Create Common 2-D Plots

Open in MATLAB Online Copy Code Copy Command

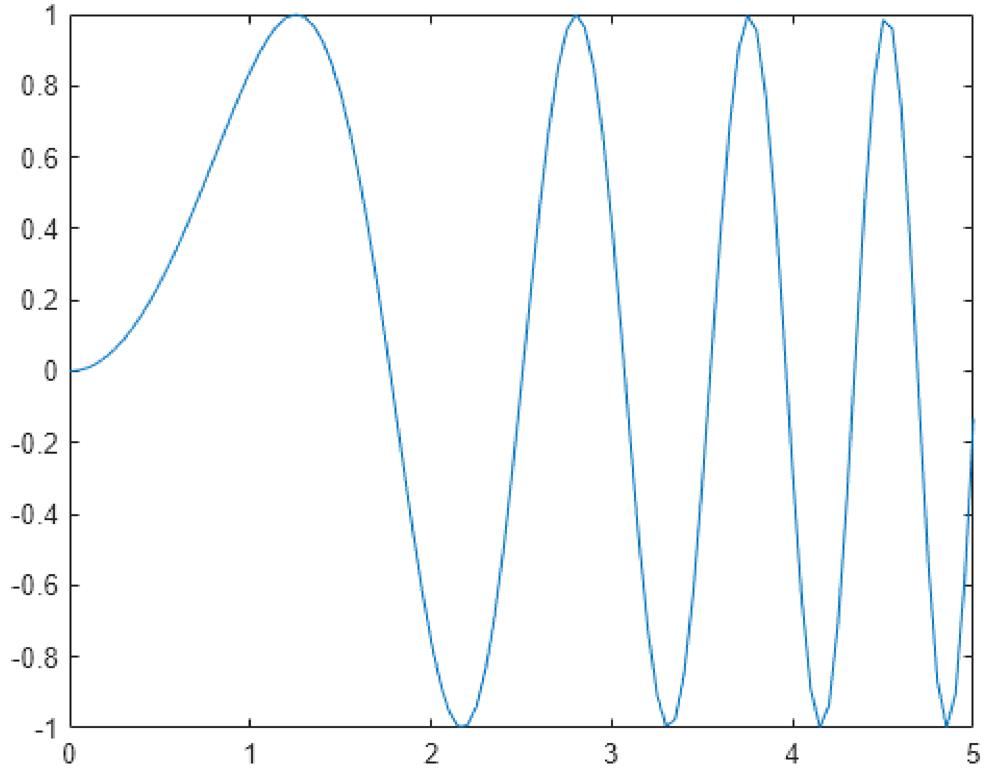
This example shows how to create a variety of 2-D plots in MATLAB®.

Line Plots

The `plot` function creates simple line plots of *x* and *y* values.

Get

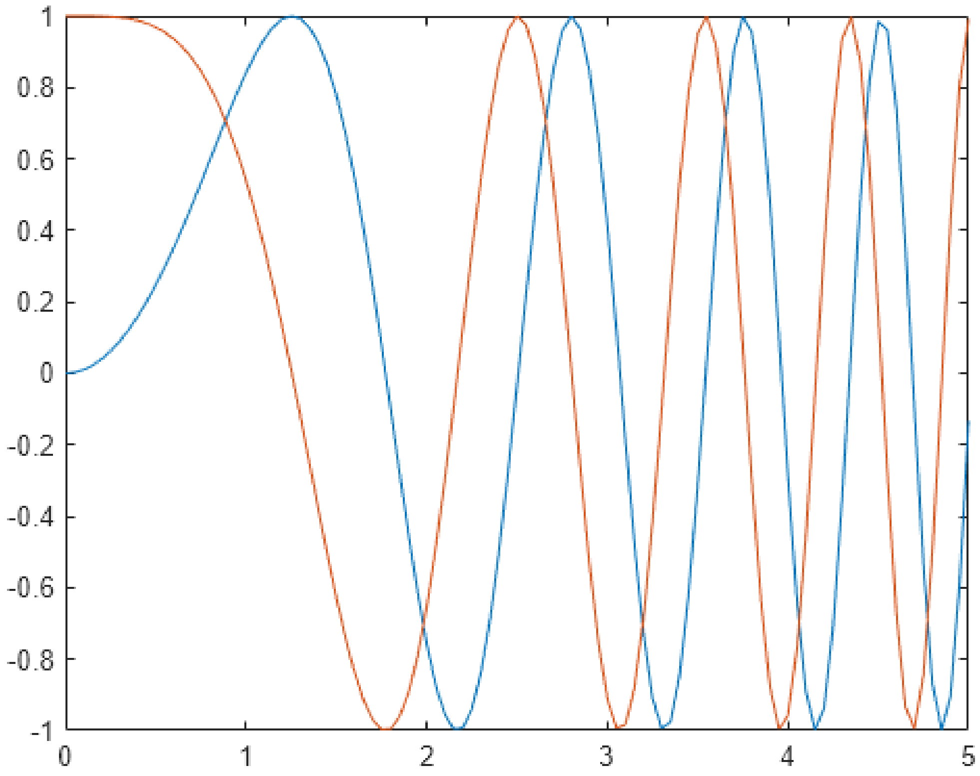
```
x = 0:0.05:5;
y = sin(x.^2);
figure
plot(x,y)
```



Line plots can display multiple sets of *x* and *y* data.

