Morteza Kazemi | 97243054

Amir Masoud Shaker | 97243081

11/14/2021

Homework #3

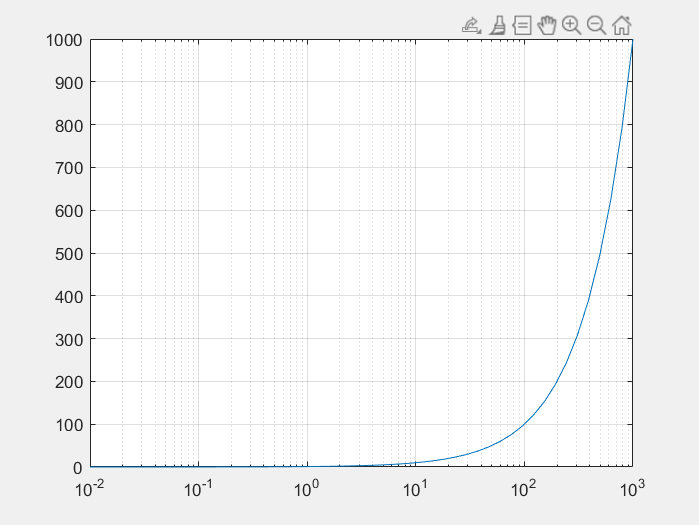
# Q1

|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. semilogx(X,Y) 2. semilogx(X,Y,LineSpec) 3. semilogx(X1,Y1,…,Xn,Yn) 4. semilogx(X1,Y1,LineSpec1,…,Xn,Yn,LineSpecn) 5. semilogx(Y) 6. semilogx(Y,LineSpec) 7. semilogx( \_\_ ,Name,Value) 8. semilogx(ax, \_\_ ) 9. lineobj = semilogx( \_\_ ) | Semilog plot (x-axis has log scale) | 1. This syntax plots x- and y-coordinates using a base-10 logarithmic scale on the x-axis and a linear scale on the y-axis. 2. This syntax creates the plot using the specified color, marker, and line style. 3. This syntax plots multiple pairs of x- and y-coordinates on the same set of axes. 4. This syntax assigns specific colors, markers, and line styles to each x-y pair. 5. This syntax plots Y against an implicit set of x-coordinates.   Note: If Y is a vector, the x-coordinates range from 1 to length(Y).  Note: If Y is a matrix, the plot contains one line for each column in Y. The x-coordinates range from 1 to the number of rows in Y.   1. This syntax specifies color, marker, and line style. 2. This syntax specifies Line properties using one or more Name,Value pair arguments. 3. This syntax displays the plot in the target axes. 4. This syntax returns a Line object or an array of Line objects. |

Examples:

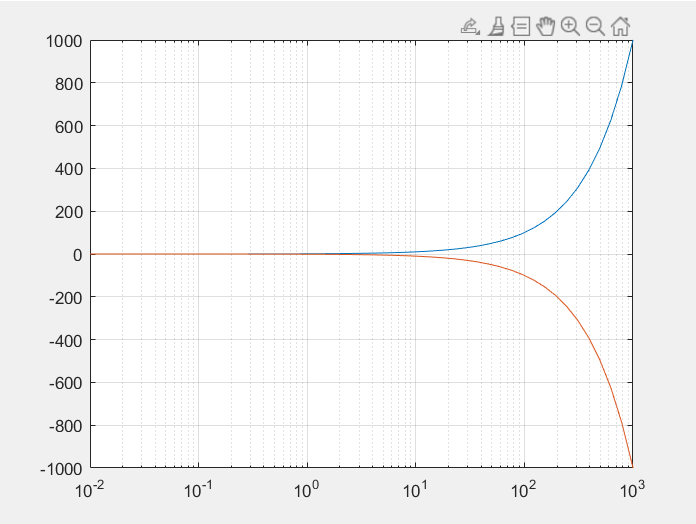
Semilogx1:





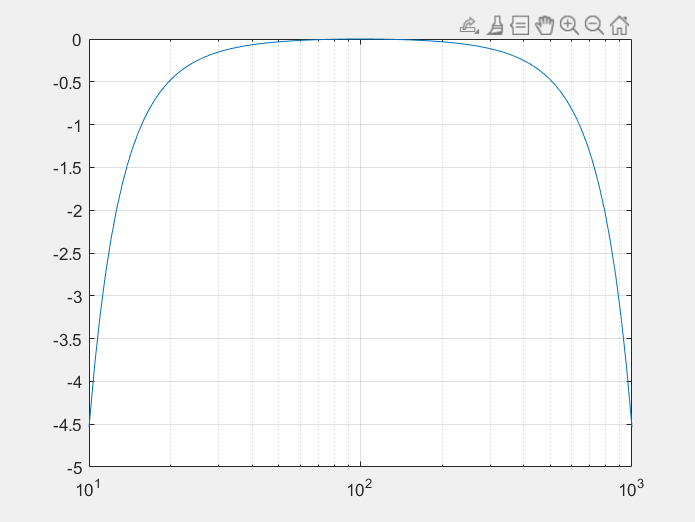
Semilogx2:





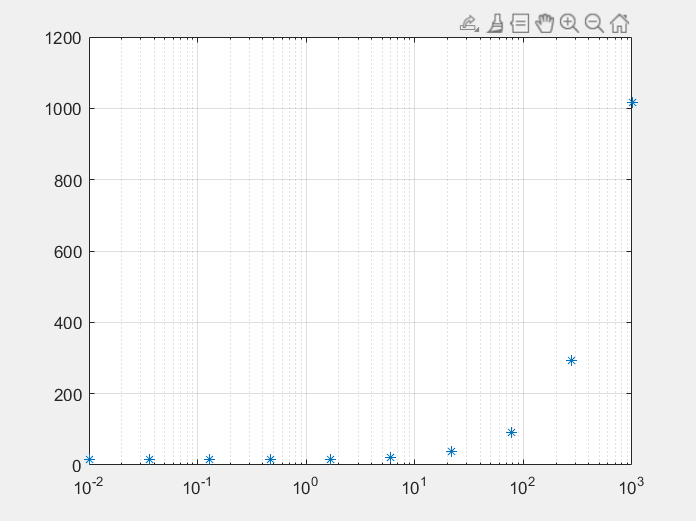
Semilogx3:





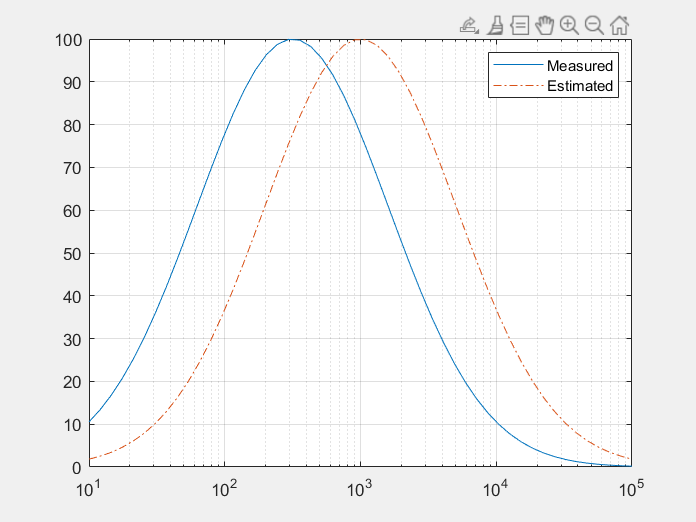
Semilogx4:





