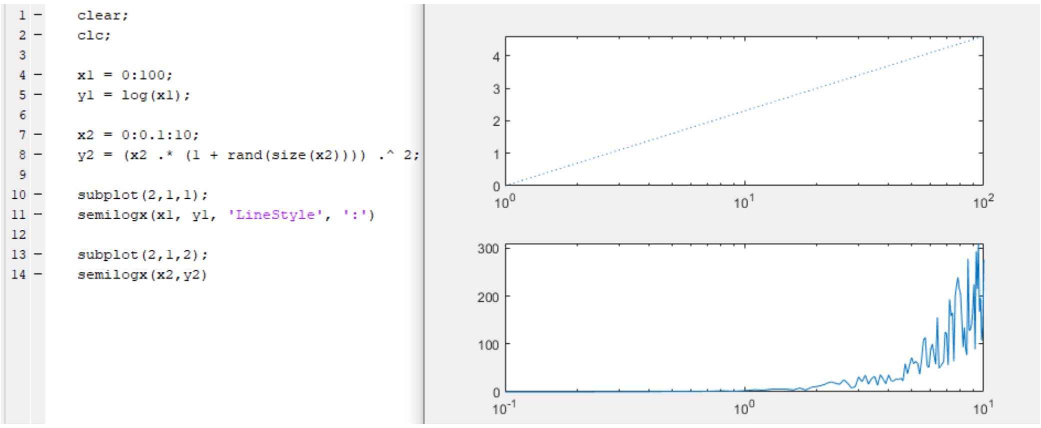
MATLAB – HW3 Upload Date: 13 Nov. 2020

Dorreen Rostami – 97243034
Zahra Hashemi – 97243072

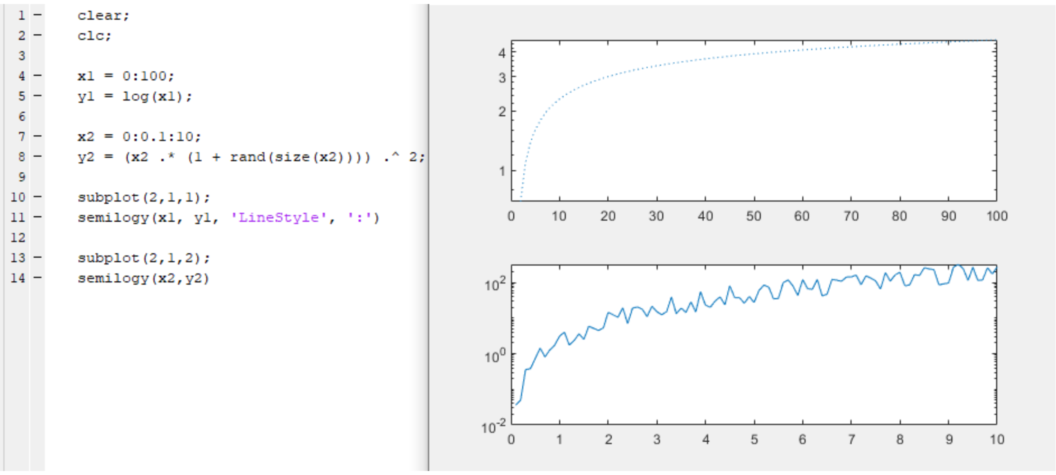
3-1)

	Syntax	Description
semilogx	semilogx(Y)	semilogx plot data as logarithmic scales for the x-axis. semilogx(Y) creates a plot using a base 10 logarithmic scale for the x-axis and a linear scale for the y-axis.
	semilogx(X1,Y1,...)	plots all Yn versus Xn pairs. If only one of Xn or Yn is a matrix, semilogx plots the vector argument versus the rows or columns of the matrix, along the dimension of the matrix whose length matches the length of the vector. If the matrix is square, its columns plot against the vector if their lengths match.
	semilogx(X1,Y1,LineSpec,...)	plots all lines defined by the Xn,Yn,LineSpec triples. LineSpec determines line style, marker symbol, and color of the plotted lines.
	semilogx(...,'PropertyName',PropertyValue,...)	sets property values for all charting lines created.
	semilogx(ax,...)	creates the line in the axes specified by ax instead of in the current axes (gca). The option ax can precede any of the input argument combinations in the previous syntaxes.
	h = semilogx(...)	returns a vector of chart line objects.



```
clear;
clc;

x1 = 0:100;
y1 = log(x1);

x2 = 0:0.1:10;
y2 = (x2 .* (1 + rand(size(x2)))) .^ 2;

subplot(2,1,1);
semilogx(x1, y1, 'LineStyle', ':')

subplot(2,1,2);
semilogx(x2,y2)
```

