

به نام خدا

آموزش متلب

مدرس:

دکتر فریده قریشی، زهرا مهدور

راه های ارتباطی:

Email : zahra.mahdevar@yahoo.com

Telegram : Z_Mhdr

جلسه سوم: آموزش مصورسازی داده ها و رسم نمودارها

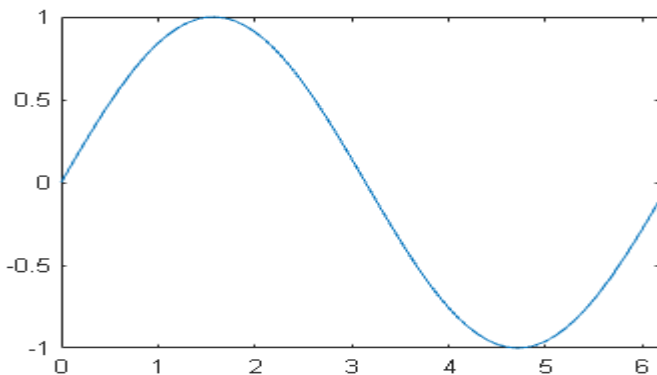
مقدمه

در متلب، رسم نمودار یکی از ابزارهای قدرتمند برای تجسم داده ها و تحلیل نتایج است. با استفاده از این ابزار، می توان اطلاعات پیچیده را به صورت تصویری نمایش داد و الگوها و روندهای موجود در داده ها را به راحتی شناسایی کرد. این ویژگی به ویژه در تحقیقات علمی و مهندسی برای بررسی نتایج و ارزیابی مدل ها بسیار مفید است. متلب انواع نمودارها مانند دو بعدی، سه بعدی و نمودارهای خاص مانند قطبی و هیستوگرام را برای تحلیل داده ها فراهم می کند

بخش اول : رسم نمودارهای دو بعدی

plot

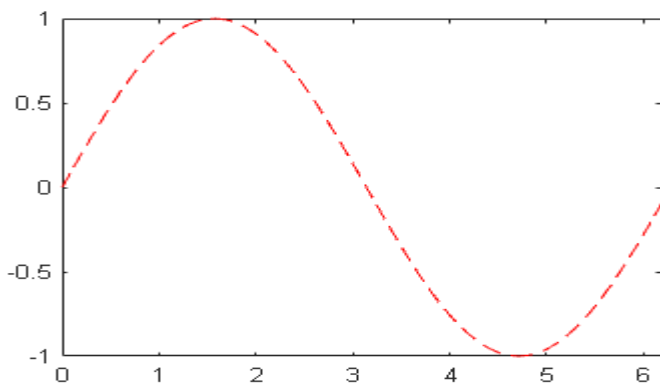
```
x = 0:0.1:2*pi; % X-axis  
y = sin(x); % Calculating the sine function  
plot(x,y) % Drawing a diagram
```



تغییر رنگ و سبک خط

```
x = 0:0.1:2*pi; % X-axis
```

```
y = sin(x);           % Calculating the sine function
plot(x,y, 'r--')      % Drawing a diagram
```



جدول رنگ‌های پرکاربرد

```
colors = {'قرمز' ; 'سبز' ; 'آبی' ; 'مشکی'};
code_colors = {'r'; 'g'; 'b'; 'k'};

T_color = table(colors, code_colors, ...
    'VariableNames', {'رنگ‌های پرکاربرد', 'کد رنگ‌ها'});

disp(T_color)
```

کد رنگ‌ها رنگ‌های پرکاربرد

{ 'قرمز' }	{ 'r' }
{ 'سبز' }	{ 'g' }
{ 'آبی' }	{ 'b' }
{ 'مشکی' }	{ 'k' }

جدول سبک خط‌ها

```
lines = {'خط پیوسته' ; 'خط چین' ; 'نقطه چین' ; 'خط نقطه'};
code_lines = {'-', '--', '...', '-.'};

T_line = table(lines, code_lines, ...
    'VariableNames', {'سبک خط‌ها', 'کد سبک خط‌ها'});

disp(T_line)
```

کد سبک خط‌ها سبک خط‌ها

```
{'خط پیوسته' } {'-' }
{'خط چین' } {'--'}
{'نقطه چین' } {'...'}
{'خط نقطه' } {'.-'}
```