Get

```
y1 = sin(x.^2);
y2 = cos(x.^2);
plot(x,y1,x,y2)
```

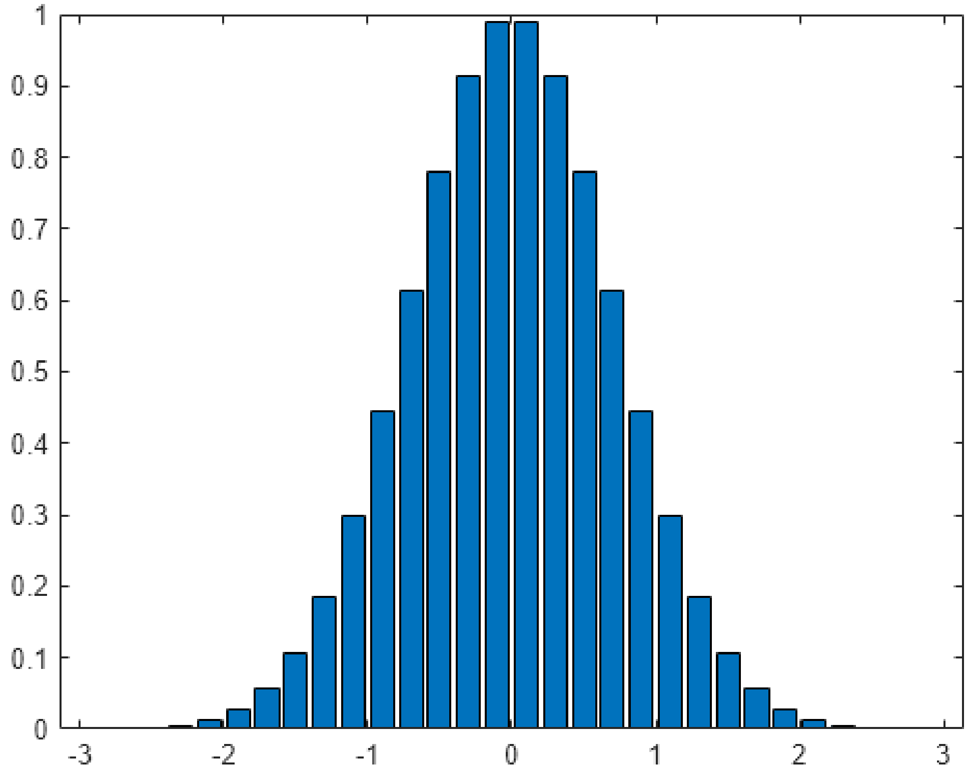


Bar Plots

The `bar` function creates vertical bar charts. The `barh` function creates horizontal bar charts.

Get

```
x = -2.9:0.2:2.9;
y = exp(-x.*x);
bar(x,y)
```

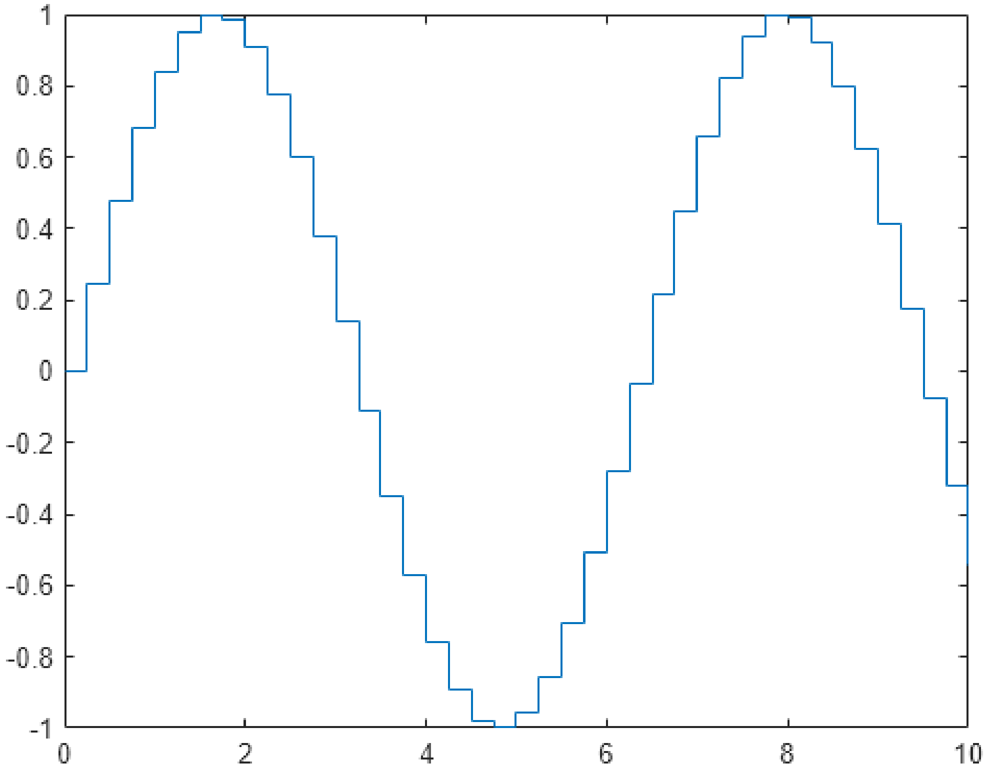


Stair step Plots

The `stairs` function creates a stair step plot. It can create a stair step plot of Y values only or a stair step plot of *x* and *y* values.

