3D Plotting

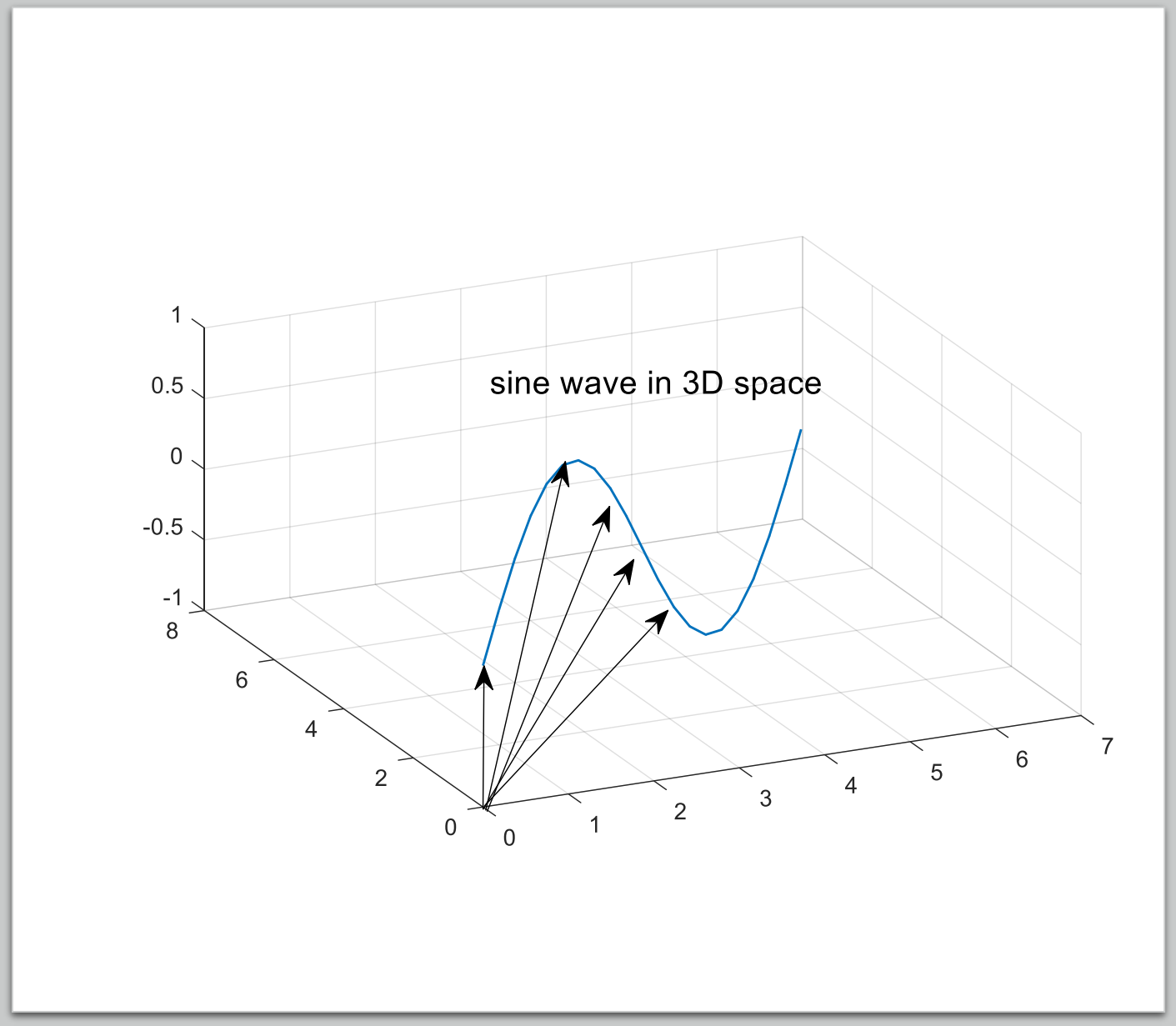
Dr. Marco A. Arocha

May 2018

Line Plots in 3D

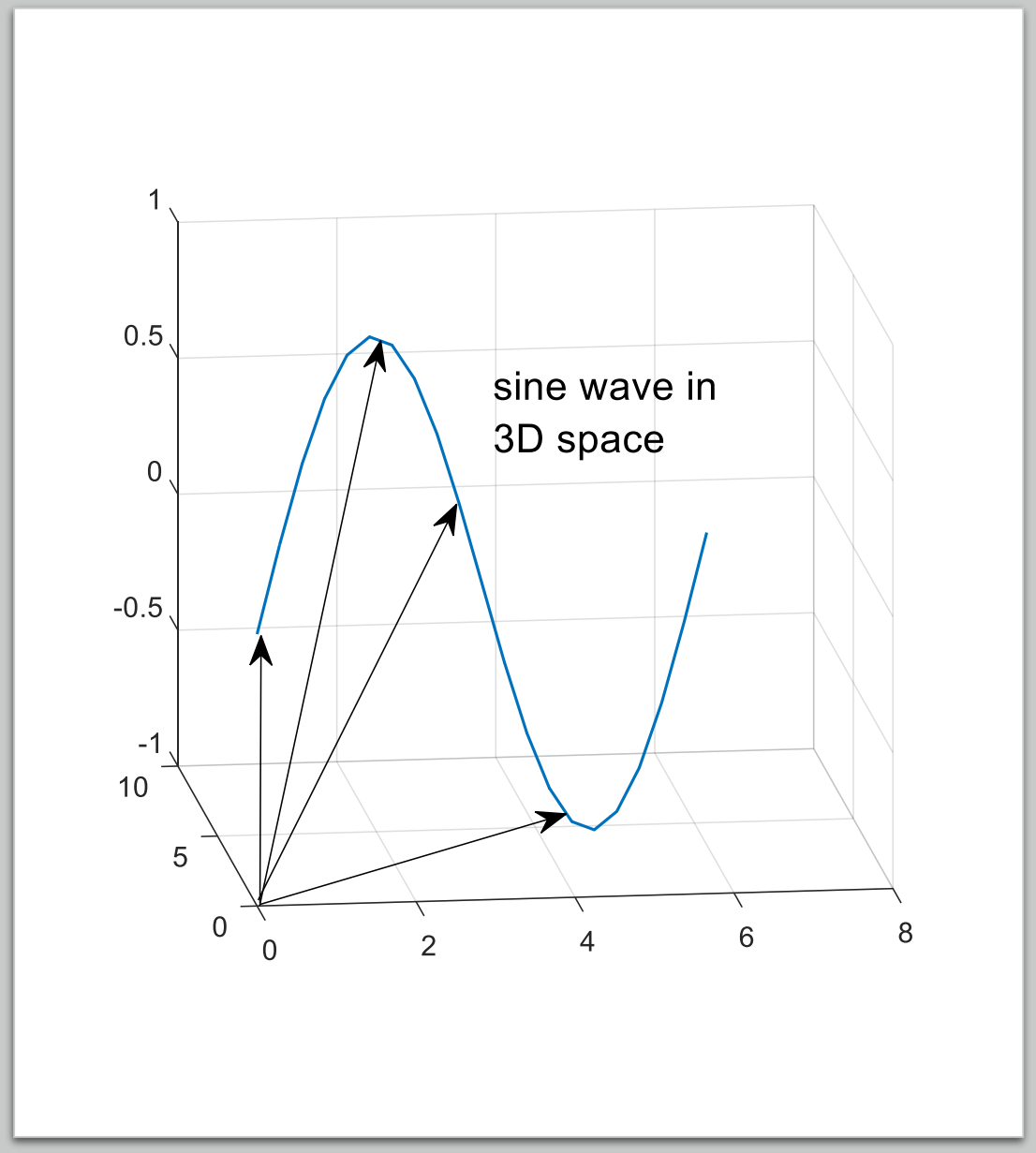
- Points in space are usually given by ordered triples (x ,y ,z). One could view this as "position vectors" starting at the point (0, 0, 0) and ending at the points (x, y, z). Then a line is just a succession of these end points.