جدول نشانگرها

```
markers = {'مربع' ; 'ستاره' ; 'جمع' ; 'دایره'};
code_markers = {'o' ; '+' ; '*' ; '□'};

T_mark = table(markers, code_markers, ...
    'VariableNames', {'کد نشانگرها', 'نشانگرها'});

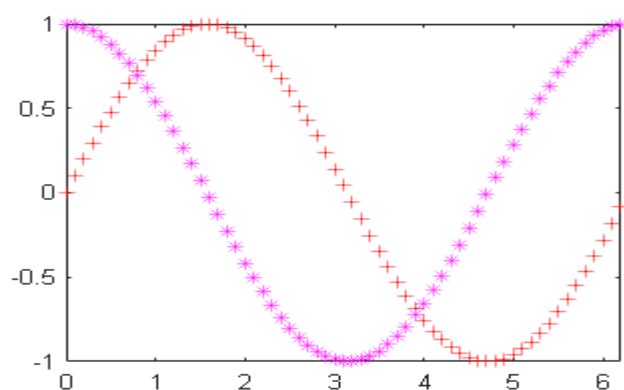
disp(T_mark)
```

کد نشانگرها نشانگرها

```
{'دایره'}      {'o'}
{'جمع' }      {'+' }
{'ستاره'}      {'*' }
{'مربع' }      {'□'}
```

رسم چند تابع در یک نمودار

```
x = 0:0.1:2*pi;
y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'r+', x, y2, 'm*')
```

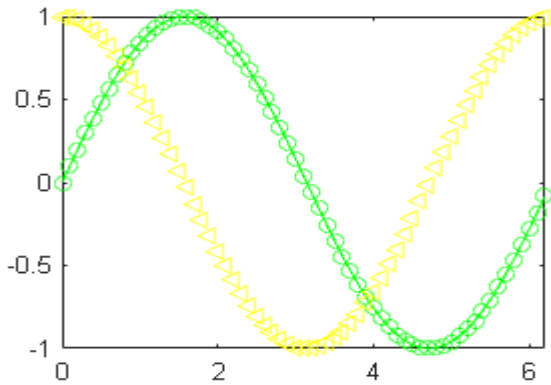


```
x = 0:0.1:2*pi;
```

```

y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'g-o')
hold on           % The hold on command allows a new graph
                  % to be drawn without erasing the previous one.
plot(x, y2, 'y<')
hold off          % At the end of the work, this mode can be
                  % deactivated with hold off.

```

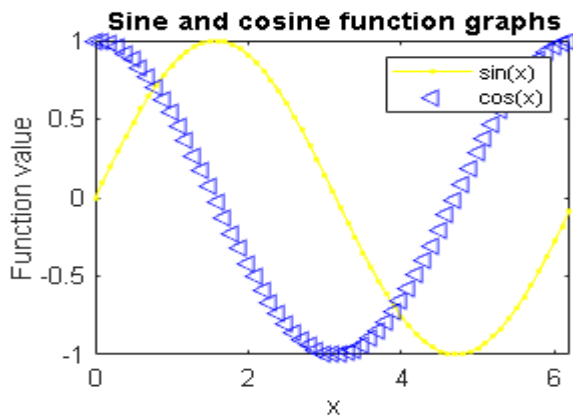


افزودن عنوان و برجسب محورها

```

x = 0:0.1:2*pi;
y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'y.-', x, y2, 'b<')
xlabel('x')
ylabel('Function value')
legend('sin(x)', 'cos(x)')
title('Sine and cosine function graphs')

```



تنظیم محورها و شبکه‌بندی

```

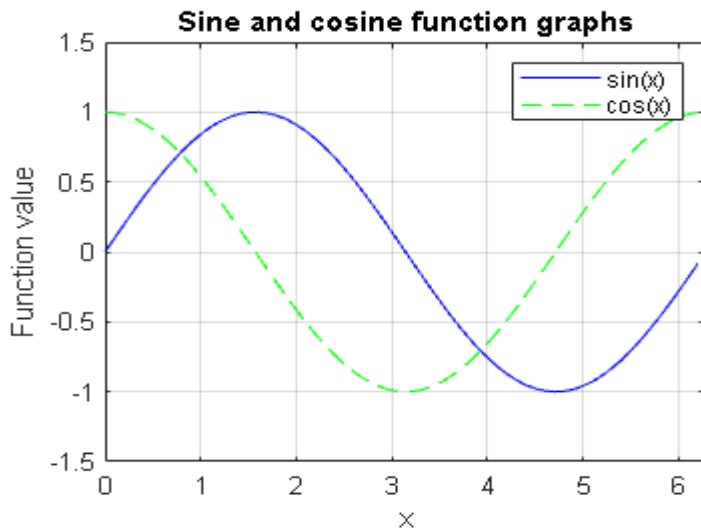
x = 0:0.1:2*pi;
y1 = sin(x);
y2 = cos(x);
plot(x, y1, 'b', x, y2, 'g--')