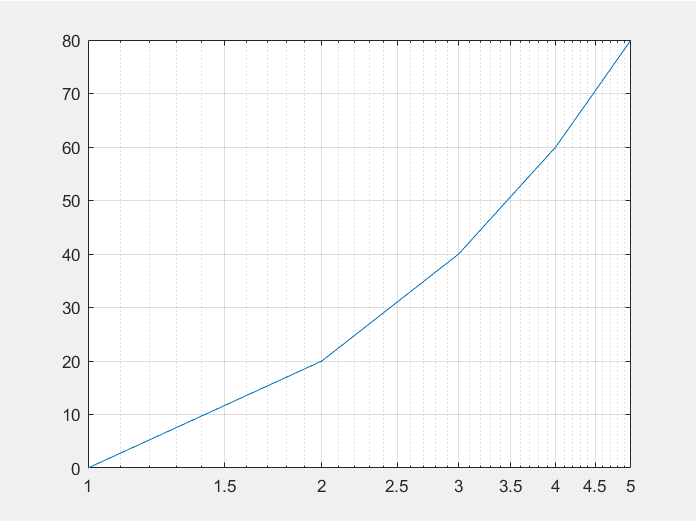
Semilogx5:





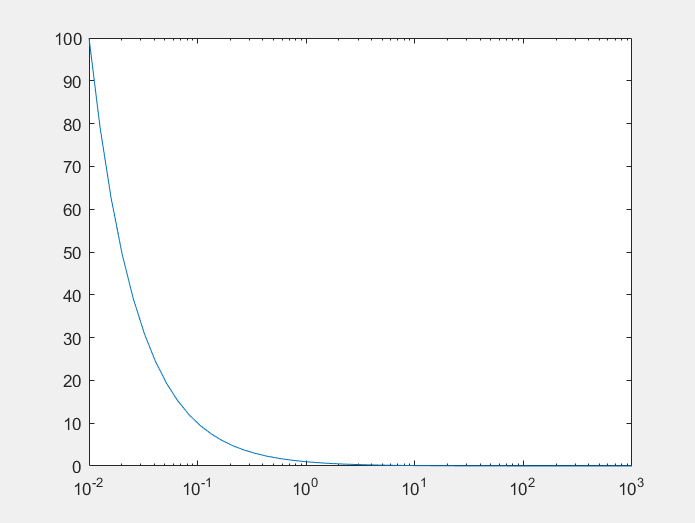
Semilogx6:





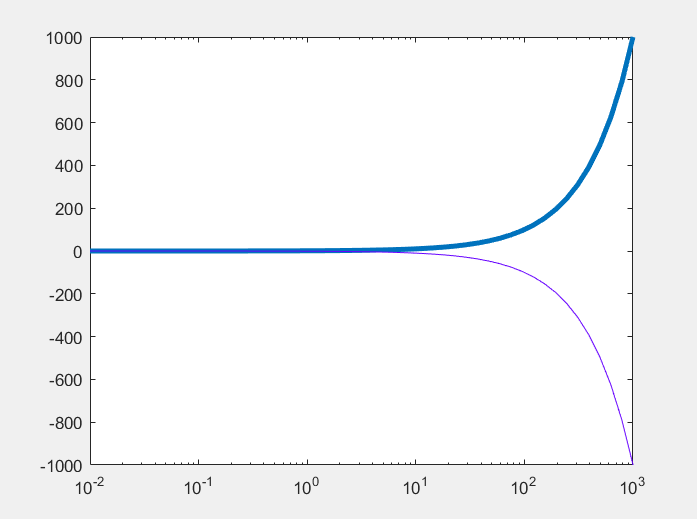
Semilogx7:





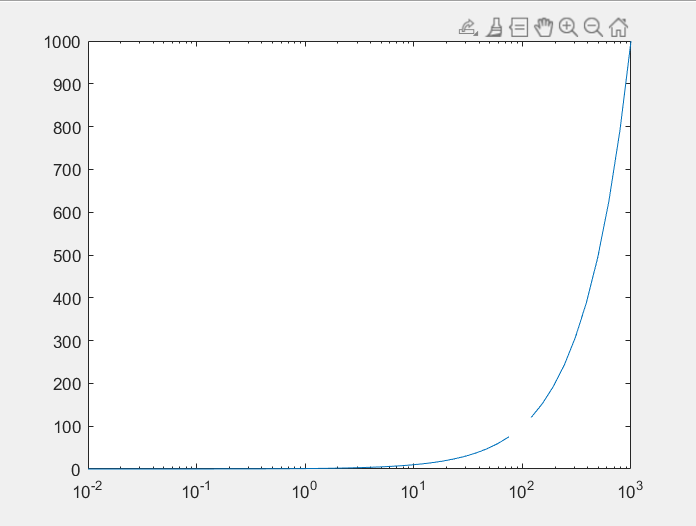
Semilogx8:





Semilogx9:



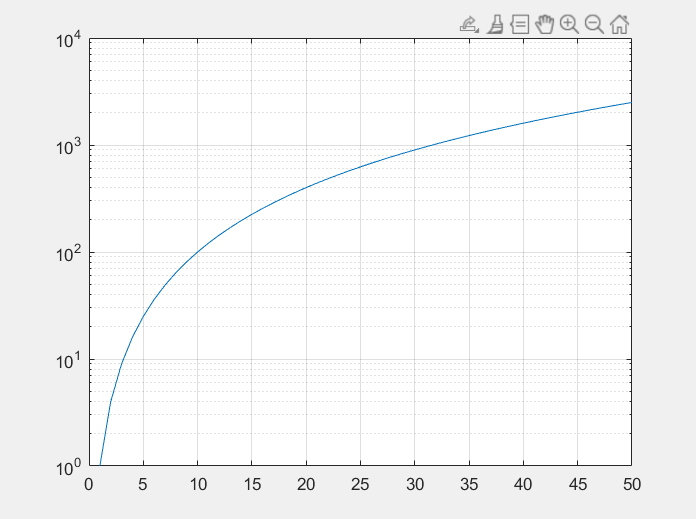


|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. semilogy(X,Y) 2. semilogy(X,Y,LineSpec) 3. semilogy(X1,Y1,…,Xn,Yn) 4. semilogy(X1,Y1,LineSpec1,…,Xn,Yn,LineSpecn) 5. semilogy(Y) 6. semilogy(Y,LineSpec) 7. semilogy( \_\_ ,Name,Value) 8. semilogy(ax, \_\_ ) 9. lineobj = semilogy( \_\_ ) | Semilog plot (y-axis has log scale) | 1. This syntax plots x- and y-coordinates using a base-10 logarithmic scale on the y-axis and a linear scale on the x-axis. 2. This syntax creates the plot using the specified color, marker, and line style. 3. This syntax plots multiple pairs of x- and y-coordinates on the same set of axes. 4. This syntax assigns specific colors, markers, and line styles to each x-y pair. 5. This syntax plots Y against an implicit set of x-coordinates.   Note: If Y is a vector, the x-coordinates range from 1 to length(Y).  Note: If Y is a matrix, the plot contains one line for each column in Y.  The x-coordinates range from 1 to the number of rows in Y.   1. This syntax specifies color, marker, and line style. 2. This syntax specifies Line properties using one or more Name,Value pair arguments. 3. This syntax displays the plot in the target axes. 4. This syntax returns a Line object or an array of Line objects. |