- Succession of points producing lines in three-dimensional space can be plotted with the **`plot3(x,y,z)`** function .



The Code:

```
% LinePlot3D.m
% Experiments in 3D

clc, clear, close, clf
t = 0:pi/10:2*pi;
x = t;
y = t;
z = sin(t);
plot3(x, y, z)
grid on
```



3D Visualization

- I have the one belief that our education is mostly taught and learned in 2D (an with scalar mathematics), therefore to develop a bit of a 3D flavor we need to start by figuring it out popular images in 2D which may upgrade ‘effortlessly’ to 3D view, so not to break too hard with our ‘habits’

- Visual learning plays a key role in the teaching and learning process.

- The 3D learning experience places students in virtual environments

Parametric Equations

- Parametric equations are a set of equations expressed as explicit functions of a number of independent variables known as "parameters."
- For example, while the equation of a circle in Cartesian coordinates can be given by,

$$r^2 = x^2 + y^2$$

- One set of parametric equations in 2D for the circle are given by:

$$x = r\cos(t)$$ Set of explicit functions of the parameter t (angle)

$$y = r\sin(t)$$ One circle or radius r is generated when you varies t=[0:2pi]

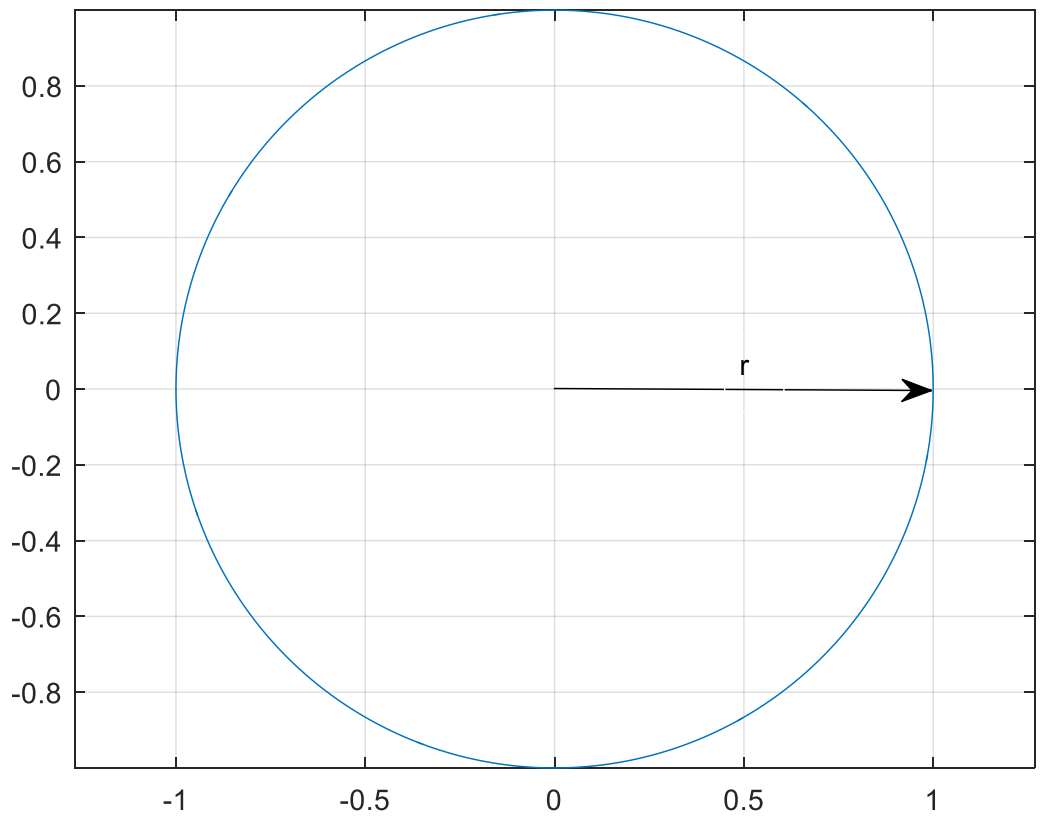
- In 3D we need (x,y,z):

$$x = r\cos(t)$$
$$y = r\sin(t)$$
$$z = t$$

- You can use these explicit equations to generate circles in 3D space.
- Parametric equations provide a convenient way to represent curves and surfaces in 3D

```
% Using a parametric equation for the circle
% Plot a circle with radius=1
clc, clear
r=1
t=linspace(0,2*pi,360);
x=r*cos(t);
y=r*sin(t);
plot(x,y);
axis('equal');
```

https://www.youtube.com/watch?v=3k4EeezUIFQ

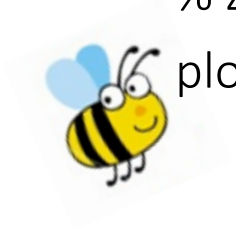


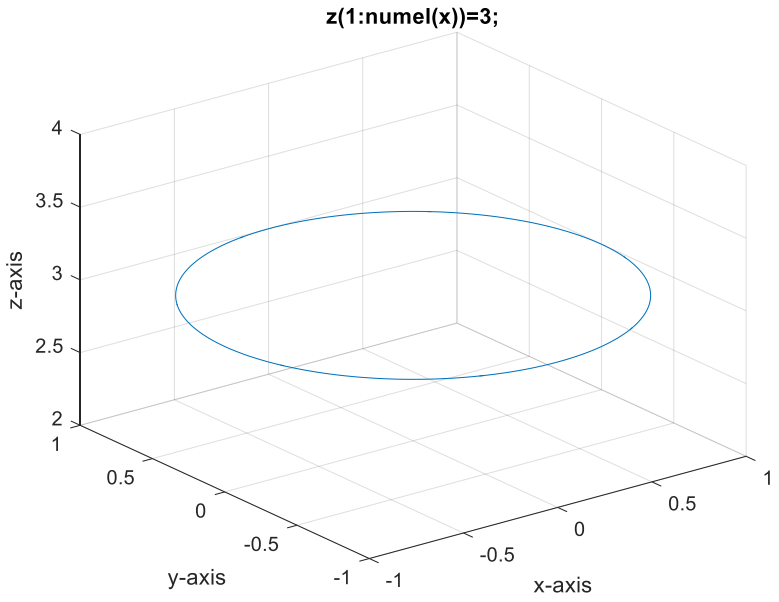
CirclePlot2D.m

3D plot of a Circle

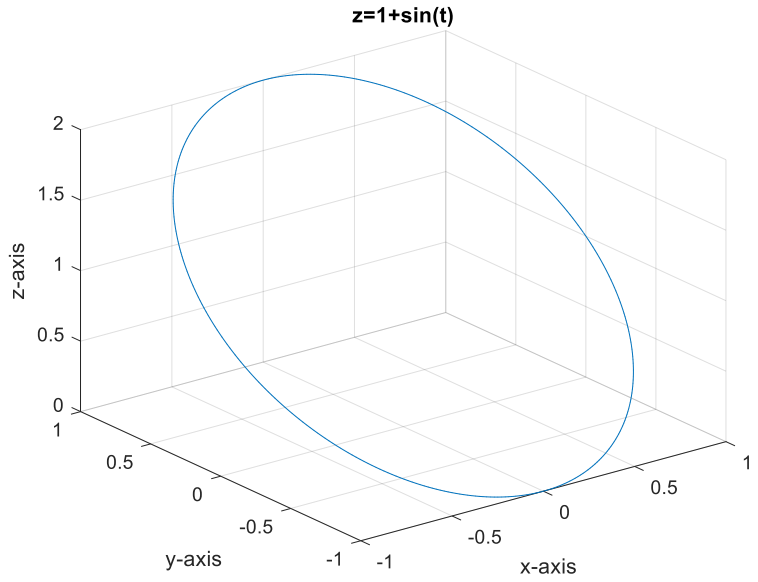
```
% CirclePlot3D.m
% By parametric equation for the circle
% Plot a circle with radius=1
clc, clear, clf
r=1;
t=linspace(0,2*pi,360);     % [angle]
x=r*cos(t);
y=r*sin(t);

% z(1:numel(x))=3;
% z=1+sin(t);                % activate one z statement
% z=t;
plot3(x,y,z);
```

```
% continued
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
title('z=t');
grid on
```