Get

```
x = 0:0.25:10;
y = sin(x);
stairs(x,y)
```

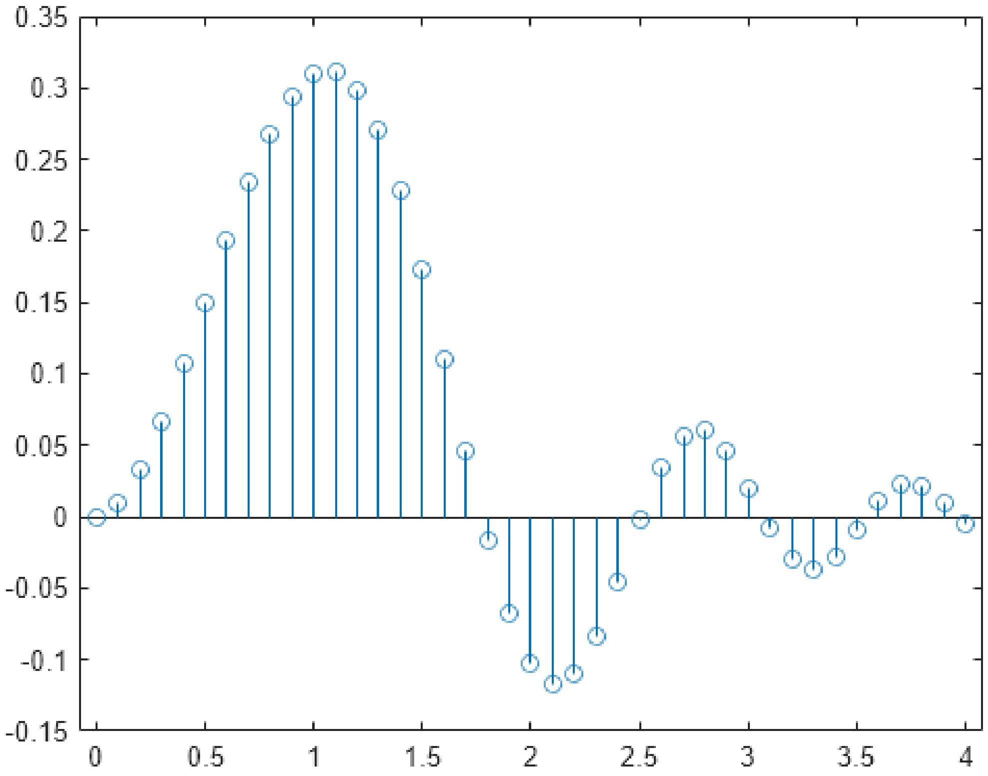


Stem Plots

The `stem` function draws a marker for each *x* and *y* value with a vertical line connected to a common baseline.

Get

```
x = 0:0.1:4;
y = sin(x.^2).*exp(-x);
stem(x,y)
```

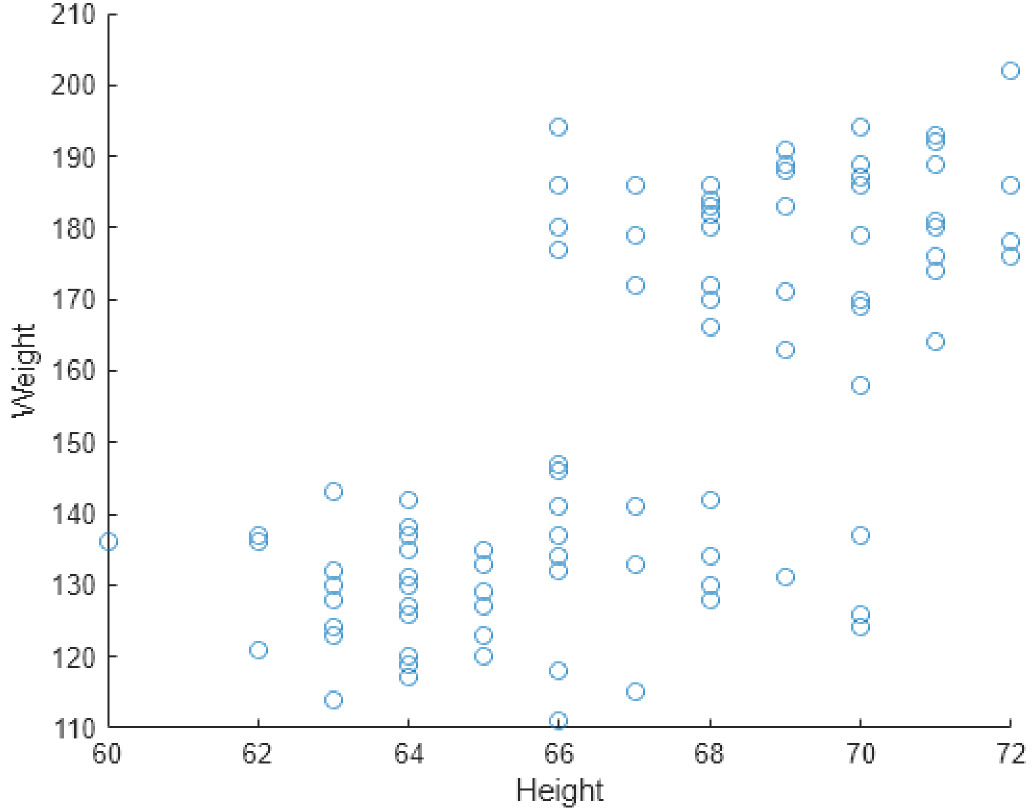


Scatter Plots

The `scatter` function draws a scatter plot of *x* and *y* values.

Get

```
load patients Height Weight Systolic
scatter(Height,Weight)
xlabel('Height')
ylabel('Weight')
```



Plot Styles (`Marker`, `LineWidth`, `Color`)

You can modify the appearance of plots using different markers, line styles, and colors.

