Practice Exercise 2.1

Type the following expressions into MATLAB® at the command prompt, and observe the results.

```
%Students should perform these tasks in the command window

clear,clc
```

2.1.1

```
5 + 2
```

```
ans =      7
```

2.1.2

```
5*2
```

```
ans =     10
```

2.1.3

```
5/2
```

```
ans =           2.5
```

2.1.4

```
3 + 2*(4 + 3)
```

```
ans =     17
```

2.1.5

```
2.54 * 8/2.6
```

```
ans =        7.8154
```

2.1.6

```
6.3 - 2.1045
```

```
ans =        4.1955
```

2.1.7

```
3.6^2
```

```
ans =          12.96
```

2.1.8

```
1 + 2^2
```

```
ans =      5
```

2.1.9

```
sqrt(5)
```

```
ans =        2.2361
```

2.1.10

```
cos(pi)
```

```
ans =     -1
```

Practice Exercise 2.2

Which of the following names are allowed in MATLAB®? Make your predictions, then test them with the `isvarname`, `iskeyword`, and `which` commands.

```
%Students should perform these tasks in the command window

clear, clc
```

2.2.1

```
isvarname test
```

```
ans =    1
```

```
iskeyword test
```

```
ans =    0
```

```
which test
```

```
C:\Program Files\MATLAB\R2016b\toolbox\stats\stats\@classregtree\test.m  % classregtree method
```

2.2.2

```
isvarname Test
```

```
ans =    1
```

```
iskeyword Test
```

```
ans =    0
```

```
which Test
```

```
C:\Program Files\MATLAB\R2016b\toolbox\stats\stats\@classregtree\test.m  % classregtree method
```

2.2.3

```
isvarname if
```

```
ans =    0
```

```
iskeyword if
```

```
ans =    1
```

```
which if
```

```
built-in (C:\Program Files\MATLAB\R2016b\toolbox\matlab\lang\if)
```

2.2.4

```
isvarname my-book
```

```
ans =    0
```

```
iskeyword my-book
```

```
ans =    0
```

```
which my-book
```

```
'my-book' not found.
```

2.2.5

```
isvarname my_book
```

```
ans =    1
```

```
iskeyword my_book
```

```
ans =    0
```

```
which my_book
```

```
'my_book' not found.
```

2.2.6

```
isvarname Thisisoneverylongnamebutisitstillallowed
```

```
ans =    1
```

```
iskeyword Thisisoneverylongnamebutisitstillallowed
```

```
ans =    0
```

```
which Thisisoneverylongnamebutisitstillallowed
```

```
'Thisisoneverylongnamebutisitstillallowed' not found.
```

2.2.7

```
isvarname 1stgroup
```

```
ans =    0
```

```
iskeyword 1stgroup
```

```
ans =    0
```

```
which 1stgroup
```

```
'1stgroup' not found.
```

2.2.8

```
isvarname group_one
```

```
ans =    1
```

```
iskeyword group_one
```

```
ans =    0
```

```
which group_one
```

```
'group_one' not found.
```

2.2.9

```
isvarname zzaAbc
```

```
ans =    1
```

```
iskeyword zzaAbc
```

```
ans =    0
```

```
which zzaAbc
```

```
'zzaAbc' not found.
```

2.2.10

```
isvarname z34wAwy?12#
```

```
ans =    0
```

```
iskeyword z34wAwy?12#
```

```
ans =    0
```

```
which z34wAwy?12#
```

```
'z34wAwy?12#' not found.
```

2.2.11

```
isvarname sin
```

```
ans =    1
```

```
iskeyword sin
```

```
ans =    0
```

```
which sin
```

```
built-in (C:\Program Files\MATLAB\R2016b\toolbox\matlab\elfun\@double\sin)  % double method
```

2.2.12

```
isvarname log
```

```
ans =    1
```

```
iskeyword log
```

```
ans =    0
```

```
which log
```

```
built-in (C:\Program Files\MATLAB\R2016b\toolbox\matlab\elfun\@double\log)  % double method
```

Practice Exercise 2.3

Predict the results of the following MATLAB® expressions, then check your predictions by keying the expressions into the command window

```
clear, clc
%Students should perform these tasks in the command window
```

2.3.1

```
6/6 + 5
```

```
ans = 6
```

2.3.2

```
2*6^2
```

```
ans = 72
```

2.3.3

```
(3+5)*2
```

```
ans = 16
```

2.3.4

```
3 + 5*2
```

```
ans = 13
```

2.3.5

```
4*3/2*8
```

```
ans = 48
```

2.3.6

```
3-2/4+6^2
```

```
ans = 38.5000
```

2.3.7

```
2^3^4
```

```
ans = 4096
```

2.3.8

```
2^(3^4)
```

```
ans = 2.4179e+24
```

2.3.9

```
3^5+2
```

```
ans = 245
```

2.3.10

```
3^(5+2)
```

```
ans = 2187
```

Create and test MATLAB® syntax to evaluate the following expressions, then check your answers with a handheld calculator.

2.3.11

```
(5+3)/(9-1)
```

```
ans = 1
```

2.3.12

```
2^3 - 4/(5+3)
```

```
ans = 7.5000
```

2.3.13

```
5^(2+1)/(4-1)
```

```
ans = 41.6667
```

2.3.14

```
(4+1/2)*(5+2/3)
```

```
ans = 25.5000
```

2.3.15

```
(5 + 6*7/3 -2^2)/(2/3*3/(3*6))
```

```
ans = 135
```

```
% or
(5 + 6*7/3 -2^2)/(2/3*3/3/6)
```

```
ans = 135
```

Practice Exercise 2.4

As you perform the following calculations, recall the difference between the * and .* operators, as well as the / and ./ and the ^and .^ operators.

```
clear, clc
%Students should perform these tasks in the command window
```

2.4. 1

Define the matrix *a* = [2.3 5.8 9] as a MATLAB® variable.

```
a=[2.3 5.8 9]
```

```
a =

    2.3000    5.8000    9.0000
```

2.4. 2

Find the sine of **a**.

```
sin(a)
```

```
ans =

    0.7457   -0.4646    0.4121
```

2.4. 3

Add 3 to every element in **a**.

```
a+3
```

```
ans =

    5.3000    8.8000   12.0000
```

2.4. 4

Define the matrix *b* = [5.2 3.14 2] as a MATLAB® variable

```
b=[5.2 3.14 2]
```

```
b =

    5.2000    3.1400    2.0000
```

2.4. 5

Add together each element in matrix **a** and in matrix **b**.

```
a+b
```

```
ans =

    7.5000    8.9400   11.0000
```

2.4.6

Multiply each element in **a** by the corresponding element in **b**.

```
a.*b
```

```
ans =

   11.9600   18.2120   18.0000
```

2.4.7

Square each element in matrix **a**.

```
a.^2
```

```
ans =

    5.2900   33.6400   81.0000
```

2.4.8

Create a matrix named **c** of evenly spaced values from 0 to 10, with an increment of 1.

```
c=0:10
```

```
c =

     0     1     2     3     4     5     6     7     8     9    10
```

2.4.9

Create a matrix named **d** of evenly spaced values from 0 to 10, with an increment of 2.

```
d=0:2:10
```

```
d =

     0     2     4     6     8    10
```

2.4.10

Use the `linspace` function to create a matrix of six evenly spaced values from 10 to 20.

```
linspace(10,20,6)
```

```
ans =

    10    12    14    16    18    20
```

2.4.11

Use the `logspace` function to create a matrix of five logarithmically spaced values between 10 and 100.

```
logspace(1,2,5)
```

```
ans =

   10.0000   17.7828   31.6228   56.2341  100.0000
```

Practice Exercise 3.1

```
clear,clc
```

3.1.1a

```
help cos
```

```
  cos    Cosine of argument in radians.
     cos(X) is the cosine of the elements of X.

     See also acos, cosd.

     Reference page for cos
     Other functions named cos
```

3.1.1b

```
help sqrt
```

```
  sqrt   Square root.
     sqrt(X) is the square root of the elements of X. Complex
     results are produced if X is not positive.

     See also sqrtm, realsqrt, hypot.

     Reference page for sqrt
     Other functions named sqrt
```

3.1.1c

```
help exp
```

```
  exp    Exponential.
     exp(X) is the exponential of the elements of X, e to the X.
     For complex Z=X+i*Y, exp(Z) = exp(X)*(COS(Y)+i*SIN(Y)).

     See also expm1, log, log10, expm, expint.

     Reference page for exp
     Other functions named exp
```

3.1.3a

```
doc cos
```

3.1.3b

```
doc sqrt
```

3.1.3c

```
doc exp
```

Practice Exercise 3.2

Consult Table 3.1 for the appropriate function to solve the following problems

```
clear,clc, format shortg
```

Exercise 3.2.1

Create a vector **x** from –2 to +2 with an increment of 1. Your vector should be

x = [**–2**, **–1**, **0**, **1**, **2**]

a. Find the absolute value of each member of the vector.

b. Find the square root of each member of the vector.

```
x= -2:1:2;
abs(x)
```

```
ans =       2      1      0      1      2
```

```
sqrt(x)
```

```
ans =             0 +      1.4142i             0 +           1i             0 +           0i
1 +            0i       1.4142 +           0i
```

Exercise 3.2.2

Find the square root of both –3 and +3.

a. Use the `sqrt` function.

```
%a
sqrt(-3)
```

```
ans =             0 +      1.7321i
```

```
sqrt(3)
```

```
ans =        1.7321
```

b. Use the `nthroot` function. (You should get an error statement for –3.)

```
%b
% nthroot(-3,2)  Does not work because n must be odd if x is negative
nthroot(3,2)
```

```
ans =        1.7321
```

c. Raise –3 and +3 to the ½ power (Don't forget parentheses around -3).

```
%c
(-3)^(1/2)
```

```
ans =              0 +     1.7321i
```

```
3^(1/2)
```

```
ans =        1.7321
```

Exercise 3.2.3

Create a vector **x** from –9 to 12 with an increment of 3.

a. Find the result of **x** divided by 2.

b. Find the remainder of **x** divided by 2.

```
x=-9:3:12
```

```
x =     -9    -6    -3     0     3     6     9    12
```

```
x/2
```

```
ans =          -4.5            -3          -1.5             0           1.5             3           4.5             6
```

```
rem(x,2)
```

```
ans =     -1     0    -1     0     1     0     1     0
```

Exercise 3.2.4

Using the vector from Exercise 3, find e^x

```
exp(x)
```

```
ans =    0.00012341    0.0024788     0.049787             1        20.086        403.43        8103.1    1.6275e+05
```

Exercise 3.2.5

Using the vector from Exercise 3:

a. Find ln (**x**) (the natural logarithm of **x**).

b. Find log10 (**x**) (the common logarithm of **x**).

Explain your results.

```
%a
log(x)
```

```
ans =        2.1972 +      3.1416i        1.7918 +      3.1416i        1.0986 +      3.1416i          -Inf +           0i        1.0986 +           0i        1.7918 +           0i        2.1972 +           0i        2.4849 +           0i
```

```
%b
log10(x)
```

```
ans =       0.95424 +      1.3644i       0.77815 +      1.3644i       0.47712 +      1.3644i          -Inf +           0i       0.47712 +           0i       0.77815 +           0i       0.95424 +           0i        1.0792 +           0i
```

Exercise 3.2.6

Use the `sign` function to determine which of the elements in vector **x** are positive.

```
sign(x)
```

```
ans =     -1    -1    -1     0     1     1     1     1
```

Exercise 3.2.7

Change the `format` to `rat`, and display the value of the **x** vector divided by 2.

```
format rat
x/2
```

```
ans =       -9/2            -3              -3/2             0               3/2             3               9/2             6
```

(Don't forget to change the format back to `format short` or `format shortg` when you are done with this exercise set.)

```
format short   % returns the format to the default
```

Practice Exercises 3.3

```
clear,clc
```

Exercise 3.3.1

Factor the number 322.

```
factor(322)
```

```
ans =

     2     7    23
```

Exercise 3.3.2

Find the greatest common denominator of 322 and 6.

```
gcd(322,6)
```

```
ans = 2
```

Exercise 3.3.3

Is 322 a prime number?

```
isprime(322)
```

```
ans =    0
```

Exercise 3.3.4

How many primes occur between 0 and 322?

```
length(primes(322))
```

```
ans = 66
```

Exercise 3.3.5

Approximate π as a rational number.

```
rats(pi)
```

```
ans =     355/113
```

Exercise 3.3.6

Find 10! (10 factorial).

```
factorial(10)
```

```
ans = 3628800
```

Exercise 3.3.7

Find the number of possible groups containing 3 people from a group of 20, when order does not matter (20 choose 3).

```
nchoosek(20,3)
```

```
ans = 1140
```

Practice Exercises 3.4

Calculate the following (remember that mathematical notation is not necessarily the same as MATLAB® notation).

Exercise 3.4.1

$\sin(2\theta)$ for $\theta = 3\pi$

```
theta = 3*pi;
sin(2*theta)
```

```
ans = -7.3479e-16
```

Exercise 3.4.2

cos(θ)for $0 \le \theta \le 2\pi$; let θ change in steps of 0.2θ.

```
theta = 0:0.2*pi:2*pi;
cos(theta)
```

```
ans =

    1.0000    0.8090    0.3090   -0.3090   -0.8090   -1.0000   -0.8090   -0.3090
```

Exercise 3.4.3

$\sin^{-1}$(1).

```
asin(1)
```

```
ans = 1.5708
```

Exercise 3.4.4

$\cos^{-1}(x)$ for $-1 \le x \le 1$; let x change in steps of 0.2

```
x = -1:0.2:1;
acos(x)
```

```
ans =

    3.1416    2.4981    2.2143    1.9823    1.7722    1.5708    1.3694    1.1593
```

Exercise 3.4.5

Find the cosine of 45°.

a. Convert the angle from degrees to radians, and then use the **cos** function.

```
cos(45*pi/180)
```

```
ans = 0.7071
```

b. Use the **cosd** function.

```
cosd(45)
```

```
ans = 0.7071
```

Exercise 3.4.6

Find the angle whose sine is 0.5. Is your answer in degrees or radians?

```
asin(0.5)
```

```
ans = 0.5236
```

Exercise 3.4.7

Find the cosecant of 60. You may have to use the help function to find the appropriate syntax.

```
cscd(60)
```

```
ans = 1.1547
```

Practice Exercises 3.5

The correct answers can be found on the Pearson website. Consider the following matrix:

```
x = [4 90 85 75;
     2 55 65 75;
     3 78 82 79;
     1 84 92 93]
```

```
x =

     4    90    85    75
     2    55    65    75
     3    78    82    79
     1    84    92    93
```

Exercise 3.5.1

What is the maximum value in each column?

```
max(x)
```

```
ans =

     4    90    92    93
```

Exercise 3.5.2

In which row does that maximum occur?

```
[maximum, row] = max(x)
```

```
maximum =

     4    90    92    93
row =

     1     1     4     4
```

Exercise 3.5.3

What is the maximum value in each row? (You'll have to transpose the matrix to answer this question or specify the dimension number.)

```
max(x')
```

```
ans =

    90    75    82    93
```

or

```
max(x,[],2)
```

```
ans =

    90
    75
    82
    93
```

Exercise 3.5.4

In which column does the maximum occur?

```
[maximum, column] = max(x')
```

```
maximum =

    90    75    82    93
column =

     2     4     3     4
```

Exercise 3.5.5

What is the maximum value in the entire table?

```
max(max(x))
```

```
ans = 93
```

```
% or
max(x(:))
```

```
ans = 93
```

Practice Exercises 3.6

Consider the following matrix:

```
clear,clc

x = [4 90 85 75;
     2 55 65 75;
     3 78 82 79;
     1 84 92 93];
```

Exercise 3.6.1

What is the mean value in each column?

```
mean(x)
```

```
ans =

    2.5000   76.7500   81.0000   80.5000
```

Exercise 3.6.2

What is the median for each column?

```
median(x)
```

```
ans =

    2.5000   81.0000   83.5000   77.0000
```

Exercise 3.6.3

What is the mean value in each row?

```
mean(x')
```

```
ans =

   63.5000   49.2500   60.5000   67.5000
```

or

```
mean(x,2)
```

```
ans =

   63.5000
   49.2500
   60.5000
   67.5000
```

Exercise 3.6.4

What is the median for each row?

```
median(x')
```

```
ans =

   80.0000   60.0000   78.5000   88.0000
```

or

```
median(x,2)
```

```
ans =

   80.0000
   60.0000
   78.5000
   88.0000
```

Exercise 3.6.5

What is returned when you request the mode?

```
mode(x)
```

```
ans =

     1    55    65    75
```

Exercise 3.6.6

What is the mean for the entire matrix?

```
mean(mean(x))
```

```
ans = 60.1875
```

```
% or
mean(x(:))
```

```
ans = 60.1875
```

Practice Exercises 3.7

Consider the following matrix:

```
clear,clc

x = [4 90 85 75;
     2 55 65 75;
     3 78 82 79;
     1 84 92 93]
```

```
x =

     4    90    85    75
     2    55    65    75
     3    78    82    79
     1    84    92    93
```

Exercise 3.7.1

Use the `size` function to determine the number of rows and columns in this matrix.

```
size(x)
```

```
ans =

     4     4
```

Exercise 3.7.2

Use the `sort` function to sort each column in ascending order.

```
sort(x)
```

```
ans =

     1    55    65    75
     2    78    82    75
     3    84    85    79
     4    90    92    93
```

Exercise 3.7.3

Use the `sort` function to sort each column in descending order.

```
sort(x,'descend')
```

```
ans =

     4    90    92    93
     3    84    85    79
     2    78    82    75
     1    55    65    75
```

Exercise 3.7.4

Use the `sortrows` function to sort the matrix so that the first column is in ascending order, but each row still retains its original data. Your matrix should look like this:

$$\begin{bmatrix} 1 & 84 & 92 & 93 \\ 2 & 55 & 65 & 75 \\ 3 & 78 & 82 & 79 \\ 4 & 90 & 85 & 75 \end{bmatrix}$$

```
sortrows(x)
```

```
ans =

     1    84    92    93
     2    55    65    75
     3    78    82    79
     4    90    85    75
```

Exercise 3.7.5

Use the `sortrows` function to sort the matrix from Exercise 4 in descending order, based on the third column.

```
sortrows(x,-3)
```

```
ans =

     1    84    92    93
     4    90    85    75
     3    78    82    79
     2    55    65    75
```

Practice Exercises 3.8

Consider the following matrix:

```
clear,clc

x = [4 90 85 75;
     2 55 65 75;
     3 78 82 79;
     1 84 92 93]
```

```
x =

     4    90    85    75
     2    55    65    75
     3    78    82    79
     1    84    92    93
```

Exercise 3.8.1

Find the standard deviation for each column.

```
std(x)
```

```
ans =

    1.2910   15.3052   11.4601    8.5440
```

Exercise 3.8.2

Find the variance for each column.

```
var(x)
```

```
ans =

    1.6667  234.2500  131.3333   73.0000
```

Exercise 3.8.3

Calculate the square root of the variance you found for each column.

```
sqrt(var(x))
```

```
ans =

    1.2910   15.3052   11.4601    8.5440
```

Practice Exercises 3.9

```
clear,clc
```

Exercise 3.9.1

Create a 3 × 3 matrix of evenly distributed random numbers.

```
rand(3)
```

```
ans =

    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
```

Exercise 3.9.2

Create a 3 × 3 matrix of normally distributed random numbers.

```
randn(3)
```

```
ans =

    2.7694    0.7254   -0.2050
   -1.3499   -0.0631   -0.1241
    3.0349    0.7147    1.4897
```

Exercise 3.9.3

Create a 100 × 5 matrix of evenly distributed random numbers. Be sure to suppress the output.

```
x = rand(100,5);
```

Exercise 3.9.4

Find the maximum, the standard deviation, the variance, and the mean for each column in the matrix that you created in Exercise 3.

```
max(x)
```

```
ans =

    0.9961    0.9619    0.9880    0.9991    0.9937
```

```
std(x)
```

```
ans =

    0.2854    0.2838    0.2843    0.2821    0.2579
```

```
var(x)
```

```
ans =

    0.0814    0.0805    0.0808    0.0796    0.0665
```

```
mean(x)
```

```
ans =

    0.5082    0.4658    0.5080    0.4754    0.4999
```

Exercise 3.9.5

Create a 100 × 5 matrix of normally distributed random numbers. Be sure to suppress the output.

```
x = randn(100,5);
```