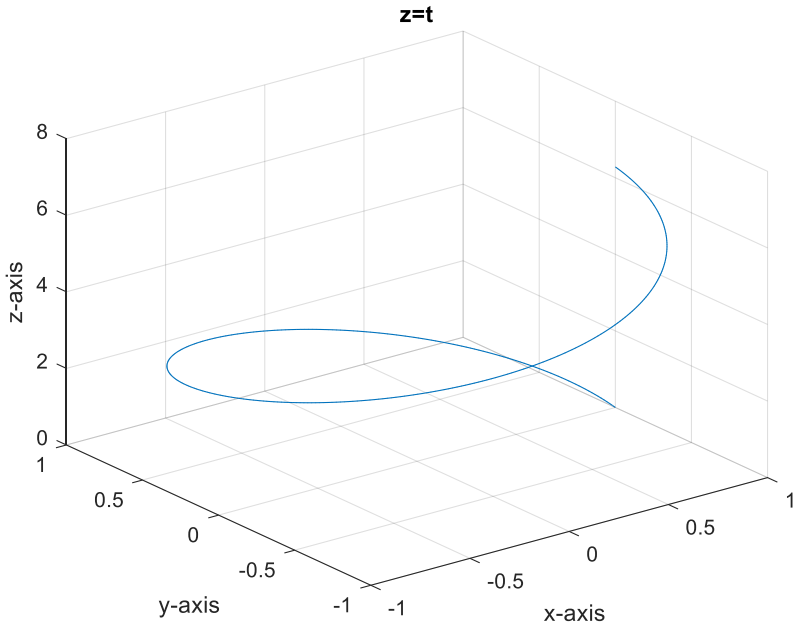




Constant z:
z(1:numel(x))=3;



Up and down z:
z=1+sin(t);



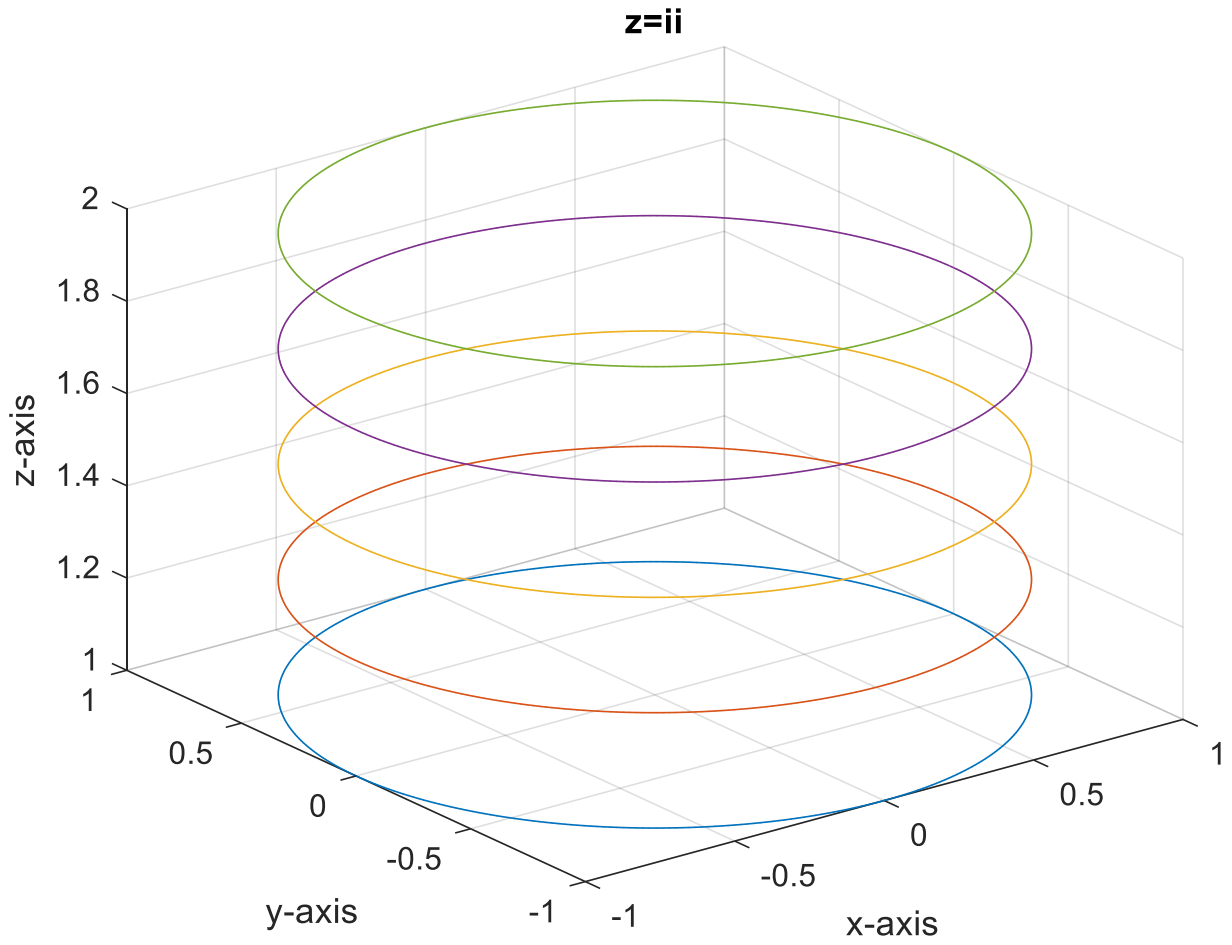
Increasing z:
z=t;
% open cycle
% i.e., spiral