Example:

```
x = 1:10;
y = x.^2;
plot(x, y, 'g-o', 'LineWidth', 2, 'MarkerSize', 10, 'MarkerFaceColor', 'r')
```

- `'g-o'` → Green line with circular markers
- `'LineWidth', 2` → Set line thickness
- `'MarkerSize', 10` → Increase marker size
- `'MarkerFaceColor', 'r'` → Red-filled markers

Customizing Plots

Changing Line Styles and Colors

```
plot(x, y, 'r--', 'LineWidth', 2); % Red dashed line
```

Common Line Styles:

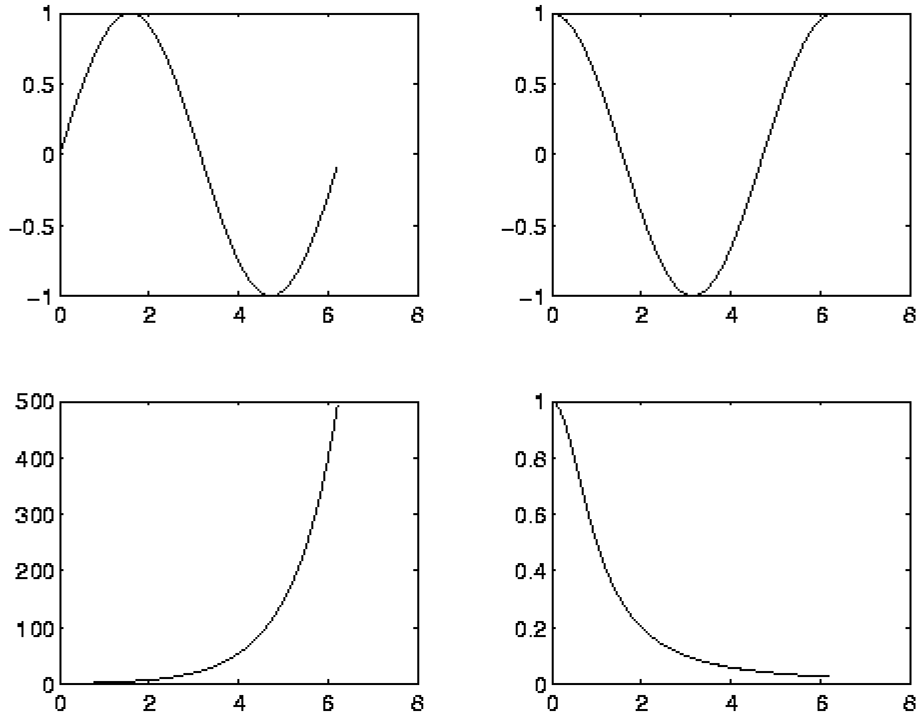
- '-' → Solid
- '--' → Dashed
- ':' → Dotted
- '-.' → Dash-dot

Common Colors:

- 'r' (Red)
- 'g' (Green)
- 'b' (Blue)
- 'k' (Black)
- 'm' (Magenta)

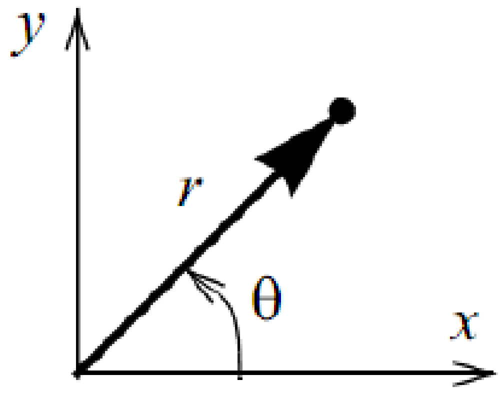
Putting several graphs in one window

- The subplot command creates several plots in a single window. Here is an example:
- >> t = (0:.1:2*pi);
- >> subplot(2,2,1)
- >> plot(t,sin(t))
- >> subplot(2,2,2)
- >> plot(t,cos(t))
- >> subplot(2,2,3)
- >> plot(t,exp(t))
- >> subplot(2,2,4)
- >> plot(t,1./(1+t.^2))



POLAR PLOTS

Polar coordinates, in which the position of a point in a plane is defined by the angle θ and the radius (distance) to the point, are frequently used in the solution of science and engineering problems. The `polar` command is used to plot functions in polar coordinates. The command has the form:




```
polar(theta,radius,'line specifiers')
```

Vector

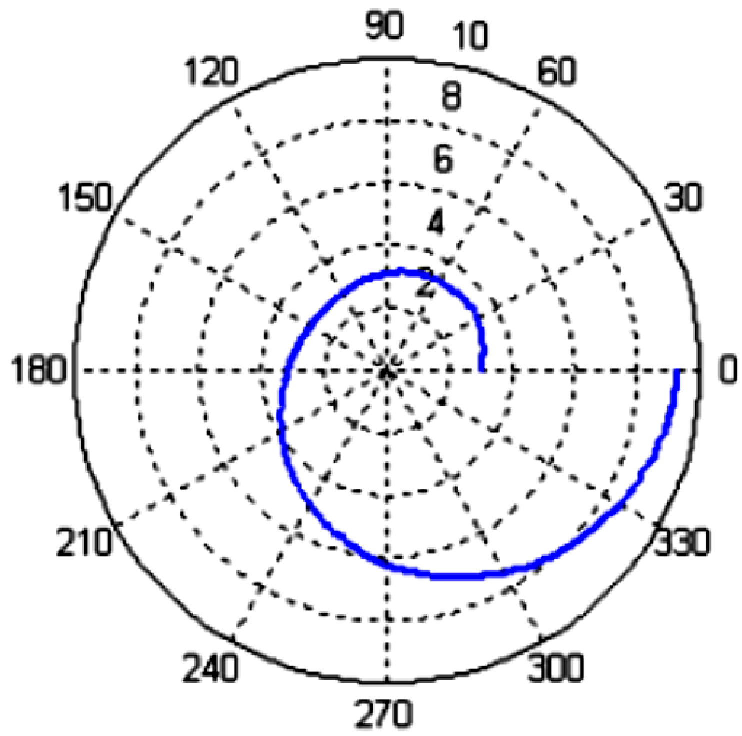
Vector

(Optional) Specifiers that define the type and color of the line and markers.

element calculations. The two vectors are then used in the `polar` command.