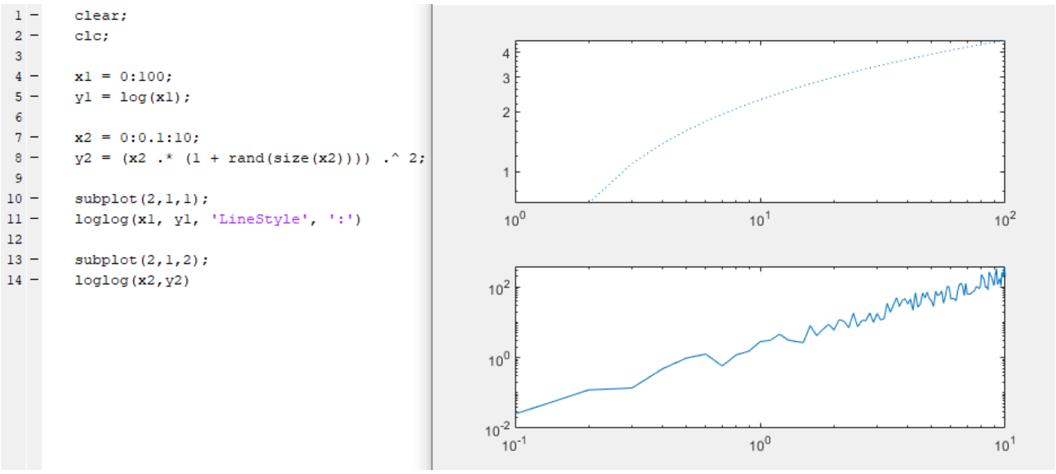
	Syntax	Description
semilogy	semilogy(Y)	semilogy plot data as logarithmic scales for the y-axis. semilogy(Y) creates a plot using a base 10 logarithmic scale for the y-axis and a linear scale for the x-axis.
	semilogy(X1,Y1,...)	plots all Yn versus Xn pairs. If only one of Xn or Yn is a matrix, semilogy plots the vector argument versus the rows or columns of the matrix, along the dimension of the matrix whose length matches the length of the vector. If the matrix is square, its columns plot against the vector if their lengths match.
	semilogy(X1,Y1,LineSpec,...)	plots all lines defined by the Xn,Yn,LineSpec triples. LineSpec determines line style, marker symbol, and color of the plotted lines.
	semilogy(...,'PropertyName',Property Value,...)	sets property values for all charting lines created.
	semilogy(ax,...)	creates the line in the axes specified by ax instead of in the current axes (gca). The option ax can precede any of the input argument combinations in the previous syntaxes.
	h = semilogy(...)	returns a vector of chart line objects.



```
clear;
clc;

x1 = 0:100;
y1 = log(x1);

x2 = 0:0.1:10;
y2 = (x2 .* (1 + rand(size(x2)))) .^ 2;

subplot(2,1,1);
semilogy(x1, y1, 'LineStyle', ':')

subplot(2,1,2);
semilogy(x2,y2)
```

3-2)

