plot можна задати стиль та колір ліній, наприклад:  
>> plot(x, f, 'k-', x, g, 'k:')  
Аргументи ‘k-‘ і ‘k:’ задають стиль і колір першої та другої ліній. Тут k означає  
чорний колір, а дефіс або двокрапка – неперервну або пунктирну лінію.



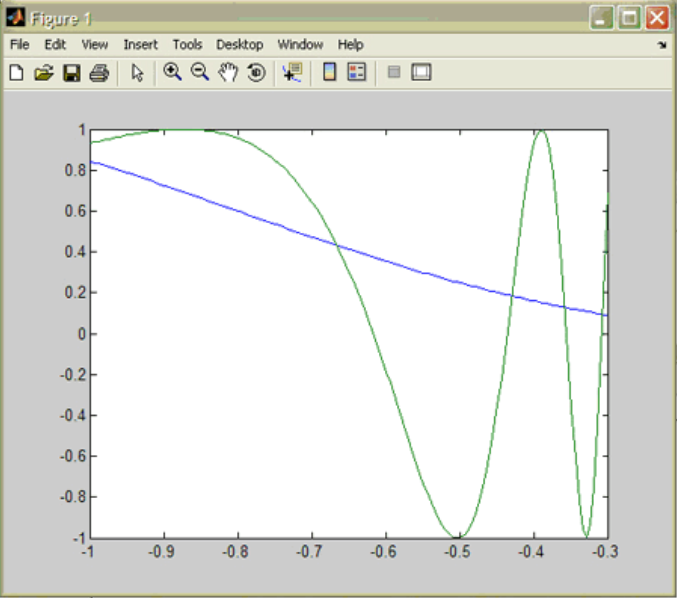
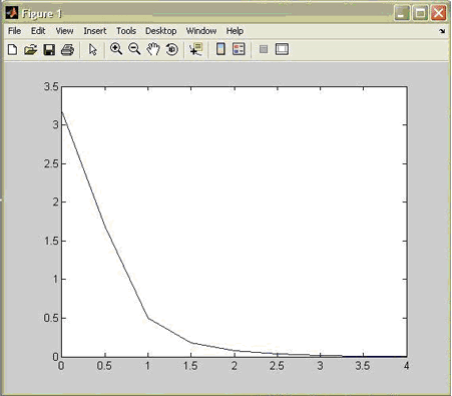
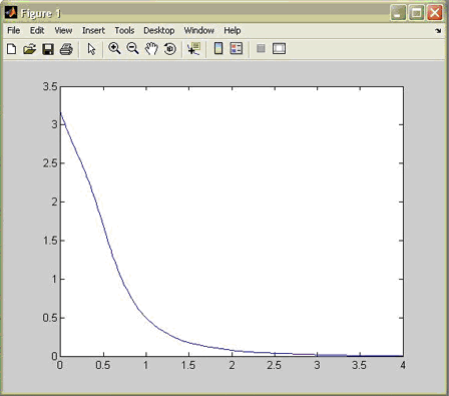
Рис. 6. Розміщення вікна з графіком у середовищі MATLAB

Рис. 7. Два графіка на одних осях

**Файли-функції**  
Для вирішення обчислювальних задач і написання власних застосувань в  
MATLAB® часто потрібно реалізовувати функції користувача, які здійснюють  
необхідні дії з вхідними аргументами і повертають результат обчислень.  
Кількість вхідних і вихідних аргументів залежить від задачі, що вирішується -  
може бути тільки один вхідний і вихідний аргумент, а може бути декілька  
вхідних і вихідних, або ж тільки вхідні аргументи. Вхідними аргументами і  
вихідними даними можуть бути масиви даних довільної розмірності.  
**Файли-функції з одним аргументом**Реалізувавши один раз файл-функцію, далі можна її використовувати всюди, де  
необхідно провести потрібні обчислення для заданого аргументу, але для цього  
файл повинен бути розміщений в робочому каталозі системи.  
Відкрийте в редакторі М-файлів новий файл і наберіть наступне:  
function f = myfun(x)  
f = exp(-x)\*sqrt((x^2 + 1)/(x^4 +0.1));  
Слово function в першій стрічці визначає, що даний файл містить функцію.  
Перша стрічка є заголовком функції, в якій знаходяться ім'я функції і списки  
вхідних і вихідних аргументів. Вхідні аргументи записуються в круглих дужках  
після імені функції. Вихідний аргумент f вказується зліва від знаку «рівне» в  
заголовку функції. При виборі імені файла необхідно пам’ятати, що воно  
повинне співпадати із назвою функції, для якої цей файл реалізовано. Також  
потрібно потурбуватися про уникнення конфліктів з зайнятими іменами  
функцій системи MATLAB®.  
Після заголовку розміщується тіло функції - один або декілька операторів, які  
реалізовують алгоритм отримання значення вихідних змінних з вхідних. Перед  
запуском файлу потрібно його зберегти в робочому каталозі (File → Save  
абоFile → Save as...).  
Тепер створену функцію можна використовувати так само, як і вмонтовані  
sin, cos і інші, наприклад:  
>> y=myfun(5)  
y =  
0.0014  
Робота файл-функції з масивом даних  
Якщо записати:  
function f = myfun(x)  
f = exp(-x).\*sqrt((x.^2 + 1)./(x.^4 +0.1));  
то тепер аргументом функції myfun може бути як число, так і вектор чи  
матриця значень, наприклад:  
>> x=[1.3 7.2];  
>> y=myfun(x)  
y =  
0.2600 0.0001  
Змінна y, в яку записується результат виклику функції myfun, автоматично стає  
вектором потрібного розміру.  
Для того, щоб побудувати графік на відрізку [0, 4] з допомогою файлу-сценарію  
програми або з командної стрічки необхідно набрати:  
>> x=0:0.5:4;  
>> y=myfun(x);  
>> plot(x,y)  
або  
fplot('myfun',[0 4])  
Отримаємо результат, зображений на рис. 8.