For example, a plot of the function $r = 3\cos^2(0.5\theta) + \theta$ for $0 \le \theta \le 2\pi$ is shown below.

```
t=linspace(0,2*pi,200);
r=3*cos(0.5*t).^2+t;
polar(t,r)
```



Command	Description
plot(x, y)	Basic 2D line plot
hold on	Plot multiple curves on the same figure
subplot(m,n,p)	Multiple subplots in one figure
scatter(x, y)	Scatter plot for individual points
bar(x, y)	Bar chart
histogram(data, bins)	Histogram

Three-Dimensional Plots (3D)

LINE PLOTS

A three-dimensional line plot is a line that is obtained by connecting points in three-dimensional space. A basic 3-D plot ‘is created with the `plot3` command, which is very similar to the `plot` command and has the form:

```
plot3(x,y,z,'line specifiers','PropertyName',property value)
```

`x`, `y`, and `z` are vectors of the coordinates of the points.

(Optional) Specifiers that define the type and color of the line and markers.

(Optional) Properties with values that can be used to specify the line width, and marker’s size and edge and fill colors.

For example, if the coordinates x, y, and z are given as a function of the parameter t by

$$x = \sqrt{t}\sin(2t)$$
$$y = \sqrt{t}\cos(2t)$$
$$z = 0.5t$$

a plot of the points for $0 \le t \le 6\pi$ can be produced by the following script file:

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
plot3(x,y,z,'k','linewidth',1)
grid on
xlabel('x'); ylabel('y'); zlabel('z')
```

The plot shown in Figure 10-1 is created when the script is executed.

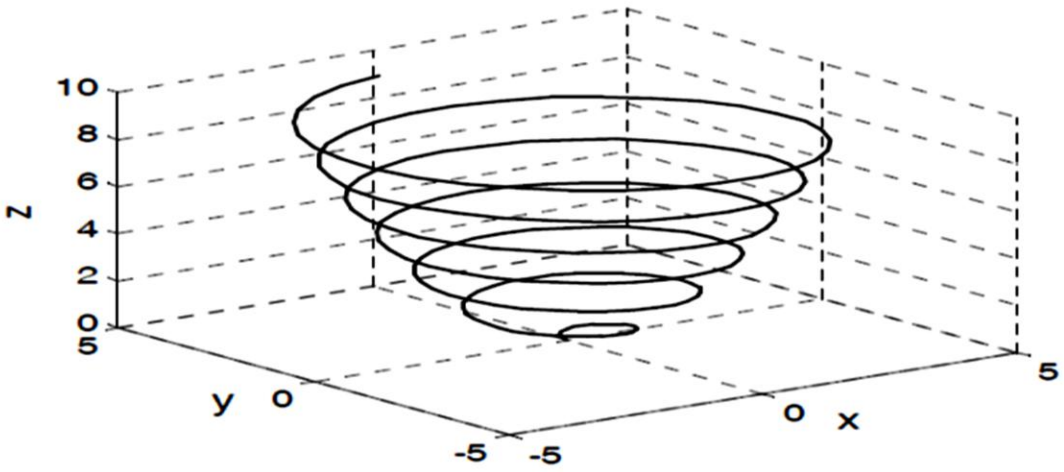


Figure 10-1: A plot of the function $x = \sqrt{t}\sin(2t)$, $y = \sqrt{t}\cos(2t)$, $z = 0.5t$ for $0 \le t \le 6\pi$.

Mesh and Surface Plots

Mesh and surface plots are three-dimensional plots used for plotting functions of
the form where *x* and *y* are the independent variables and *z* is the dependent variable. It means that within a given domain the value of *z* can be calculated
for any combination of *x* and *y*. Mesh and surface plots are created in three

steps. The first step is to create a grid in the *x y* plane that covers the domain of the
function. The second step is to calculate the value of *z* at each point of the grid.
The third step is to create the plot. The three steps are explained next.

Creating a grid in the *x y* plane (Cartesian coordinates):

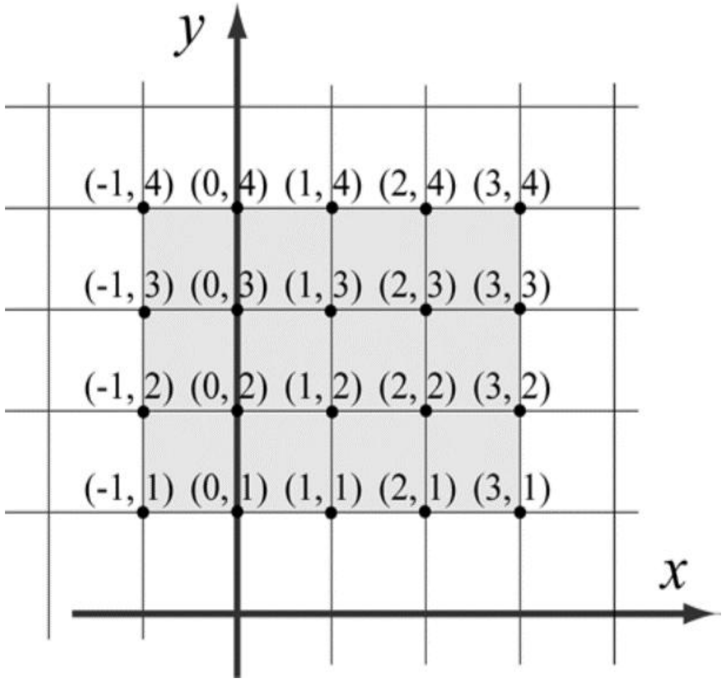
The grid is a set of points in the *x y* plane in the domain of the function. The density of the grid (number of points used to define the domain) is defined by the user. Figure 10-2 shows a grid in the domain $-1 \le x \le 3$ *and* $1 \le y \le 4$. In this grid the distance between the points is one unit. The points of the grid can be defined by two matrices, *X* and *Y*. Matrix *X* has the *x* coordinates of all the points, and matrix *Y* has the *y* coordinates of all the points: and The *X* matrix is made of identical rows since in each row of the grid the points have the same *x* coordinate. In the same way the *Y* matrix is made of identical columns since in each column of the grid the *y* coordinate of the points is the same. MATLAB has a built-in function, called **meshgrid**, that can be used for

creating the *X* and *Y* matrices. The form of the `meshgrid` function is:

```
[X,Y] = meshgrid(x,y)
```

`X` is the matrix of the *x* coordinates of the grid points.
`Y` is the matrix of the *y* coordinates of the grid points.

`x` is a vector that divides the domain of *x*.
`y` is a vector that divides the domain of *y*.



```
>> x=-1:3;
>> y=1:4;
>> [X,Y]=meshgrid(x,y)
X =
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
Y =
     1     1     1     1     1
     2     2     2     2     2
     3     3     3     3     3
     4     4     4     4     4
```

Once the grid matrices exist, they can be used for calculating the value of z at each grid point.

Calculating the value of z at each point of the grid:

The value of z at each point is calculated by using element-by-element calculations in the same way it is used with vectors. When the independent variables x and y are matrices (they must be of the same size), the calculated dependent variable is also a matrix of the same size. The value of z at each address is calculated from the corresponding values of x and y. For example, if z is given by

$$z = \frac{xy^2}{x^2 + y^2}$$

the value of z at each point of the grid above is calculated by:

```
>> Z = X.*Y.^2./(X.^2 + Y.^2)
```

```
Z =
   -0.5000         0    0.5000    0.4000    0.3000
   -0.8000         0    0.8000    1.0000    0.9231
   -0.9000         0    0.9000    1.3846    1.5000