Examples:

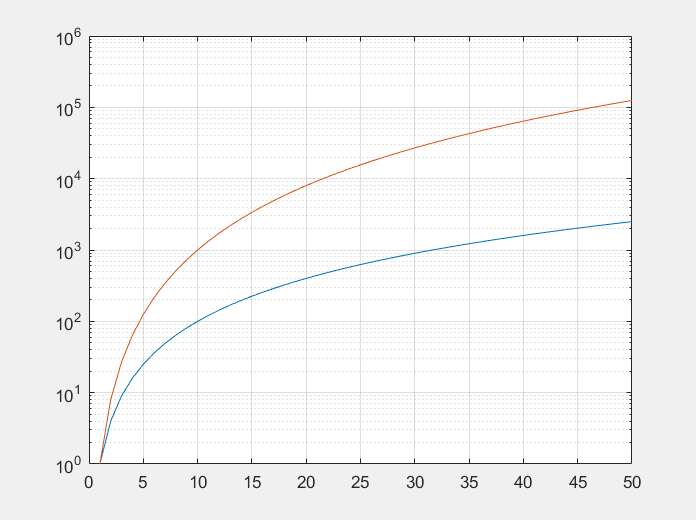
Semilogy1:





Semilogy2:





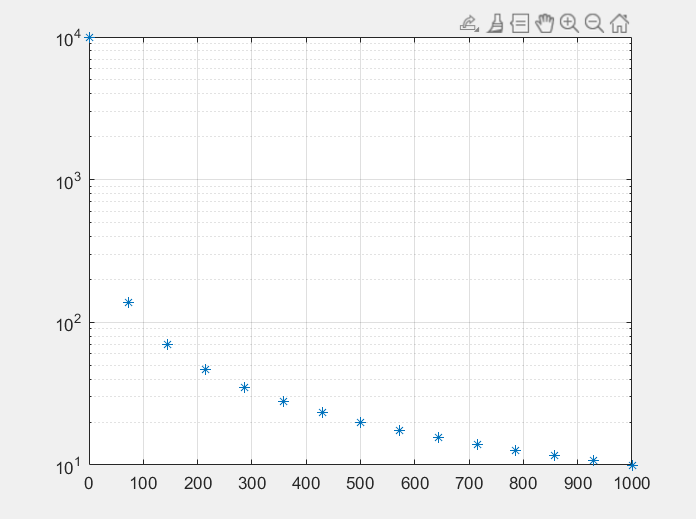
Semilogy3:



# 

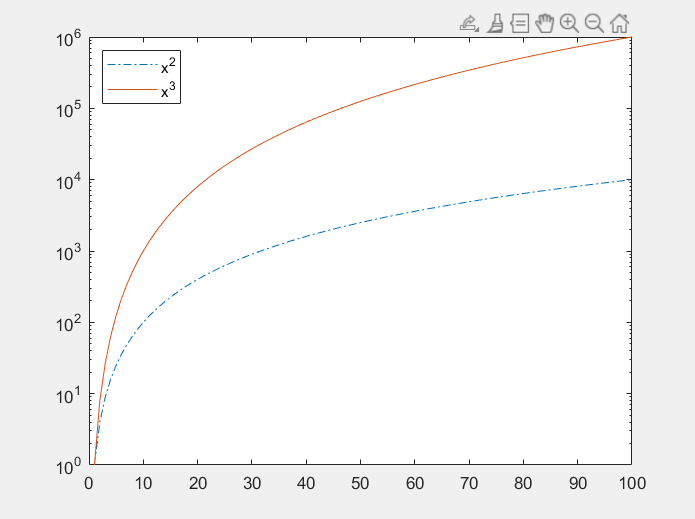
Semilogy4:





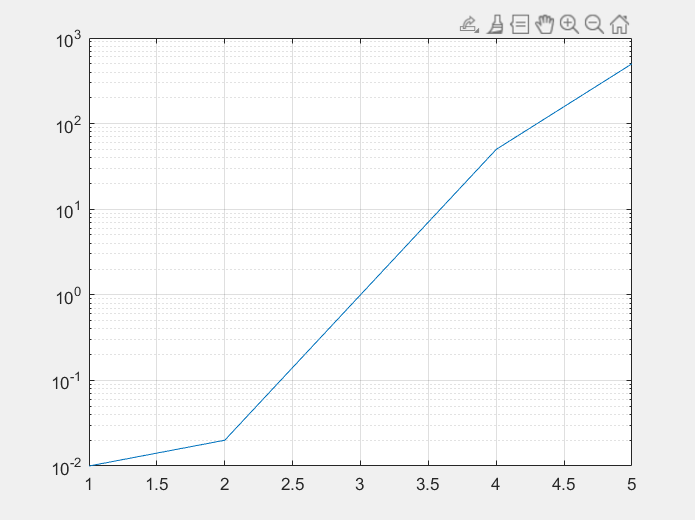
Semilogy5:





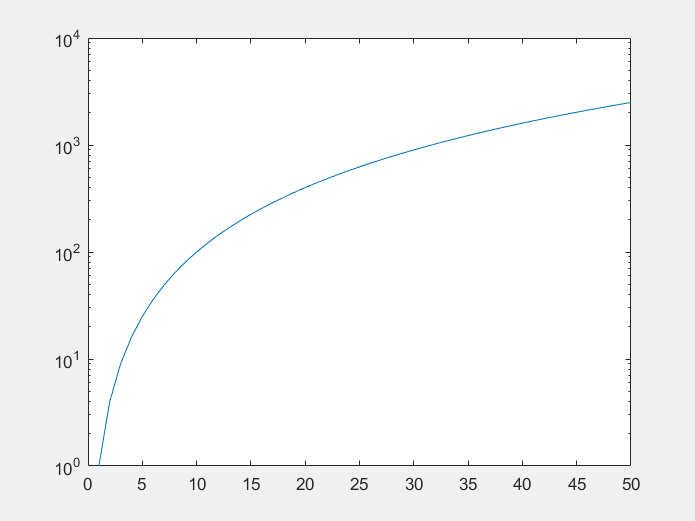
Semilogy6:





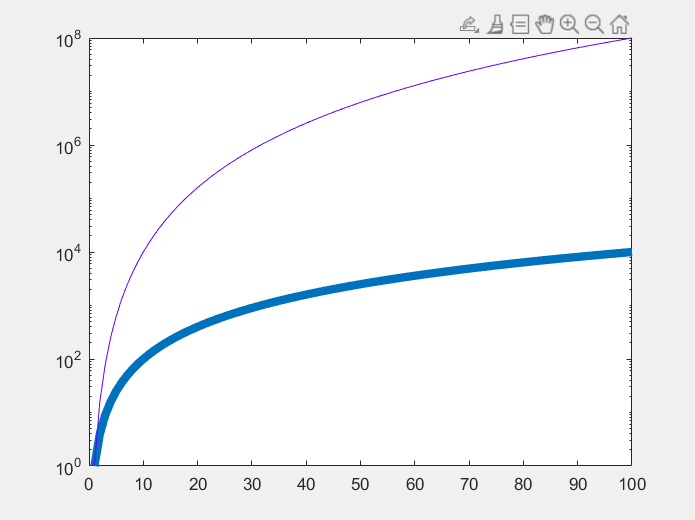
Semilogy7:





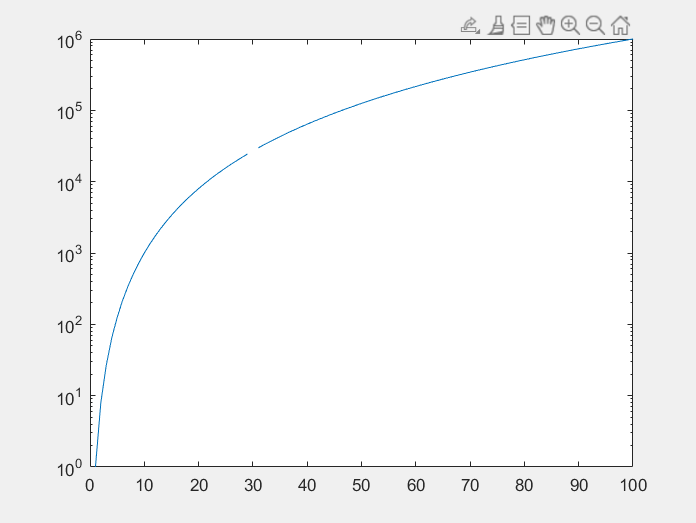
Semilogy8:





Semilogy9:





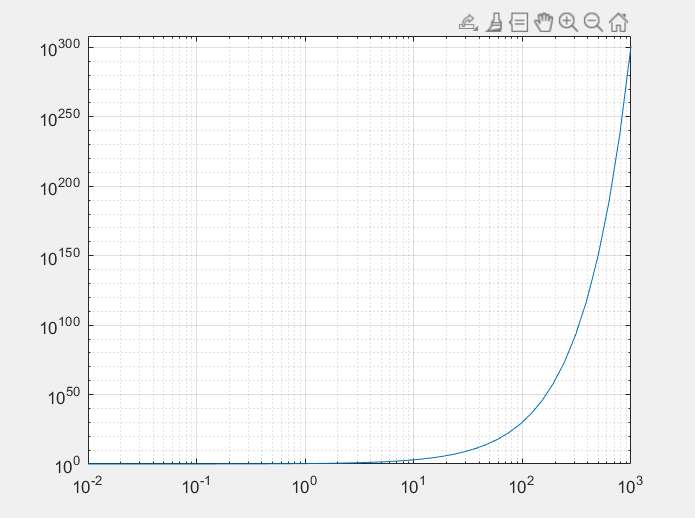
# Q2

|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. loglog(X,Y) 2. loglog(X,Y,LineSpec) 3. loglog(X1,Y1,…,Xn,Yn) 4. loglog(X1,Y1,LineSpec1,…,Xn,Yn,LineSpecn) 5. loglog(Y) 6. loglog(Y,LineSpec) 7. loglog( \_\_ ,Name,Value) 8. loglog(ax, \_\_ ) 9. lineobj = loglog( \_\_ ) | Log-log scale plot | 1. This syntax plots x- and y-coordinates using a base-10 logarithmic scale on the x-axis and the y-axis. 2. This syntax creates the plot using the specified color, marker, and line style. 3. This syntax plots multiple pairs of x- and y-coordinates on the same set of axes. 4. This syntax assigns specific colors, markers, and line styles to each x-y pair. 5. This syntax plots Y against an implicit set of x-coordinates.   Note: If Y is a vector, the x-coordinates range from 1 to length(Y).  Note: If Y is a matrix, the plot contains one line for each column in Y.  The x-coordinates range from 1 to the number of rows in Y.   1. This syntax specifies color, marker, and line style. 2. This syntax specifies Line properties using one or more Name,Value pair arguments. 3. This syntax displays the plot in the target axes. 4. This syntax returns a Line object or an array of Line objects. |

Examples:

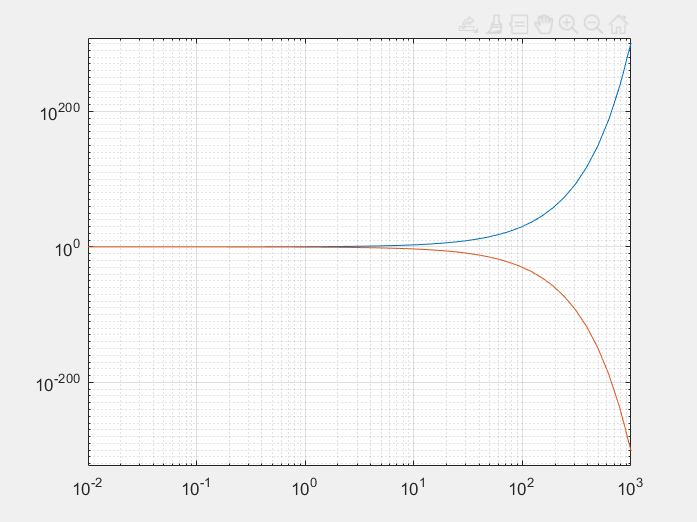
Loglog1:





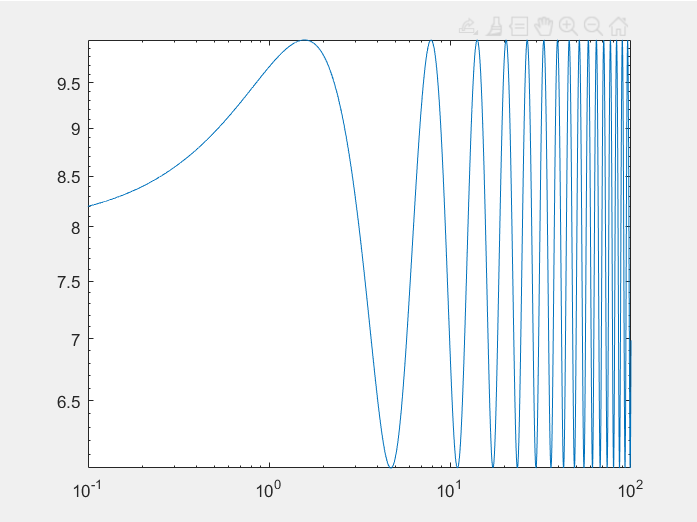
Loglog2:





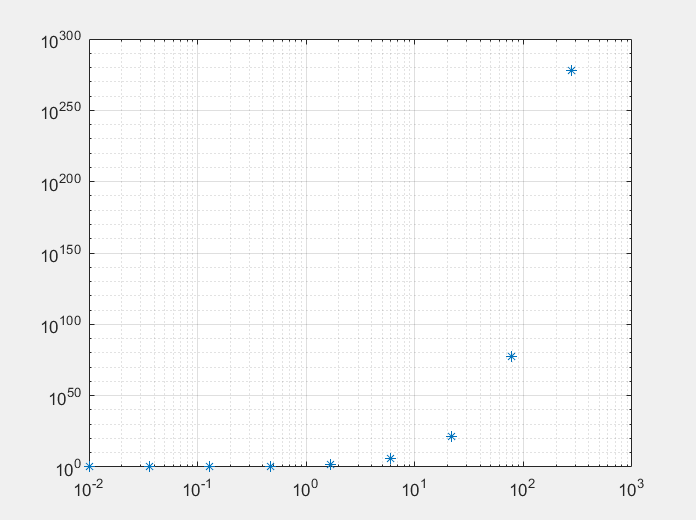
Loglog3:





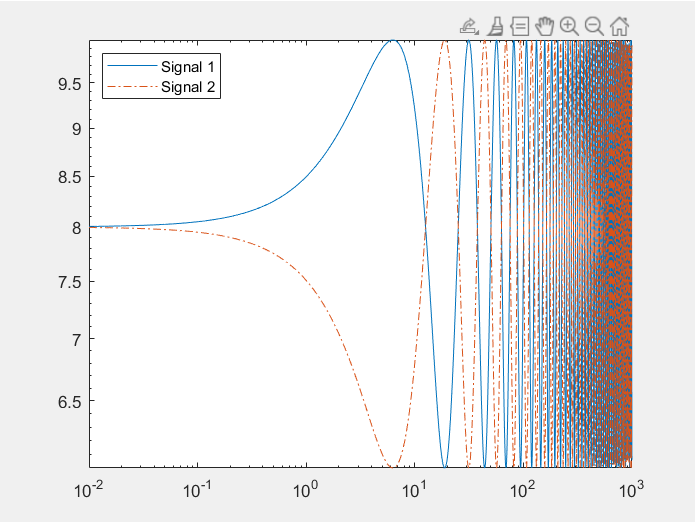
Loglog4:





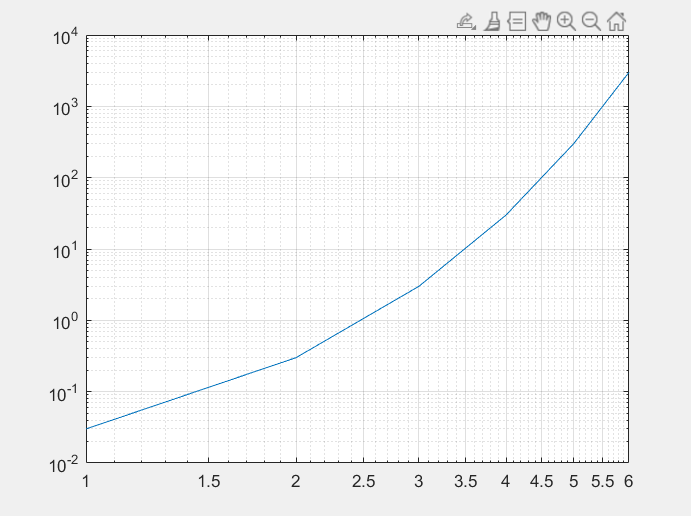
Loglog5:





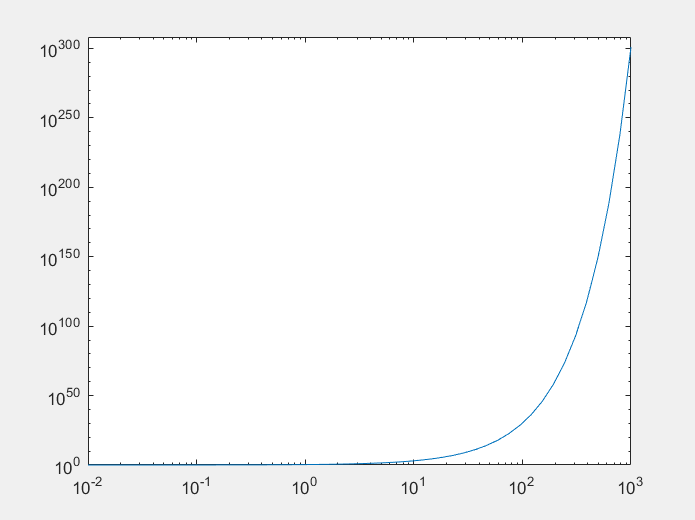
Loglog6:





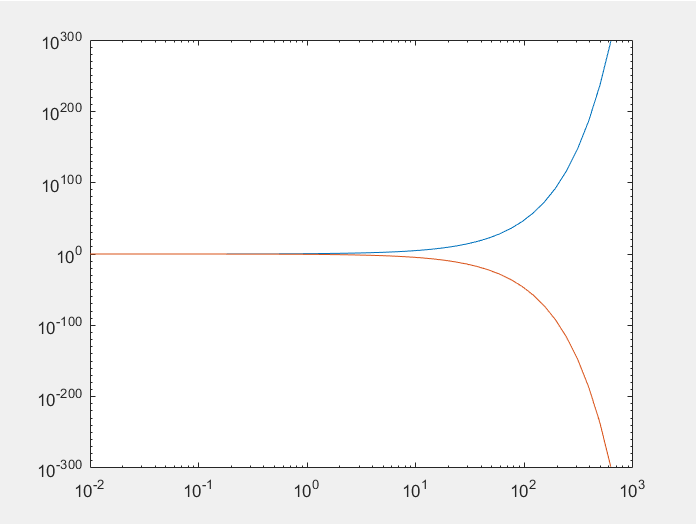
Loglog7:





Loglog8:





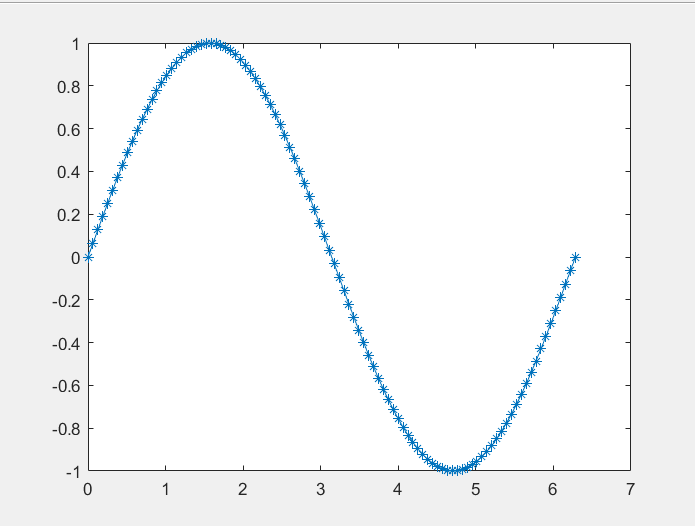
# Q3

|  |  |  |
| --- | --- | --- |
| Syntax | Meaning | Description |
| 1. axis(limits) 2. axis style 3. axis mode 4. axis ydirection 5. axis visibility 6. lim = axis 7. [m,v,d] = axis(‘state’) 8. \_\_ = axis(ax, \_\_ ) | Log-log scale plot | 1. This syntax specifies the limits for the current axes (as a vector of four, six, or eight elements). 2. This syntax uses a predefined style to set the limits and scaling. 3. This syntax sets whether MATLAB automatically chooses the limits or not. 4. This syntax places the origin at the upper left corner of the axes. 5. This syntax turns off the display of the axes background. The default for visibility is on. 6. This syntax returns the x-axis and y-axis limits for the current axes.   Note: For polar axes, returns the theta-axis and r-axis limits.   1. This syntax returns the current settings for the axis limit selection, the axes visibility, and the y-axis direction. 2. This syntax uses the axes or polar axes specified by ax instead of the current axes. |

Examples:

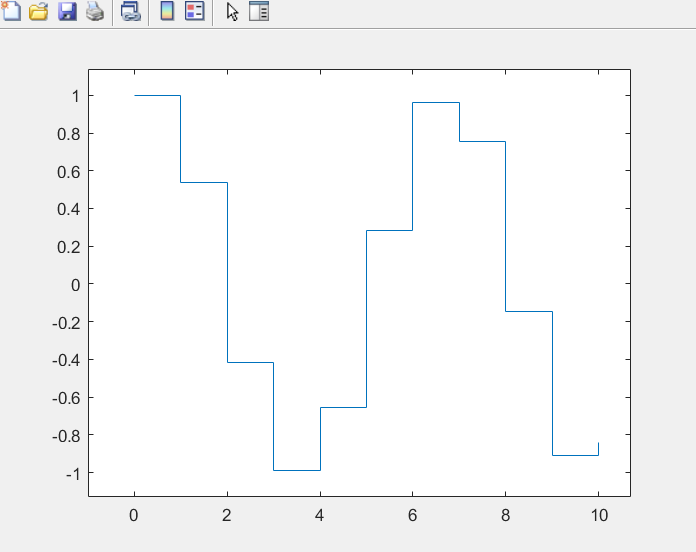
Axis1::





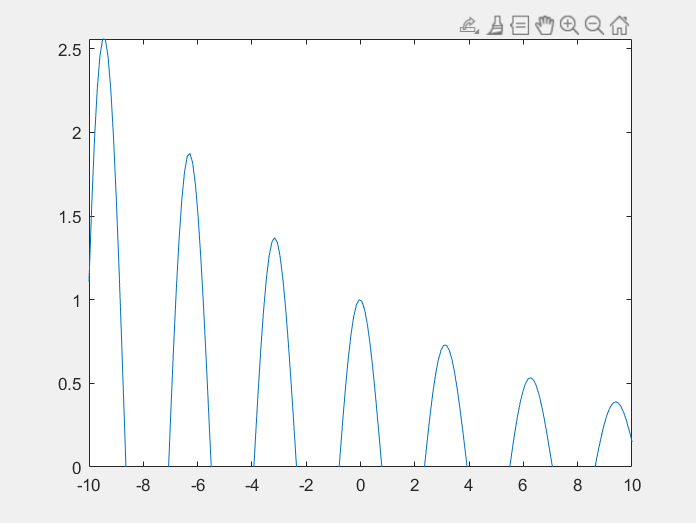
Axis2:





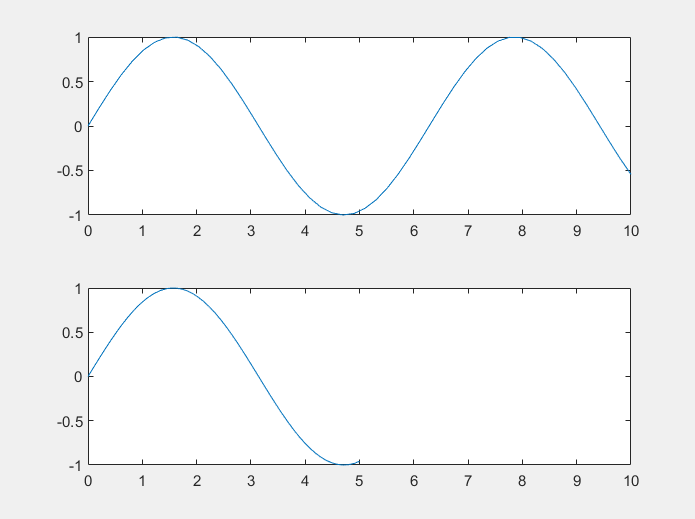
Axis3:





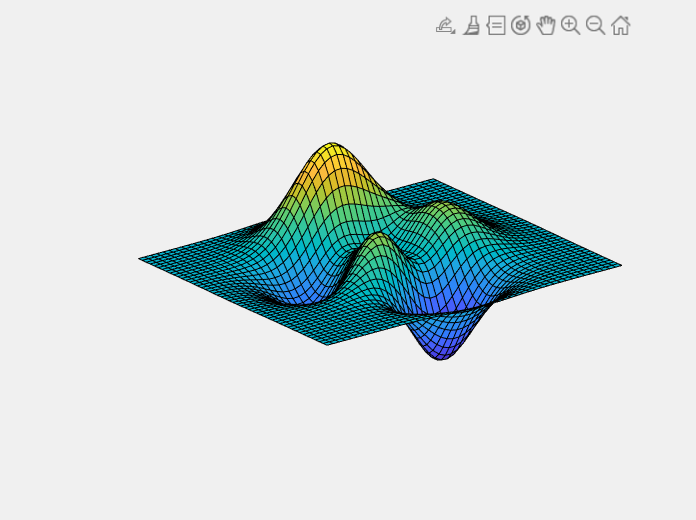
Axis4:





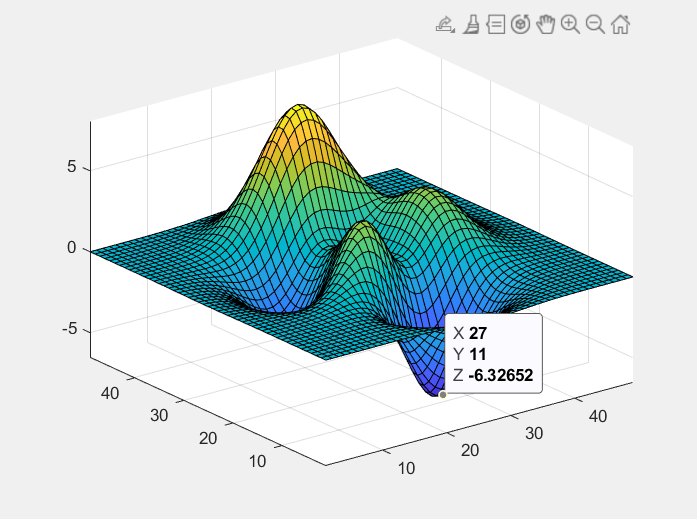
Axis5:





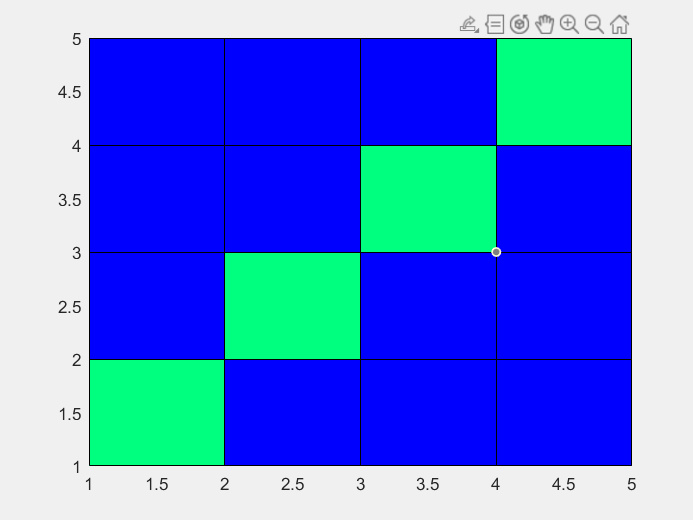
Axis6:





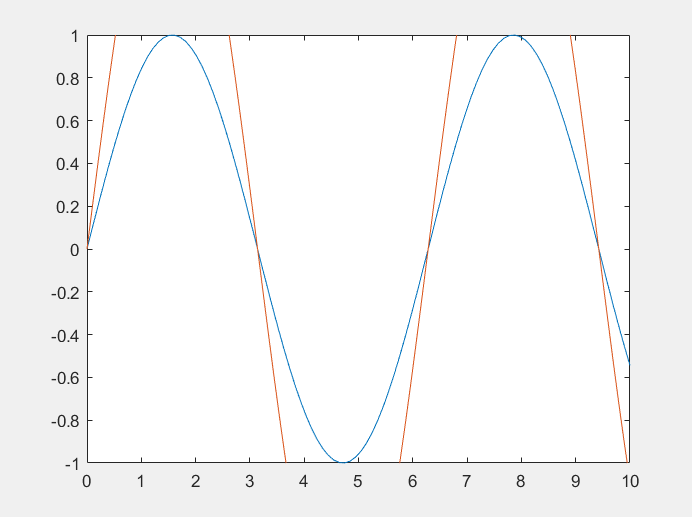
Axis7:





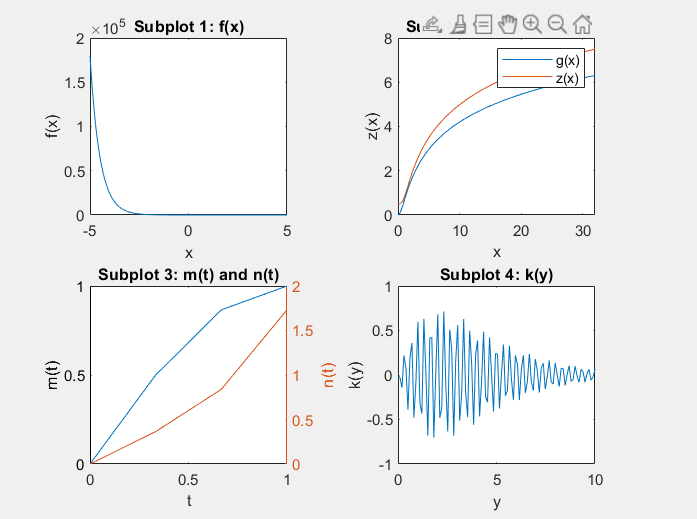
Axis8:





# Q4

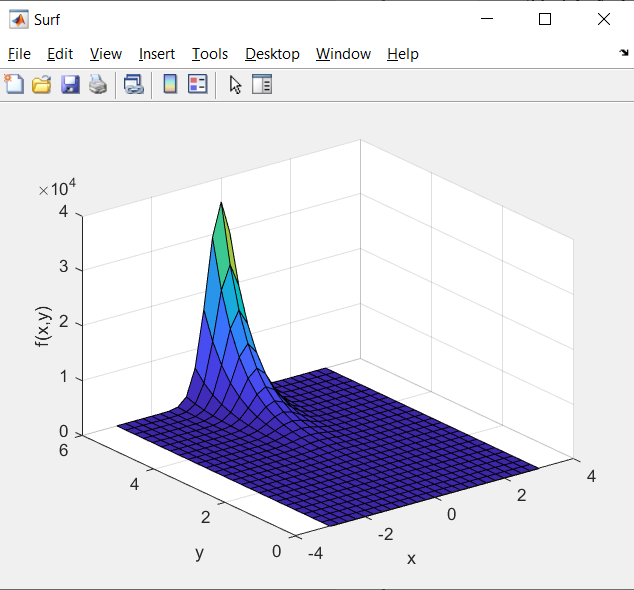


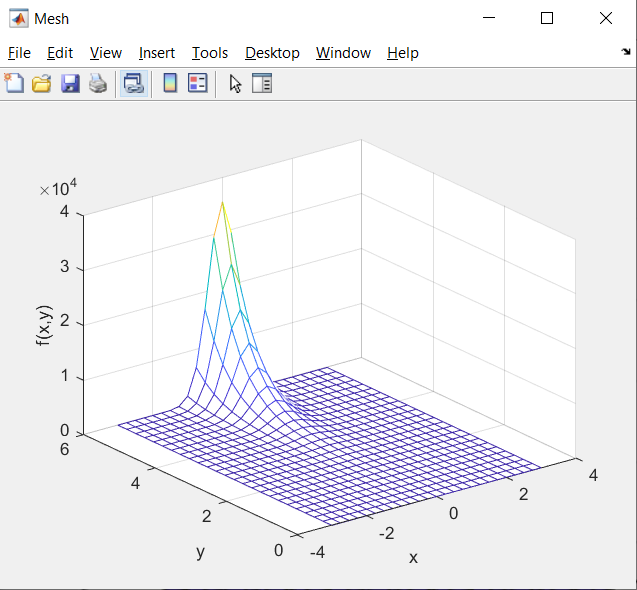


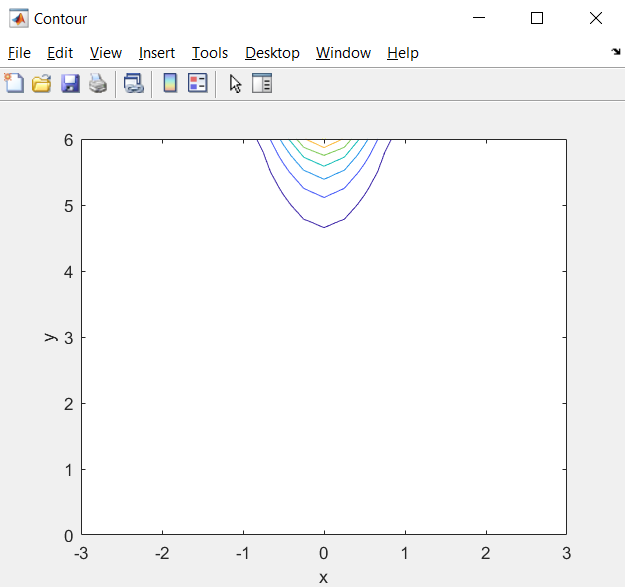
# Q5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Syntax | | Meaning | | Description |
| 1. [X,Y] = meshgrid(x,y) 2. [X,Y] = meshgrid(x) 3. [X,Y,Z] = meshgrid(x,y,z) 4. [X,Y,Z] = meshgrid(x) | 2-D and 3-D grids | | 1. This returns 2-D grid coordinates based on the coordinates contained in vectors x and y. X is a matrix where each row is a copy of x, and Y is a matrix where each column is a copy of y. The grid is represented by the coordinates X and Y has length(y) rows and length(x) columns. 2. This is the same as [X,Y] = meshgrid(x,x). 3. returns 3-D grid coordinates defined by the vectors x, y, and z. The grid represented by X, Y, and Z has size length(y)-by-length(x)-by-length(z). 4. is the same as [X,Y,Z] = meshgrid(x,x,x). | |

For the first function:

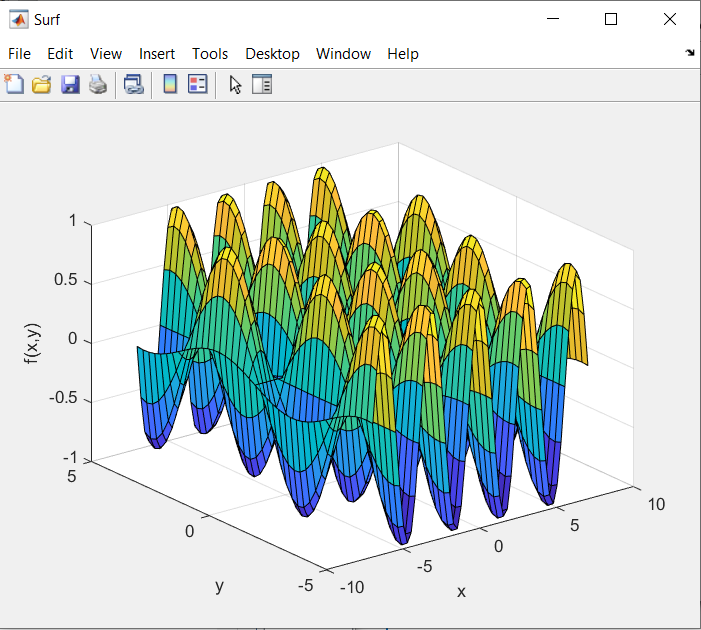


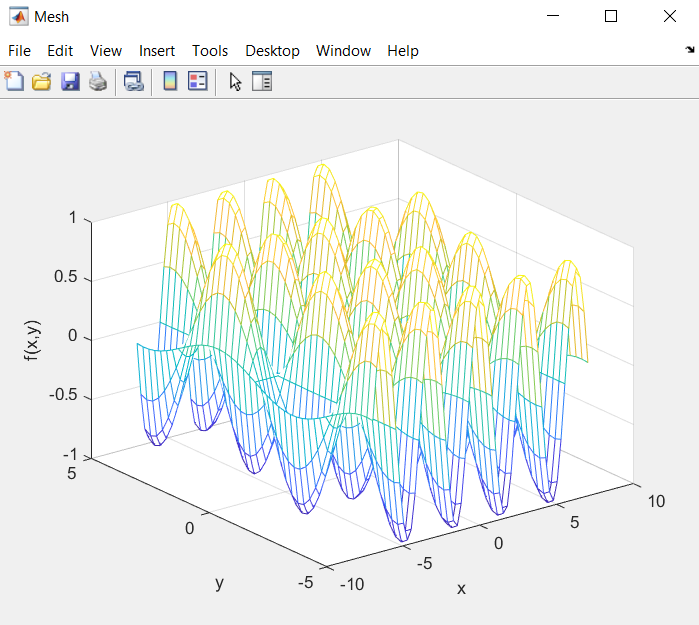


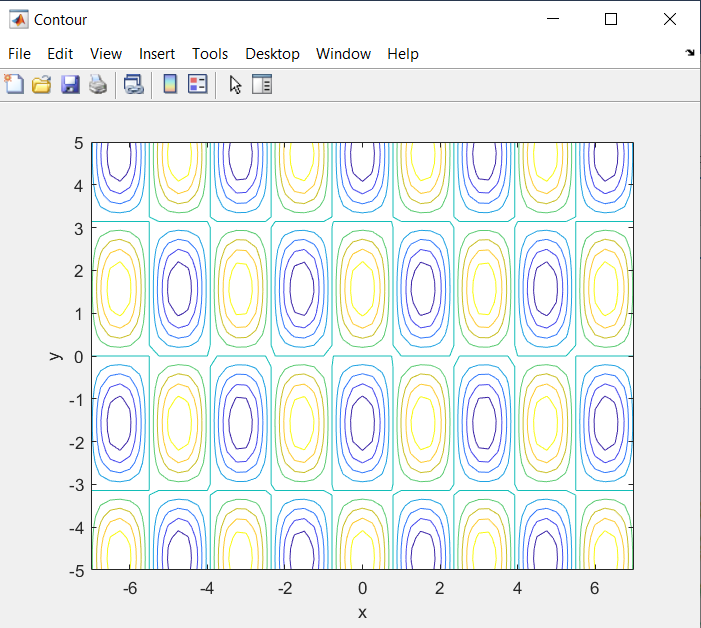


**The plots are in separate figures as stated in the question.**

For the second function:







The code generating these figures is in the next page.



# Q6

plot3() plots some markers (‘.’, ‘\*’, etc.) at the specified points without making a solid surface between them. However surf() creates a solid surface between the points. The notable difference is that mesh draws a three-dimensional surface but plot3 only draws three-dimensional lines.

Mesh() creates a three-dimensional surface that has solid edge colors and no face colors so still, the difference between plot3() and surf() exists because the difference between mesh and surf is only about the surface color.

A contour plot is used to represent a 3-dimensional surface by plotting constant z slices, called contours, on a 2-dimensional format. This is like looking to the plot from upside however, plot3() is creating a 3d plot that we can look at from any angle.