```

```

xlabel('x')
ylabel('Function value')
legend('sin(x)', 'cos(x)')
title('Sine and cosine function graphs')
axis([0 2*pi -1.5 1.5])      % x and y axis range
grid on                      % Creating grid lines on a chart

```

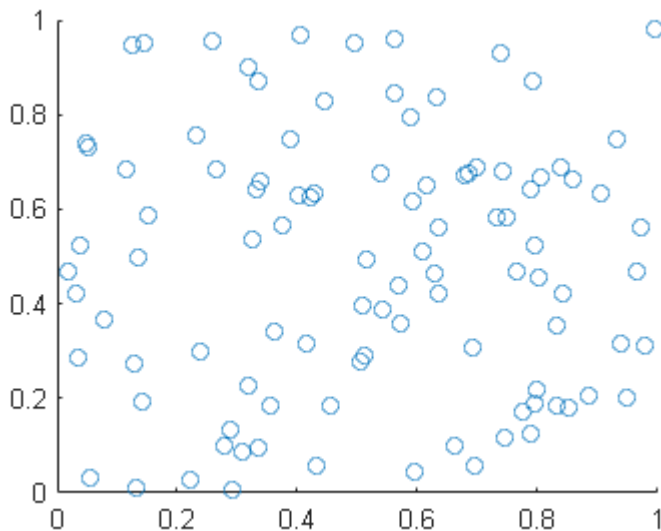


رسم نمودار نقطه‌ای

```

a = rand(1, 100);
b = rand(1, 100);
scatter(a, b)

```



رسم چند نمودار در یک شکل

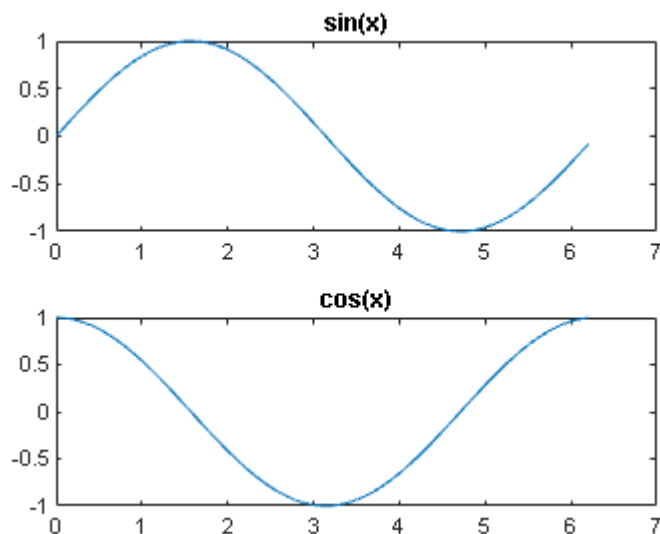
```

%subplot(m, n, p) divides the window shape into m×n sections
% and draws the graph in section number p.
x = 0:0.1:2*pi;
subplot(2, 1, 1)

```

```
plot(x, sin(x))
title('sin(x)')

subplot(2, 1, 2)
plot(x, cos(x))
title('cos(x)')
```



ذخیره‌سازی نمودار

```
saveas(gcf, 'sin_cos.png') % Saved as PNG image
print('-dpdf', 'sin_cos.pdf') % Save as PDF
```

practice 1: Write a MATLAB script to plot the following three functions on the same graph over the interval -5 to 5 :

1. $y_1 = \sin(10x) \cdot e^{-x^2}$
2. $y_2 = |x| \cdot \sin(x)$
3. $y_3 = 1 / (1 + e^{-x})$

Make sure to use different line styles or colors for each function, add a legend, label the axes, and include a title for the plot.