   -0.9412         0    0.9412    1.6000    1.9200
```

Once the three matrices have been created, they can be used to plot mesh or surface plots.

Making mesh and surface plots:

A mesh or surface plot is created with the `mesh` or `surf` command, which has the form:

`mesh(X,Y,Z)`   `surf(X,Y,Z)`

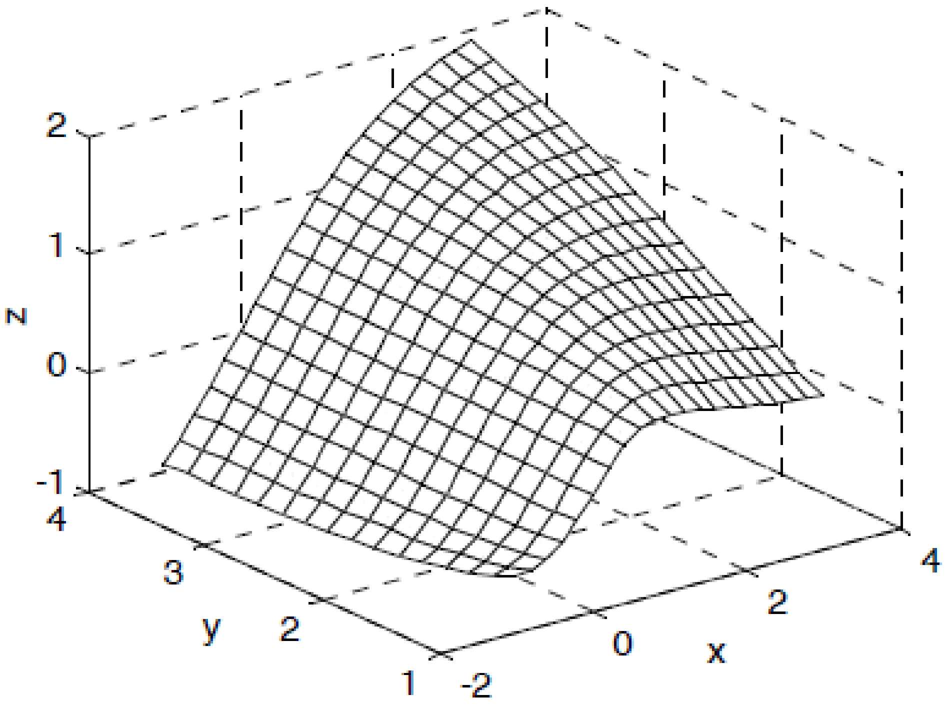
where `X` and `Y` are matrices with the coordinates of the grid and `Z` is a matrix with the value of z at the grid points. The mesh plot is made of lines that connect the points. In the surface plot, areas within the mesh lines are colored.

As an example, the following script file contains a complete program that creates the grid and then makes a mesh (or surface) plot of the function $z = \frac{xy^2}{x^2+y^2}$ over the domain $-1 \le x \le 3$ and $1 \le y \le 4$.

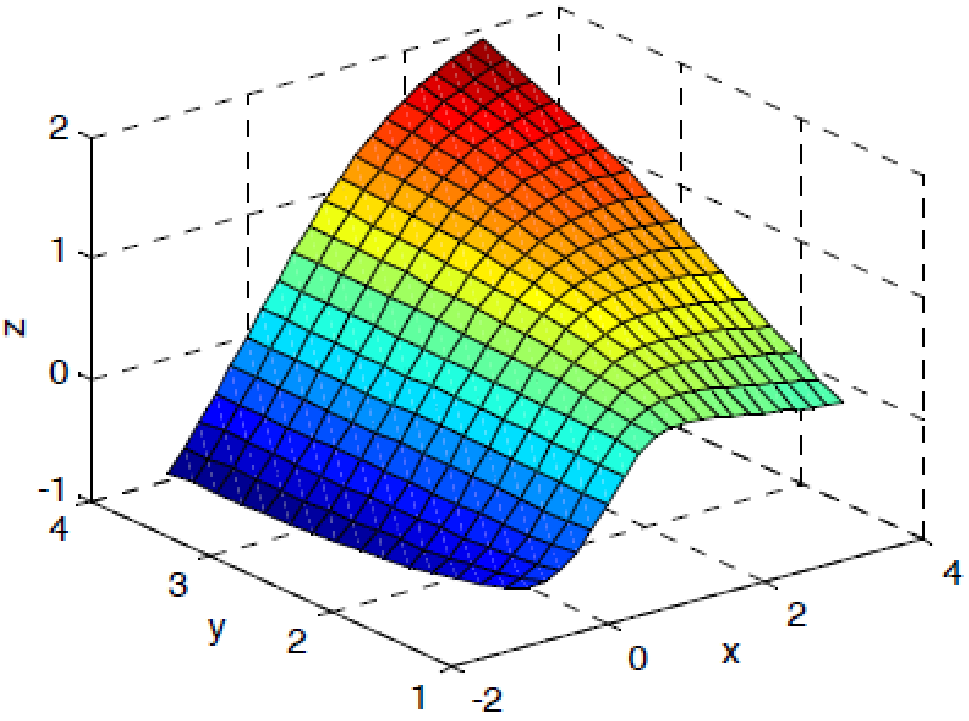
```
x=-1:0.1:3;
y=1:0.1:4;
[X,Y]=meshgrid(x,y);
Z=X.*Y.^2./(X.^2+Y.^2);
mesh(X,Y,Z)
xlabel('x'); ylabel('y'); zlabel('z')
```

Type `surf(X,Y,Z)` for surface plot.

The figures created by the program are:



Mesh plot



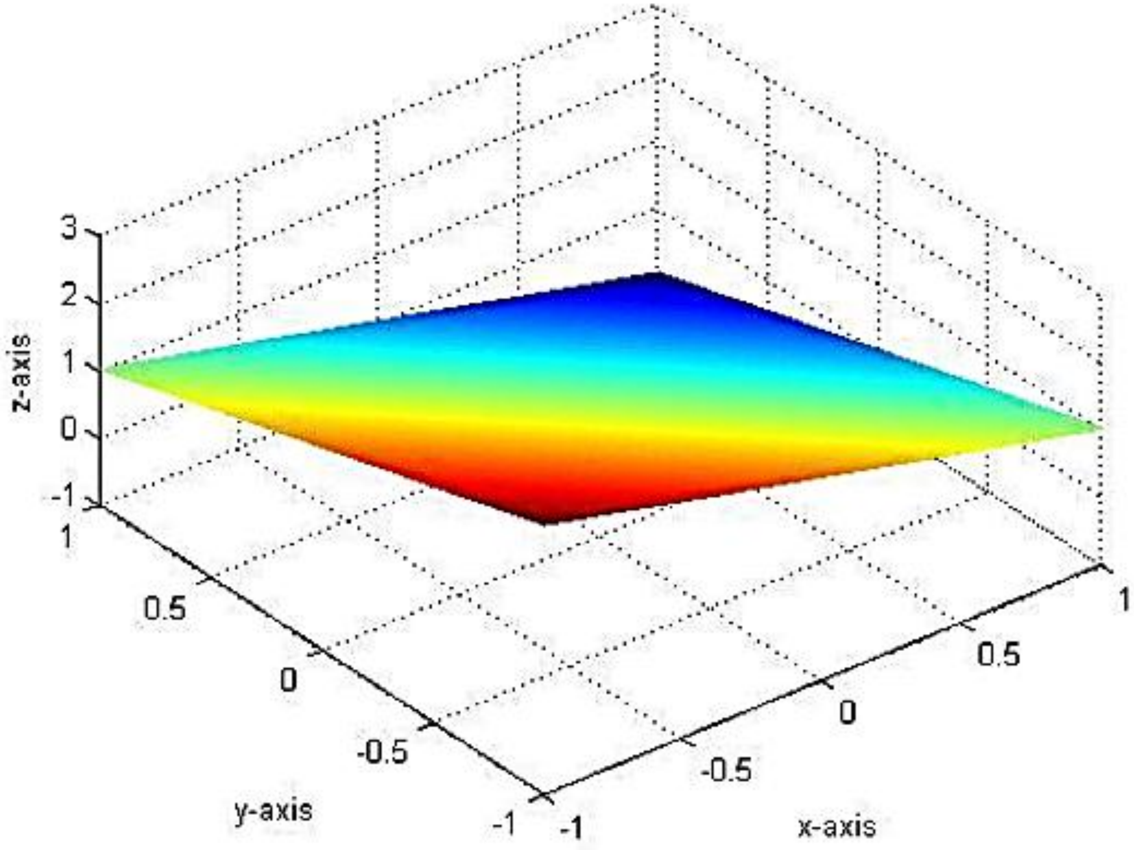
Surface plot

Ex.

1. plot x+y+z=1 in 3D?

Solution:

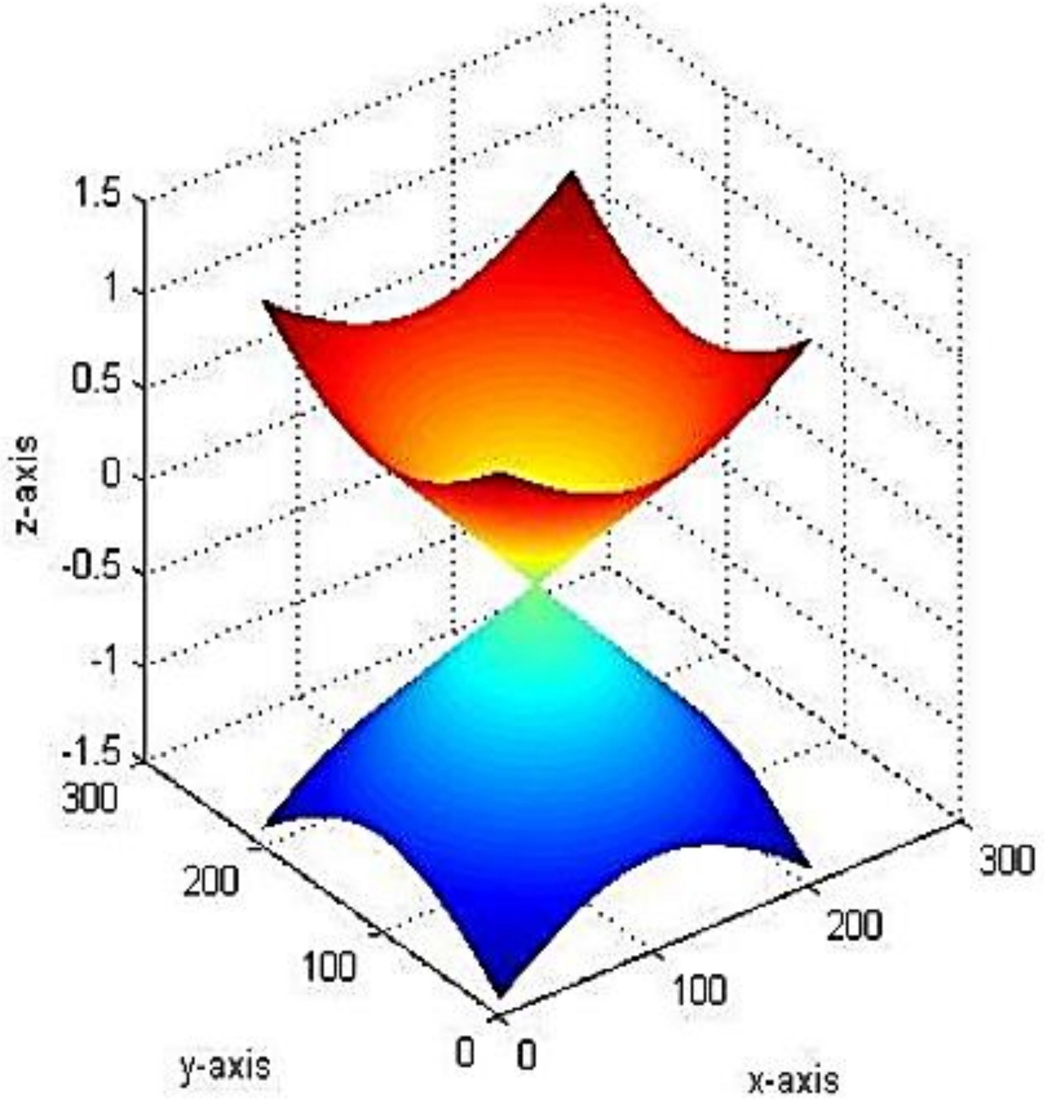
```
x=-1:.01:1;
y=x;
[x y]=meshgrid(x,y);
z=1-x-y;
mesh(x,y,z)
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```



