**Лабораторна робота №1**

**Варіант №7**

**Тема:** Основи програмування MATLAB. Рівняння, масиви та функції.

**Мета:** ознайомитися з основами програмування MATLAB; розглянути рівняння, масиви та функції.

**Теоретичні відомості**

Інтегрована система автоматизації математичних та науково-технічних розрахунків MATLAB є добре апробованою надійною системою розв‘язання математичних задач з поданням даних у вигляді матриць. Вона розроблена фірмою MathWorks Inc.

При створенні багатьох автоматизованих систем математичних обрахунків розробники прагнули уникнути необхідності написання користувачем власних програмних кодів. Але такий підхід значно обмежував можливості користувача при розв‘язанні задач, що використовують нові методи та алгоритми.

Розробники MATLAB провели величезну роботу по створенню операторів і функцій для розв‘язання багатьох практичних задач. До таких функцій належать як прості (транспонування матриць, обчислення косинуса та ін.) так і складні функції, що реалізують певні методи розв‘язання задач. Число таких функцій сягає багатьох тисяч і невпинно збільшується.

А для вирішення задач з використанням алгоритмів користувача, MATLAB має потужну математично-орієнтовану мову програмування високого рівня. Тому інтегрована система автоматизації математичних MATLAB з успіхом застосовується в багатьох країнах Європи для розв‘язання нових, нестандартних та найбільш складних математичних задач.

Запис коду програми відбувається у редакторі системи, що викликається з рядка головного меню MATLAB командами **File → New → M-file**, але попередньо може бути написаний у будь-якому текстовому редакторі. Файл коду програми називається m-файлом і має розширення .m, тобто можемо записати: filename.m Система MATLAB має власний редактор і відладчик. Формули створюваних методів записуються у формі, притаманній мовам програмування, але ця незручність компенсується високою швидкістю обрахунків, яка наприклад на порядок вища ніж у системи MathCad.

#### Вбудовані елементарні функції

Вбудовані елементарні функції MATLAB включають тригонометричні, гіперболічні, експоненціальні і логарифмічні функції, для округлення різними способами.

Тригонометричні і зворотні до них функції:

* **sin, соs, tan, cot** - синус, косинус, тангенс і котангенс;
* **sec, csc** - секанс, косеканс;
* **asin, acos, atan, acot** - арксинус, арккосинус, арктангенс і арккотангенс;
* **asec, acsc** - арксеканс, арккосеканс.

Аргументи тригонометричних функцій повинні бути виражені в радіанах. Обернені тригонометричні функції повертають результат також в радіанах.

Експоненціальна функція, логарифми, степеневі функції:

* **exp** – експоненціальна функція;
* **log** – натуральний логарифм;
* **log10** – десятковий логарифм;
* **log2** – логарифм за основою 2;
* **pow2** – піднесення до степеня 2;
* **sqrt** – квадратний корінь
* **nextpow2** – степінь, до якої потрібно піднести число 2, щоб отримати найближче число (більше або рівне аргументу)

**Виведення графіків. Процедура plot**

Виведення графіків у системі Matlab є настільки простою і зручною процедурою, що нею можна користуватися навіть при обчисленнях у режимі калькулятора. Основною функцією, яка забезпечує побудову графіків на екрані дисплея, є функція plot. Загальна форма звернення до цієї процедури така:

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

Тут x1, y1 – задані вектори, елементами яких є масиви значень аргументу (х1) і функції (у1), що відповідають першої кривої графіка; x2, y2 – масиви значень аргументу й функції другої кривої і т.д. При цьому передбачається, що значення аргументу відкладаються уздовж горизонтальної осі графіка, а значення функції – уздовж вертикальної осі. Змінні s1, s2,... є символьними (їхня вказівка не є обов'язковою). Кожна з них може містити до трьох спеціальних символів, що визначають відповідно: а) тип лінії, що з'єднує окремі точки графіка; б) тип точки графіка; в) колір лінії. Якщо змінні s не зазначені, то тип лінії за умовчанням - відрізок прямої, тип точки – піксель, а колір установлюється за такою черговістю: - синій, зелений, червоний, блакитний, фіолетовий, жовтий, чорний і білий - у залежності від того, яка по черзі лінія виводиться на графік.

Якщо потрібно другий графік провести "поверх першого графіка", то перед виконанням другої графічної команди plot, потрібно виконати команду **hold on**, яка призначена для утримання поточного графічного вікна.

ezplot('f(x)',limits) будує графік f(x) на зазначеному інтервалі і в заданому вікні.