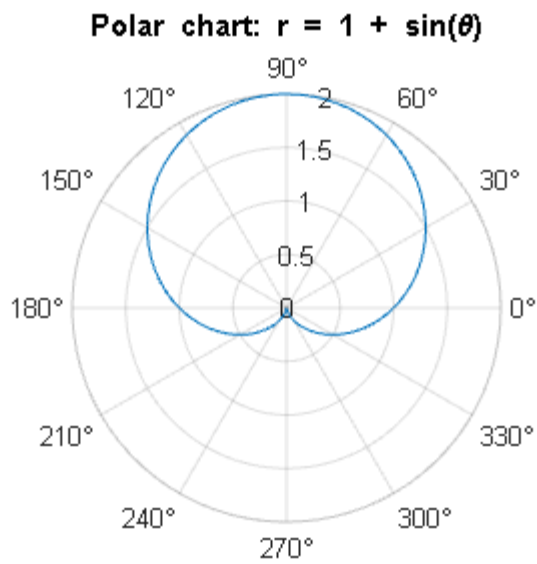
بخش دوم : رسم نمودارهای ویژه

در کنار نمودارهای خطی دوبعدی، متلب ابزارهایی برای رسم نمودارهای ویژه فراهم کرده است؛ مانند نمودارهای قطبی، سه‌بعدی، هیستوگرام، نمودارهای میله‌ای و دایره‌ای. در ادامه با رایج‌ترین آن‌ها آشنا می‌شویم

نمودار قطبی

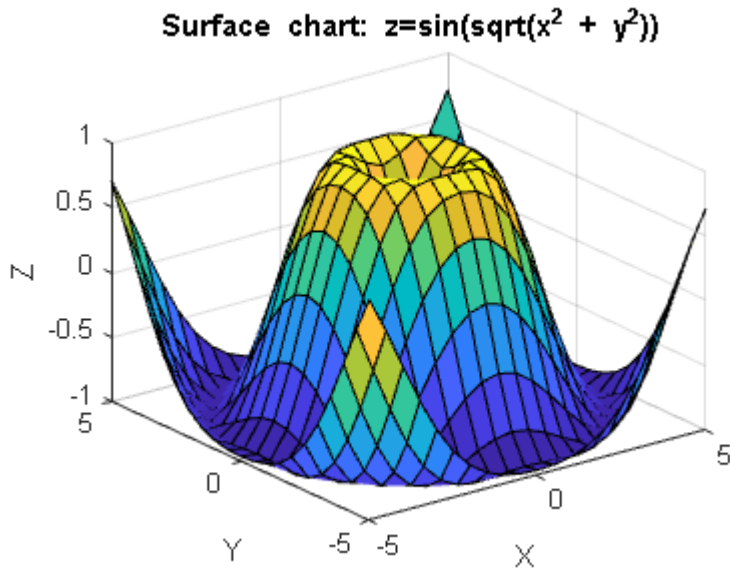
```
theta = linspace(0, 2*pi, 100); % Generating uniformly spaced vectors
r = 1 + sin(theta);
polarplot(theta, r)
```

```
title('Polar chart:  $r = 1 + \sin(\theta)$ ')
```

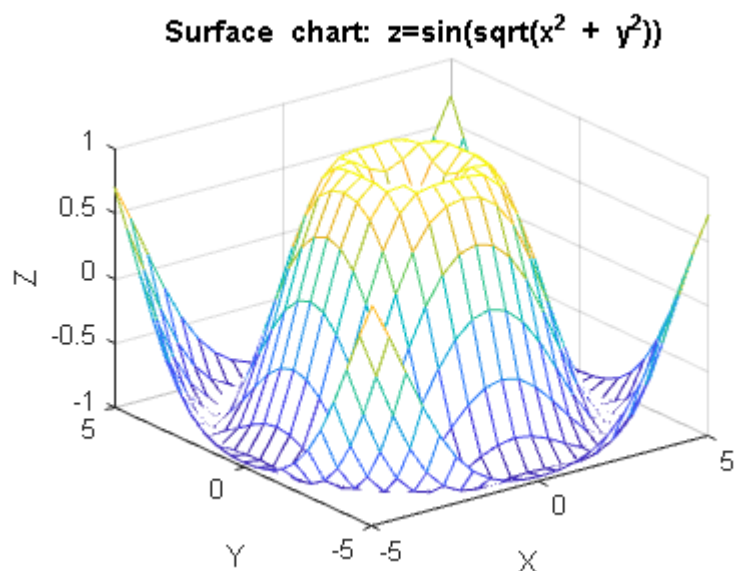


نمودار سه بعدی سطحی

```
[X, Y] = meshgrid(-5:0.5:5, -5:0.5:5);
Z = sin(sqrt(X.^2 + Y.^2));
surf(X, Y, Z)
title('Surface chart:  $z = \sin(\sqrt{x^2 + y^2})$ ')
```