Circles

QUIZ

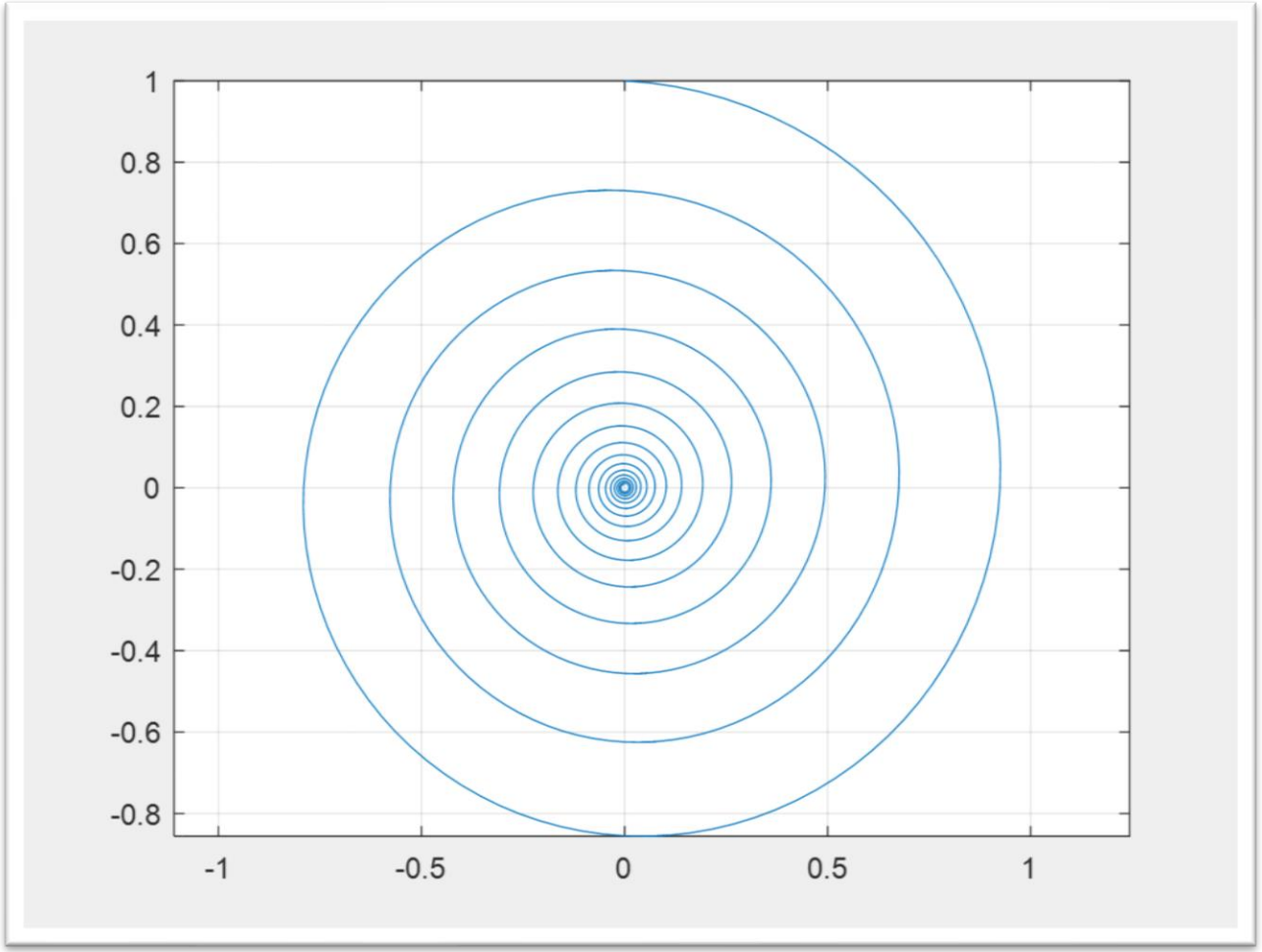
Exercise

Produce a pile of circles of radius=1 and center at (x=0,y=0) at different levels of z=1,1.25,1.5,1.75,2



Spiral Plot: 2D

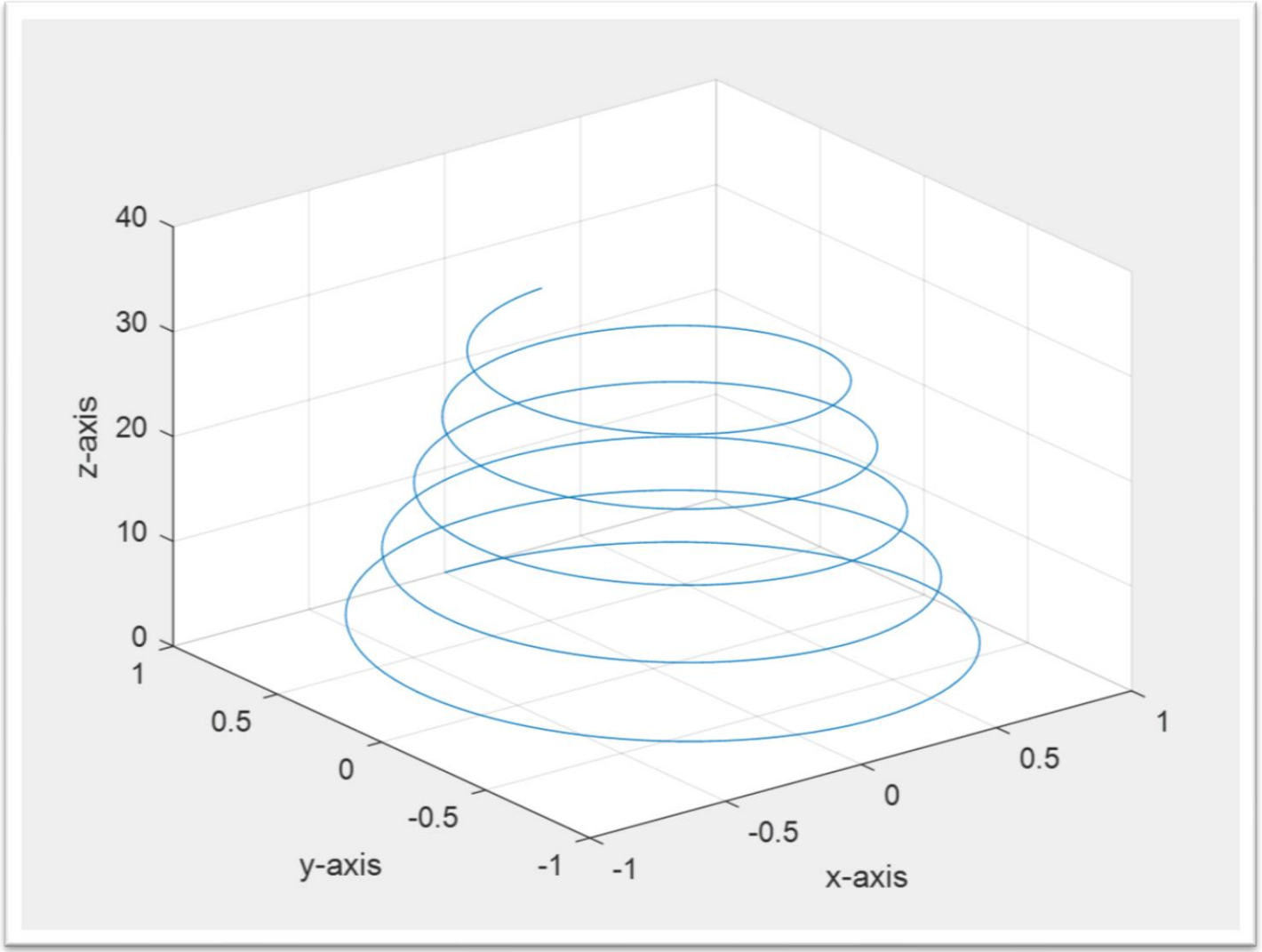
```
% SpiralPlot2D.m
% Using parametric equation
% for the modified circle
clc, clear, clf

t = 0:pi/50:30*pi;      % angle, 15 cycles
x=exp(-0.05*t).*sin(t);
y=exp(-0.05*t).*cos(t);
% exp(-0.05*t) representa the radious
% which is increasing
plot(x,y);
axis('equal');
grid
```



Explanation: If t vary from t=0 to 30pi, the sine and cosine functions will vary through 15 cycles, while the absolute values of x and y become smaller as t increases.

