

## Лекція 6

### Основи роботи з системою інженерних та науково-технічних обчислень MATLAB

#### 1. Склад та функціональне призначення системи MATLAB

Сучасна комп'ютерна математика пропонує цілий набір інтегрованих програмних систем та пакетів програм для автоматизації математичних обчислень: Eureka, Gauss, TK Solver!, Derive, Mathcad, Mathematica, Maple та ін. Виникає питання: «Яке місце у цьому займає система MATLAB?» MATLAB – одна з найстарших, ретельно пророблених та перевічених часом систем автоматизації математичних обчислень, побудована на розширеному подаванні та застосуванні матричних операцій. Це знайшло відображення в назві системи MATLAB – MATrix LABoratory (MATрична ЛАБораторія).

Типове використання MATLAB – це:

- створення алгоритмів;
- математичні обчислення
- моделювання;
- аналіз даних, дослідження і візуалізація;
- наукова і інженерна графіка;
- розробка додатків, включаючи створення графічного інтерфейсу.

Дана система може бути використовувана в областях науки та техніки, таких як електро- і радіотехніка, динаміка, акустика, енергетика, економіка. MATLAB дозволяє виконувати матричний аналіз, розв'язання задач математичної фізики, статистичних, оптимізаційних та фінансово-економічних задач, дослідження та обробку сигналів та зображень, обробку та візуалізацію інформації.

Але найбільшою мірою це середовище орієнтоване на виконання інженерних задач, тому що його математичний апарат базується на обчисленнях з матрицями та комплексними числами. MATLAB містить велику кількість процедур та функцій, необхідних інженеру та науковому працівнику для виконання складних обчислень та моделювання поведінки технічних та фізичних систем.

Система MATLAB складається з п'яти основних частин:

1) MATLAB – це мова матриць і масивів високого рівня з управлінням потоками, функціями, структурами даних, введенням та виведенням даних і особливостями об'єктно-орієнтованого програмування;

2) середовище MATLAB. Це набір інструментів і пристосувань, з якими працює користувач або програміст MATLAB. Воно включає засоби для управління змінними в робочій області пам'яті MATLAB, введення і виведення даних, а також створення, контролю і відлагодження m-файлів і додатків MATLAB;

3) графічна система MATLAB включає команди високого рівня для візуалізації дво- і тривимірних даних, обробки зображень, анімації і

ілюстрованої графіки. Вона також включає команди низького рівня, що дозволяють повністю редагувати зовнішній вигляд графічної інформації таким чином, як при створенні графічного інтерфейсу користувача (*GUI*) для MATLAB додатків;

4) бібліотека математичних функцій – це велика та різноманітна колекція обчислювальних алгоритмів від елементарних функцій, таких як сума, синус, косинус, комплексна арифметика, до складніших, таких як обернення матриць, знаходження власних значень, функції Беселя, швидке перетворення Фур'є;

5) програмний інтерфейс – це бібліотека, що дозволяє писати програми на C і Fortran, які взаємодіють з MATLAB. Вона включає засоби для виклику програм з MATLAB (динамічний зв'язок): MATLAB викликається як обчислювальний інструмент і для читання або запису *m*-файлів.

*Simulink*, супутня MATLAB програма, – це інтерактивна система для моделювання нелінійних динамічних систем. Вона є середовищем, яке управляється мишкою, що дозволяє моделювати процес шляхом перетягування і маніпуляції блоків діаграм на екрані. *Simulink* працює з лінійними, нелінійними, безперервними, дискретними, багатовимірними системами.

*Blocksets* – це доповнення до *Simulink*, які забезпечують бібліотеки блоків для спеціалізованих додатків, таких як зв'язок, обробка сигналів, енергетичні системи.

## 2. Режими роботи у середовищі MATLAB

Сеанс роботи з MATLAB прийнято називати сесією (*session*). Сесія, по суті, є поточним документом, що відображає роботу користувача з системою MATLAB. У ній є рядки введення, виведення і повідомлень про помилки. Визначення змінних і функцій, які входять у сесію і розташовані в робочій області пам'яті, але не саму сесію, можна записати на диск (файли формату *.mat*), використовуючи команду *save* (зберегти). Команда *load* (завантажити) дозволяє зчитати з диска дані робочої області. Фрагменти сесії можна оформити у вигляді щоденника за допомогою команди *diary* (щоденник).