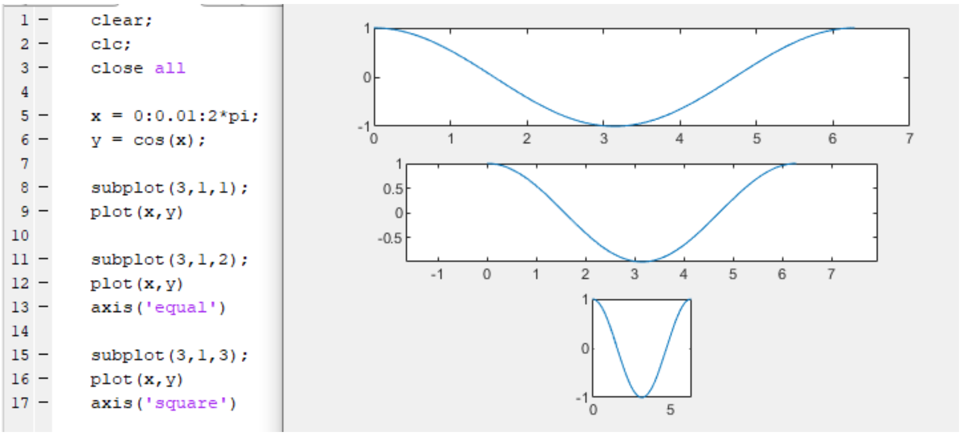
	Syntax	Description
loglog	loglog(Y)	plots the columns of Y versus their index if Y contains real numbers. If Y contains complex numbers, loglog(Y) and loglog(real(Y),imag(Y)) are equivalent. loglog ignores the imaginary component in all other uses of this function.
	loglog(X1,Y1,...)	plots all Yn versus Xn pairs. If only one of Xn or Yn is a matrix, loglog plots the vector argument versus the rows or columns of the matrix, along the dimension of the matrix whose length matches the length of the vector. If the matrix is square, its columns plot against the vector if their lengths match.
	loglog(X1,Y1,LineSpec,...)	plots all lines defined by the Xn,Yn,LineSpec triples, where LineSpec determines line type, marker symbol, and color of the plotted lines.
	loglog(...,'PropertyName',PropertyValue,...)	sets property values for all charting lines created.
	loglog(ax,...)	creates the line in the axes specified by ax instead of in the current axes (gca). The option ax can precede any of the input argument combinations in the previous syntaxes.
	h = loglog (...)	returns a vector of chart line objects.



```
clear;
clc;

x1 = 0:100;
y1 = log(x1);

x2 = 0:0.1:10;
y2 = (x2 .* (1 + rand(size(x2)))) .^ 2;

subplot(2,1,1);
loglog(x1, y1, 'LineStyle', ':')

subplot(2,1,2);
loglog(x2,y2)
```

3-3)

	Syntax	Description
axis	axis(limits)	specifies the limits for the current axes as a vector of four, six, or eight elements.
	axis style	uses a predefined style to set the limits and scaling. For example, specify the style as equal to use *equal* data unit lengths along each axis.
	axis mode	sets whether MATLAB automatically chooses the limits or not. Specify the mode as *manual*, *auto*, or one of the semiautomatic options, such as '*auto x*'.
	axis ydirection	where ydirection is *ij*, places the origin at the upper left corner of the axes.
	axis visibility	where visibility is *off*, turns off the display of the axes background. Plots in the axes still display.
	lim = axis	returns the x-axis and y-axis limits for the current axes. For 3-D axes, it also returns the z-axis limits. For polar axes, it returns the theta-axis and r-axis limits.
	[m,v,d] = axis('state')	returns the current settings for the axis limit selection, the axes visibility, and the y-axis direction.
	____ = axis(ax,____)	uses the axes or polar axes specified by ax instead of the current axes. Specify ax as the first input argument for any of the previous syntaxes. Use single quotes around input arguments that are character vectors, such as axis(ax,'equal').