Spiral Plot 3D

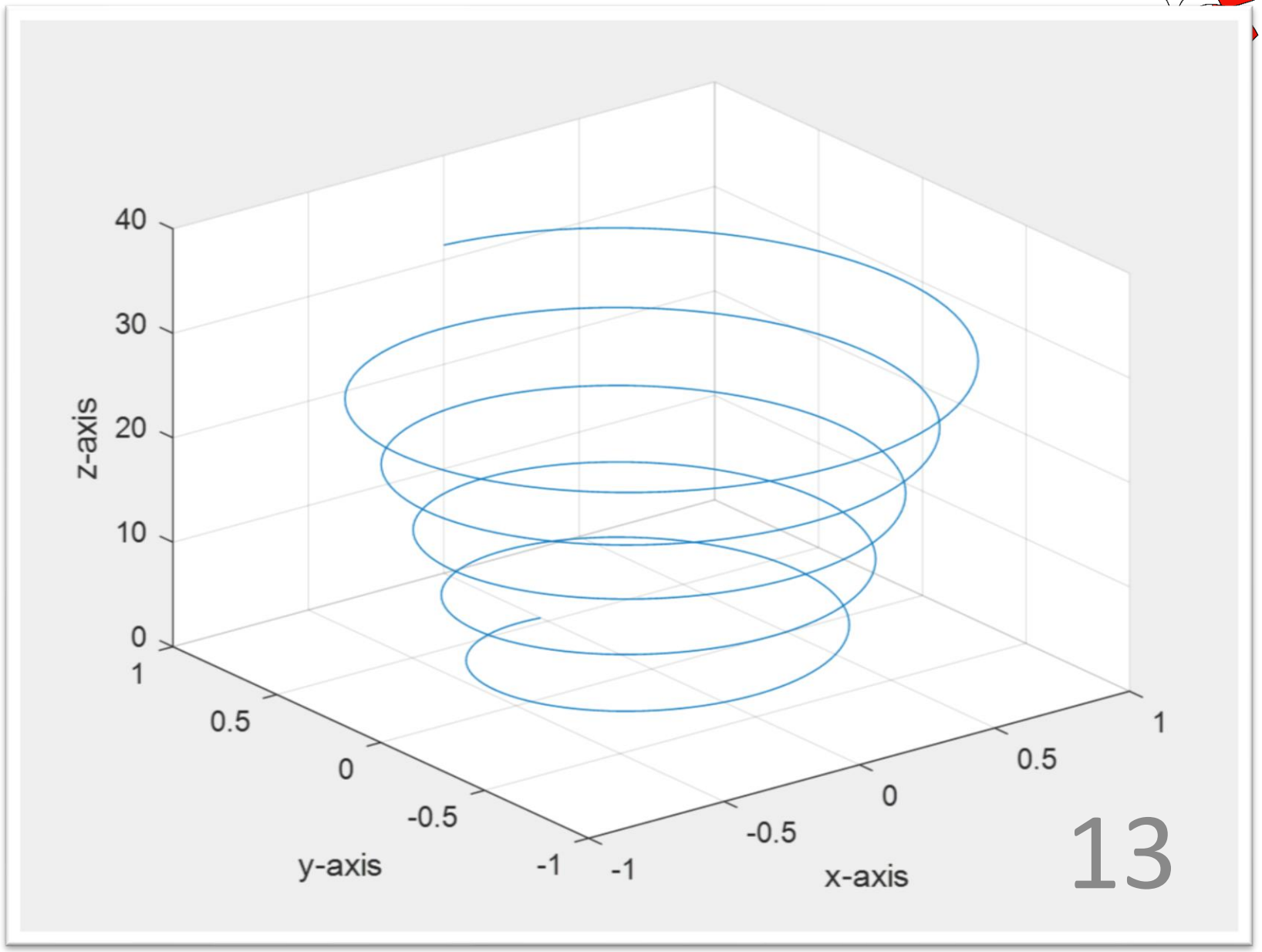
```
% tresD002.m
clc, clear, clf
t = 0:pi/50:10*pi;
x=exp(-0.02*t).*sin(t);
y=exp(-0.02*t).*cos(t);
z=t
plot3(x,y,z);
xlabel('x-axis’);
ylabel('y-axis’);
zlabel('z-axis');
grid on
```



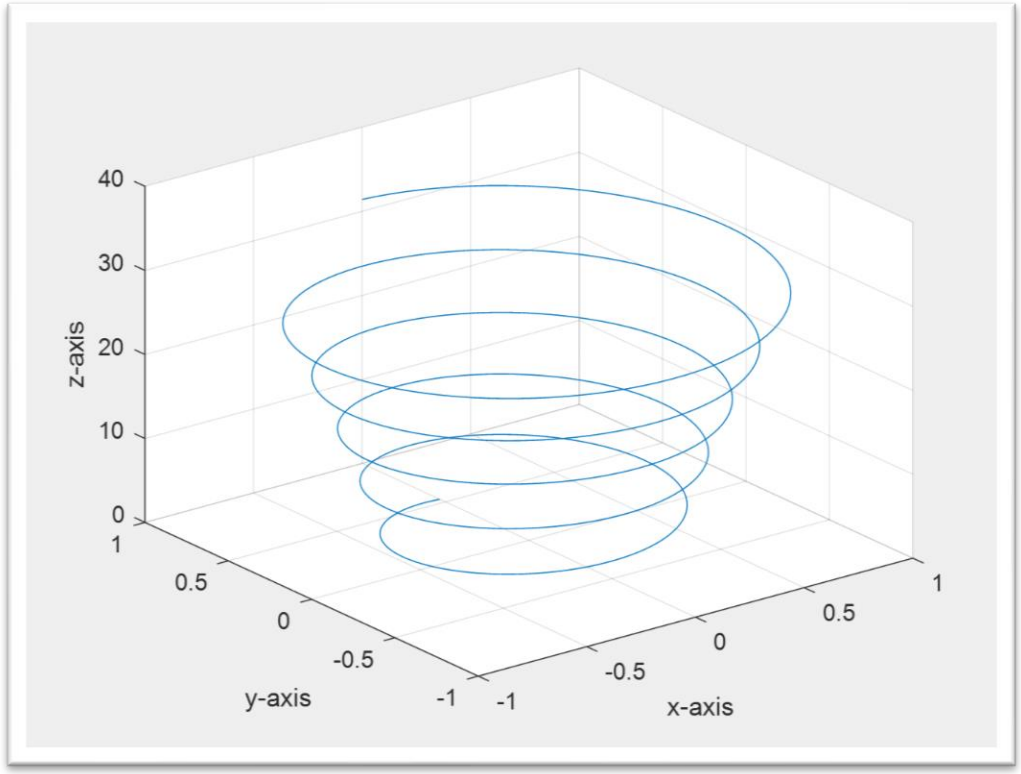
Explanation: If t vary from t=0 to 10pi, the sine and cosine functions will vary though five cycles, while the absolute values of x and y become smaller as t increases.

Quiz

- Inspired in the previous plot, write code to construct the 3D plot



Quiz-Solution




```
% tresD003.m
clc, clear, clf
t = 0:pi/50:10*pi;
x=exp(-0.02*t).*sin(t);
y=exp(-0.02*t).*cos(t);
z=10*pi-t;
plot3(x,y,z);
%plot3(exp(-0.02*t).*sin(t), exp(-0.02*t).*cos(t), t)
xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis');
grid on
```

Mesh Plots

The function $z = f(x, y)$ represents a surface when plotted on *xyz* axes

Construct x and y axes as two vectors:

x = xmin:xspacing:xmax

y = ymin:yspacing:ymax

A grid of points in the *xy* plane are generated with:

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

where

x = xmin:xspacing:xmax

y = ymin:yspacing:ymax

Resulting 2D matrices *X* and *Y* contain the coordinate pairs of every point in the grid. These pairs are used to evaluate the math function z=*f* (*x*, *y*) in the form of a 2D array equation: Z=f(X,Y)

The *mesh(X,Y,Z)* function generates the surface plots.

Explanation: Coordinates of a rectangular grid with one corner at (*xmin, ymin*) and the opposite corner at (*xmax, ymax*) are generated. Each rectangular panel in the grid will have a width equal to *xspacing* and a depth equal to *yspacing.*

Example:

Create a surface plot (mesh plot) of the function:

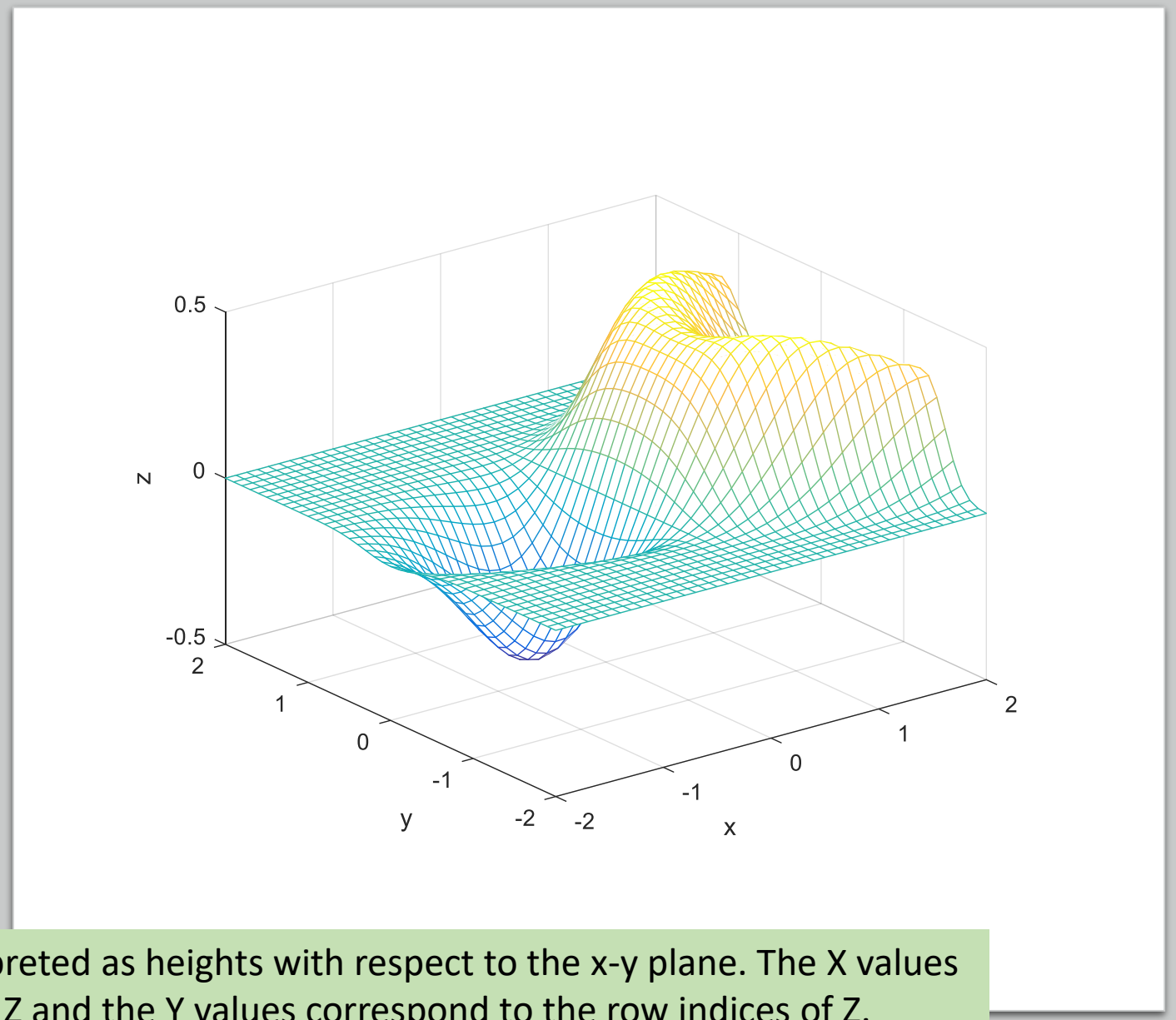
$$z = xe^{-\left[\left(x-y^2\right)^2+y^2\right]}$$

For

$-2 \leq x \leq 2$ and
$-2 \leq y \leq 2$ with spacing of 0.1

Mesh Plot

```
% MeshPlot3D.m

% Construct the grid in the plane x-y
x=-2:0.1:2; y=x;
[X,Y] = meshgrid(x,y);

% Evaluate the function z=f(x,y)
Z = X.*exp(-((X-Y.^2).^2+Y.^2));

% Plot the surface plot
mesh(X,Y,Z)
xlabel('x'),ylabel('y'),zlabel('z');
```



The matrix Z in a 3-D view. Z is interpreted as heights with respect to the x-y plane. The X values correspond to the column indices of Z and the Y values correspond to the row indices of Z.

% Construct the grid in the plane x-y
x=-2:1:2; y=x;
[X,Y] = meshgrid(x,y);

X=

-2.0	-1.0	0.0	1.0	2.0
-2.0	-1.0	0.0	1.0	2.0
-2.0	-1.0	0.0	1.0	2.0
-2.0	-1.0	0.0	1.0	2.0
-2.0	-1.0	0.0	1.0	2.0

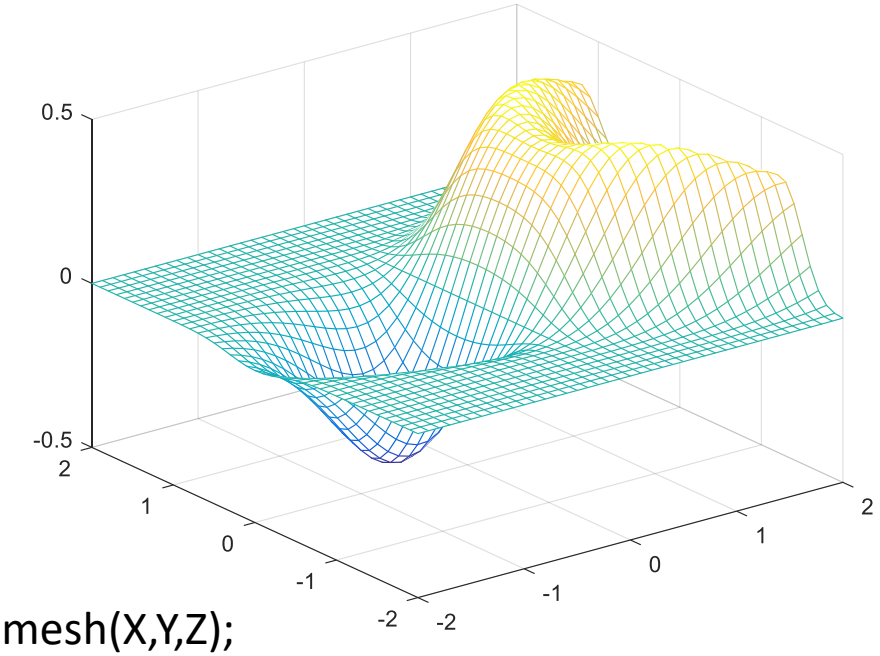
Row-1 Row-2 Row-3 Row-4 Row-5

INDICES representation into TWO 2D arrays

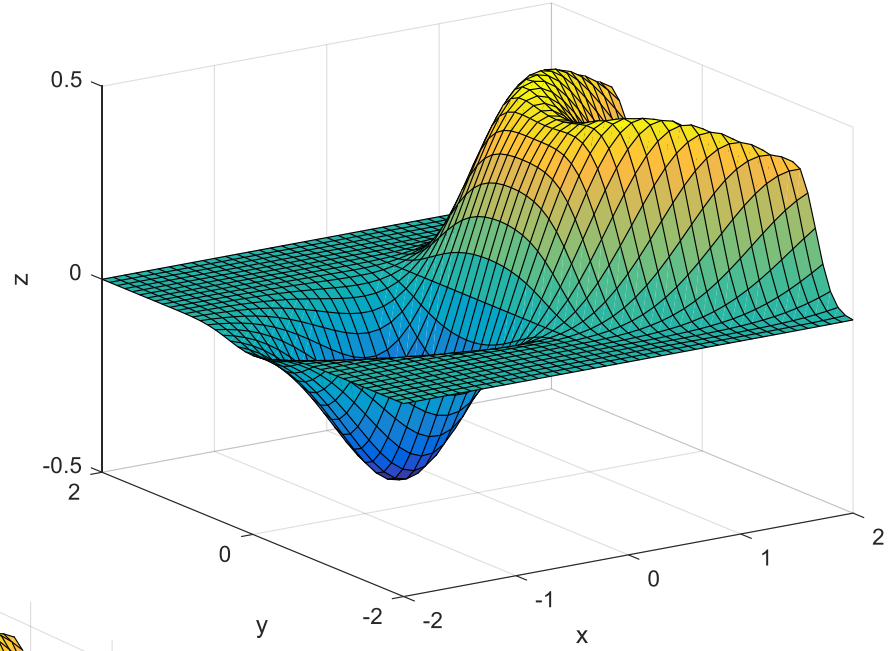
Y=

-2.0	-2.0	-2.0	-2.0	-2.0	Col-1
-1.0	-1.0	-1.0	-1.0	-1.0	Col-2
0.0	0.0	0.0	0.0	0.0	Col-3
1.0	1.0	1.0	1.0	1.0	Col-4
2.0	2.0	2.0	2.0	2.0	Col-5

$$XYgridValues = \begin{bmatrix} (-2,-2) & (-2,-1) & (-2,0) & (-2,1) & (-2,2) \\ (-1,-2) & (-1,-1) & (-1,0) & (-1,1) & (-1,2) \\ (0,-2) & (0,-1) & (0,0) & (0,1) & (0,2) \\ (1,-2) & (1,-1) & (1,0) & (1,1) & (1,2) \\ (2,-2) & (2,-1) & (2,0) & (2,1) & (2,2) \end{bmatrix}$$