Робота в середовищі MATLAB може здійснюватися у двох режимах:

1) режим калькулятора. Обчислення можна виконувати у режимі прямих обчислень, тобто без підготовки програми. Робота з системою у такому режимі носить характер діалогу і відбувається згідно правилу «поставив питання, отримав відповідь». Користувач набирає на клавіатурі обчислювальний вираз, редагує його (коли треба) у командному рядку та завершує введення інформації натисканням клавіші *<Enter>*.

2) програмний режим. Це режим виклику програми, що створена та записана на диск мовою MATLAB і містить усі необхідні команди, які забезпечують введення даних, організацію обчислень та виведення результатів на екран.

У командному вікні у режимі діалогу проводяться обчислення та активізуються елементи середовища MATLAB. Виконавець завдання вводить команди або запускає на виконання файли з текстами на мові MATLAB.

Інтерпретатор здійснює обробку введених даних та формує результати: числові та текстові дані, попередження та інформацію про помилки.

В обох режимах користувачу доступні практично всі обчислювальні можливості системи, у тому числі виведення інформації у графічній формі. Програмний режим дозволяє зберігати розроблені обчислювальні алгоритми і без додаткових операцій повторювати обчислення з іншими початковими даними.

### 3. Інтерфейс системи MATLAB

Інтерфейс MATLAB відповідає сучасним канонам. Він багатовіконний і має низку засобів прямого доступу до різних компонентів системи (рис. 6.1).

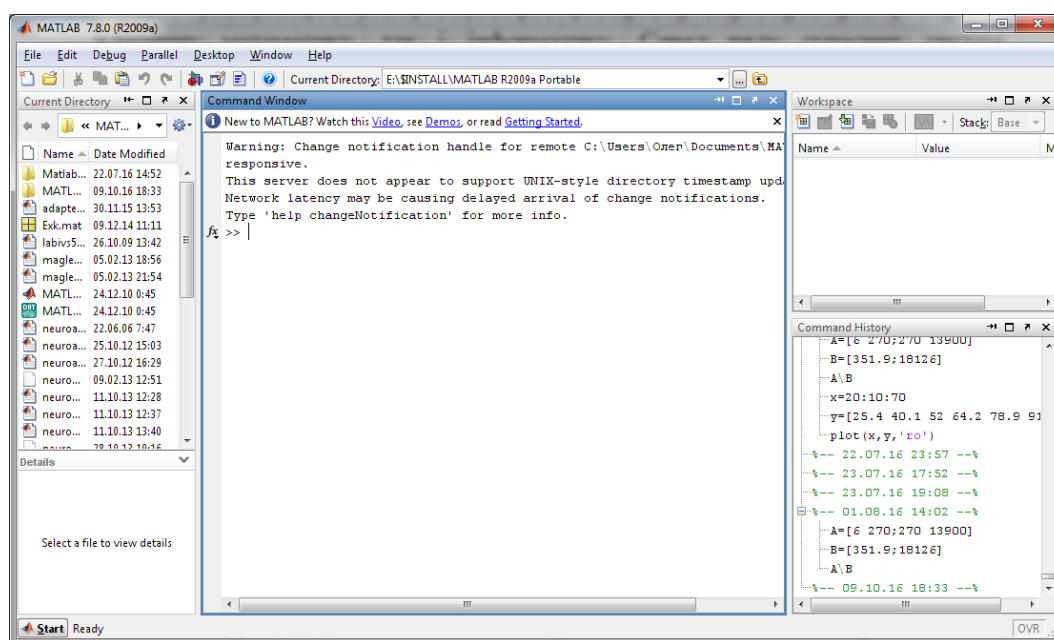


Рисунок 6.1 – Робоче середовище системи MATLAB

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 командного рядка), призначене для введення команд і виведення результатів;
- вікно *Command History* – для перегляду і повторного виконання раніше введених команд;
- рядок стану та кнопка *Start*.

При натисканні на кнопку *Start* відкривається меню, приклад якого зображено на рис. 6.2, з його допомогою забезпечується доступ до усіх основних засобів системи.