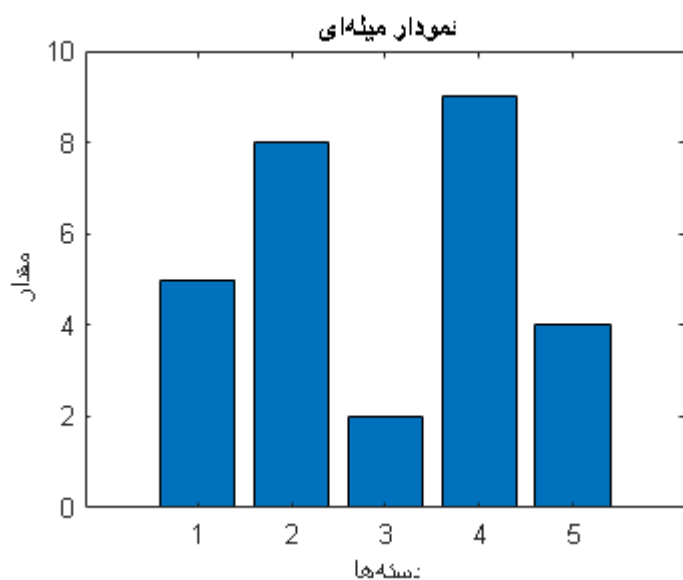


```
[X, Y] = meshgrid(-5:0.5:5, -5:0.5:5);
Z = sin(sqrt(X.^2 + Y.^2));
mesh(X, Y, Z)
title('Surface chart:  $z = \sin(\sqrt{x^2 + y^2})$ ')
```

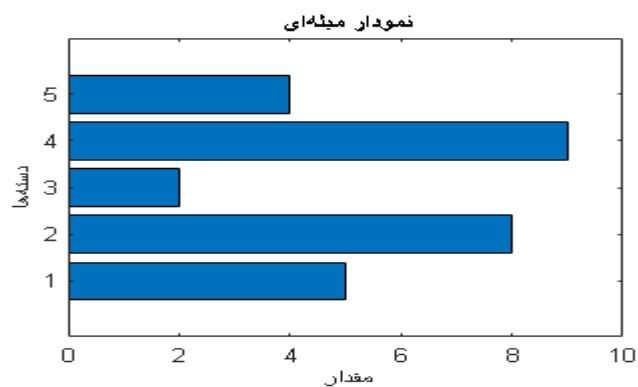


نمودار میله‌ای

```
data = [5 8 2 9 4];
bar(data)
title('نمودار میله‌ای')
xlabel('دسته‌ها'); ylabel('مقدار')
```



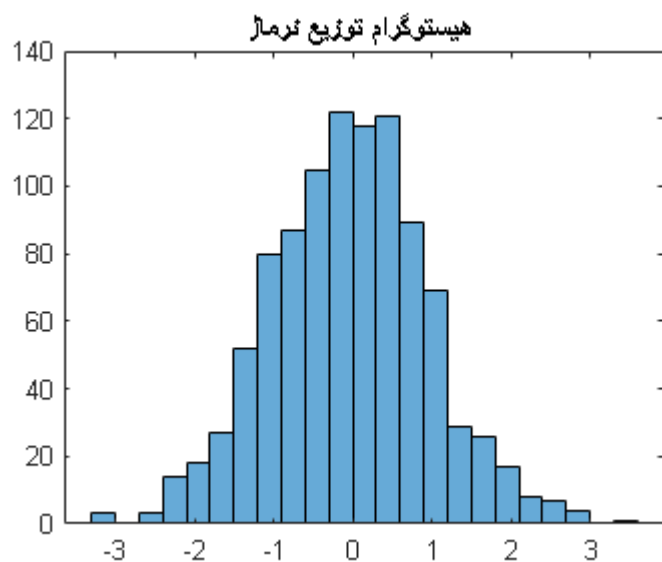
```
data = [5 8 2 9 4];
barh(data)
title('نمودار میله‌ای')
ylabel('دسته‌ها'); xlabel('مقدار')
```