Графік побудований з допомогою fplot більш точно відображає поведінку  
функції, тому що алгоритм автоматично підбирає крок аргументу, зменшуючи  
його на відрізках швидкої зміни досліджуваної функції.  
**Файл-функція з декількома вхідними аргументами**Написання файлу-функції з декількома вхідними аргументами практично не  
відрізняється від написання файлу-функції з одним вхідним аргументом.  
Нижче наведено приклад файлу-функції з трьома вхідними аргументами для  
обчислення радіус-вектора точки тривимірного простору:  
function r=radius3(x,y,z)  
r=sqrt(x.^2+y.^2+z.^2);  
Для обчислення радіус-вектора точки тривимірного простору можна  
скористатися функцією radius3:  
>> R=radius3(1,1,1)  
R =  
1.7321

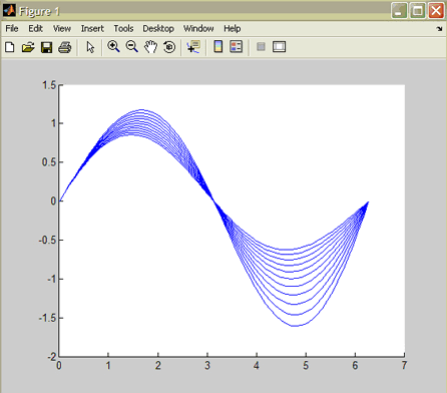


а)

б)

Рис. 8. Порівняння plot і fplot

**Оператори циклу**  
Схожі і повторювані дії виконуються з допомогою операторів циклу for та while.  
Цикл for призначений для виконання заданої кількості повторюваних дій,   
while - для дій, кількість яких наперед невідома, але відома умова виконання і  
закінчення циклу.  
**Цикл for**  
Використання for здійснюється наступним чином:  
for count = start:step:final  
і команди MATLAB  
end  
Тут count – змінна циклу, start – її початкове значення, final – кінцеве значення, а step – крок, на який збільшується count при кожному наступному заході у цикл.  
Цикл закінчується, як тільки значення count стає більше ніж final. Змінна циклу  
може приймати не тільки цілі числа, а й дійсні значення з будь-яким знаком.  
Наприклад, для виводу графіків кривих для , яке задано функцією  
, що залежить віл параметра а, для значень параметра а від  
-0,1 до 0,1 з кроком 0,002. В редакторі М-файлів потрібно набрати код:  
figure %  
x = 0:pi/30:2\*pi; %  
for a = -0.1:0.02:0.1  
y = exp(-a\*x).\*sin(x); %  
hold on;  
plot(x, y);  
end  
В результаті виконання даного фрагменту коду з’явиться графічне вікно, яке  
зображене на рис. 9.

  
Рис. 9. Сімейство кривих

**Цикл while**  
Використання while здійснюється наступним чином:  
while (умова повторення циклу)  
Команди MATLAB®  
end  
Команди MATLAB® виконуються, поки виконується умова  
повторення циклу.

Умовний оператор if  
Умовний оператор if в загальному вигляді записується таким чином:  
if Умова1  
Інструкція\_1  
else іf Умова2  
Інструкція\_2  
else  
Інструкція\_3  
end  
Наприклад, поки Умова1 повертає логічне значення 1 (тобто «істина»),  
виконується Інструкція\_1, що складає тіло структури if...else if.  
Оператор end вказує на кінець переліку інструкцій. Інструкції в списку  
розділяються оператором , (кома) або ; (крапка з комою). Якщо Умова не  
виконується (дає логічне значення 0, «не істина»), то виконуються інструкції,  
що записані після ключового слова else.  
Ще одна конструкція:  
if Умова  
Інструкція\_1  
else  
Інструкція\_2  
end  
Виконується Інструкція\_1, якщо виконується Умова, або Інструкція\_2 в  
протилежному випадку.  
Умова записується у вигляді:  
Вираз\_1 Оператор\_відношення Вираз\_2, в якості Операторів\_відношення  
використовуються наступні оператори:

|  |  |
| --- | --- |
| Позначення | Операція відношення |
| == | Дорівнює |
| < | Менше |
| > | Більше |
| <= | Менше-рівне |
| >= | Більше-рівне |
| ~= | Не дорівнює |

**Завдання на роботу:** Запустити на виконання програму MATLAB®, послідовно задати та виконати у m-файлі всі пункти, розглянуті в даній теоретичній частині.

**Зміст звіту:**

- назву та мету роботи;

- m-файли з наведеними прикладами;

- скріншоти результатів виконання завдань;

- висновки по роботі з аналізом операторів та розроблених програм.