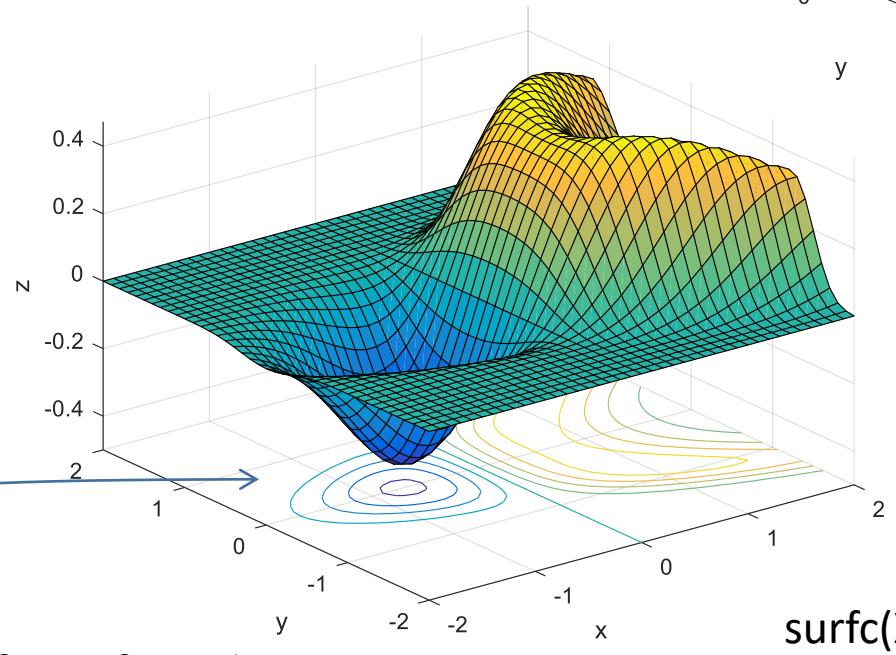


mesh(X,Y,Z);



surface(X,Y,Z);

Other functions for 3D plots



surfc(X,Y,Z);

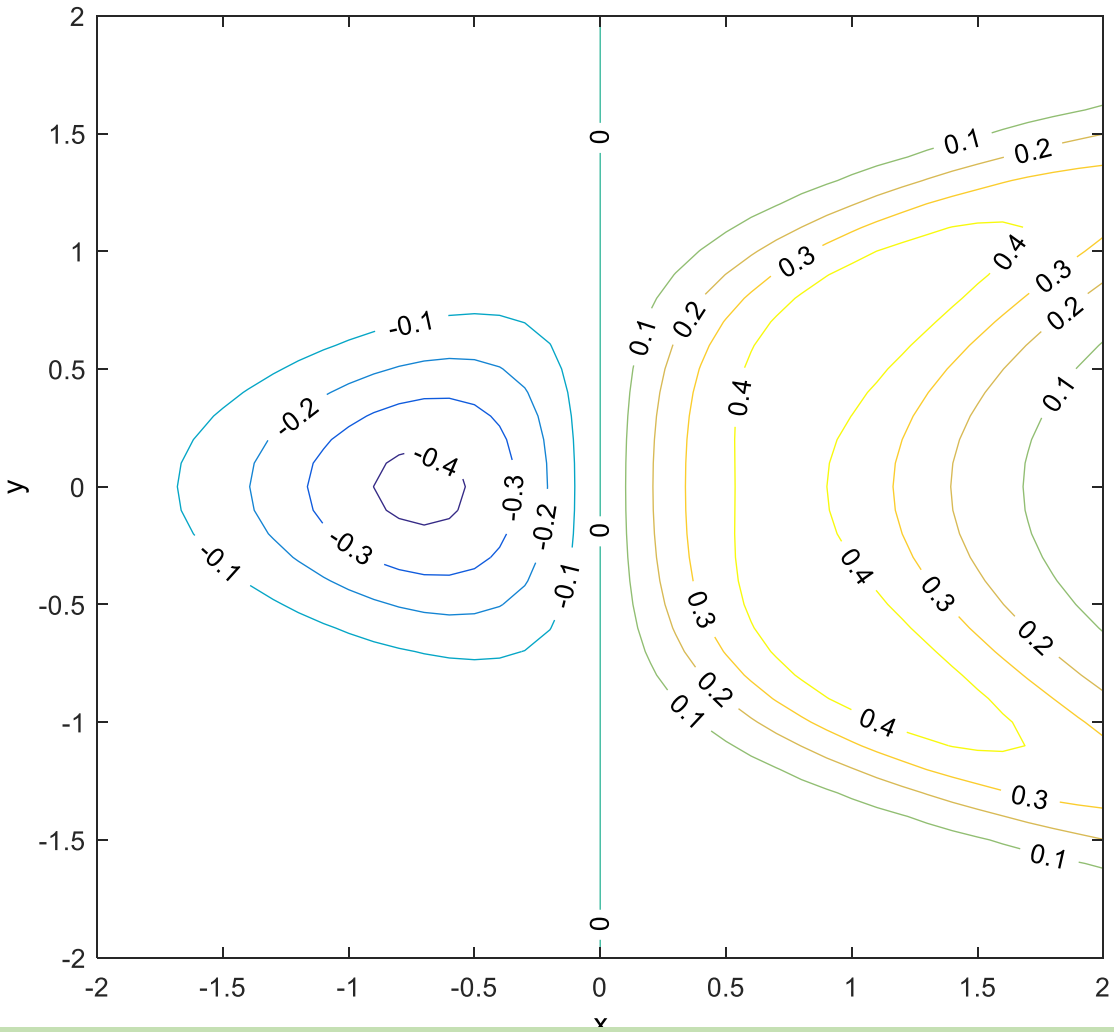
Contour plot at the bottom of a surface plot

Contour Plots

- Topographic plots show the contours of the land by means of constant elevation lines.
- These lines are also called *contour lines,* and such a plot is called a *contour plot*.
- If you walk along a contour line, you remain at the same elevation. Contour plots can help you visualize the shape of a function.
- The contour(X,Y,Z) function is used.
- You use this function the same way you use the mesh function; that is, first use meshgrid to generate the grid [X,Y], after that, generate the function values [z=f(x,y)], then generate contour(X,Y,Z).