Exercise 3.9.6

Find the maximum, the standard deviation, the variance, and the mean for each column in the matrix you created in Exercise 5.

```
max(x)
```

```
ans =

    2.2272    2.5088    2.7891    2.6052    3.5267
```

```
std(x)
```

```
ans =

    0.9283    0.9743    1.0061    0.9682    1.0162
```

```
var(x)
```

```
ans =

    0.8618    0.9493    1.0123    0.9373    1.0326
```

```
mean(x)
```

```
ans =

    0.0815   -0.0908   -0.1185   -0.2511    0.1266
```

Practice Exercises 3.10

```
clear,clc
```

Exercise 3.10.1

Create the following complex numbers:

a. $A = 1 + i$

b. $B = 2 - 3i$

c. $C = 8 + 2i$

```
A = 1+i
```

```
A =    1.0000 + 1.0000i
```

```
B = 2-3*i
```

```
B =    2.0000 - 3.0000i
```

```
C = 8+2*i
```

```
C =    8.0000 + 2.0000i
```

Exercise 3.10.2

Create a vector `D` of complex numbers whose real components are 2, 4, and 6 and whose imaginary components are −3, 8, and −16.

```
realD = [2 4 6]
```

```
realD =

     2     4     6
```

```
imagD = [-3 8 -16]
```

```
imagD =

    -3     8   -16
```

```
D = complex(realD, imagD)
```

```
D =    2.0000 - 3.0000i   4.0000 + 8.0000i   6.0000 -16.0000i
```

Exercise 3.10.3

Find the magnitude (absolute value) of each of the vectors you created in Exercises 1 and 2.

```
abs(A)
```

```
ans = 1.4142
```

```
abs(B)
```

```
ans = 3.6056
```

```
abs(C)
```

```
ans = 8.2462
```

```
abs(D)
```

```
ans =

    3.6056    8.9443   17.0880
```

Exercise 3.10.4

Find the angle from the horizontal of each of the complex numbers you created in Exercises 1 and 2.

```
angle(A)
```

```
ans = 0.7854
```

```
angle(B)
```

```
ans = -0.9828
```

```
angle(C)
```

```
ans = 0.2450
```

```
angle(D)
```

```
ans =

   -0.9828    1.1071   -1.2120
```

Exercise 3.10.5

Find the complex conjugate of vector `D`.

```
conj(D)
```

```
ans =    2.0000 + 3.0000i   4.0000 - 8.0000i   6.0000 +16.0000i
```

Exercise 3.10.6

Use the transpose operator to find the complex conjugate of vector D.

```
D'
```

```
ans =
   2.0000 + 3.0000i
   4.0000 - 8.0000i
   6.0000 +16.0000i
```

Exercise 3.10.7

Multiply `A` by its complex conjugate, and then take the square root of your answer. How does this value compare against the magnitude (absolute value) of `A`?

```
sqrt(A.*A')
```

```
ans = 1.4142
```

```
abs(A)
```

```
ans = 1.4142
```

Practice Exercises 3.11

```
clear,clc, format shortg
```

Exercise 3.11.1

Use the `clock` function to add the time and date to your work sheet.

```
clock
```

```
ans =          2016             7            21            16             3        48.207
```

Exercise 3.11.2

Use the `date` function to add the date to your work sheet.

```
date
```

```
ans = 21-Jul-2016
```

Exercise 3.11.3

Convert the following calculations to MATLAB® code and explain your results:

a. 322! (Remember that, to a mathematician, the symbol ! means factorial.)

```
factorial(322)
```

```
ans =    Inf
```

b. $5 * 10^{500}$

```
5e500  % or
```

```
ans =    Inf
```

```
5*10^500
```

```
ans =    Inf
```

c. $1/5 * 10^{500}$

```
1/5 * 10^500
```

```
ans =    Inf
```

d. 0/0

```
0/0
```

```
ans =    NaN
```

Practice Exercises 4.1

Create MATLAB ® variables to represent the following matrices, and use them in the exercises that follow.

```
a=[12 17 3 6]
```

```
a =      12      17       3       6
```

```
b=[5 8 3; 1 2 3; 2 4 6]
```

```
b =
     5     8     3
     1     2     3
     2     4     6
```

```
c=[22;17;4]
```

```
c =
    22
    17
     4
```

Exercise 4.1.1

Assign to the variable `x1` the value in the second column of matrix `a`. This is sometimes represented in mathematics textbooks as element $a_{1,2}$ and could be expressed as `x1` $= a_{1,2}$

```
x1=a(1,2)
```

```
x1 =      17
```

Exercise 4.1.2

Assign to the variable `x2` the third column of matrix `b`.

```
x2=b(:,3)
```

```
x2 =
     3
     3
     6
```

Exercise 4.1.3

Assign to the variable `x3` the third row of matrix `b`.

```
x3=b(3,:)
```

```
x3 =        2       4       6
```

Exercise 4.1.4

Assign to the variable `x4` the values in matrix `b` along the diagonal (i.e., elements $b_{1,1}$, $b_{2,2}$, and $b_{3,3}$).

```
x4=[b(1,1), b(2,2), b(3,3)]
```

```
x4 =        5       2       6
```

Exercise 4.1.5

Assign to the variable `x5` the first three values in matrix `a` as the first row and all the values in matrix `b` as the second through the fourth row.

```
x5 = [a(1:3);b]
```

```
x5 =
    12    17     3
     5     8     3
     1     2     3
     2     4     6
```

Exercise 4.1.6

Assign to the variable `x6` the values in matrix `c` as the first column, the values in matrix `b` as columns 2, 3, and 4, and the values in matrix `a` as the last row.

```
x6 = [c,b;a]
```

```
x6 =
    22     5     8     3
    17     1     2     3
     4     2     4     6
    12    17     3     6
```

Exercise 4.1.7

Assign to the variable `x7` the value of element 8 in matrix `b`, using the single-index-number identification scheme.

```
x7=b(8)
```

```
x7 =        3
```

Exercise 4.1.8

Convert matrix `b` to a column vector named `x8`.

```
x8=b(:)
```

```
x8 =
     5
     1
     2
     8
     2
     4
     3
     3
     6
```

Practice Exercises 4.2

```
clear,clc
```

Exercise 4.2.1

The area of a rectangle (Figure 4.2) is length times width (area = length ´ width). Find the areas of rectangles with lengths of 1, 3, and 5 cm and with widths of 2, 4, 6, and 8 cm. (You should have 12 answers.)

```
length = [1, 3, 5];
width = [2,4,6,8];
[L,W]=meshgrid(length,width);
area = L.*W
```

```
area =
     2     6    10
     4    12    20
     6    18    30
     8    24    40
```

Exercise 4.2.2

The volume of a circular cylinder is, volume = $\pi r2h$. Find the volume of cylindrical containers with radii from 0 to 12 m and heights from 10 to 20 m. Increment the radius dimension by 3 m and the height by 2 m as you span the two ranges.

```
radius = 0:3:12;
height = 10:2:20;
[R,H] = meshgrid(radius,height);
volume = pi*R.^2.*H
```

```
volume =
            0       282.74         1131       2544.7       4523.9
            0       339.29       1357.2       3053.6       5428.7
            0       395.84       1583.4       3562.6       6333.5
            0       452.39       1809.6       4071.5       7238.2
            0       508.94       2035.8       4580.4         8143
            0       565.49       2261.9       5089.4       9047.8
```

Practice Exercises 4.3

```
clear,clc
```

Exercise 4.3.1

Create a 3 x 3 matrix of zeros.

```
zeros(3)
```

```
ans =
     0     0     0
     0     0     0
     0     0     0
```

Exercise 4.3.2

Create a 3 x 4 matrix of zeros.

```
zeros(3,4)
```

```
ans =
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

Exercise 4.3.3

Create a 3 x 3 matrix of ones.

```
ones(3)
```

```
ans =
     1     1     1
     1     1     1
     1     1     1
```

Exercise 4.3.4

Create a 5 x 3 matrix of ones.

```
ones(5,3)
```

```
ans =
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

Exercise 4.3.5

Create a 4 x 6 matrix in which all the elements have a value of pi.

```
ones(4,6)*pi  % or
```

```
ans =
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
```

```
zeros(4,6) + pi
```

```
ans =
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
      3.1416        3.1416        3.1416        3.1416        3.1416        3.1416
```

Exercise 4.3.6

Use the `diag` function to create a matrix whose diagonal has values of 1, 2, 3.

```
x = [1,2,3];
diag(x)
```

```
ans =
     1     0     0
     0     2     0
     0     0     3
```

Exercise 4.3.7

Create a 10 x 10 magic matrix.

```
x = magic(10)
```

```
x =
    92    99     1     8    15    67    74    51    58    40
    98    80     7    14    16    73    55    57    64    41
     4    81    88    20    22    54    56    63    70    47
    85    87    19    21     3    60    62    69    71    28
    86    93    25     2     9    61    68    75    52    34
    17    24    76    83    90    42    49    26    33    65
    23     5    82    89    91    48    30    32    39    66
    79     6    13    95    97    29    31    38    45    72
    10    12    94    96    78    35    37    44    46    53
    11    18   100    77    84    36    43    50    27    59
```

a. Extract the diagonal from this matrix.

```
diag(x)
```

```
ans =
```

```
    92
    80
    88
    21
     9
    42
    30
    38
    46
    59
```

b. Extract the diagonal that runs from lower left to upper right from this matrix.

```
diag(fliplr(x))
```

```
ans =
    40
    64
    63
    62
    61
    90
    89
    13
    12
    11
```

c. Confirm that the sums of the rows, columns, and diagonals are all the same.

```
sum(x)
```

```
ans =    505   505   505   505   505   505   505   505   505   505
```

```
sum(x')
```

```
ans =    505   505   505   505   505   505   505   505   505   505
```

```
sum(diag(x))
```

```
ans =    505
```

```
sum(diag(fliplr(x)))
```

```
ans =    505
```

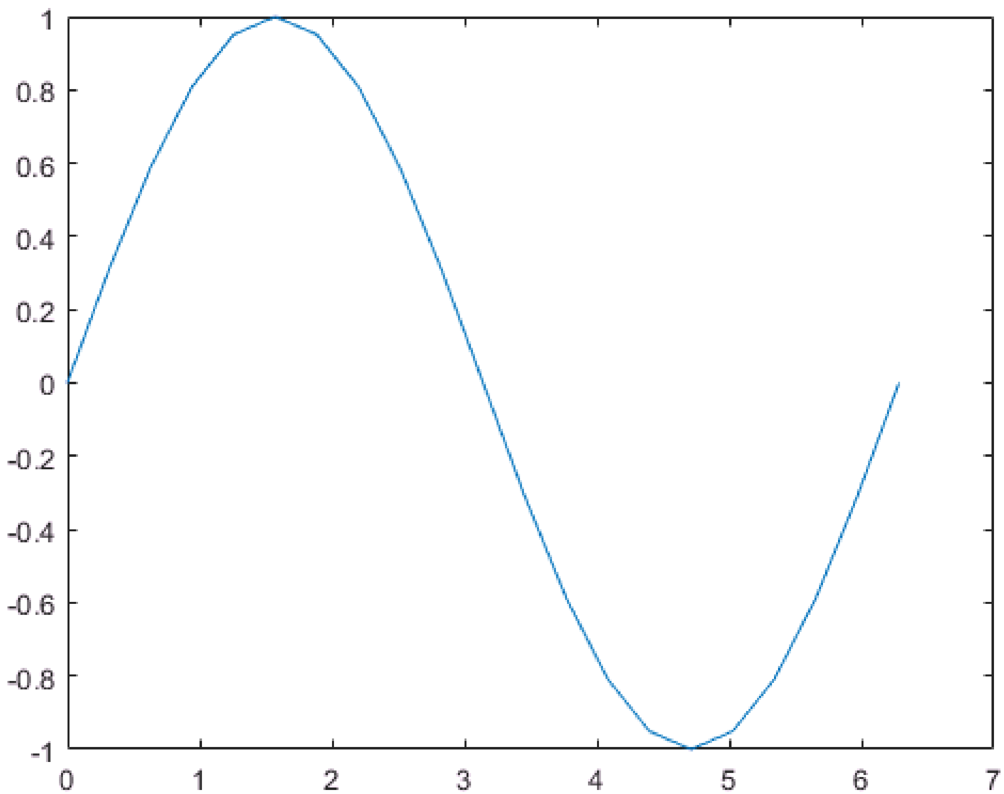
Practice Exercises 5.1

```
clear,clc
```

Exercise 5.1.1

Plot *x* versus *y* for *y* = sin(*x*). Let *x* vary from 0 to 2π in increments of 0.1π.

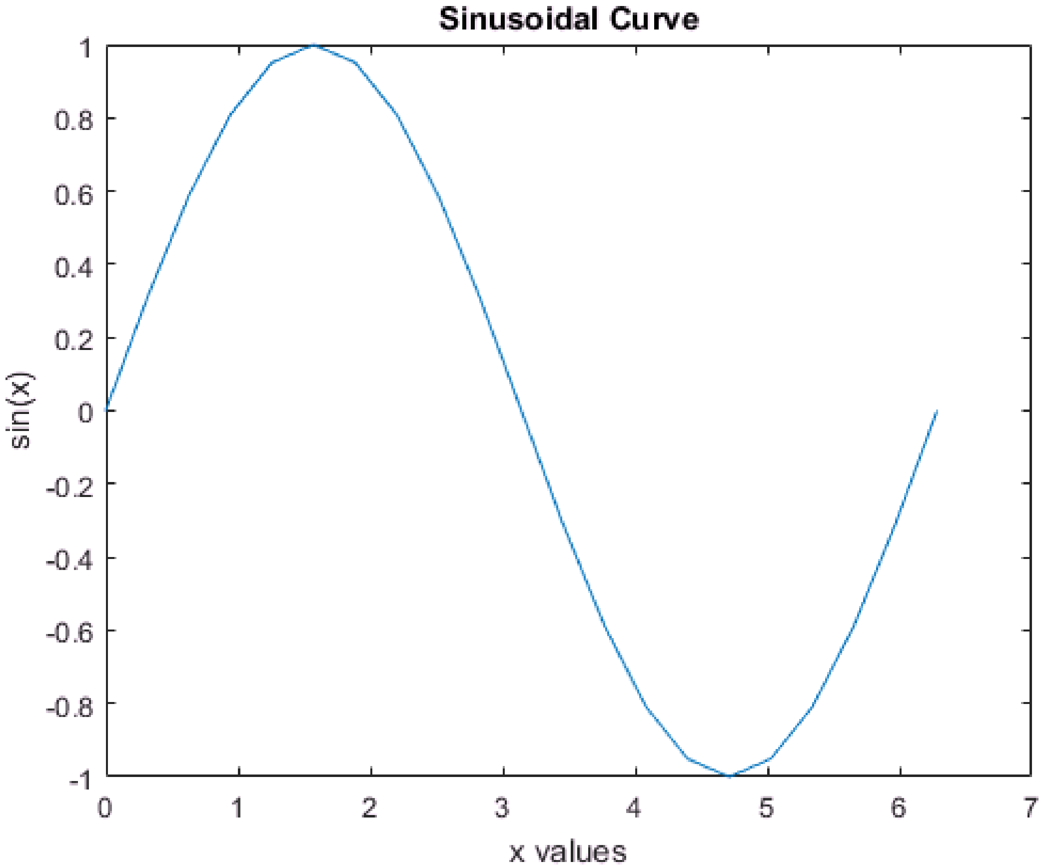
```
x=0:0.1*pi:2*pi;
y=sin(x);
plot(x,y)
```



Exercise 5.1.2

Add a title and labels to your plot.

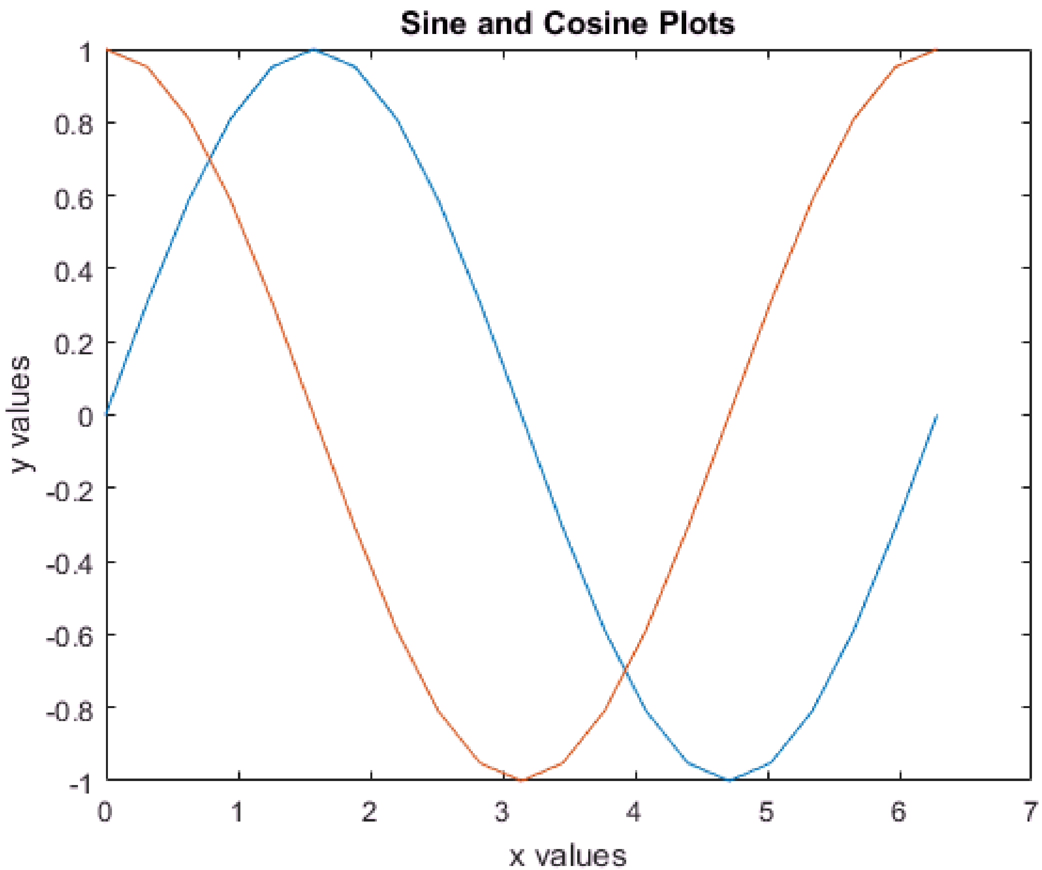
```
title('Sinusoidal Curve')
xlabel('x values')
ylabel('sin(x)')
```



Exercise 5.1.3

Plot x versus y_1 and y_2 for y_1 = sin(x) and y_2 = cos(x). Let x vary from 0 to 2π in increments of 0.1π. Add a title and labels to your plot.