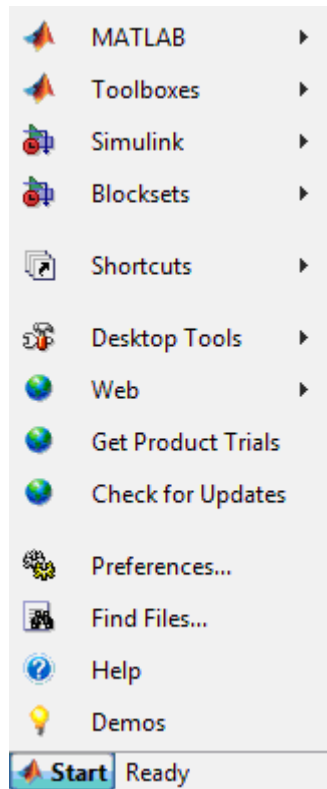


Рисунок 6.2 – Меню, яке відкривається при натисканні на кнопку *Start*

Вікно *Command Window* складається з наступних елементів:

- заголовка з назвою вікна і двома кнопками справа;
- робочої області з командним рядком, в якій знаходиться мигаючий вертикальний курсор;
- смуг прокрутки.

В полі назви кожного вікна, поряд з кнопкою закриття, 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 програми в робочій області вікна *Command Window* з'являються два посилання: *MATLAB Help* і *Demos* – для виклику довідкової системи або демонстраційних прикладів.

В довідковій системі MATLAB є декілька способів одержання інформації про команди та роботу з ними:

- команда *help*;
- операція меню *Help*;
- гіпертекстова система *Help Desk*.

Основний доступ до довідкової інформації забезпечує операція меню *Help*. Воно відкриває доступ до довідкової системи MATLAB та до інформації про виробника. Меню *Help* складається з команд: *Full Product Family Help*, *Help MATLAB*, *Using the Desktop*, *Using the Command Windows*, *Demos*, *About MATLAB*.

У вікні *About MATLAB* можна знайти інформацію про застосовувану версію системи.

Команда *Demos* відкриває вікно *MATLAB Demo Window*, у якому можна ознайомитись з прикладами основних команд або процедур системи MATLAB, її обчислювальними та графічними можливостями, а також з текстами програм, за допомогою яких створюються продемонстровані результати. Подібне вікно можна створити при виконанні команди *demo* у режимі діалога.

Запуск довідкової системи здійснюється за допомогою команди *Help* меню MATLAB. При цьому запускається браузер довідкової системи. Кожен розділ довідкової системи представлений у правому вікні у вигляді гіпертекстового посилання, активізація якого приводить до переходу на відповідну *html*-сторінку. За допомогою лінійки прокручування можна переміщатися по списку документів довідкової системи та вибирати різні теми довідки.

#### 4. Типи даних, з якими оперує MATLAB

Система MATLAB є інтерактивною системою для виконання інженерних та наукових розрахунків, орієнтованою на роботу з масивами даних. Система використовує математичний співпроцесор і дає можливість звертання до програм, написаних на мовах Fortran, C та C++.

MATLAB підтримує виконання операцій з векторами, матрицями і масивами даних (включаючи операції з комплексними числами), реалізує сингулярне та спектральне розкладання, обчислення рангу та чисел зумовленості матриць, підтримує роботу з алгебраїчними поліномами, розв'язання нелінійних рівнянь і задач оптимізації, розв'язання диференціальних і різницевих рівнянь, побудову різноманітних видів графіків, тривимірних поверхонь та ліній рівня. В системі реалізовано зручне операційне середовище, яке дозволяє формулювати проблеми і отримувати рішення у звичній математичній формі, не звертаючись до рутинного програмування.

MATLAB оперує з такими типами даних:

- дійсні числа;
- комплексні числа;
- змінні, які, в свою чергу, бувають:
  - а) числові;
  - б) векторні;
  - в) символьні;
  - г) системні (*i, j, pi, e, ans, Inf, NaN, realmin, realmax*).

Імена змінних і функцій можуть складатися з будь-яких символів алфавіту системи MATLAB, окрім спеціальних. Ім'я обов'язково повинно починатися з букви, за якою може йти довільне сполучення букв та цифр. Будь-яка змінна інтерпретується в системі MATLAB як двовимірний масив чисел (матриця розмірності  $(m \times n)$ ).

## 5. Виконання обчислень в MATLAB

Основним об'єктом системи MATLAB є прямокутний числовий масив, що допускає комплексні елементи і введення матриць без явного указування їх розмірів. Система дозволяє вирішувати багато обчислювальних проблем за більш короткий час, ніж той, який був би потрібен для написання відповідних програм на мовах Fortran, Basic чи C.

Розглянемо основні прийоми програмування в системі MATLAB. Мову, що використовується в MATLAB, можна порівняти з мовою Basic за простотою її застосування і принципом безпосереднього виконання. Мова MATLAB – це мова операторів. Оператори, які вводяться користувачем в командному рядку, виконуються в режимі інтерпретації. Оператори мають дві форми запису:

1) з явним присвоюванням:

змінна = вираз;

2) з неявним присвоюванням:

вираз.