هیستوگرام

برای نمایش توزیع داده‌ها از هیستوگرام استفاده می‌شود

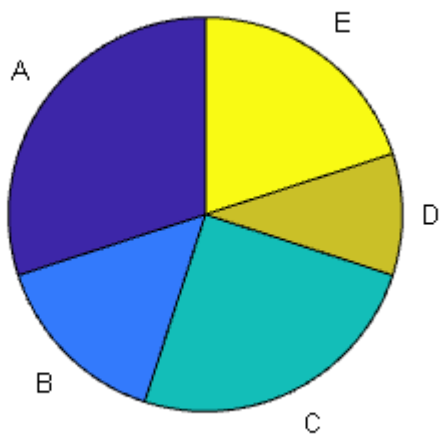
```
x = randn(1, 1000); % داده‌های تصادفی نرمال
histogram(x)
title('هیستوگرام توزیع نرمال')
```



نمودار دایره‌ای

```
values = [30 15 25 10 20];
labels = {'A', 'B', 'C', 'D', 'E'};
pie(values, labels)
title('نمودار دایره‌ای')
```

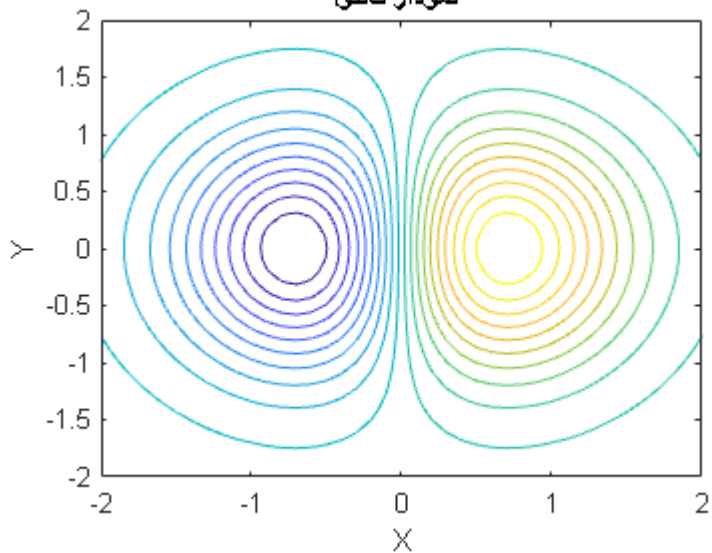
نمودار دایره‌ای



نمودار کانتور

```
[X, Y] = meshgrid(-2:0.1:2, -2:0.1:2);
Z = X .* exp(-X.^2 - Y.^2);
contour(X, Y, Z, 20)
title('نمودار کانتور')
xlabel('X'); ylabel('Y')
```

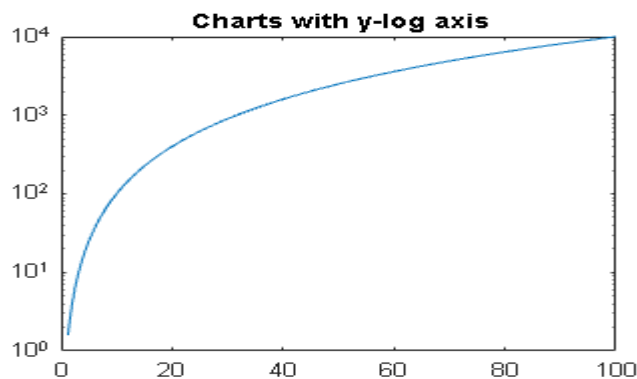
نمودار گانتو



نمودارهای لگاریتمی

```
% logspace(a, b, n)
x = logspace(0.1, 2, 100);
y = x.^2;
semilogy(x, y)
```

```
title('Charts with y-log axis')
```



practice 2 : Write a MATLAB script that includes the following tasks:

- 1. Plot the polar equation $r(\theta) = 10 \times \cos(10\theta)$ in MATLAB using the `polarplot` function. Plot the curve over the interval $\theta \in [0, 2\pi]$.
- 2. Plot the 3D surface of the Rastrigin function given by:

$$z = a \cdot n + \sum_{i=1}^n (x_i^2 - a \cdot \cos(2\pi x_i))$$

Where $a=10$, and n represents the number of dimensions. For this example, consider the 2D case where x and y both range from -5.12 to 5.12 . Use a mesh grid to generate the plot and label the axes appropriately. Visualize the surface with shading and color mapping for better clarity.