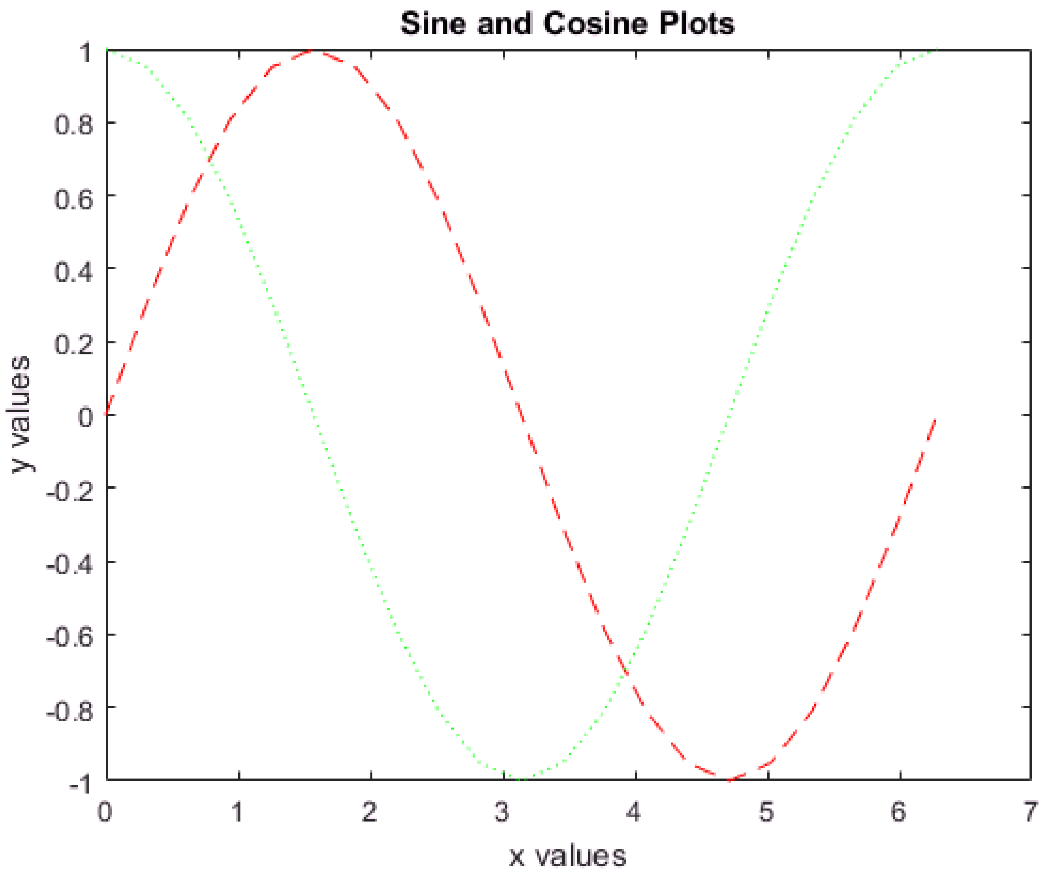
```
figure(2)
y1=sin(x);
y2=cos(x);
plot(x,y1,x,y2)
title('Sine and Cosine Plots')
xlabel('x values')
ylabel('y values')
```



Exercise 5.1.4

Re-create the plot from Exercise 3, but make the sin(*x*) line dashed and red. Make the cos(*x*) line green and dotted.

```
figure(3)
plot(x,y1,'-- r',x,y2,': g')
title('Sine and Cosine Plots')
xlabel('x values')
ylabel('y values')
```



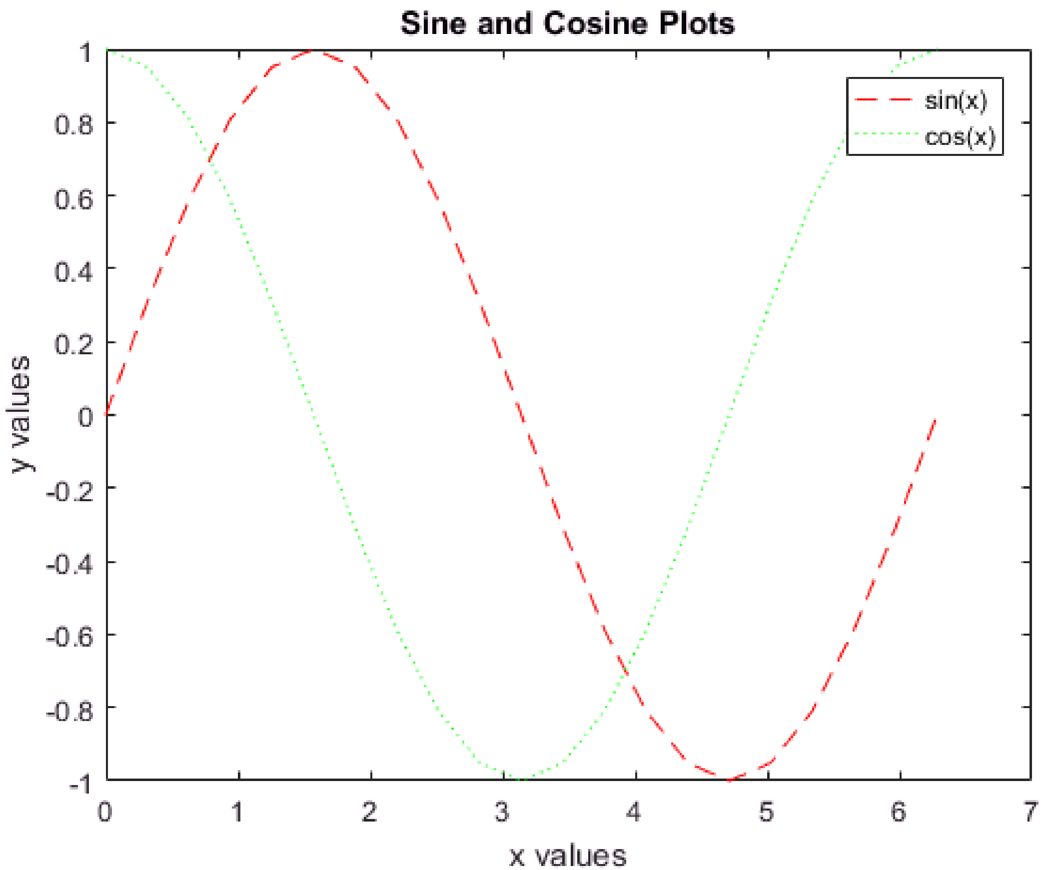

```

```

Exercise 5.1.5

Add a legend to the graph in Exercise 4.

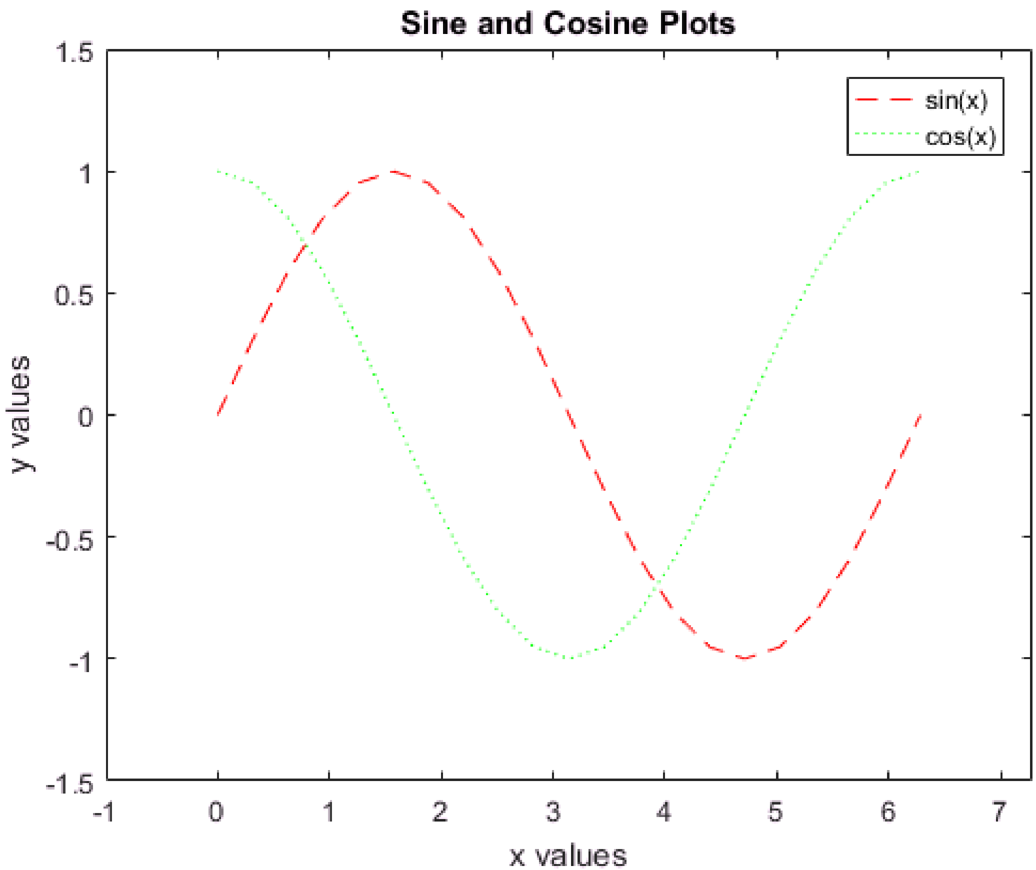
```
legend('sin(x)','cos(x)')
```



Exercise 5.1.6

Adjust the axes so that the x-axis goes from –1 to $2\pi + 1$ and the *y*-axis from –1.5 to + 1.5.

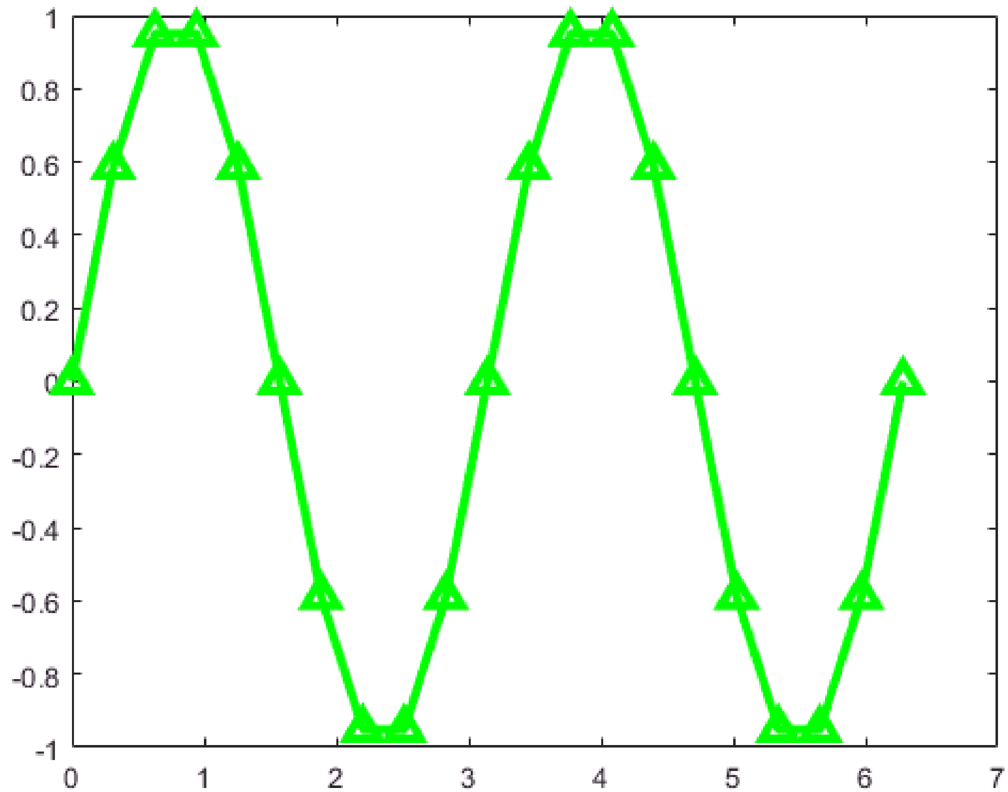
```
axis([-1,2*pi+1,-1.5,1.5])
```



Exercise 5.1.7

Plot *x* versus *y* for *y* = sin(2**x*), with a line weight of 3, a line color of green and with markers as triangles with a weight of 10.

```
figure(4)
y = sin(2*x);
plot(x,y,'LineWidth',3,'Color','g','Marker','^','MarkerSize',10)
```



1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7

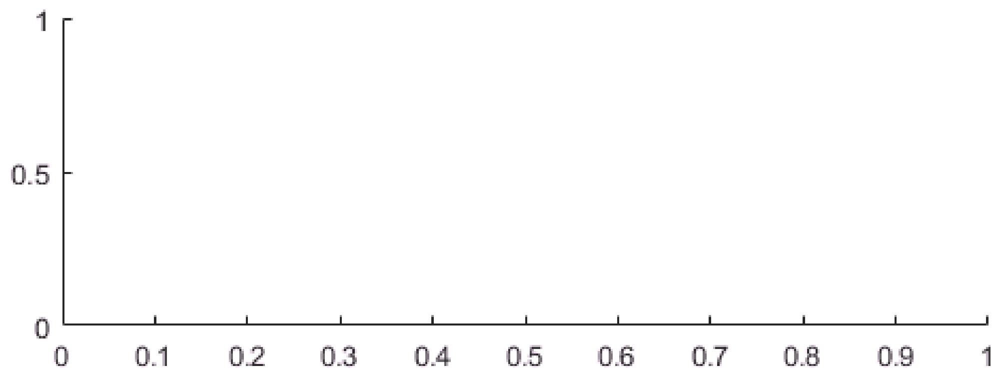
Practice Exercise 5.2

```
clear,clc, clf
```

Exercise 5.2.1

Subdivide a figure window into two rows and one column.

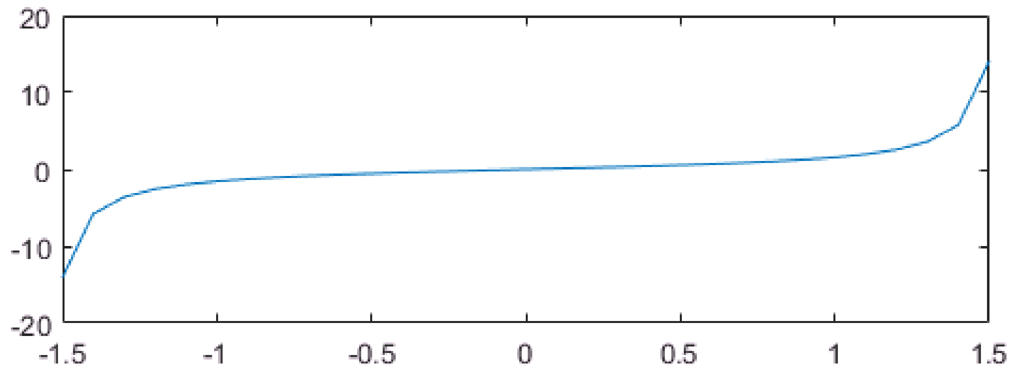
```
figure(1)
subplot(2,1,1)
```



Exercise 5.2.2

In the top window, plot $y = \tan(x)$ for $-1.5 \le x \le 1.5$. Use an increment of 0.1.

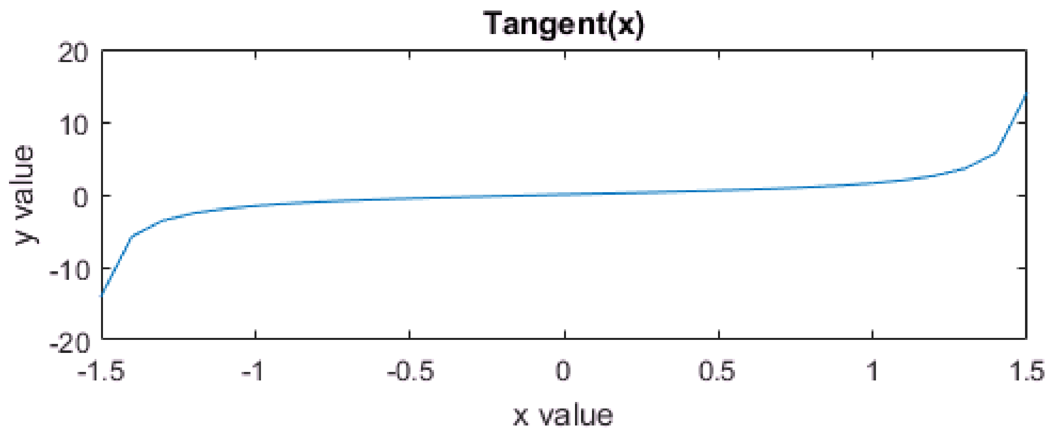
```
x=-1.5:0.1:1.5;
y=tan(x);
plot(x,y)
```



Exercise 5.2.3

Add a title and axis labels to your graph.

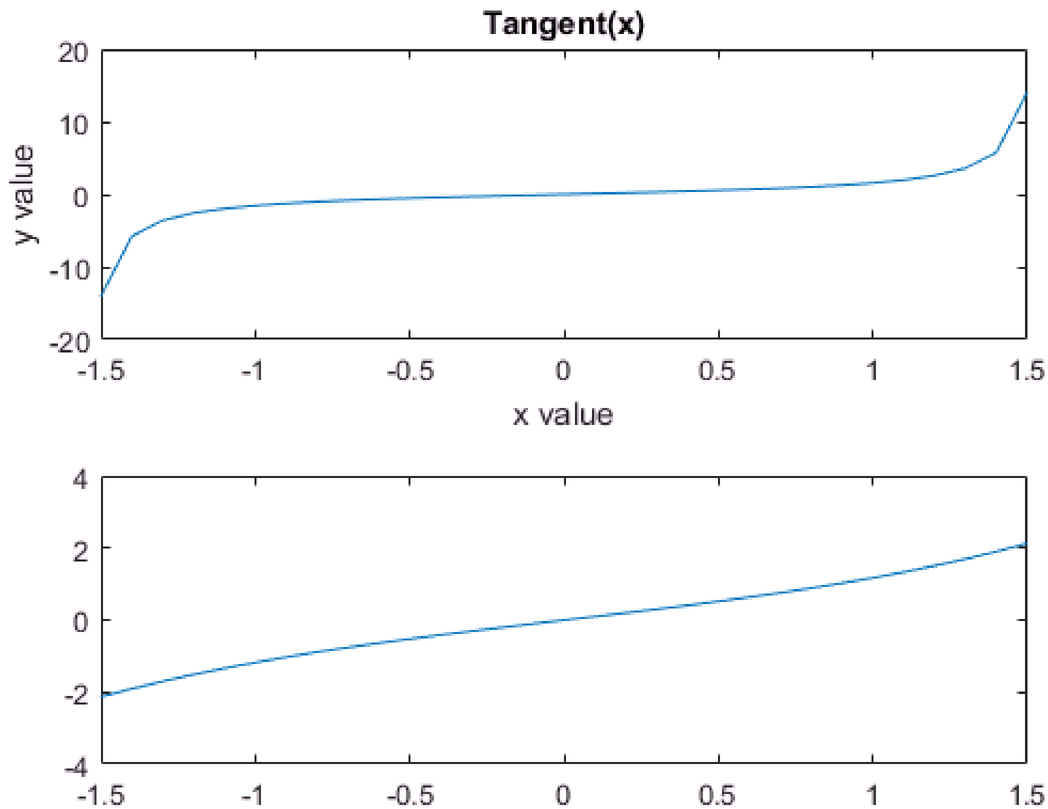
```
title('Tangent(x)')
xlabel('x value')
ylabel('y value')
```



Exercise 5.2.4

In the bottom window, plot *y* = sinh (*x*) for the same range.

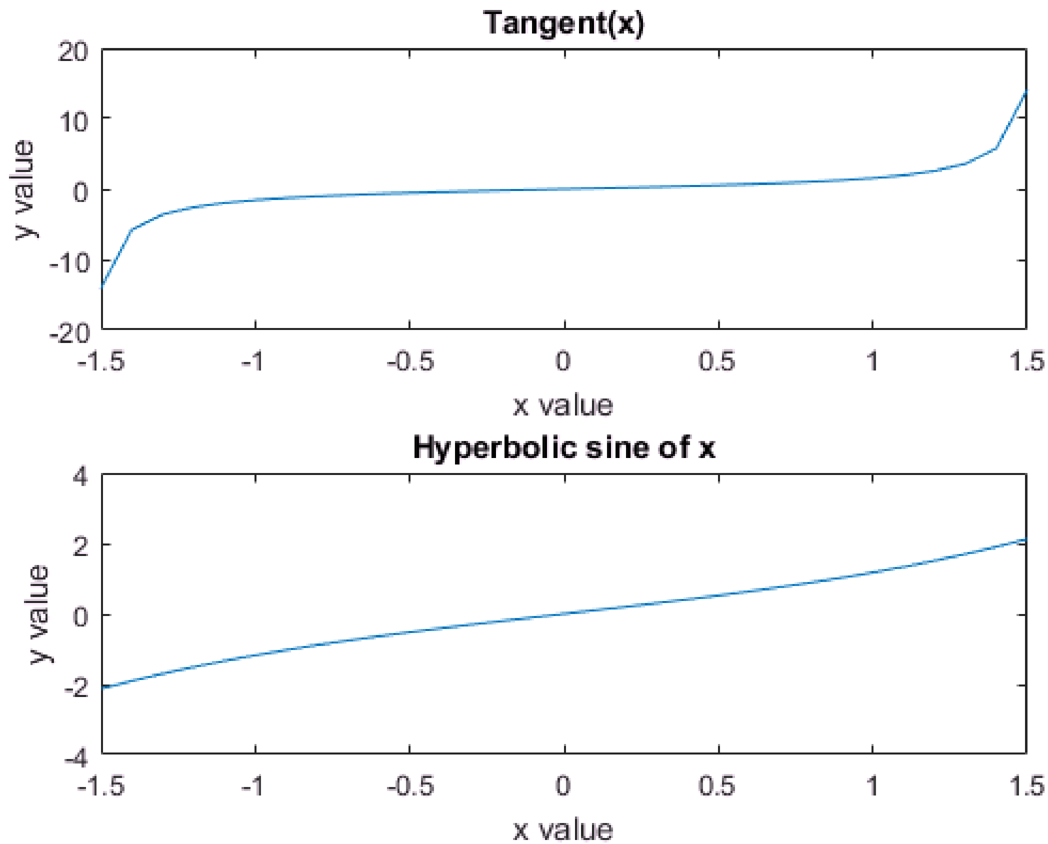
```
subplot(2,1,2)
y=sinh(x);
plot(x,y)
```



Exercise 5.2.5

Add a title and labels to your graph.