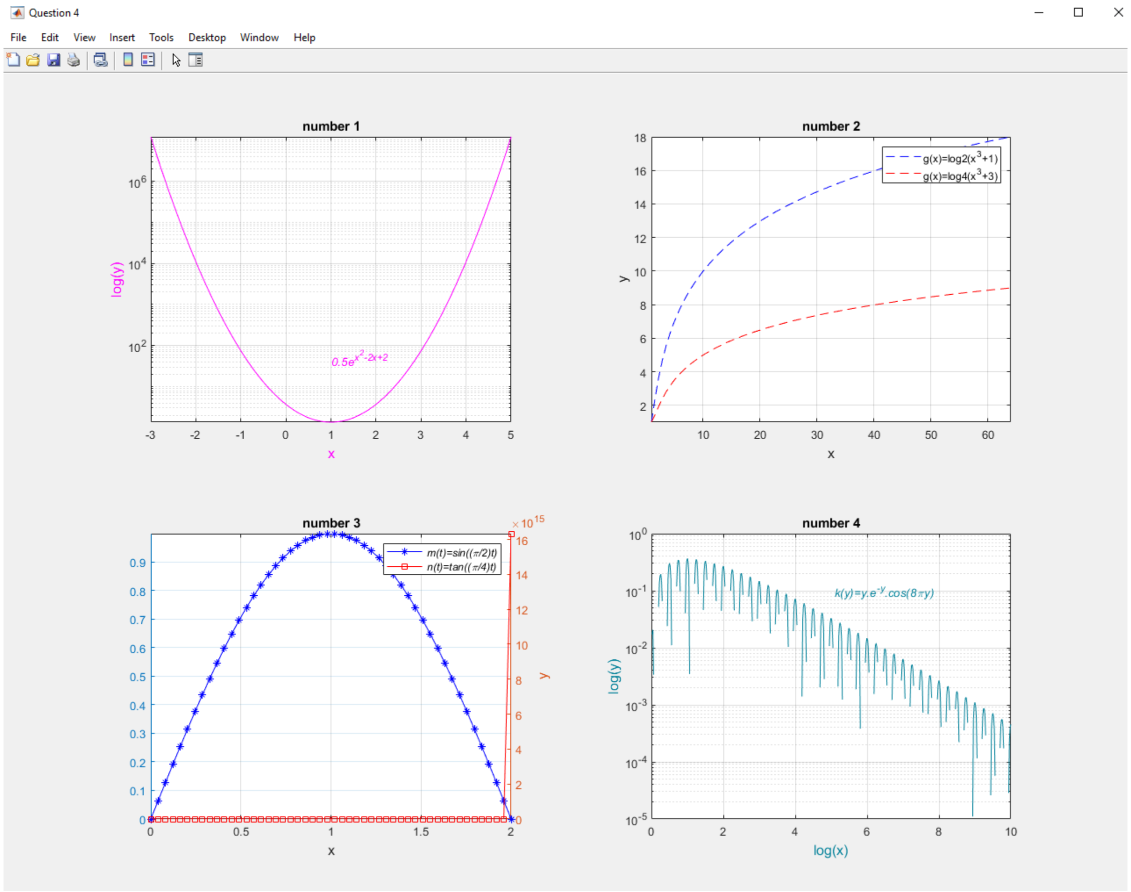


```
clear;
clc;
close all

x = 0:0.01:2*pi;
y = cos(x);

subplot(3,1,1);
plot(x,y)

subplot(3,1,2);
plot(x,y)
axis('equal')

subplot(3,1,3);
plot(x,y)
axis('square')
```

3-4)

Code is in “Q4.m”

Plots:

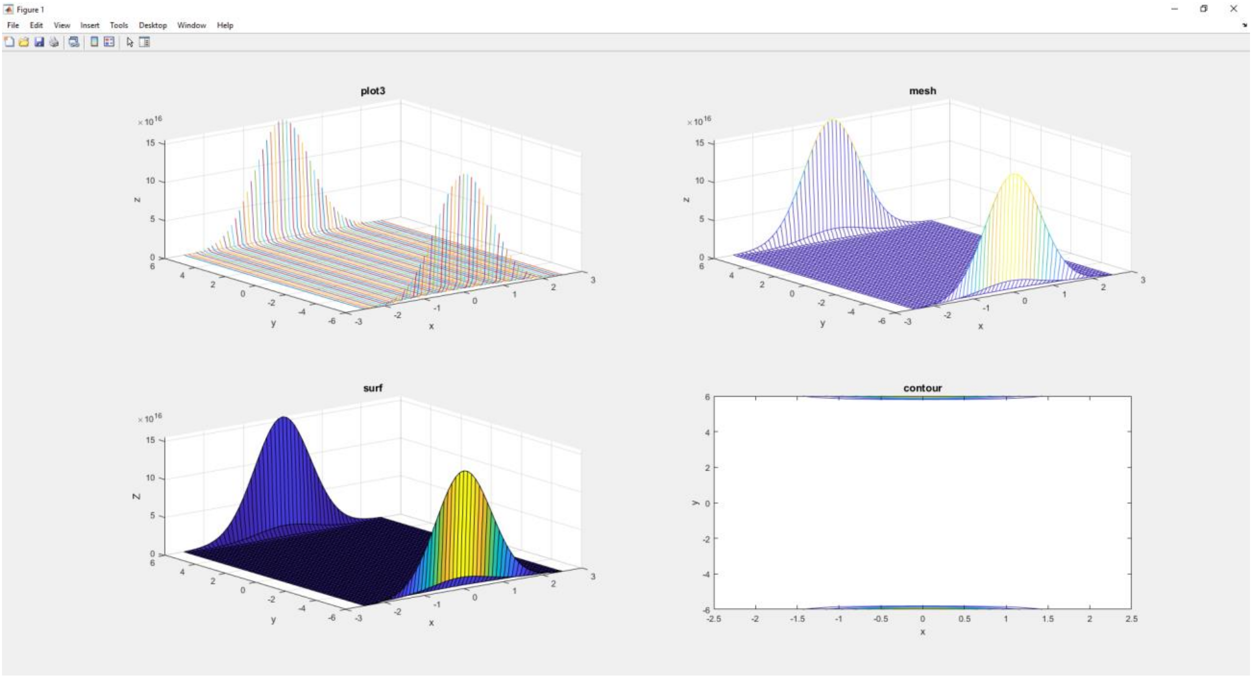


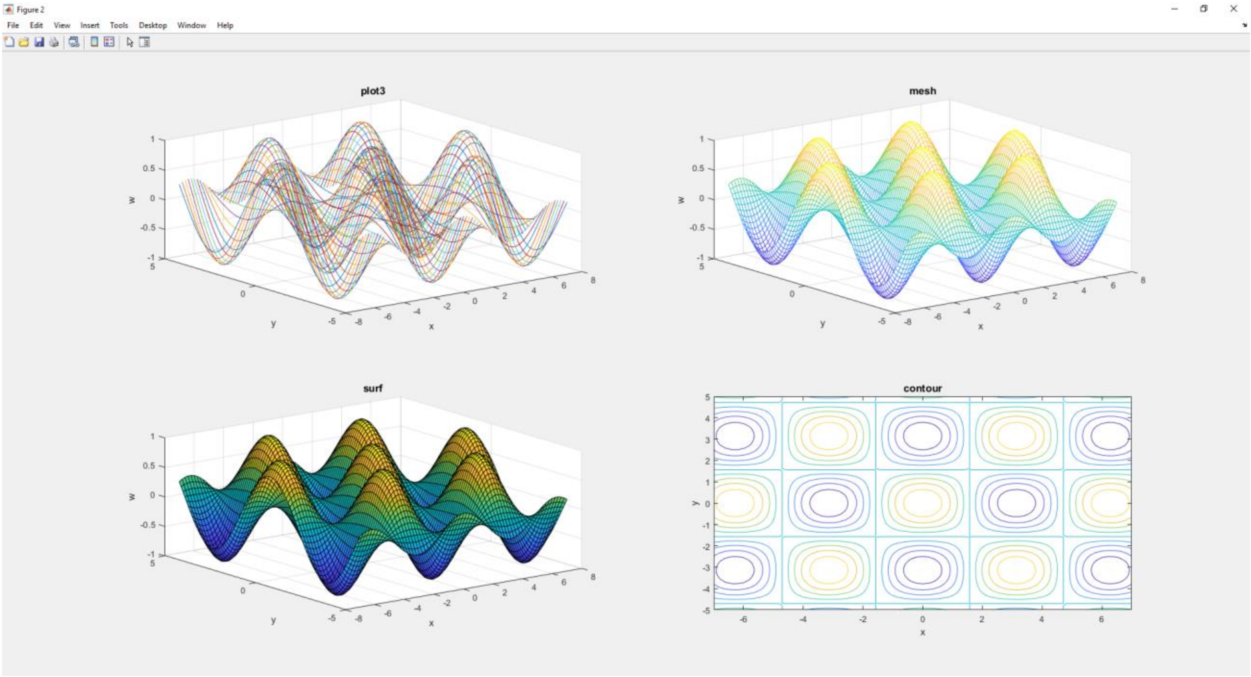
3-5)

- [X,Y] = meshgrid(x,y) returns 2-D grid coordinates based on the coordinates contained in vectors x and y.

Code is in “Q5.m”

Plots:





3-6)

- The *plot3* function displays a plot of a set of data points meaning there is no solid surface between them.
- The *surf* function creates a surface plot.
- The *mesh* function draws a wireframe mesh with color proportional to the surface height.
- The *contour* function creates a contour plot containing the isolines of matrix Z, where Z contains height values on the x-y plane. A contour line (also isoline, isopleth, or isarithm) of a function of two variables is a curve along which the function has a constant value, so that the curve joins points of equal value. It is a plane section of the three-dimensional graph of the function f(x, y) parallel to the (x, y)-plane.