**Порядок виконання**

1. Розв’язання функції:

Код:

x=3.2:0.6:6.2;

y=(a.^x-b.^x)/(log10(a/b))\*sqrt(a\*b).^(1/3);

Відповідь:

y =

1.1989 1.0922 0.9805 0.8721 0.7712 0.6794

2. Квадратичне рівняння:

Приклад розв’язку квадратичного рівняння: *x2 - 2x - 4 = 0*

Код: Відповідь:

solve('x^(2)-2\*x-4') ans =

1 - 5^(1/2)

5^(1/2) + 1

Завдання до варіанту: *x2 - 3x - 4 = 0*

Код: Відповідь:

solve('x^(2)-3\*x-4') ans =

-1

4

3. Побудова графіку з використанням функції plot:

Код:

x=[1 2 3 4 5];

y=[0.3 5 7 8 9.5];

plot(x,y)

Графік:

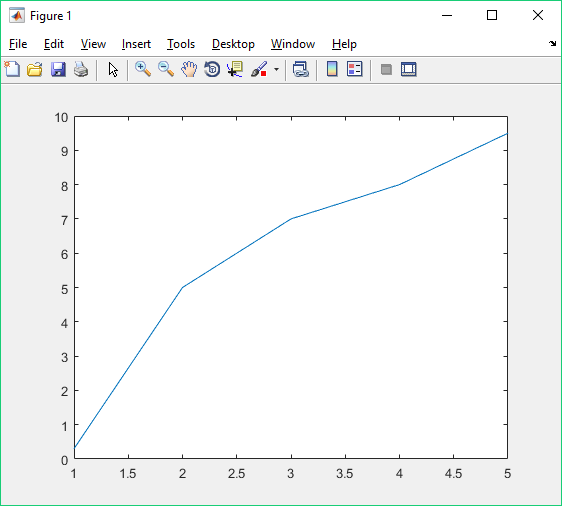


Рис. 1. Побудова графіку за допомогою функції plot

4. Побудова графіку з використанням функції ezplot:

Код:

ezplot('x^(2)-3\*x-4',[-17;21])

Графік:

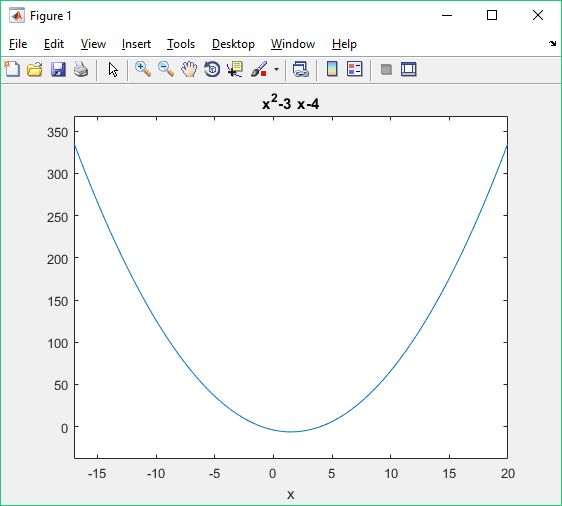


Рис. 2. Побудова графіку за допомогою функції ezplot

5. Побудова декількох кривих:

Код:

ezplot('exp(-x)',[0;10])

hold on

ezplot('sin(x)',[0;10])

hold off

title ('exp(-x),sin(x)')

Графік:

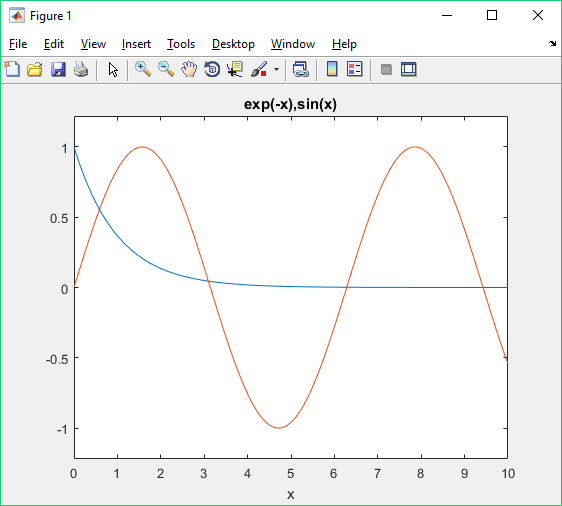


Рис. 3. Побудова декількох кривих

**Висновок.** Під час виконання лабораторної роботи я ознайомилась з основами програмування MATLAB, а також розглянула рівняння, масиви та функції.