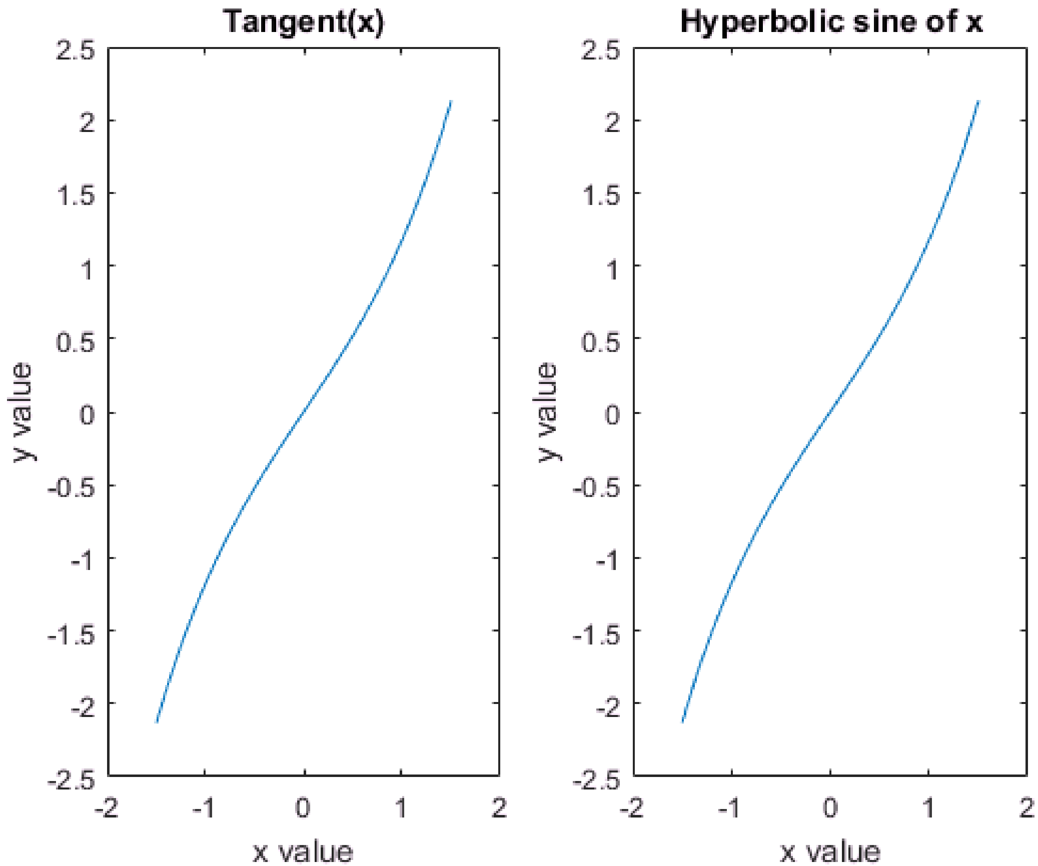
```
title('Hyperbolic sine of x')
xlabel('x value')
ylabel('y value')
```



Exercise 5.2.6

Try the preceding exercises again, but divide the figure window vertically instead of horizontally.

```
figure(2)
subplot(1,2,1)
plot(x,y)
title('Tangent(x)')
xlabel('x value')
ylabel('y value')
subplot(1,2,2)
y=sinh(x);
plot(x,y)
title('Hyperbolic sine of x')
xlabel('x value')
ylabel('y value')
```


Tangent(x)
Hyperbolic sine of x
y value
x value
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2
-1
0
1
2

Practice Exercises 5.3

Use the polarplot function, not the older polar function.

```
clear,clc,close all
```

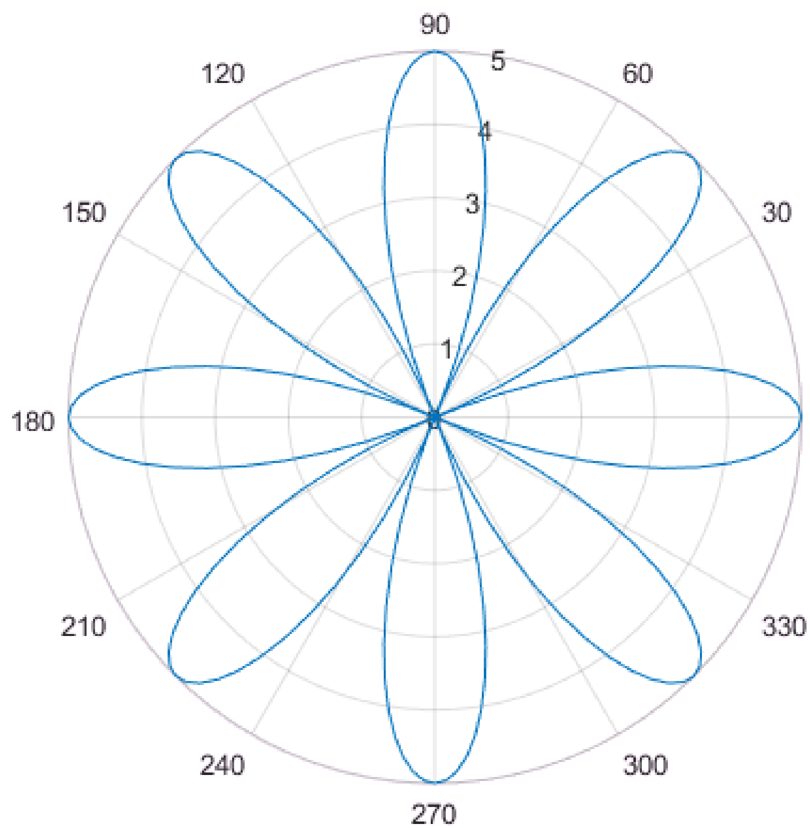
Exercise 5.3.1

Define an array called theta, from 0 to 2π, in steps of 0.01π.

Define an array of distances `r = 5*cos(4*theta)`.

Make a polar plot of `theta` versus `r`.

```
figure(1)
theta = 0:0.01*pi:2*pi;
r = 5*cos(4*theta);
polarplot(theta,r)
```

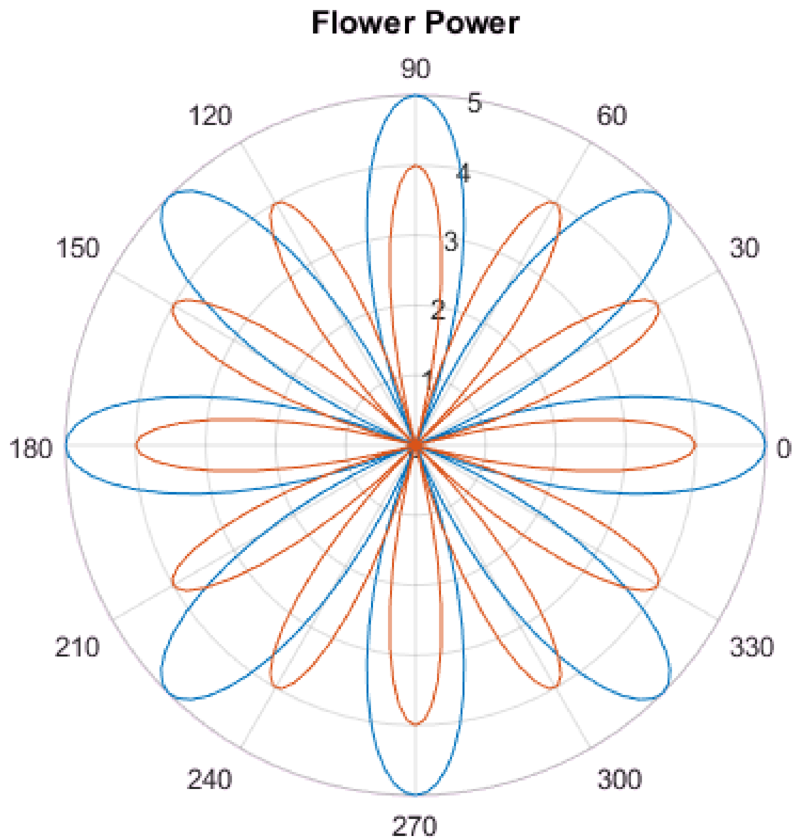


Exercise 5.3.2

Use the hold on command to freeze the graph. Assign `r = 4*cos(6*theta)` and plot. Add a title.

```
hold on
r=4*cos(6*theta);
polarplot(theta,r)
```

```
title('Flower Power')
```



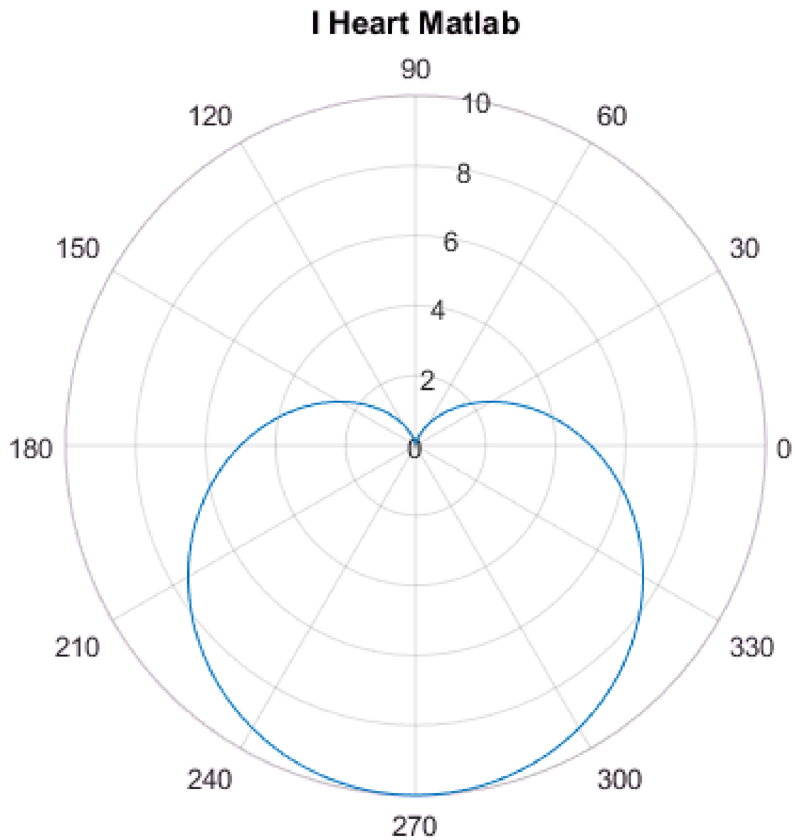
Exercise 5.3.3

Create a new figure.

Use the `theta` array from the preceding exercises.

Assign `r = 5 – 5*sin(theta)` and create a new polar plot.

```
figure(2)
r=5-5*sin(theta);
polarplot(theta,r)
title('I Heart Matlab')
```



Exercise 5.3.4

Create a new figure.

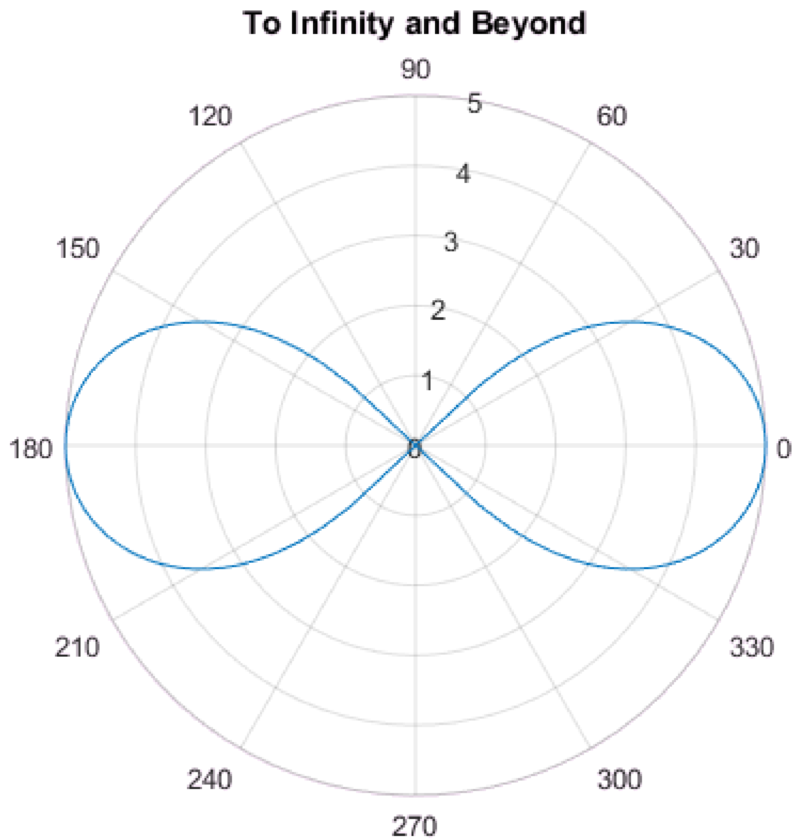
Use the `theta` array from the preceding exercises.

Assign `r = sqrt(5^2*cos(2*theta))` and create a new polar plot.

```
figure(3)
r = sqrt(5^2*cos(2*theta));
polarplot(theta,r)
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

```
title('To Infinity and Beyond')
```



Exercise 5.3.5

Create a new figure.

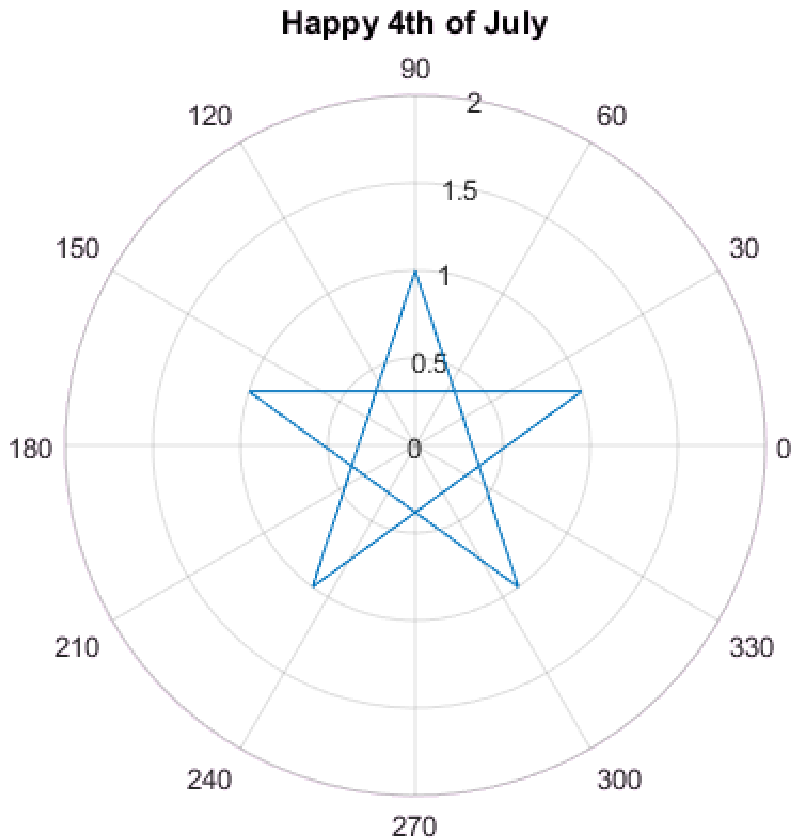
Define a theta array such that

`theta = pi/2:4/5*pi:4.8*pi;`

Create a six-member array of ones called `r`.

Create a new polar plot of `theta` versus `r`.

```
figure(4)
theta = pi/2:4/5*pi:4.8*pi;
r=ones(1,6);
polarplot(theta,r)
title('Happy 4th of July')
```


Happy 4th of July
90
60
30
0
330
300
270
240
210
180
150
120
0
0.5
1
1.5
2

Practice Exercises 5.4

Create appropriate `x` and `y`arrays to use in plotting each of the expressions that follow. Use the `subplot` command to divide your figures into four sections, and create each of these four graphs for each expression:

· Rectangular

· Semilogx

· Semilogy

· Loglog

```
clear, clc
```

Exercise 5.4.1

$y = 5x + 3$

```
figure(1)
x=-1:0.1:1;
y=5*x+3;
subplot(2,2,1)
plot(x,y)
title('Rectangular Coordinates'), ylabel('y-axis'),grid on
subplot(2,2,2)
semilogx(x,y)
```

```
Warning: Negative data ignored
```

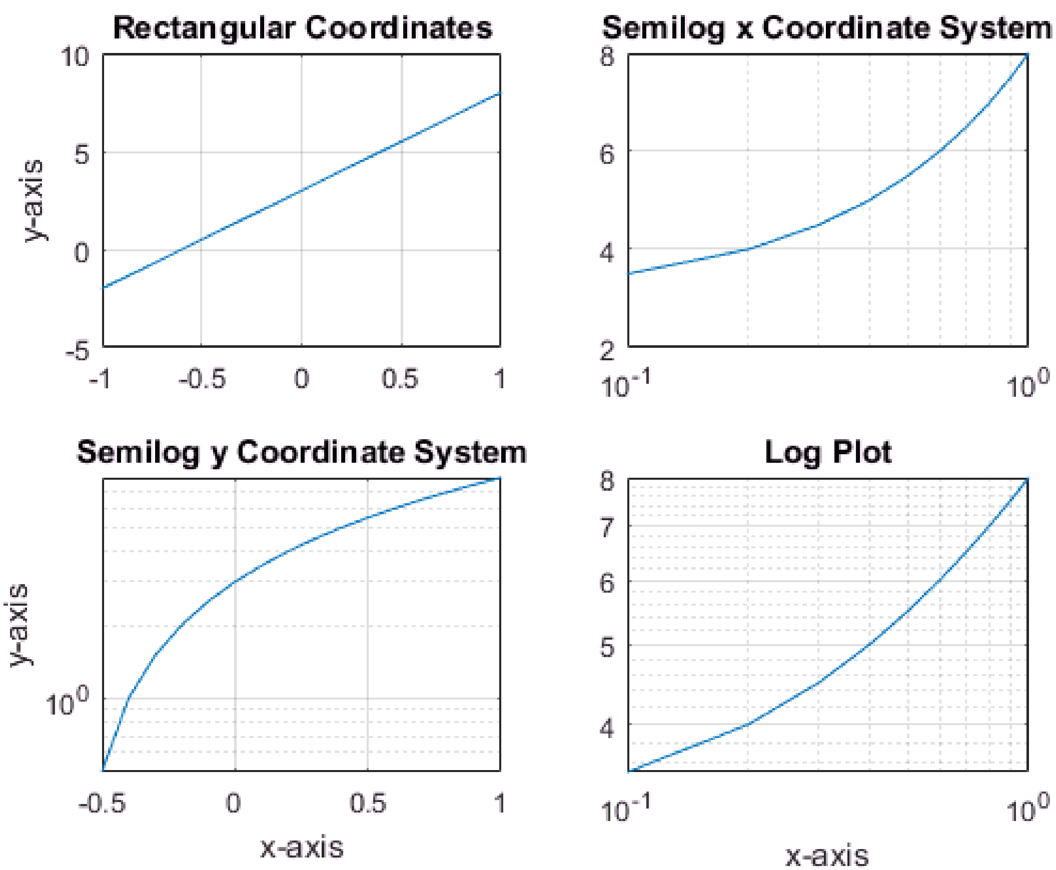
```
title('Semilog x Coordinate System'),grid on
subplot(2,2,3)
semilogy(x,y)
```