Якщо обчислюваний вираз не присвоюється змінній, то створюється змінна *ans* (від слова *answer* – відповідь), у яку і записується результат виконання команди. Оператори складаються із спеціальних символів, імен функцій і змінних, а також числових констант, і можуть закінчуватися комою або крапкою з комою. Якщо після оператора не стоїть крапка з комою, то результат виконання команди виводиться у командному вікні *Command Window*. Якщо ж крапка з комою стоїть в кінці оператора, то результат не виводиться в командному вікні, але (як і в попередньому випадку) записується в робочу область пам'яті *Workspace*.

Приклад:

1) визначення скаляра

```
>> x=5.213
```

```
x =
```

```
5.2130
```

2) визначення масива з 5 елементів (матриці розмірності (1×5))

```
>> x=[1 2 3 4 5]
```

```
x =
```

```
1         2         3         4         5
```

### 3) визначення матриці розмірності (3×4)

```
>> A=[0.1 -2 0 3; -1 0 2 4; 7.5 -4.25 8 256]
```

```
A =
```

```
    0.1000    -2.0000         0     3.0000  
   -1.0000         0     2.0000     4.0000  
    7.5000   -4.2500     8.0000   256.0000
```

Існує можливість визначення масивів з використанням символу <: >

```
1) >> x=1:7
```

```
x =
```

```
    1     2     3     4     5     6     7
```

```
2) >> x=0:0.1:0.5
```

```
x =
```

```
    0    0.1000    0.2000    0.3000    0.4000    0.5000
```

Символ <: > використовується також для індексації елементів масивів.

Слід зазначити, що всі арифметичні операції над матрицями (масивами) в MATLAB виконуються за правилами лінійної алгебри. Додавання і віднімання матриць здійснюється поелементно

```
>> A=[1 2 3;4 5 6]
```

```
A =
```

```
    1     2     3  
    4     5     6
```

```
>> B=[0 -1 5;4 2 -11]
```

```
B =
```

```
    0    -1     5  
    4     2   -11
```

```
>> A+B
```

```
ans =
```

```
    1    1    8  
    8    7   -5
```

```
>> A-B
```

```
ans =
```

```
    1    3   -2  
    0    3   17
```

Проте, якщо ми спробуємо перемножити матриці A і B у наведеному вище прикладі, то отримаємо повідомлення про помилку

```
>> A*B
```

```
??? Error using ==> mtimes  
Inner matrix dimensions must agree.
```

Така ситуація виникає тому, що для перемноження матриць в лінійній алгебрі необхідно, щоб кількість стовпчиків першої матриці дорівнювала кількості рядків другої матриці, а у нашому прикладі ця умова не виконується. Для того, щоб поелементно перемножити або розділити одну матрицю на іншу, перед знаком операції необхідно ставити крапку

```
>> A.*B
```

```
ans =
```

```
    0   -2   15  
   16   10  -66
```

```
>> B./A
```

```
ans =
```

```
    0   -0.5000    1.6667  
   1.0000    0.4000   -1.8333
```

Для виконання операції транспонування матриці застосовується знак '

```
>> A=[1 2 3;4 5 6]
```

```
A =
```



```
1 2 3
4 5 6
```

```
>> A'
```

```
ans =
```

```
1 4
2 5
3 6
```

Визначник матриці обчислюється за допомогою функції `det`, а обернена матриця – за допомогою функції `inv`

```
>> G_15=[1 3 5;-4 7 9;12 -6 3]
```

```
G_15 =
```

```
1 3 5
-4 7 9
12 -6 3
```

```
>> det(G_15)
```

```
ans =
```

```
135
```

```
>> inv(G_15)
```

```
ans =
```

```
0.5556 -0.2889 -0.0593
0.8889 -0.4222 -0.2148
-0.4444 0.3111 0.1407
```

У командному рядку можна вводити різноманітні математичні вирази з використанням вбудованих у MATLAB функцій. Наприклад, необхідно обчислити наступний вираз:

$$e^{-2.5}(\ln 11.3)^{0.3} - \sqrt{\frac{\sin 2.45\pi + \cos 3.78\pi}{\operatorname{tg} 3.3}}$$

Для цього в командному рядку 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))  
  
ans =  
  
-3.2105
```

Арифметичні операції в MATLAB виконуються в звичайному порядку, властивому для більшості мов програмування.

При проведенні обчислень з комплексними числами в командному рядку 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
```

## 6 Побудова графіків

Система MATLAB має широкі можливості для графічного представлення результатів обчислень і візуалізації даних.