- 3. Create a visually appealing pie chart in MATLAB that displays the distribution of interests among four categories: Science, Art, Technology, and Literature. Customize the chart with specific colors and add a legend and title.

حل چند مثال

Example 1: Solve the quadratic equation $ax^2 + bx + c = 0$

```
% Define the coefficients
a = input('Enter the value of a: '); % Coefficient of x^2
b = input('Enter the value of b: '); % Coefficient of x
c = input('Enter the value of c: '); % Constant term

% Compute the discriminant (delta)
delta = b^2 - 4*a*c;

% Check the value of delta to determine the type of roots
if delta > 0
    % Two distinct real roots
    x1 = (-b + sqrt(delta)) / (2*a);
    x2 = (-b - sqrt(delta)) / (2*a);
    disp('This equation has two real roots:');
    disp(['x1 = ' num2str(x1)])
    disp(['x2 = ' num2str(x2)])
```

```

elseif delta == 0
    % One real and repeated root
    x = -b / (2*a);
    disp('This equation has two identical real roots:');
    disp(['x1 = x2 = ' num2str(x)])
else
    % Two complex (imaginary) roots
    disp('This equation has no real roots')
end

```

Example 2: Solve the roots of a quadratic equation as a function.

```

function roots = solveQuadratic(a, b, c)
% solveQuadratic --> Solves a quadratic equation of the form ax^2 + bx + c = 0
% Input:  a, b, c --> coefficients of the quadratic equation
% Output: roots --> a vector containing the roots (real or complex)

    if a == 0
        error('Coefficient "a" cannot be zero. This is not a quadratic equation.');
```

```

    end

    % Calculate the discriminant
    D = b^2 - 4*a*c;

    % Check the nature of the discriminant
    if D > 0
        % Two distinct real roots
        root1 = (-b + sqrt(D)) / (2*a);
        root2 = (-b - sqrt(D)) / (2*a);
        disp('The equation has two distinct real roots.')
    elseif D == 0
        % Two equal real roots
        root1 = -b / (2*a);
        root2 = root1;
        disp('The equation has two equal real roots.')
    else
        % Two complex conjugate roots
        realPart = -b / (2*a);
        imagPart = sqrt(-D) / (2*a);
        root1 = realPart + 1i*imagPart;
        root2 = realPart - 1i*imagPart;
        disp('The equation has two complex conjugate roots.')
    end

    % Return the roots as a vector
    roots = [root1, root2];
end

```

فراخوانی تابع

```

solveQuadratic(1, -3, 2)    % ریشه واقعی متمایز ( $x^2 - 3x + 2 = 0$ )

```

```

solveQuadratic(1, 2, 1)    % دو ریشه واقعی برابر ( $x^2 + 2x + 1 = 0$ )
solveQuadratic(1, 2, 5)    % دو ریشه مختلط مزدوج ( $x^2 + 2x + 5 = 0$ )

```

Example 3: Write a program that takes the month number (1 to 12) and prints the corresponding season.

```

month = input('Enter month number (1-12): ');

switch month
    case {1, 2 , 3}
        disp('Spring');
    case {4, 5 , 6}
        disp('Summer');
    case {7, 8 , 9}
        disp('Autumn');
    case {10 , 11 , 12}
        disp('Winter');
    otherwise
        disp('Invalid month number');
end

```

Exercise 1: Finding the roots of the equation $x=\cos(x)$, using the simple iteration method (fixed point).

Initial guess : 0.5

Stop condition: $\|x_{n+1} - x_n\| < 10^{-6}$

max_iter = 100

Exercise 2: Finding the Root of a Nonlinear Equation Using the Bisection Method

$$f(x) = x^3 - 4x + 1 = 0$$

$$a = 0;$$

$$b = 1;$$

$$\text{tol} = 1e-5;$$

$$\text{max_iter} = 100;$$