```
Warning: Negative data ignored
```

```
title('Semilog y Coordinate System')
ylabel('y-axis'), xlabel('x-axis'), grid on
subplot(2,2,4)
loglog(x,y)
```

```
Warning: Negative data ignored
```

```
title('Log Plot'), xlabel('x-axis'), grid on
```



Exercise 5.4.2

$y = 3x^2$

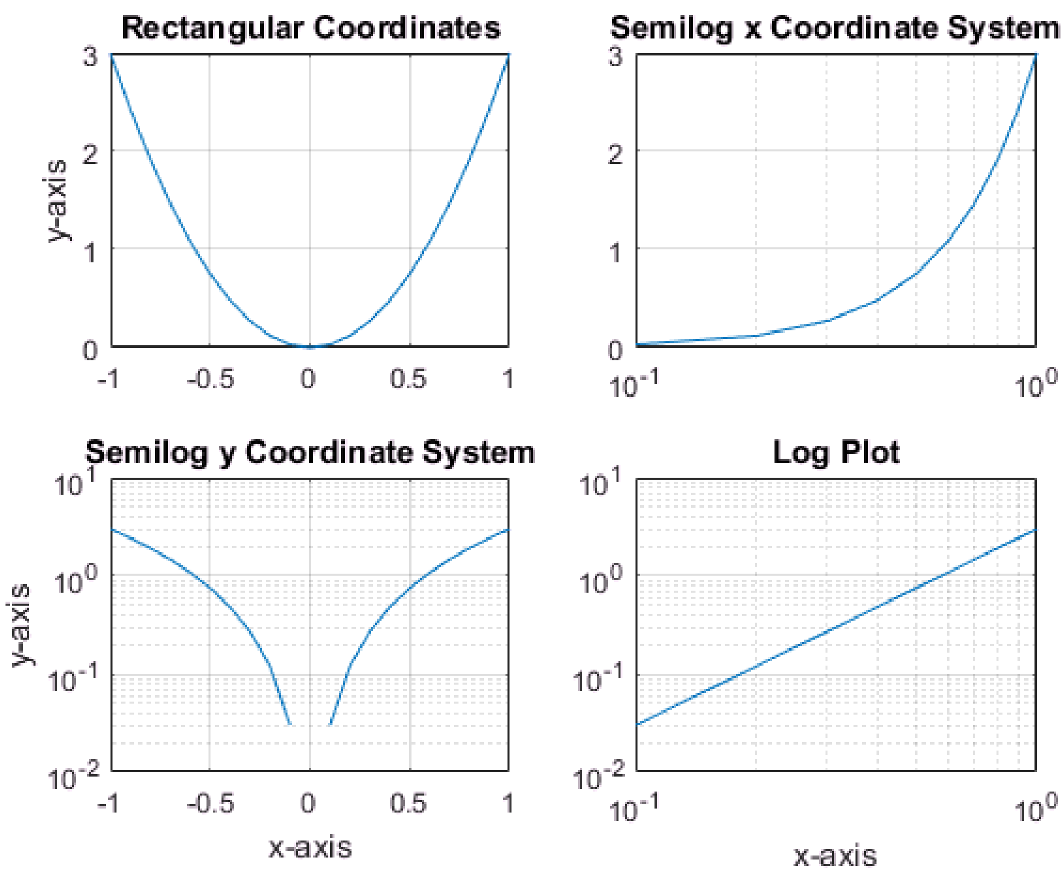
```
figure(2)
x=-1:0.1:1;
y=3*x.^2;
subplot(2,2,1)
plot(x,y)
title('Rectangular Coordinates'), ylabel('y-axis')
grid on
subplot(2,2,2)
semilogx(x,y)
```

```
Warning: Negative data ignored
```

```
title('Semilog x Coordinate System'), grid on
subplot(2,2,3)
semilogy(x,y)
title('Semilog y Coordinate System')
ylabel('y-axis'), xlabel('x-axis'), grid on
subplot(2,2,4)
loglog(x,y)
```

```
Warning: Negative data ignored
```

```
title('Log Plot'), xlabel('x-axis'), grid on
```



Exercise 5.4.3

$y = 12e^{(x+2)}$

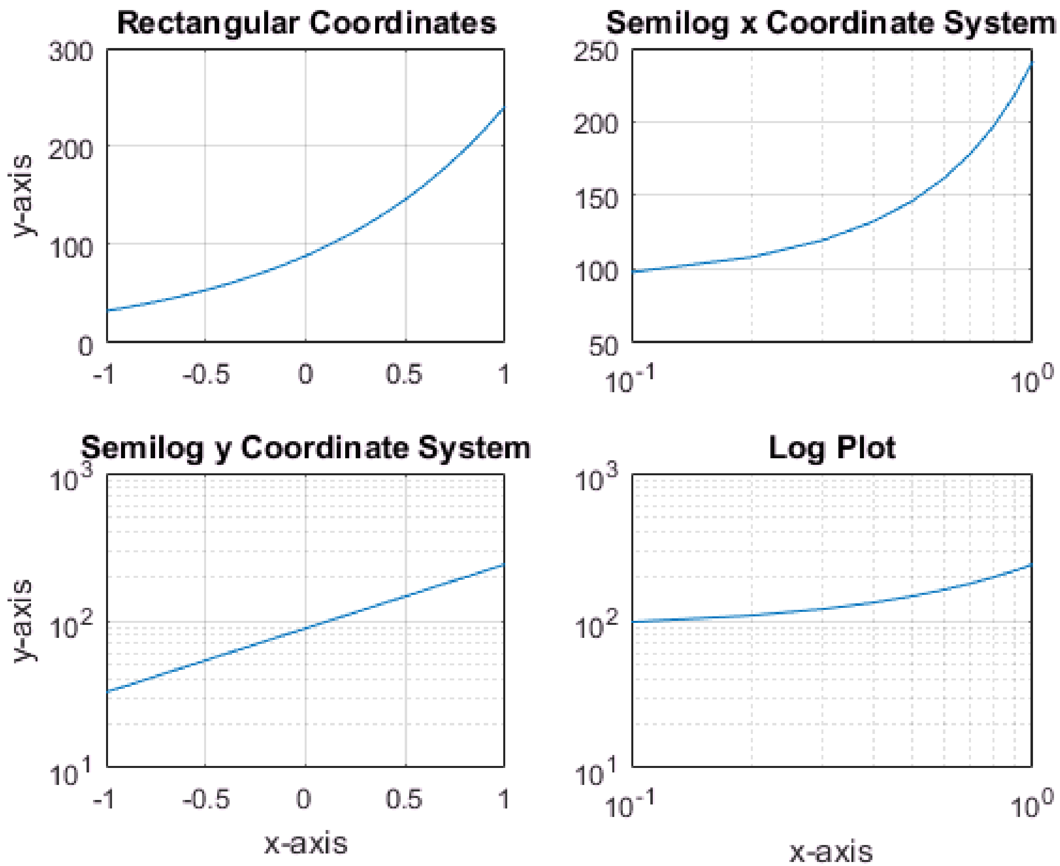
```
figure(3)
x=-1:0.1:1;
y=12*exp(x+2);
subplot(2,2,1)
plot(x,y)
title('Rectangular Coordinates'),ylabel('y-axis'),grid on
subplot(2,2,2)
semilogx(x,y)
```

```
Warning: Negative data ignored
```

```
title('Semilog x Coordinate System'),grid on
subplot(2,2,3)
semilogy(x,y)
title('Semilog y Coordinate System')
ylabel('y-axis'), xlabel('x-axis'), grid on
subplot(2,2,4)
loglog(x,y)
```

```
Warning: Negative data ignored
```

```
title('Log Plot'), xlabel('x-axis'), grid on
```



Exercise 5.4.4

y = 1/*x*

```
figure(4)
x=-1:0.01:1;
y=1./x;
subplot(2,2,1)
plot(x,y)
title('Rectangular Coordinates'), ylabel('y-axis'), grid on
subplot(2,2,2)
semilogx(x,y)
```

```
Warning: Negative data ignored
```

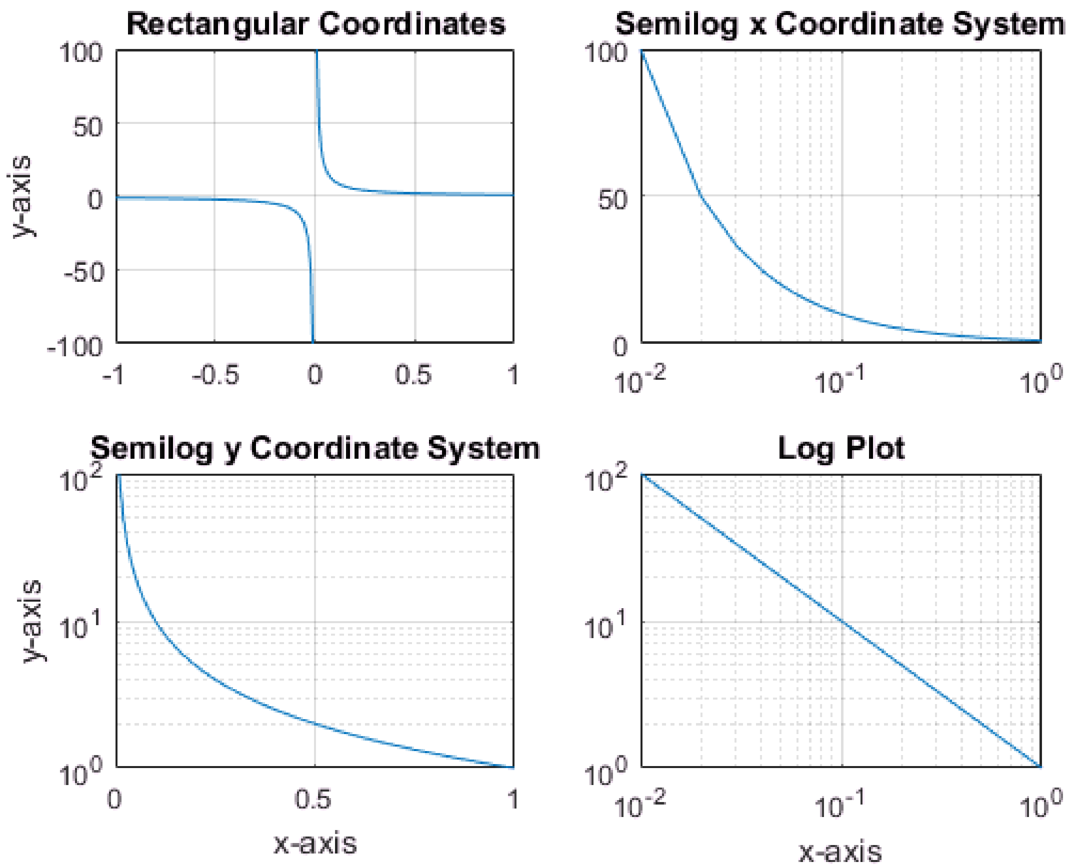
```
title('Semilog x Coordinate System'), grid on
subplot(2,2,3)
semilogy(x,y)
```

```
Warning: Negative data ignored
```

```
title('Semilog y Coordinate System')
ylabel('y-axis'), xlabel('x-axis'), grid on
subplot(2,2,4)
loglog(x,y)
```

```
Warning: Negative data ignored
```

```
title('Log Plot'), xlabel('x-axis'), grid on
```



Practice Exercises 5.5

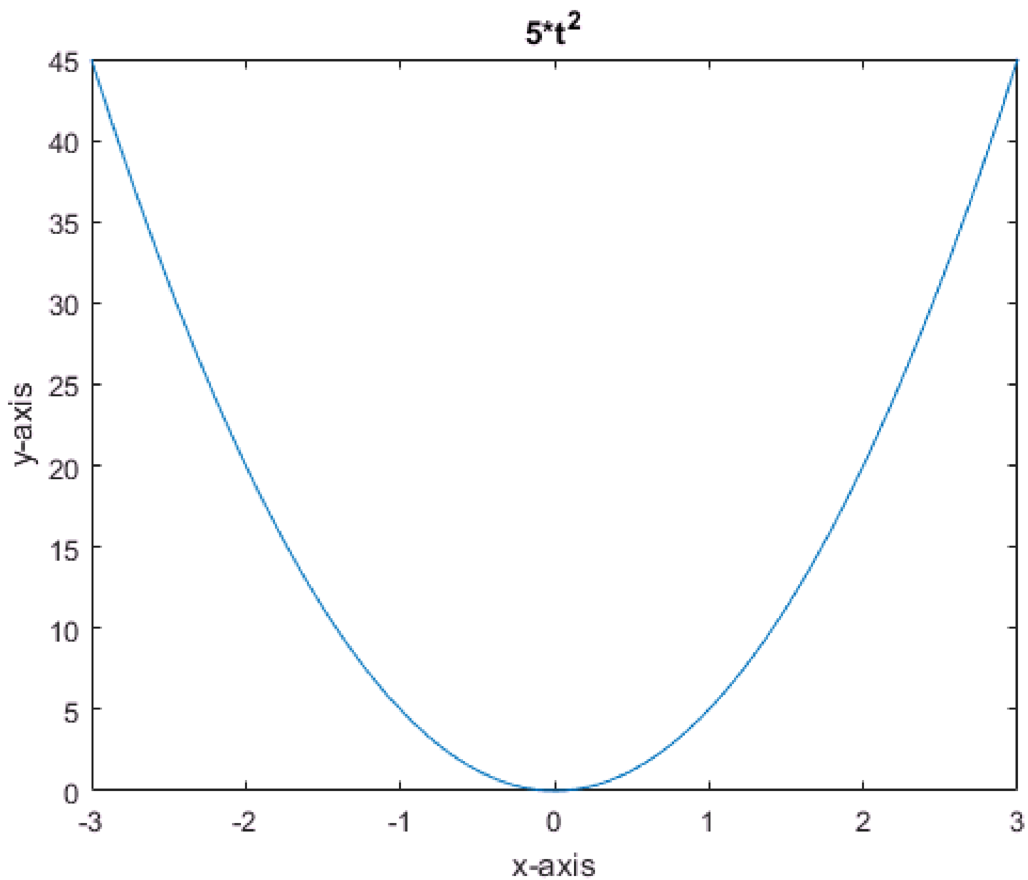
Create a plot of the functions that follow, using `fplot`. You'll need to select an appropriate range for each plot. Don't forget to title and label your graphs.

```
clear,clc, close all
```

Exercise 5.5.1

$f(t) = 5t^2$

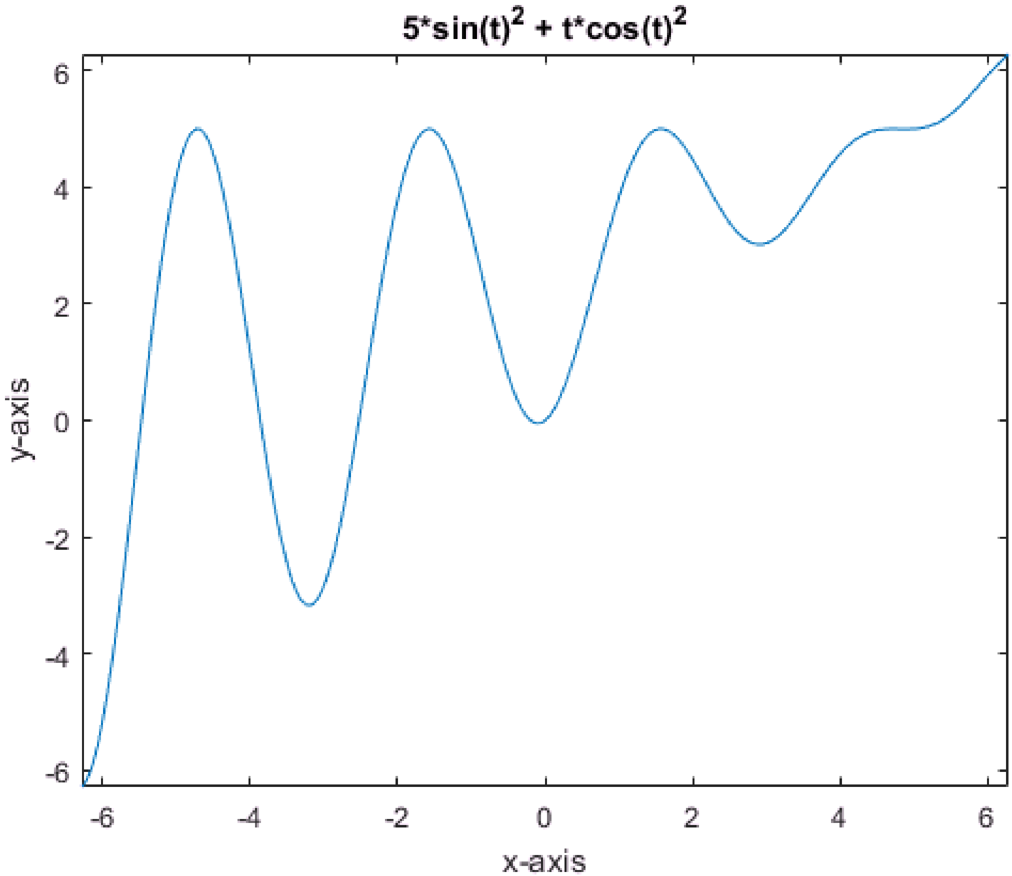
```
fplot(@(t) 5*t.^2,[-3,+3])
title('5*t^2')
xlabel('x-axis')
ylabel('y-axis')
```



Exercise 5.5.2

$f(t) = 5\ \sin^2(t) + t\cos^2(t)$

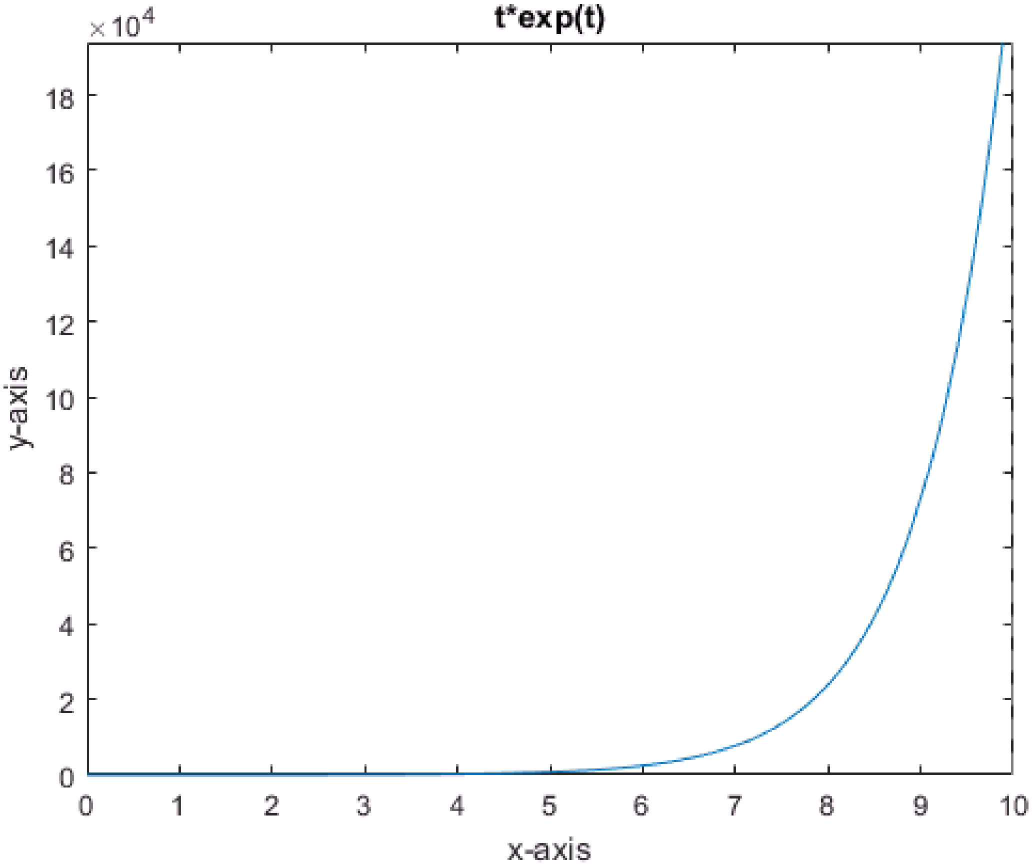
```
fplot(@(t) 5*sin(t).^2 + t.*cos(t).^2,[-2*pi,2*pi])
title('5*sin(t)^2 + t*cos(t)^2')
xlabel('x-axis')
ylabel('y-axis')
```



Exercise 5.5.3

$f(t) = te^t$

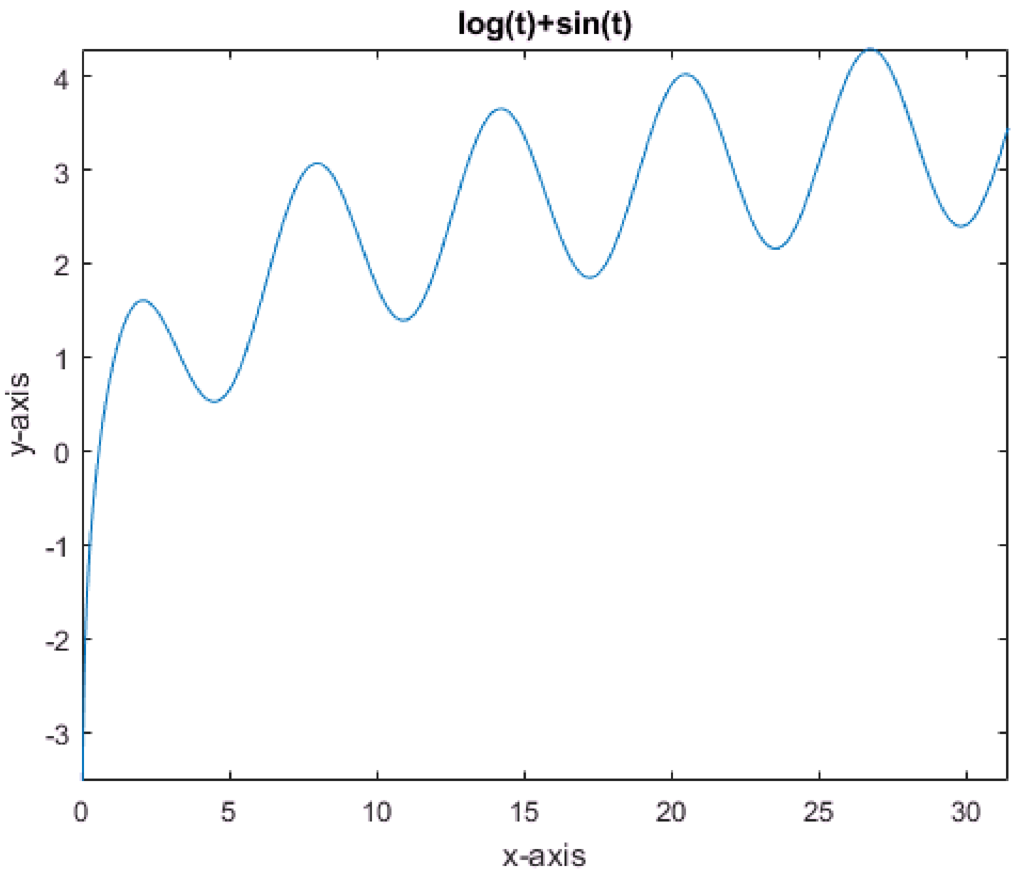
```
fplot(@(t) t.*exp(t),[0,10])
title('t*exp(t)')
xlabel('x-axis')
ylabel('y-axis')
```



Exercise 5.5.4

$f(t) = \ln(t) + \sin(t)$