Виведення відображення функції  $y(x) = e^{-x} \sin 10x$  у вигляді графіка складається з наступних етапів:

- задання вектора значень аргумента  $x$ ;
- обчислення вектора  $y$  значень функції  $y(x)$ ;
- виклик команди `plot` для побудови графіка.

Команди для задання 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. 6.3).

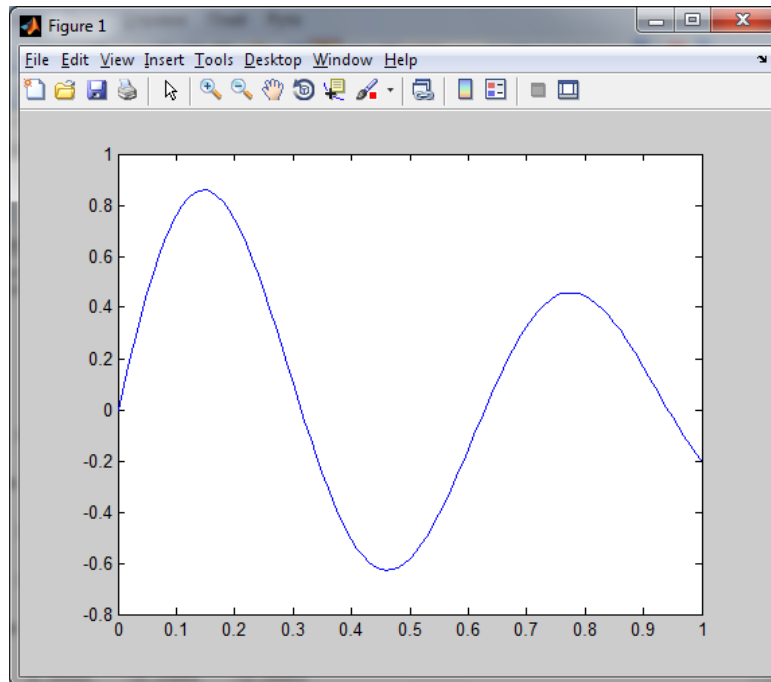


Рисунок 6.3 – Графік функції

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. Для цього використовується такий формат функції `plot`:

```
plot(x, y, s);
```

де  $x$  – одновимірний масив значень аргументу,

$y$  – одновимірний масив значень функції,

$s$  – рядкова змінна, яка задає спосіб відображення графіка, колір ліній і маркерів; може включати до трьох символів із табл. 6.1.

Якщо необхідно побудувати графіки декількох функцій в одній системі координат, можна скористатися таким форматом функції `plot`:

```
plot(x1, y1, s1, x2, y2, s2, ...)
```

Приклад: побудова графіків функцій  $y = \sin x$  та  $y = e^{-x}$


```
>> x=0:pi/30:2*pi;
```

```
>> y1=sin(x); y2=exp(-x);
```

```
>> plot(x, y1, 'b-*', x, y2, 'r--');
```

Таблиця 6.1 – Можливі значення параметра  $s$  функції plot

Тип лінії		Тип маркера		Колір	
Суцільна	-	Точка •	.	Жовтий	y
Штрихова	--	Плюс +	+	Фіолетовий	m
Подвійний пунктир	:	Зірочка *	*	Голубий	b
Штрих-пунктирна	-. .	Кружок ○	o	Червоний	r
		Хрестик ×	x	Зелений	g
		Квадрат ■	s	Синій	b
		Ромб ◆	D	Білий	w
		Трикутник ▼	V	Чорний	k
		Трикутник ▲	A		
		Трикутник ◀	<		
		Трикутник ▶	>		
		П'ятикутник ◆	P		
		Шестикутник ◆	H		

Результат виконання цієї послідовності команд показано на рис. 6.4. Для того, щоб змінити параметри відображення графічних зображень у інтерактивному режимі, необхідно в меню графічного вікна натиснути на зображення стрілочки , після цього вибрати на полі графіка потрібний об'єкт і подвійним клацанням мишки на ньому увійти в режим редагування графічних властивостей (рис. 6.5). Тут за допомогою зручного графічного інтерфейсу можна вибрати тип, колір та товщину лінії, якою відображається графік, тип, розмір, колір заливки та колір обрамлення маркера у вузлових точках та інші параметри графічного об'єкта.