2. plot $x^2+y^2=z^2$ (Cone) in 3D

```
x=-1:.01:1;
y=x;
[x y]=meshgrid(x,y);
z1=sqrt(x.^2+y.^2);
z2=-sqrt(x.^2+y.^2);
     mesh(z1)
  hold on
   mesh(z2)

xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```



H.W

PROBLEMS

1. Plot the function $f(t) = \dfrac{5\sin(x)}{x + e^{-0.75x}} - \dfrac{3x}{5}$ for $-5 \le x \le 10$.

2. Make two separate plots of the function $f(x) = (x+1)(x-2)(2x-0.25) - e^x$ one plot for $0 \le x \le 3$ and one for $-3 \le x \le 6$.

3. Plot $x=1$ in 2D and 3D.

4. Plot $x^2+y^2=z^2+1$ (Hyperbloid of one sheet) in 3D.

5. plot $x^2-y^2=z$ (Elliptic paraboloid) in 3D.

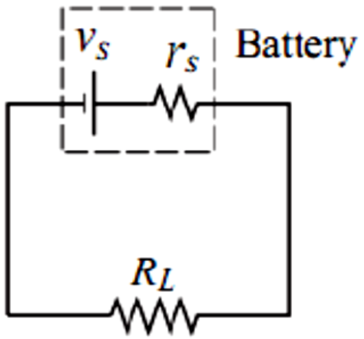
6. Prove (by plot) the equation ($x^2+y^2=16$) is the circle in 2D , and cylindrical in 3D .

7.plot the data x=(3 4 6 8 10), y=(12 14 16 20 22) ,using Bars,Stairs and stem plots.

8. An electrical circuit that includes a voltage source v_S with an internal resistance r_S and a load resistance R_L is shown in the figure. The power P dissipated in the load is given by

$$P = \frac{v_S^2 R_L}{(R_L + r_S)^2}$$

Plot the power P as a function of R_L for $1 \le R_L \le 10\,\Omega$, given that $v_S = 12$ V and $r_S = 2.5\,\Omega$.



9. The position of a moving particle as a function of time is given by:

$$x = (4 - 0.1t)\sin(0.8t) \qquad y = (4 - 0.1t)\cos(0.8t) \qquad z = 0.4t^{(3/2)}$$

Plot the position of the particle for $0 \le t \le 30$.

10. Use the `fplot` command to plot the function $f(x) = \sqrt{|\cos(3x)|} + \sin^2(4x)$ in the domain $-2 \le x \le 2$.

Any question?

Thanks