```
fplot(@(t) log(t)+sin(t),[0,10*pi])
title('log(t)+sin(t)')
xlabel('x-axis')
ylabel('y-axis')
```


log(t)+sin(t)
y-axis
x-axis
4
3
2
1
0
-1
-2
-3
0
5
10
15
20
25
30

Practice Exercise 6.1

Create MATLAB® functions to evaluate the following mathematical functions (make sure you select meaningful function names) and test them. To test your functions you'll need to call them from the command window, or use them in a script M-file program. Remember, each function requires its own M-file.

```
clear,clc
```

Exercise 6.1.1

$y(x) = x^2$

```
x=-3:3;
quadratic(x)
```

```
ans =      9     4     1     0     1     4     9
```

Exercise 6.1.2

$y(x) = e^{\frac{1}{x}}$

```
one_over(x)
```

```
ans =      0.71653      0.60653      0.36788          Inf       2.7183       1.6487       1.3956
```

Exercise 6.1.3

$y(x) = \sin\left(x^2\right)$

```
sin_x_squared(x)
```

```
ans =      0.41212      -0.7568      0.84147            0      0.84147      -0.7568      0.41212
```

Create MATLAB®functions for the following unit conversions (you may need to consult a textbook or the Internet for the appropriate conversion factors). Be sure to test your functions, either from the command window, or by using them in a script M-file program.

Exercise 6.1.4

Inches to feet

```
inches = 0:4:24;
feet = in_to_ft(inches)
```

```
feet =         0    0.33333    0.66667          1    1.3333    1.6667         2
```

Exercise 6.1.5

Calories to joules

```
cal = 0:3;
joules = cal_to_joules(cal)
```

```
joules =         0       4.2       8.4      12.6
```

Exercise 6.1.6

Watts to BTU/hr

```
Watts = 0:5;
Power = Watts_to_Btu_per_hour(Watts)
```

```
Power =         0     3.412     6.824    10.236    13.648     17.06
```

Exercise 6.1.7

Meters to miles

```
meters = 0:100:1000;
miles = meters_to_miles(meters)
```

```
miles =         0   0.06214   0.12428   0.18642   0.24856    0.3107   0.37284   0.43498   0.49712   0.55926    0.6214
```

Exercise 6.1.8

Miles per hour (mph) to ft/s

```
mph=0:10:60;
fps = mph_to_fps(mph)
```

```
fps =         0    14.667    29.333        44    58.667    73.333        88
```

Functions

```
function output = quadratic(x)
output = x.^2;
end
```

```matlab
function output=one_over(x)
output = exp(1./x);
end

function output = sin_x_squared(x)
output = sin(x.^2);
end

function result = in_to_ft(x)
result = x./12;
end

function result=cal_to_joules(x)
result = 4.2.*x;
end

function output = Watts_to_Btu_per_hour(x)
output = x.*3.412;
end

function output = meters_to_miles(x)
output = x./1000.*.6214;
end

function output = mph_to_fps(x)
output = x.*5280/3600;
end
```

Practice Exercise 6.2

Assuming that the matrix dimensions agree, create and test MATLAB® functions to evaluate the following simple mathematical functions with multiple input vectors and a single output vector. In this solution the following test values were used as input to the functions.

```
clear,clc
x=0:4;
y=1:5;
w=0:4;
```

Exercise 6.2.1

$z(x, y) = x + y$

```
answer1=z1(x,y)
```

```
answer1 =      1      3      5      7      9
```

Exercise 6.2.2

$z(a, b, c) = ab^c$

```
answer2 = z2(x,y,w)
```

```
answer2 =            0            2           18          192         2500
```

Exercise 6.2.3

$z(w, x, y) = we^{\frac{x}{y}}$

```
answer3 = z3(x,y,w)
```

```
answer3 =          NaN       7.3891       8.9634       11.381       13.961
```

Exercise 6.2.4

$z(p, t) = p/\sin(t)$

```
answer4 = z4(x,y)
```

```
answer4 =            0       1.0998       14.172       -3.964      -4.1713
```

Assuming that the matrix dimensions agree, create and test MATLAB® functions to evaluate the following simple mathematical functions with a single input vector and multiple output vectors:

Exercise 6.2.5

$f(x) = \cos(x)$

$f(x) = \sin(x)$

```
[answer5a,answer5b] = f5(x)
```

```
answer5a =             1       0.5403     -0.41615     -0.98999     -0.65364
answer5b =             0      0.84147       0.9093      0.14112      -0.7568
```

Exercise 6.2.6

f(x) = $5x^2 + 2$

f(x) = $\sqrt{5x^2 + 2}$

```
[answer6a,answer6b] = f6(x)
```

```
answer6a =      2     7    22    47    82
answer6b =        1.4142       2.6458       4.6904       6.8557       9.0554
```

Exercise 6.2.7

```
[answer7a,answer7b] = f7(x)
```

```
answer7a =             1       2.7183       7.3891       20.086       54.598
answer7b =          -Inf            0      0.69315       1.0986       1.3863
```

Assuming that the matrix dimensions agree, create, and test MATLAB®functions to evaluate the following simple mathematical functions with multiple input vectors and multiple output vectors:

Exercise 6.2.8

$f(x) = \exp(x)$

$f(x) = \ln(x)$

```
[answer8a,answer8b] = f8(x,y)
```

```
answer8a =      1     3     5     7     9
answer8b =     -1    -1    -1    -1    -1
```

Exercise 6.2.9

$f(x, y) = y e^{x}$

$f(x, y) = x e^{y}$

```
[answer9a,answer9b] = f8(x,y)
```

```
answer9a =      1     3     5     7     9
answer9b =     -1    -1    -1    -1    -1
```

Functions

```
function output = z1(x,y)
% summation of x and y
% the matrix dimensions must agree
output = x+y;
end

function output = z2(a,b,c)
% finds a.*b.^c
% the matrix dimensions must agree
output = a.*b.^c;
end

function output = z3(w,x,y)
% finds w.*exp(x./y)
% the matrix dimensions must agree
output = w.*exp(x./y);
end

function output = z4(p,t)
% finds p./sin(t)
% the matrix dimensions must agree
output = p./sin(t);
end

function [a,b]=f5(x)
a = cos(x);
b = sin(x);
end

function [a,b] = f6(x)
a = 5.*x.^2 + 2;
b = sqrt(5.*x.^2 + 2);
end

function [a,b] = f7(x)
a = exp(x);
b = log(x);
end

function [a,b] = f8(x,y)
```

```
a = x+y;
b = x-y;
end

function [a,b] = f9(x,y)
a = y.*exp(x);
b = x.*exp(y);
end
```

Practice Exercise 7.1

```
clear,clc
```

Exercise 7.1.1

Create an M-file to calculate the area *A* of a triangle: $A = \frac{1}{2} * \text{base} * \text{height}$

Prompt the user to enter the values for the base and for the height.

```
b = input('Enter the length of the base of the triangle: ')
```

```
b =     4
```

```
h = input('Enter the height of the triangle: ')
```

```
h =     3
```

```
Area = 1/2*b*h
```

```
Area =     6
```

Exercise 7.1.2

Create an M-file to find the volume *V* of a right circular cylinder:

$V = \pi r^2 h$

Prompt the user to enter the values of *r* and *h*.

```
r = input('Enter the radius of the cylinder: ')
```

```
r =     5
```

```
h = input('Enter the height of the cylinder: ')
```

```
h =     6
```

```
Volume = pi*r.^2*h
```

```
Volume =       471.24
```

Exercise 7.1.3

Create a vector from 0 to *n*, allowing the user to enter the value of *n*.

```
n = input('Enter a value of n: ')
```

```
n =      4
```

```
vector = 0:n
```

```
vector =      0     1     2     3     4
```

Exercise 7.1.4

Create a vector that starts at *a*, ends at *b*, and has a spacing of *c*. Allow the user to input all of these parameters.

```
a = input('Enter the starting value: ')
```

```
a =      0
```

```
b = input('Enter the ending value: ')
```

```
b =    100
```

```
c = input('Enter the vector spacing: ')
```

```
c =     20
```

```
vector = a:c:b
```

```
vector =      0    20    40    60    80   100
```

Practice Exercise 7.2

```
clear,clc
```

Exercise 7.2.1 to 7.6

1. Use the `disp` command to create a title for a table that converts inches to feet.

2. Use the `disp` command to create column headings for your table.

3. Create an `inches` vector from 0 to 120 with an increment of 10.

4. Calculate the corresponding values of `feet`.

5. Group the `inch` vector and the `feet` vector together into a `results` matrix.

6. Use the `fprintf` command to send your results to the command window.

```
% Exercise 7.2.1
disp('Inches to Feet Conversion Table')
```

```
Inches to Feet Conversion Table
```

```
% Exercise 7.2.2
disp('      Inches     Feet')
```

```
      Inches     Feet
```

```
% Exercise 7.2.3
inches = 0:10:120;
% Exercise 7.2.4
feet = inches./12;
% Exercise 7.2.5
results = [inches; feet];
% Exercise 7.2.6
fprintf('  %8.0f  %8.2f \n',results)
```

```
         0      0.00
        10      0.83
        20      1.67
        30      2.50
        40      3.33
        50      4.17
        60      5.00
        70      5.83
        80      6.67
        90      7.50
       100      8.33
       110      9.17
       120     10.00
```

Practice Exercise 8.1

Consider the following matrices:

```
x=[1 10 42 6
    5 8 78 23
    56 45 9 13
    23 22 8 9];

y=[1 2 3; 4 10 12; 7 21 27];

z=[10 22 5 13];
```

Exercise 8.1.1

Using single-index notation, find the index numbers of the elements in each matrix that contain values greater than 10.

```
elements_x = find(x>10)
```

```
elements_x =
     3
     4
     7
     8
     9
    10
    14
    15
```

```
elements_y = find(y>10)
```

```
elements_y =
     6
     8
     9
```

```
elements_z = find(z>10)
```

```
elements_z =      2     4
```

Exercise 8.1.2

Find the row and column numbers (sometimes called subscripts) of the elements in each matrix that contain values greater than 10.

```
[rows_x, cols_x]=find(x>10)
```

```
rows_x =
     3
     4
     3
     4
     1
     2
```

```
     2
     3
cols_x =
     1
     1
     2
     2
     3
     3
     4
     4
```

```
[rows_y, cols_y]=find(y>10)
```

```
rows_y =
     3
     2
     3
cols_y =
     2
     3
     3
```

```
[rows_z, cols_z]=find(z>10)
```

```
rows_z =      1     1
cols_z =      2     4
```

Exercise 8.1.3

Find the values in each matrix that are greater than 10.

```
x(elements_x)
```

```
ans =
    56
    23
    45
    22
    42
    78
    23
    13
```

```
y(elements_y)
```

```
ans =
    21
    12
    27
```

```
z(elements_z)
```

```
ans =     22    13
```

Exercise 8.1.4

Using single-index notation, find the index numbers of the elements in each matrix that contain values greater than 10 and less than 40.

```
elements_x = find(x>10 & x< 40)
```

```
elements_x =
     4
     8
    14
    15
```

```
elements_y = find(y>10 & y< 40)
```

```
elements_y =
     6
     8
     9
```

```
elements_z = find(z>10 & z< 40)
```

```
elements_z =      2     4
```

Exercise 8.1.5

Find the row and column numbers for the elements in each matrix that contain values greater than 10 and less than 40.

```
[rows_x, cols_x]=find(x>10 & x<40)
```

```
rows_x =
     4
     4
     2
     3
```

```
cols_x =
     1
     2
     4
     4
```

```
[rows_y, cols_y]=find(y>10 & y<40)
```

```
rows_y =
     3
     2
     3
```

```
cols_y =
     2
     3
     3
```

```
[rows_z, cols_z]=find(z>10 & z<40)
```

```
rows_z =      1     1
```

```
cols_z =      2     4
```

Exercise 8.1.6

Find the values in each matrix that are greater than 10 and less than 40.

```
x(elements_x)
```

```
ans =
    23
    22
    23
    13
```

```
y(elements_y)
```

```
ans =
    21
    12
    27
```

```
z(elements_z)
```

```
ans =     22    13
```

Exercise 8.1.7

Using single-index notation, find the index numbers of the elements in each matrix that contain values between 0 and 10 or between 70 and 80.

```
elements_x = find((x>0 & x<10) | (x>70 & x<80))
```

```
elements_x =
     1
     2
     6
    10
    11
    12
    13
    16
```

```
elements_y = find((y>0 & y<10) | (y>70 & y<80))
```

```
elements_y =
     1
     2
     3
     4
     7
```

```
elements_z = find((z>0 & z<10) | (z>70 & z<80))
```

```
elements_z =      3
```

Exercise 8.1.8

Use the `length` command together with results from the `find` command to determine how many values in each matrix are between 0 and 10 or between 70 and 80.

```
length_x = length(find((x>0 & x<10) | (x>70 & x<80)))
```

```
length_x =      8
```

```
length_y = length(find((y>0 & y<10) | (y>70 & y<80)))
```

```
length_y =      5
```

```
length_z = length(find((z>0 & z<10) | (z>70 & z<80)))
```

```
length_z =      1
```

Practice Exercise 8.2

The `if` family of functions is particularly useful in functions. Write and test a function for each of these problems, assuming that the input to the function is a scalar.

clear,clc

Exercise 8.2.1

Suppose the legal drinking age is 21 in your state. Write and test a function to determine whether a person is old enough to drink. If the user enters a negative age, use the error function to exit the function and to return an error message.

```
drink(22)
```

```
ans = You can drink
```

```
drink(18)
```

```
ans = Wait 'till you're older
```

Exercise 8.2.2

Many rides at amusement parks require riders to be a certain minimum height. Assume that the minimum height is 48" for a certain ride. Write and test a function to determine whether the rider is tall enough. If the user enters a negative height, use the error function to exit the function and to return an error message.

```
height_requirement(50)
```

```
ans = You may ride
```

```
height_requirement(46)
```

```
ans = You're too short
```

Exercise 8.2.3

When a part is manufactured, the dimensions are usually specified with a tolerance. Assume that a certain part needs to be 5.4 cm long, plus or minus 0.1 cm (5.4 ± 0.1 cm). Write a function to determine whether a part is within these specifications. If the user enters a negative length, use the error function to exit the function and to return an error message.

```
spec(5.6)
```

```
ans =  out of spec
```

```
spec(5.45)
```

```
ans = in spec
```

```
spec(5.2)
```

```
ans =  out of spec
```

Exercise 8.2.4

Unfortunately, the United States currently uses both metric and English units. Suppose the part in Exercise 3 was inspected by measuring the length in inches instead of centimeters. Write and test a function that determines whether the part is within specifications and that accepts input into the function in inches. If the user enters a negative length, use the error function to exit the function and to return an error message.

```
metric_spec(2)
```

```
ans =  out of spec
```

```
metric_spec(2.2)
```

```
ans =  out of spec
```

```
metric_spec(2.4)
```

```
ans =  out of spec
```

Exercise 8.2.5

Many solid-fuel rocket motors consist of three stages. Once the first stage burns out, it separates from the missile and the second stage lights. Then the second stage burns out and separates, and the third stage lights. Finally, once the third stage burns out, it also separates from the missile. Assume that the following data approximately represent the times during which each stage burns:

Stage 1 0–100 seconds

Stage 2 100–170 seconds

Stage 3 170–260 seconds

Write and test a function to determine whether the missile is in Stage 1 flight, Stage 2 flight, Stage 3 flight, or free flight (unpowered). If the user enters a negative time, use the error function to exit the function and to return an error message.

```
flight(50)
```

```
ans = first stage
```

```
flight(110)
```

```
ans = second stage
```

```
flight(200)
```

```
ans = third stafe
```

```
flight(300)
```

```
ans = free flight
```

Functions

These functions must be stored as separate m-files, or may be part of a single live script file.

```
function output = drink(x)
if x<0
    error('Age can not be negative')
end
if x>=21
    output = 'You can drink';
else
    output = 'Wait ''till you''re older';
end
end

function output = height_requirement(x)
if x<0
    error('Height can not be negative')
end
if x>=48
    output='You may ride';
else
    output = 'You''re too short';
end
end

function output = spec(x)
if x<0
    error('The length can not be negative')
end
if x>=5.3 & x<=5.5
    output = 'in spec';
else
    output = ' out of spec';
end
end

function output = metric_spec(x)
if x<0
    error('The length can not be negative')
end
if x>=5.3/2.54 & x<=5.5/2.54
    output = 'in spec';
else
    output = ' out of spec';
end
end
```

```
function output = flight(x)
if x<0
    error('The time can not be negative')
end
if x>=0 & x<=100
    output='first stage';
elseif x<=170
    output = 'second stage';
elseif x<260
    output = 'third stafe';
else
    output = 'free flight';
end
end
```

Practice Exercise 8.3

Use the `switch/case` structure to solve these problems.

```
clear,clc
```

Exercise 8.3.1

Create a program that prompts the user to enter his or her year in school—freshman, sophomore, junior, or senior. The input will be a string. Use the `switch/case` structure to determine which day finals will be given for each group—Monday for freshmen, Tuesday for sophomores, Wednesday for juniors, and Thursday for seniors.