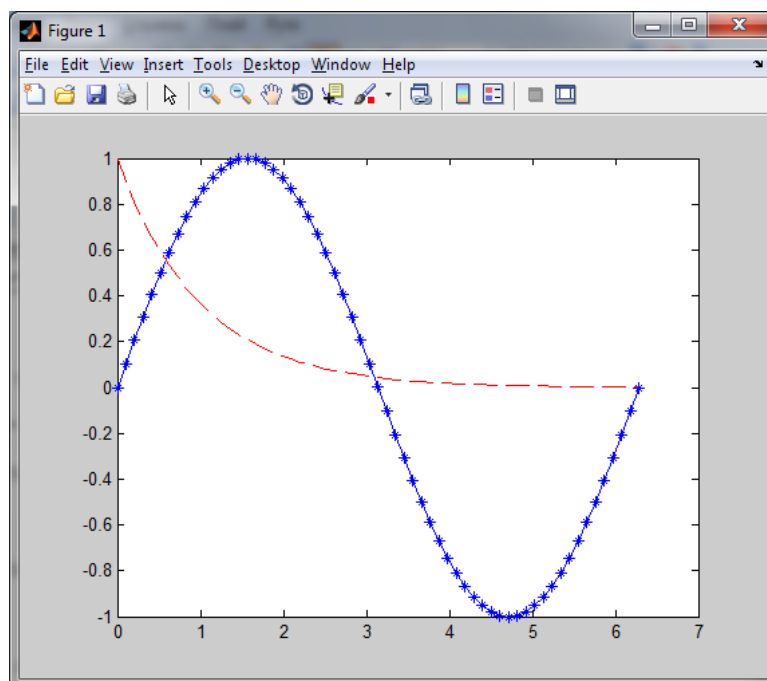


Рисунок 6.4 – Побудова графіків декількох функцій

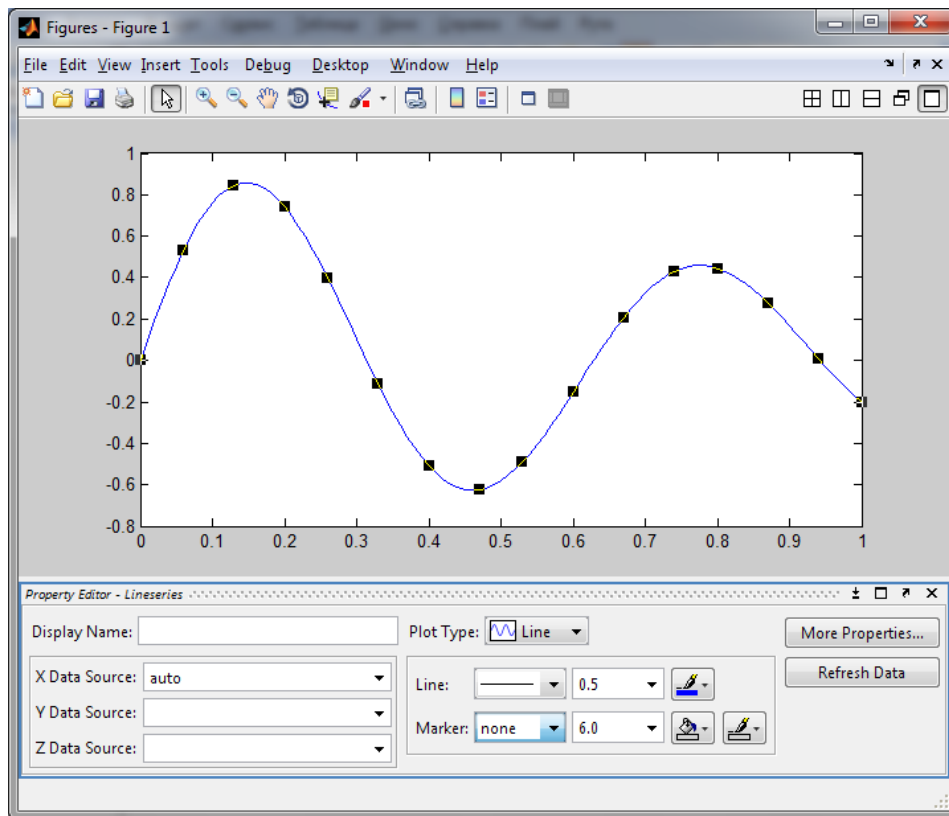


Рисунок 6.5 – Інтерактивне редагування графічних властивостей