Contour

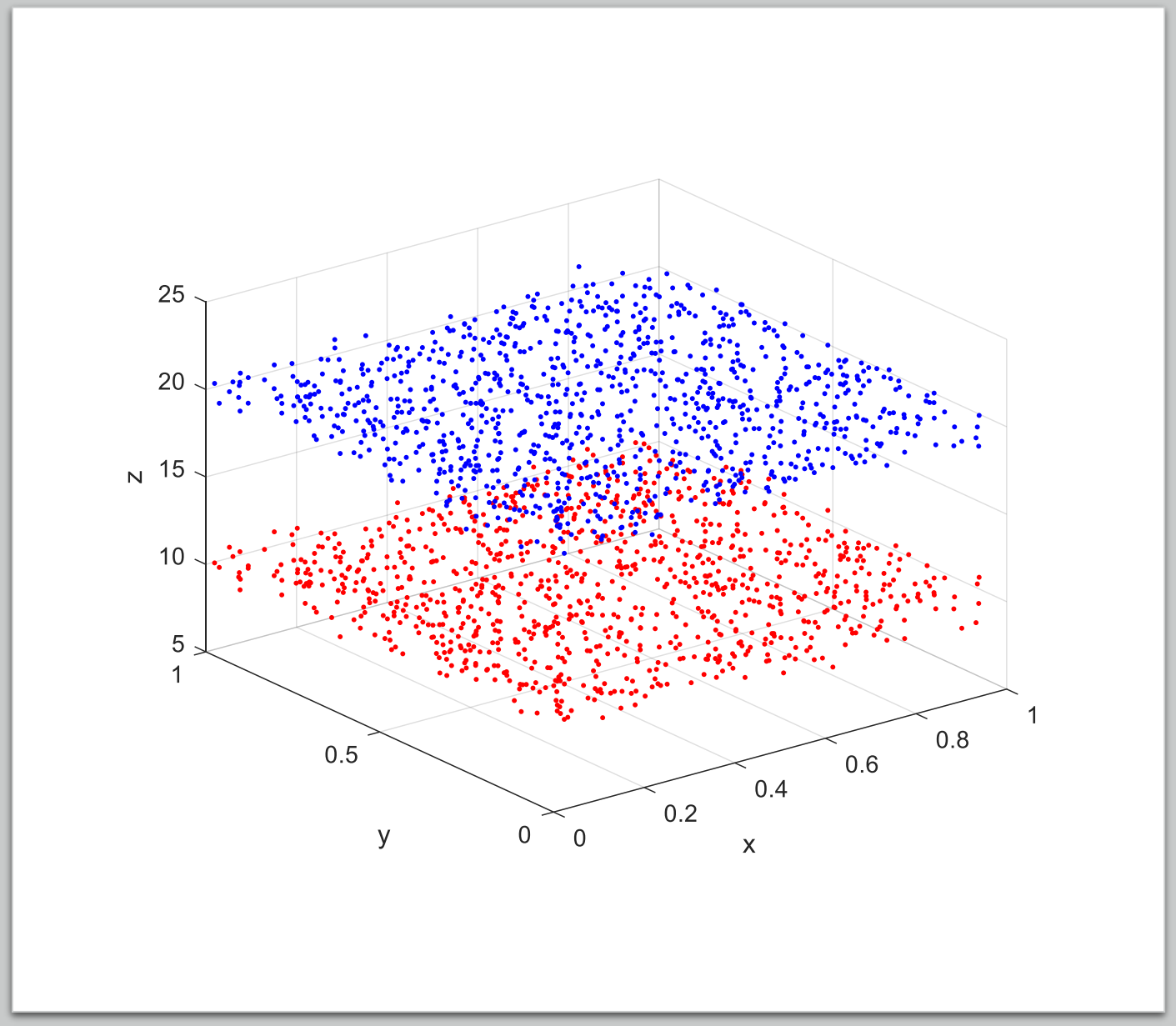
```
% contourPlot3D_v2.m
clc, clear, clf
% Construct the grid in the plane x-y
x=-2:0.1:2; y=x;
[X,Y] = meshgrid(x,y);

% Evaluate the function z=f(x,y)
Z = X.*exp(-((X-Y.^2).^2+Y.^2));

% contour plot
[C,h]=contour(X,Y,Z);
clabel(C,h);
xlabel('x'),ylabel('y'),zlabel('z');
```



clabel(C,h) labels the current contour plot with rotated text inserted into each contour line. The contour lines must be long enough to fit the label, otherwise clabel does not insert a label. C is a two row matrix that contains the data that define the contour lines. Here h is the contour object (lines)

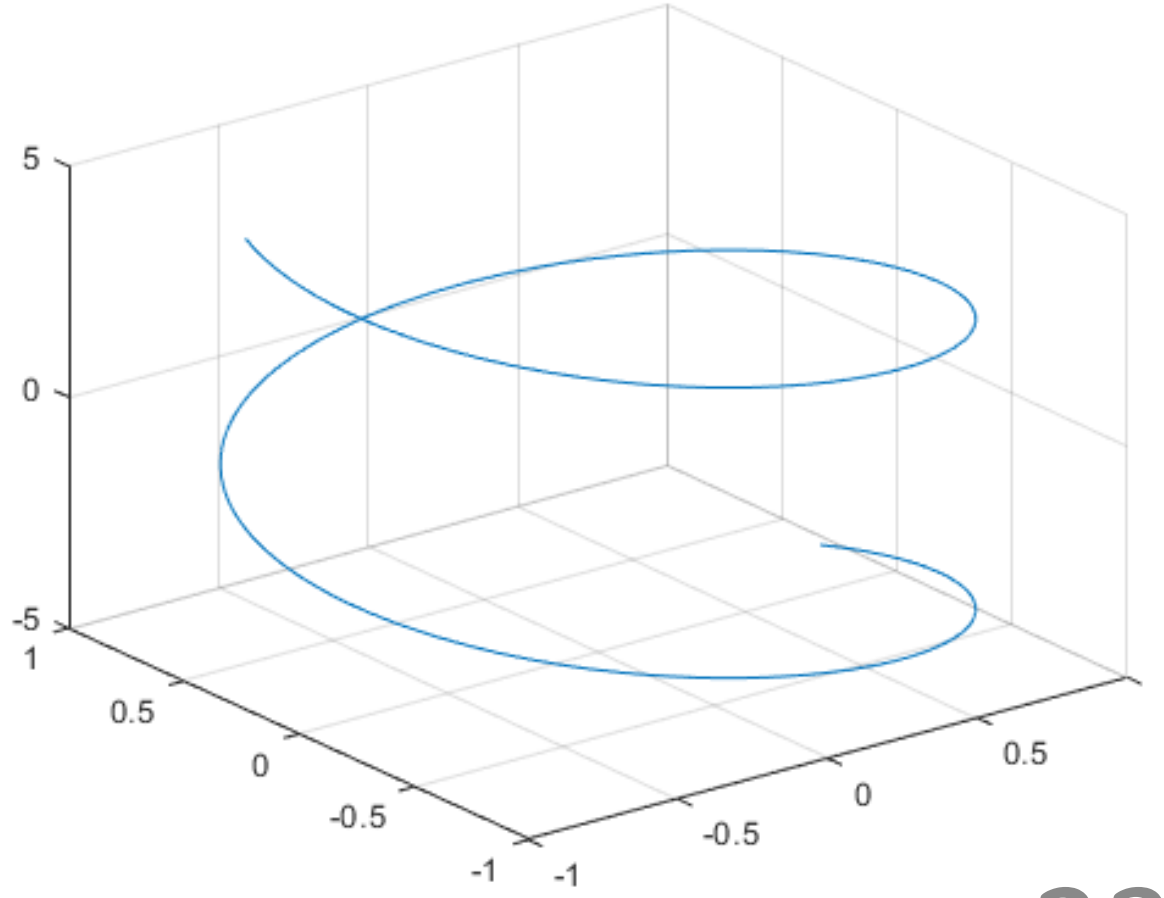
Scatter Plot

```
N = 1000;
x = rand(N,1);
y = rand(N,1);
z1 = 20+0.5*randn(N, 1);
z2 = 10+0.5*randn(N, 1);
figure(1)
scatter3(x, y, z1, '.b')
hold on
scatter3(x, y, z2, '.r')
hold off
grid on
xlabel('x')
ylabel('y')
zlabel('z')
```



fplot3: Used to Plot functions (as opposite to discrete triplets)

```
% fplot3Example.m
% Plot an spiral
clc, clear, clf


xt=@(t) sin(t);
yt=@(t) cos(t);
zt=@(t) t;
fplot3(xt,yt,zt)  % not available in R2015b
```



SYNTAX

fplot3(funx, funy, funz)

Where funx, funy, funz are function names or function handlers. Fplot3 makes 3D plots in a default range of [-5,5] of the arguments.

https://www.mathworks.com/help/matlab/ref/fplot3.html

References

On the Parametric Equations

- http://tutorial.math.lamar.edu/Classes/CalcII/ParametricEqn.aspx

On the equations of lines in 3D space:

- http://tutorial.math.lamar.edu/Classes/CalcIII/EqnsOfLines.aspx