```
year = input('Enter the name of your year in school:  ','s')
```

```
year = sophomore
```

```
switch year
    case 'freshman'
        day='Monday';
    case 'sophomore'
        day = 'Tuesday';
    case 'junior'
        day = 'Wednesday';
    case 'senior'
        day = 'Thursday';
    otherwise
        day = 'I don''t know that year';
end
disp(['Your finals are on ',day])
```

```
Your finals are on Tuesday
```

Exercise 8.3.2

Repeat Exercise 1, but this time with a menu.

```
disp('What year are you in school?')
```

```
What year are you in school?
```

```
disp('Use the menu box to make your selection ')
```

```
Use the menu box to make your selection
```

```
    choice = menu('Year in School','freshman','sophomore','junior', 'senior')
```

```
choice =      4
```

```

switch choice
    case 1
```

```
        day = 'Monday';

    case 2
        day = 'Tuesday';
    case 3
        day = 'Wednesday';
    case 4
        day = 'Thursday';
end
disp(['Your finals are on ',day])
```

```
Your finals are on Thursday
```

Exercise 8.3.3

Create a program to prompt the user to enter the number of candy bars he or she would like to buy. The input will be a number. Use the `switch/case` structure to determine the bill, where

1 bar = $0.75

2 bars = $1.25

3 bars = $1.65

more than 3 bars = $1.65 + $0.30 (number ordered – 3)

```
num = input('How many candy bars would you like? ')
```

```
num =     5
```

```
switch num
    case 1
        bill = 0.75;
    case 2
        bill = 1.25;
    case 3
        bill = 1.65;
    otherwise
        bill = 1.65 + (num-3)*0.30;
end
fprintf('Your bill is %5.2f \n',bill)
```

```
Your bill is  2.25
```

Practice Exercise 9.1

Use a `for` loop to solve the following problems.

Exercise 9.1.1

Create a table that converts inches to feet.

```
inches = 0:3:24;
for k=1:length(inches)
    feet(k) = inches(k)/12;
end
result=[inches',feet']
```

```
result =
           0           0
           3        0.25
           6         0.5
           9        0.75
          12           1
          15        1.25
          18         1.5
          21        1.75
          24           2
```

Exercise 9.1.2

Consider the following matrix of values:

```
x = [ 45,23,17,34,85,33];
```

How many values are greater than 30? (Use a counter.)

```
count=0;
for k=1:length(x)
    if x(k)>30
        count = count+1;
    end
end
fprintf('The are %4.0f values greater than 30 \n',count)
```

```
The are    4 values greater than 30
```

Problem 9.1.3

Repeat Exercise 2, this time using the `find` command.

```
num = length(find(x>30));
fprintf('The are %4.0f values greater than 30 \n',num)
```

```
The are    4 values greater than 30
```

Exercise 9.1.4

Use a `for` loop to sum the elements of the matrix in Problem 2. Check your results with the `sum` function. (Use the `help` feature if you don't know or remember how to use `sum`.)

```
total = 0;
for k=1:length(x)
    total = total + x(k);
end
disp('The total is: ')
```

```
The total is:
```

```
disp(total)
```

```
   237
```

```
sum(x)
```

```
ans =    237
```

Exercise 9.1.5

Use a `for` loop to create a vector containing the first 10 elements in the harmonic series, i.e.,

1/1 1/2 1/3 1/4 1/5… 1/10

```
for k=1:10
    x(k)=1/k;
end
format rat
disp(x)
```

```
  Columns 1 through 5
       1              1/2            1/3            1/4            1/5
  Columns 6 through 10
       1/6            1/7            1/8            1/9            1/10
```

Exercise 9.1.6

Use a `for` loop to create a vector containing the first 10 elements in the alternating harmonic series, i.e.,

1/1 -1/2 1/3 -1/4 1/5… -1/10

```
for k=1:10
    x(k)=(-1)^(k+1)/k;
end
format rat
disp(x)
```

```
  Columns 1 through 5
       1             -1/2            1/3           -1/4            1/5
  Columns 6 through 10
```

```
-1/6          1/7          -1/8          1/9          -1/10
```

Practice Exercise 9.2

Use a `while` loop to solve the following problems.

Exercise 9.2.1

Create a conversion table of inches to feet.

```
inches = 0:3:24;
k=1;
while k<=length(inches)
    feet(k) = inches(k)/12;
    k=k+1;
end
disp('      Inches     Feet')
```

```
      Inches     Feet
```

```
fprintf('  %8.0f   %8.2f \n',[inches;feet])
```

```
           0       0.00
           3       0.25
           6       0.50
           9       0.75
          12       1.00
          15       1.25
          18       1.50
          21       1.75
          24       2.00
```

Exercise 9.2.2

Consider the following matrix of values:

```
x = [ 45,23,17,34,85,33];
```

How many values are greater than 30? (Use a counter.)

```
k=1;
count = 0;
while k<=length(x)
    if x(k)>=30;
        count = count +1;
    end
    k=k+1;
end
fprintf('There are %4.0f values greater than 30 \n',count)
```

```
There are    4 values greater than 30
```

Exercise 9.2.3

Compare your solution to Exercise 2 to the solutions you created in Practice Exercise 9.1, where you used both a for loop and the find function to solve the same problem.

```
count = length(find(x>30))
```

```
count =        4
```

Exercise 9.2.4

Use a `while` loop to sum the elements of the matrix in Exercise 2. Check your results with the `sum` function. (Use the `help` feature if you don't know or remember how to use `sum`.)

```
k=1;
total = 0;
while k<=length(x)
    total = total + x(k);
    k=k+1;
end
disp(total)
```

```
   237
```

```
sum(x)
```

```
ans =      237
```

Exercise 9.2.5

Use a `while` loop to create a vector containing the first 10 elements in the harmonic series, i.e.,

1/1 1/2 1/3 1/4 1/5... 1/10

```
k=1;
while k<=10
    x(k)=1/k;
    k=k+1;
end
format rat
disp(x)
```

```
  Columns 1 through 5
       1              1/2            1/3            1/4            1/5
  Columns 6 through 10
       1/6            1/7            1/8            1/9            1/10
```

Exercise 9.2.6

Use a `while` loop to create a vector containing the first 10 elements in the alternating harmonic series, i.e.,

1/1 -1/2 1/3 -1/4 1/5... -1/10

```
k=1;
while k<=10
```

```
    x(k) = (-1)^(k+1)/k;
    k = k+1;
end
format rat
disp(x)
```

```
  Columns 1 through 5
       1              -1/2            1/3            -1/4            1/5
  Columns 6 through 10
      -1/6             1/7           -1/8             1/9           -1/10
```

Practice Exercise 10.1

Exercise 10.1.1

Use the `dot` function to find the dot product of the following vectors:

$$\vec{A} = [1\ 2\ 3\ 4]$$
$$\vec{B} = [12\ 20\ 15\ 7]$$

```
A = [ 1 2 3 4];
B = [ 12  20  15  7];
dot(A,B)
```

```
ans =    125
```

Exercise 10.1.2

Find the dot product of $\vec{A}$ and $\vec{B}$

by summing the array products of $\vec{A}$ and $\vec{B}$

(`sum(A.*B)`).

```
sum(A.*B)
```

```
ans =    125
```

Exercise 10.1.3

```
price=[0.99, 1.49, 2.50, 0.99, 1.29];
num = [4, 3, 1, 2, 2];
total=dot(price,num)
```

```
total =         15.49
```

Practice Exercise 10.2

Which of the following sets of matrices can be multiplied together?

Exercise 10.2.1

```
A=[2 5;
   2 9;
   6 5];
B=[2 5;
   2 9;
   6 5];
% These can not be multiplied because the number of columns in A does not
% equal the number of rows in B
```

Exercise 10.2.2

```
A=[2 5;
   2 9;
   6 5];
B=[1 3 12;
   5 2 9];
% Since A is a 3 x 2 matrix and B is a 2 x 3 matrix, they can be multiplied
A*B
```

```
ans =
    27    16    69
    47    24   105
    31    28   117
```

```
% However A*B does not equal B*A
B*A
```

```
ans =
    80    92
    68    88
```

Exercise 10.2.3

```
A=[5 1 9;
   7 2 2];
B = [8 5;
     4 2;
     8 9];
% Since A is a 2x3 matrix and B is a 3x2 matrix, they can be multiplied
A*B
```

```
ans =
   116   108
    80    57
```

```
%However, A*B does not equal B*A
```

```
B*A
```

```
ans =
    75    18    82
    34     8    40
   103    26    90
```

Exercise 10.2.4

```
A=[1 9 8;
   8 4 7;
   2 5 3];
B=[7;
   1;
   5];
% Since A is a 3x3 matrix and B is a 3x1 matrix they can be multiplied
A*B
```

```
ans =
    56
    95
    34
```

```
% However B*A won't work
```

Practice Exercise 10.3

Exercise 10.3.1

Find the inverse of the following magic matrices, both by using the `inv` function and by raising the matrix to the –1 power:

a

```
a=magic(3)
```

```
a =
     8     1     6
     3     5     7
     4     9     2
```

```
inv(magic(3))
```

```
ans =
      0.14722     -0.14444     0.063889
    -0.061111     0.022222      0.10556
    -0.019444      0.18889     -0.10278
```

```
magic(3)^-1
```

```
ans =
      0.14722     -0.14444     0.063889
    -0.061111     0.022222      0.10556
    -0.019444      0.18889     -0.10278
```

b

```
b=magic(4)
```

```
b =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
inv(b)
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
1.306145e-17.
```

```
ans =
   9.3825e+13   2.8147e+14  -2.8147e+14  -9.3825e+13
   2.8147e+14   8.4442e+14  -8.4442e+14  -2.8147e+14
  -2.8147e+14  -8.4442e+14   8.4442e+14   2.8147e+14
  -9.3825e+13  -2.8147e+14   2.8147e+14   9.3825e+13
```

```
b^-1
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
1.306145e-17.
```

```
ans =
   9.3825e+13   2.8147e+14  -2.8147e+14  -9.3825e+13
   2.8147e+14   8.4442e+14  -8.4442e+14  -2.8147e+14
  -2.8147e+14  -8.4442e+14   8.4442e+14   2.8147e+14
  -9.3825e+13  -2.8147e+14   2.8147e+14   9.3825e+13
```

c

```
c=magic(5)
```

```
c =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
inv(magic(5))
```

```
ans =
   -0.0049359     0.051154    -0.035385    0.0011538    0.0033974
     0.043141    -0.037308   -0.0046154     0.012692    0.0014744
    -0.030256    0.0030769    0.0030769    0.0030769      0.03641
    0.0046795   -0.0065385     0.010769     0.043462    -0.036987
    0.0027564        0.005     0.041538       -0.045      0.01109
```

```
magic(5)^-1
```

```
ans =
   -0.0049359     0.051154    -0.035385    0.0011538    0.0033974
     0.043141    -0.037308   -0.0046154     0.012692    0.0014744
    -0.030256    0.0030769    0.0030769    0.0030769      0.03641
    0.0046795   -0.0065385     0.010769     0.043462    -0.036987
    0.0027564        0.005     0.041538       -0.045      0.01109
```

Exercise 10.3.2

Find the determinant of each of the matrices in Exercise 1.

```
det(a)
```

```
ans =   -360
```

```
det(b)
```

```
ans =   -1.4495e-12
```

```
det(c)
```

```
ans =      5.07e+06
```

Exercise 10.3.3

Consider the following matrix:

```
A=[1 2 3;2 4 6;3 6 9];
```

Would you expect it to be singular or not? (Recall that singular matrices have a determinant of 0 and do not have an inverse.)

```
det(A)
```

```
ans =      0
```

```
inv(A)
```

```
Warning: Matrix is singular to working precision.
```

```
ans =
   Inf   Inf   Inf
   Inf   Inf   Inf
   Inf   Inf   Inf
```

```
%Notice that the three lines are just multiples of each other, and
%therefore do not represent independent equations
```

Practice Exercise 11.1

```
clear,clc
```

Exercise 11.1.1

Enter the following list of numbers into arrays of each of the numeric data types [1, 4, 6; 3, 15, 24; 2, 3, 4]:

(**a)** Double-precision floating point—name this array `A`

(**b)** Single-precision floating point—name this array `B`

(**c)** Signed integer (pick a type)—name this array `C`

(**d)** Unsigned integer (pick a type)—name this array `D`

```
A = [1,4,6; 3 15, 24; 2, 3,4];
B=single(A)
```

```
B =
     1     4     6
     3    15    24
     2     3     4
```

```
C=int8(A)
```

```
C =
    1    4    6
    3   15   24
    2    3    4
```

```
D=uint8(A)
```

```
D =
    1    4    6
    3   15   24
    2    3    4
```

Exercise 11.1.2

Create a new matrix `E` by adding `A` to `B`:

E = A + B

What data type is the result?

```
E = A+B
```

```
E =
     2     8    12
     6    30    48
     4     6     8
```

```
% The result is a single precision array
```

Exercise 11.1.3

Define x as an integer data type equal to 1 and y as an integer data type equal to 3.

(**a**) What is the result of the calculation x/y?

(**b**) What is the data type of the result?

(**c**) What happens when you perform the division when x is defined as the integer 2 and y as the integer 3?

```
x=int8(1)
```

```
x =    1
```

```
y=int8(3)
```

```
y =    3
```

```
result1=x./y
```

```
result1 =    0
```

```
% This calculation returns the integer 0
x=int8(2)
```

```
x =    2
```

```
result2=x./y
```

```
result2 =    1
```

```
% This calculation returns the integer 1 - It appears that Matlab rounds
% the answer
```

Exercise 11.1.4

Use intmax to determine the largest number you can define for each of the numeric data types. (Be sure to include all eight integer data types.)

```
intmax('int8')
```

```
ans =    127
```

```
intmax('int16')
```

```
ans =    32767
```

```
intmax('int32')
```

```
ans =    2147483647
```

```
intmax('int64')
```

```
ans =    9223372036854775807
```

```
intmax('uint8')
```

```
ans =    255
```

```
intmax('uint16')
```

```
ans =    65535
```

```
intmax('uint32')
```

```
ans =    4294967295
```

```
intmax('uint64')
```

```
ans =    18446744073709551615
```

Exercise 11.1.5

Use MATLAB® to determine the smallest number you can define for each of the numeric data types. (Be sure to include all eight integer data types.)

```
intmin('int8')
```

```
ans =    -128
```

```
intmin('int16')
```

```
ans =    -32768
```

```
intmin('int32')
```

```
ans =    -2147483648
```

```
intmin('int64')
```

```
ans =    -9223372036854775808
```

```
intmin('uint8')
```

```
ans =    0
```

```
intmin('uint16')
```

```
ans =    0
```

```
intmin('uint32')
```

```
ans =    0
```

```
intmin('uint64')
```

```
ans =    0
```

Practice Exercise 11.2

Exercise 11.2.1

Create a character array consisting of the letters in your name.

```
name ='Holly'
```

```
name = Holly
```

Exercise 11.2.2

What is the decimal equivalent of the letter *g* ?

```
G=double('g')
```

```
G =    103
```

```
fprintf('The decimal equivalent of the letter g is %5.0f \n',G)
```

```
The decimal equivalent of the letter g is   103
```

Exercise 11.2.3

Upper- and lowercase letters are 32 apart in decimal equivalent. (Uppercase comes first.) Using nested functions, convert the string “matlab” to the uppercase equivalent, “MATLAB.”

```
m='matlab'
```

```
m = matlab
```

```
M=char(double(m)-32)
```

```
M = MATLAB
```

Practice Exercise 11.4

```
clear,clc
```

Exercise 11.4.1

Create a character matrix called `names` of the names of all the planets. Your matrix should have nine rows.

```
names=char('Mercury','Venus','Earth','Mars','Jupiter','Saturn','Uranus','Neptune','Pluto')
```

```
names =
Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune
Pluto
```

Exercise 11.4.2

Some of the planets can be classified as rocky midgets and others as gas giants. Create a character matrix called `type`, with the appropriate designation on each line.

```
R='rocky';
G='gas giants';
type=char(R,R,R,R,G,G,G,G,R)
```

```
type =
rocky
rocky
rocky
rocky
gas giants
gas giants
gas giants
gas giants
rocky
```

Exercise 11.4.3

Create a character matrix of nine spaces, one space per row.

```
space =[' ';' ';' ';' ';' ';' ';' ';' ';' '];
```

Exercise 11.4.4

Combine your matrices to form a table listing the names of the planets and their designations, separated by a space.

```
planets =[names,space,type]
```

```
planets =
Mercury rocky
Venus   rocky
Earth   rocky
Mars    rocky
Jupiter gas giants
Saturn  gas giants
Uranus  gas giants
Neptune gas giants
Pluto   rocky
```

Exercise 11.4.5

Use the Internet to find the mass of each of the planets, and store the information in a matrix called `mass`. (Or use the data from Example 11.3.) Use the `num2str` function to convert the numeric array into a character array, and add it to your table.

This data was found at http://sciencepark.etacude.com/astronomy/pluto.php Similar data is found at many websites

```
mercury=3.303e23; % kg
venus = 4.869e24; % kg
earth = 5.976e24;  % kg
mars = 6.421e23;   % kg
jupiter=1.9e27;   % kg
saturn = 5.69e26;  % kg
uranus = 8.686e25; % kg
neptune = 1.024e26; % kg
pluto = 1.27e22;    % kg
mass = [mercury,venus,earth,mars,jupiter, saturn,uranus,neptune,pluto]';
newtable=[planets,space,num2str(mass)]
```

```
newtable =
Mercury rocky      3.303e+23
Venus   rocky      4.869e+24
Earth   rocky      5.976e+24
Mars    rocky      6.421e+23
Jupiter gas giants   1.9e+27
Saturn  gas giants  5.69e+26
Uranus  gas giants 8.686e+25
Neptune gas giants 1.024e+26
Pluto   rocky       1.27e+22
```

Exercise 11.6

Recreate the table from problems 1 through 5, using strings arrays instead of character arrays. Use the MATLAB help feature to determine the appropriate syntax.

```
Names = string({'Mercury','Venus','Earth','Mars','Jupiter','Saturn','Uranus','Neptue','Pluto'});
Type = string({R,R,R,R,G,G,G,G,R});
newertable = [Names',Type',mass]
```

```
newertable =
    "Mercury"    "rocky"         "3.303e+23"
    "Venus"      "rocky"         "4.869e+24"
    "Earth"      "rocky"         "5.976e+24"
    "Mars"       "rocky"         "6.421e+23"
    "Jupiter"    "gas giants"    "1.9e+27"
    "Saturn"     "gas giants"    "5.69e+26"
    "Uranus"     "gas giants"    "8.686e+25"
    "Neptue"     "gas giants"    "1.024e+26"
    "Pluto"      "rocky"         "1.27e+22"
```

For a nicer presentation use fprintf

```
fprintf('%15s   %15s  %15s \n',newertable')
```

```
        Mercury              rocky        3.303e+23
          Venus              rocky        4.869e+24
          Earth              rocky        5.976e+24
           Mars              rocky        6.421e+23
        Jupiter         gas giants          1.9e+27
         Saturn         gas giants         5.69e+26
         Uranus         gas giants        8.686e+25
         Neptue         gas giants        1.024e+26
          Pluto              rocky         1.27e+22
```

Practice Exercise 11.4

```
clear,clc
```

Exercise 11.3.1

Create a three-dimensional array consisting of a 3 x 3 magic square, a 3 x 3 matrix of zeros, and a 3 x 3 matrix of ones.

```
a=magic(3)
```

```
a =
     8     1     6
     3     5     7
     4     9     2
```

```
b=zeros(3)
```

```
b =
     0     0     0
     0     0     0
     0     0     0
```

```
c=ones(3)
```

```
c =
     1     1     1
     1     1     1
     1     1     1
```

```
x(:,:,1)=a
```

```
x =
     8     1     6
     3     5     7
     4     9     2
```

```
x(:,:,2)=b
```

```
x =
(:,:,1) =
     8     1     6
     3     5     7
     4     9     2
(:,:,2) =
     0     0     0
     0     0     0
     0     0     0
```

```
x(:,:,3)=c
```

```
x =
(:,:,1) =
     8     1     6
     3     5     7
     4     9     2
```

```
(:,:,2) =
     0     0     0
     0     0     0
     0     0     0
(:,:,3) =
     1     1     1
     1     1     1
     1     1     1
```

Exercise 11.3.2

Use triple indexing such as `A(m,n,p)` to determine what number is in row 3, column 2, page 1 of the matrix you created in Exercise 1.

```
x(3,2,1)
```

```
ans =      9
```

Exercise 11.3.3

Find all the values in row 2, column 3 (on all the pages) of the matrix.

```
x(2,3,:)
```

```
ans =
(:,:,1) =
     7
(:,:,2) =
     0
(:,:,3) =
     1
```

Exercise 11.3.4

Find all the values in all the rows and pages of column 3 of the matrix.

```
x(:,3,:)
```

```
ans =
(:,:,1) =
     6
     7
     2
(:,:,2) =
     0
     0
     0
(:,:,3) =
     1
     1
     1
```

Practice Exercise 12.1

```
clear,clc
```

Exercise 12.1.1

Create the following symbolic variables, using either the `sym` or `syms` command:

`x, a, b, c, d`

```
syms x a b c
%or
d=sym('d')  %etc
```

d $= d$

Exercise 12.1.2

Verify that the variables you created in Exercise 1 are listed in the workspace window as symbolic variables. Use them to create the following symbolic ***expressions***:

```
ex1 = x^2-1
```

ex1 $= x^2 - 1$

```
ex2 = (x+1)^2
```

ex2 $= (x + 1)^2$

```
ex3 = a*x^2-1
```

ex3 $= a\,x^2 - 1$

```
ex4 = a*x^2 + b*x + c
```

ex4 $= a\,x^2 + b\,x + c$

```
ex5 = a*x^3 + b*x^2 + c*x + d
```

ex5 $= a\,x^3 + b\,x^2 + c\,x + d$

```
ex6 = sin(x)
```

ex6 $= \sin(x)$

Exercise 12.1.3

Create the following symbolic ***equations.***

```
eq1 =  x^2==1
```

eq1 $= x^2 = 1$

```
eq2 = (x+1)^2==0
```

eq2 $= (x+1)^2 = 0$

```
eq3 = a*x^2==1
```

eq3 $= a\,x^2 = 1$

```
eq4 = a*x^2 + b*x + c==0
```

eq4 $= a\,x^2 + b\,x + c = 0$

```
eq5 = a*x^3 + b*x^2 + c*x + d==0
```

eq5 $= a\,x^3 + b\,x^2 + c\,x + d = 0$

```
eq6 = sin(x)==0
```

eq6 $= \sin(x) = 0$

Save the variables, expressions, and equations you created in this practice to use in later practice exercises in the chapter.

Practice Exercise 12.2

Use the variables defined in Practice Exercises 12.1 in these exercises.

Exercise 12.2.1

Multiply `ex1`by `ex2`, and name the result `y1`.

```
y1=ex1*ex2
```

y1 = $\left(x^2-1\right)\ \left(x+1\right)^2$

Exercise 12.2.2

Divide `ex1` by `ex2`, and name the result `y2`.

```
y2=ex1/ex2
```

y2 =

$$\frac{x^2-1}{\left(x+1\right)^2}$$

Exercise 12.2.3

Use the `numden` function to extract the numerator and denominator from `y1` and `y2`.

```
[num1,den1]=numden(y1)
```

num1 = $\left(x^2-1\right)\ \left(x+1\right)^2$

den1 = 1

```
[num2,den2]=numden(y2)
```

num2 = $x-1$

den2 = $x+1$

Exercise 12.2.4

Use the `factor`, `expand`, `collect`, and `simplify` functions on `y1`, `y2`.

a

```
factor(y1)
```

ans $= \begin{pmatrix} x-1 & x+1 & x+1 & x+1 \end{pmatrix}$

```
expand(y1)
```

ans $= x^4 + 2\,x^3 - 2\,x - 1$

```
collect(y1)
```

ans $= x^4 + 2\,x^3 - 2\,x - 1$

```
simplify(y1)
```

ans $= \left(x^2 - 1\right)\,\left(x + 1\right)^2$

b

```
factor(y2)
```

ans =

$$\begin{pmatrix} x-1 & \frac{1}{x+1} \end{pmatrix}$$

```
expand(y2)
```

ans =

$$\frac{x^2}{x^2 + 2\,x + 1} - \frac{1}{x^2 + 2\,x + 1}$$

```
collect(y2)
```

ans =

$$\frac{x-1}{x+1}$$

```
simplify(y2)
```

ans =

$$\frac{x-1}{x+1}$$

Practice Exercise 12.3

Use the variables and expressions you defined in Practice Exercises 12.1 to solve these exercises

Exercise 12.3.1

Use the `solve` function to solve both `ex1` and `eq1`.

```
solve(ex1)
```

ans =

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

```
solve(eq1)
```

ans =

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Exercise 12.3.2

Use the `solve` function to solve both `ex2` and `eq2`.

```
solve(ex2)
```

ans =

$$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

```
solve(eq2)
```

ans =

$$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

Exercise 12.3.3

Use the `solve` function to solve both `ex3` and `eq3 for` both `x` and `a`.

```
solve(ex3,x)
```

ans =

$$\begin{pmatrix} -\frac{1}{\sqrt{a}} \\ \frac{1}{\sqrt{a}} \end{pmatrix}$$

```
solve(ex3,a)
```

ans =

$$\frac{1}{x^2}$$

```
solve(eq3,x)
```

ans =

$$\begin{pmatrix} -\frac{1}{\sqrt{a}} \\ \frac{1}{\sqrt{a}} \end{pmatrix}$$

```
solve(eq3,a)
```

ans =

$$\frac{1}{x^2}$$

Exercise 12.3.4

Use the `solve` function to solve `ex4` and `eq4` for both `x` and `a`.

```
solve(ex4,x)
```

ans =

$$\begin{pmatrix} -\frac{b+\sqrt{b^2-4\,a\,c}}{2\,a} \\ -\frac{b-\sqrt{b^2-4\,a\,c}}{2\,a} \end{pmatrix}$$

```
solve(ex4,a)
```

ans =

$$-\frac{c+b\,x}{x^2}$$

```
solve(eq4,x)
```

ans =

$$\begin{pmatrix} -\frac{b+\sqrt{b^2-4\,a\,c}}{2\,a} \\ -\frac{b-\sqrt{b^2-4\,a\,c}}{2\,a} \end{pmatrix}$$

```
solve(eq4,a)
```

ans =

$$-\frac{c+b\,x}{x^2}$$

Exercise 12.3.5

Both `ex4` and `eq4` represent the quadratic equation—the general form of a second-order polynomial. The solution for *x* is usually memorized by students in early algebra classes. Expression/equation 5 in these exercises is the general form of a third-order polynomial. Use the `solve` function to solve these expressions/equations, and comment on why students do not memorize the general solution of a third-order polynomial.

```
solve(ex5,x)
```

ans =

$$\begin{pmatrix} \mathrm{root}(a\,z^3+b\,z^2+c\,z+d,z,1) \\ \mathrm{root}(a\,z^3+b\,z^2+c\,z+d,z,2) \\ \mathrm{root}(a\,z^3+b\,z^2+c\,z+d,z,3) \end{pmatrix}$$

Exercise 12.3.6

Use the `solve` function to solve both `ex1` and `eq1`. On the basis of your knowledge of trigonometry, comment on this solution.

```
solve(ex6)
```

ans = 0

```
solve(eq6)
```

ans = 0

The sin function is equal to 0 at a number of x values, but just one is reported

Practice Exercises 12.4

Consider the following system of linear equations to use in Exercises 12.1 through 12.5:

$5x + 6y - 3z = 10$

$3x - 3y + 2z = 14$

$2x - 4y - 12z = 24$

```
clear,clc
```

Exercise 12.4.1

Solve this system of equations by means of the linear algebra techniques discussed in Chapter 10.

```
coef = [5 6 -3; 3 -3 2; 2 -4 -12];
result=[10; 14; 24];
answer=coef\result
```

```
answer =
      3.5314
     -1.6987
    -0.84519
```

Exercise 12.4.2

Define a symbolic equation representing each equation in the given system of equations. Use the `solve` function to solve for *x*, *y*, and *z*.

```
syms x y z
a = 5*x + 6*y - 3*z==10
```

a $= 5x + 6y - 3z = 10$

```
b = 3*x - 3*y + 2*z==14
```

b $= 3x - 3y + 2z = 14$

```
c = 2*x - 4*y - 12*z==24
```

c $= 2x - 4y - 12z = 24$

```
answer=solve(a,b,c)
```

```
answer =
    x: [1×1 sym]
    y: [1×1 sym]
    z: [1×1 sym]
```

Exercise 12.4.3

Display the results from Exercise 2 by using the structure array syntax.

```
answer.x
```

```
ans =
```

$$\frac{844}{239}$$

```
answer.y
```

```
ans =
```

$$-\frac{406}{239}$$

```
answer.z
```

```
ans =
```

$$-\frac{202}{239}$$

Convert to a double to make comparisons easier

```
double(answer.x)
```

```
ans =        3.5314
```

```
double(answer.y)
```

```
ans =       -1.6987
```

```
double(answer.z)
```

```
ans =      -0.84519
```

Exercise 12.4.4

Display the results from Exercise 2 by specifying the output names.

```
[X,Y,Z]=solve(a,b,c)
```

```
X =
```

$$\frac{844}{239}$$

```
Y =
```

$$-\frac{406}{239}$$

```
Z =
```

$-\frac{202}{239}$

Convert to a double to make comparisons easier

```
double(X)
```

```
ans =        3.5314
```

```
double(Y)
```

```
ans =       -1.6987
```

```
double(Z)
```

```
ans =      -0.84519
```

Exercise 12.4.5

Consider the following nonlinear system of equations:

*x*2 + 5*y* – 3*z*3= 15

4*x* + *y*2– *z* = 10

x + *y* + *z*= 15

Solve the nonlinear system with the `solve` function. Use the `double` function on your results to simplify the answer. Why can't you solve this system of equations using linear algebra techniques?

```
A = x^2 +5*y -3*z^3==15
```

A = $x^2 - 3\,z^3 + 5\,y = 15$

```
B = 4*x + y^2 -z == 10
```

B = $y^2 + 4\,x - z = 10$

```
C = x + y + z ==15
```

C = $x + y + z = 15$

```
[X,Y,Z]=solve(A,B,C)
```

X =

$$\begin{pmatrix} \frac{4225\,\mathrm{root}(\sigma_1, z_1, 1)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 1)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 1)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 1)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 1)}{27721} + \frac{191575}{27721} \\ \frac{4225\,\mathrm{root}(\sigma_1, z_1, 2)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 2)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 2)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 2)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 2)}{27721} + \frac{191575}{27721} \\ \frac{4225\,\mathrm{root}(\sigma_1, z_1, 3)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 3)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 3)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 3)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 3)}{27721} + \frac{191575}{27721} \\ \frac{4225\,\mathrm{root}(\sigma_1, z_1, 4)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 4)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 4)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 4)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 4)}{27721} + \frac{191575}{27721} \\ \frac{4225\,\mathrm{root}(\sigma_1, z_1, 5)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 5)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 5)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 5)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 5)}{27721} + \frac{191575}{27721} \\ \frac{4225\,\mathrm{root}(\sigma_1, z_1, 6)^3}{27721} - \frac{299\,\mathrm{root}(\sigma_1, z_1, 6)^2}{27721} + \frac{354\,\mathrm{root}(\sigma_1, z_1, 6)^4}{27721} + \frac{36\,\mathrm{root}(\sigma_1, z_1, 6)^5}{27721} - \frac{11876\,\mathrm{root}(\sigma_1, z_1, 6)}{27721} + \frac{191575}{27721} \end{pmatrix}$$

where

$$\sigma_1 = z_1^6 - \frac{2\,z_1^5}{3} + \frac{127\,z_1^4}{9} - \frac{770\,z_1^3}{9} + \frac{427\,z_1^2}{3} - 1225\,z_1 + \frac{34210}{9}$$

Y =

$$\begin{pmatrix} \frac{299\,\mathrm{root}(\sigma_1, z_1, 1)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 1)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 1)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 1)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 1)}{27721} + \frac{224240}{27721} \\ \frac{299\,\mathrm{root}(\sigma_1, z_1, 2)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 2)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 2)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 2)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 2)}{27721} + \frac{224240}{27721} \\ \frac{299\,\mathrm{root}(\sigma_1, z_1, 3)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 3)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 3)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 3)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 3)}{27721} + \frac{224240}{27721} \\ \frac{299\,\mathrm{root}(\sigma_1, z_1, 4)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 4)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 4)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 4)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 4)}{27721} + \frac{224240}{27721} \\ \frac{299\,\mathrm{root}(\sigma_1, z_1, 5)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 5)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 5)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 5)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 5)}{27721} + \frac{224240}{27721} \\ \frac{299\,\mathrm{root}(\sigma_1, z_1, 6)^2}{27721} - \frac{4225\,\mathrm{root}(\sigma_1, z_1, 6)^3}{27721} - \frac{354\,\mathrm{root}(\sigma_1, z_1, 6)^4}{27721} - \frac{36\,\mathrm{root}(\sigma_1, z_1, 6)^5}{27721} - \frac{15845\,\mathrm{root}(\sigma_1, z_1, 6)}{27721} + \frac{224240}{27721} \end{pmatrix}$$

where

$$\sigma_1 = z_1^6 - \frac{2\,z_1^5}{3} + \frac{127\,z_1^4}{9} - \frac{770\,z_1^3}{9} + \frac{427\,z_1^2}{3} - 1225\,z_1 + \frac{34210}{9}$$

Z =

$$\begin{pmatrix} \mathrm{root}(\sigma_1, z_1, 1) \\ \mathrm{root}(\sigma_1, z_1, 2) \\ \mathrm{root}(\sigma_1, z_1, 3) \\ \mathrm{root}(\sigma_1, z_1, 4) \\ \mathrm{root}(\sigma_1, z_1, 5) \\ \mathrm{root}(\sigma_1, z_1, 6) \end{pmatrix}$$

where

$$\sigma_1 = {z_1}^6 - \frac{2\,{z_1}^5}{3} + \frac{127\,{z_1}^4}{9} - \frac{770\,{z_1}^3}{9} + \frac{427\,{z_1}^2}{3} - 1225\,z_1 + \frac{34210}{9}$$

```
double(X)
```

```
ans =
      16.889 +     4.2178i
      16.889 -     4.2178i
       11.56 +     11.183i
       11.56 -     11.183i
      10.217 -     4.7227i
      10.217 +     4.7227i
```

```
double(Y)
```

```
ans =
     0.84987 -     7.8114i
     0.84987 +     7.8114i
      3.5094 -     6.9733i
      3.5094 +     6.9733i
      1.6407 +     5.5153i
      1.6407 -     5.5153i
```

```
double(Z)
```

```
ans =
      -2.739 +     3.5936i
      -2.739 -     3.5936i
   -0.069692 -     4.2102i
   -0.069692 +     4.2102i
       3.142 -    0.79261i
       3.142 +    0.79261i
```

Practice Exercise 12.5

You need the variables defined in Practice Exercise 12.1 to do these problems

Exercise 12.5.1

Using the `subs` function, substitute `4` into each expression/equation defined in Practice Exercises 12.1 for `x` . Comment on your results.

```
eq1
```

eq1 $= x^2 = 1$

```
subs(eq1,x,4)
```

ans $= 16 = 1$

```
ex1
```

ex1 $= x^2 - 1$

```
subs(ex1,x,4)
```

ans $= 15$

```
eq2
```

eq2 $= (x + 1)^2 = 0$

```
subs(eq2,x,4)
```

ans $= 25 = 0$

```
ex2
```

ex2 $= (x + 1)^2$

```
subs(ex2,x,4)
```

ans $= 25$

```
eq3
```

eq3 $= a\,x^2 = 1$

```
subs(eq3,x,4)
```

ans $=16\,a = 1$

```
ex3
```

ex3 $=a\,x^2 - 1$

```
subs(ex3,x,4)
```

ans $=16\,a - 1$

```
eq4
```

eq4 $=a\,x^2 + b\,x + c = 0$

```
subs(eq4,x,4)
```

ans $=16\,a + 4\,b + c = 0$

```
ex4
```

ex4 $=a\,x^2 + b\,x + c$

```
subs(ex4,x,4)
```

ans $=16\,a + 4\,b + c$

```
eq5
```

eq5 $=a\,x^3 + b\,x^2 + c\,x + d = 0$

```
subs(eq5,x,4)
```

ans $=64\,a + 16\,b + 4\,c + d = 0$

```
ex5
```

ex5 $=a\,x^3 + b\,x^2 + c\,x + d$

```
subs(ex5,x,4)
```

ans $=64\,a + 16\,b + 4\,c + d$

```
eq6
```

eq6 $=\sin(x) = 0$

```
subs(eq6,x,4)
```

ans = $\sin(4) = 0$

```
ex6
```

ex6 = $\sin(x)$

```
subs(ex6,x,4)
```

ans = $\sin(4)$

Exercise 12.5.2

Create a symbolic function called `g` for the following:

`x^2+sin(x)*x`

and use it to evaluate `g('a')`, `g(3)`, and `g([1:5])`.

```
syms x
g(x) = x^2 + sin(x)*x
```

g(x) = $x\sin(x) + x^2$

```
g('a')
```

ans = $a\sin(a) + a^2$

```
g(3)
```

ans = $3\sin(3) + 9$

```
g([1:5])
```

ans = $\left(\sin(1) + 1 \quad 2\sin(2) + 4 \quad 3\sin(3) + 9 \quad 4\sin(4) + 16 \quad 5\sin(5) + 25\right)$

Convert to a double to make interpreting the results easier

```
double(g(1:5))
```

```
ans =         1.8415         5.8186         9.4234         12.973         20.205
```

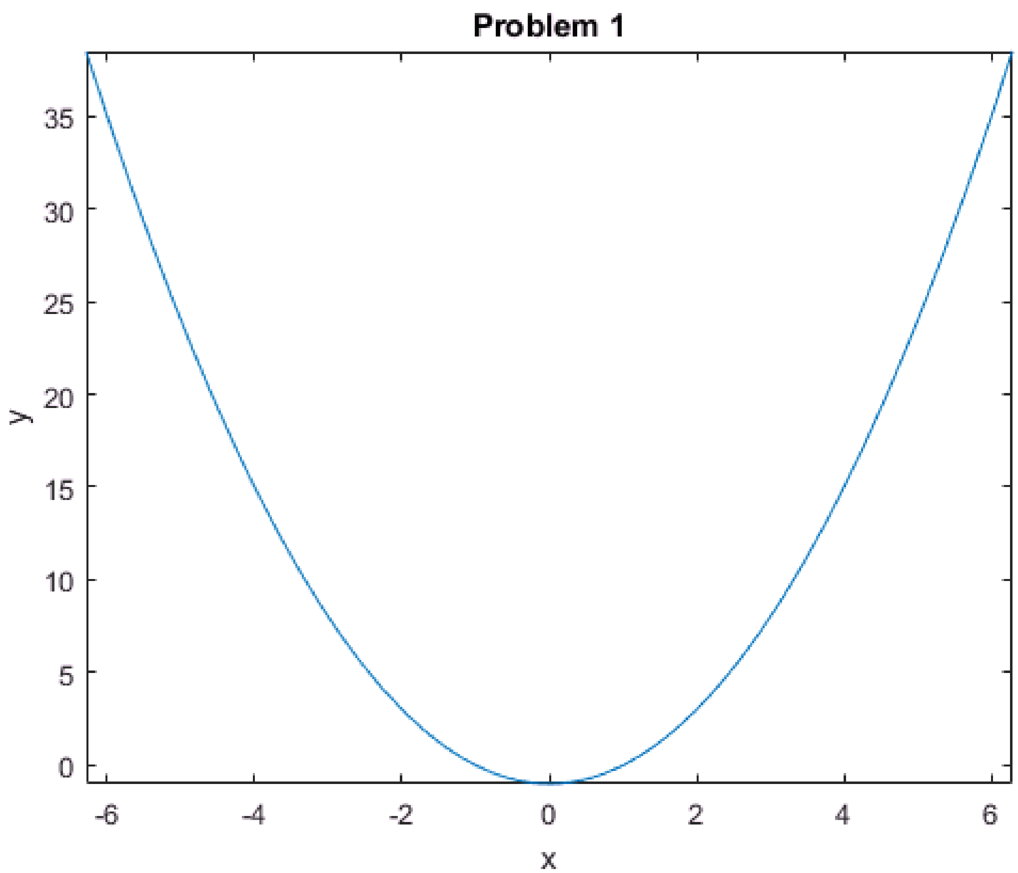
Practice Exercise 12.6

Be sure to add titles and axis labels to all your plots.

Exercise 12.6.1

Use `fplot` to plot `ex1` from -2π to $+2\pi$.

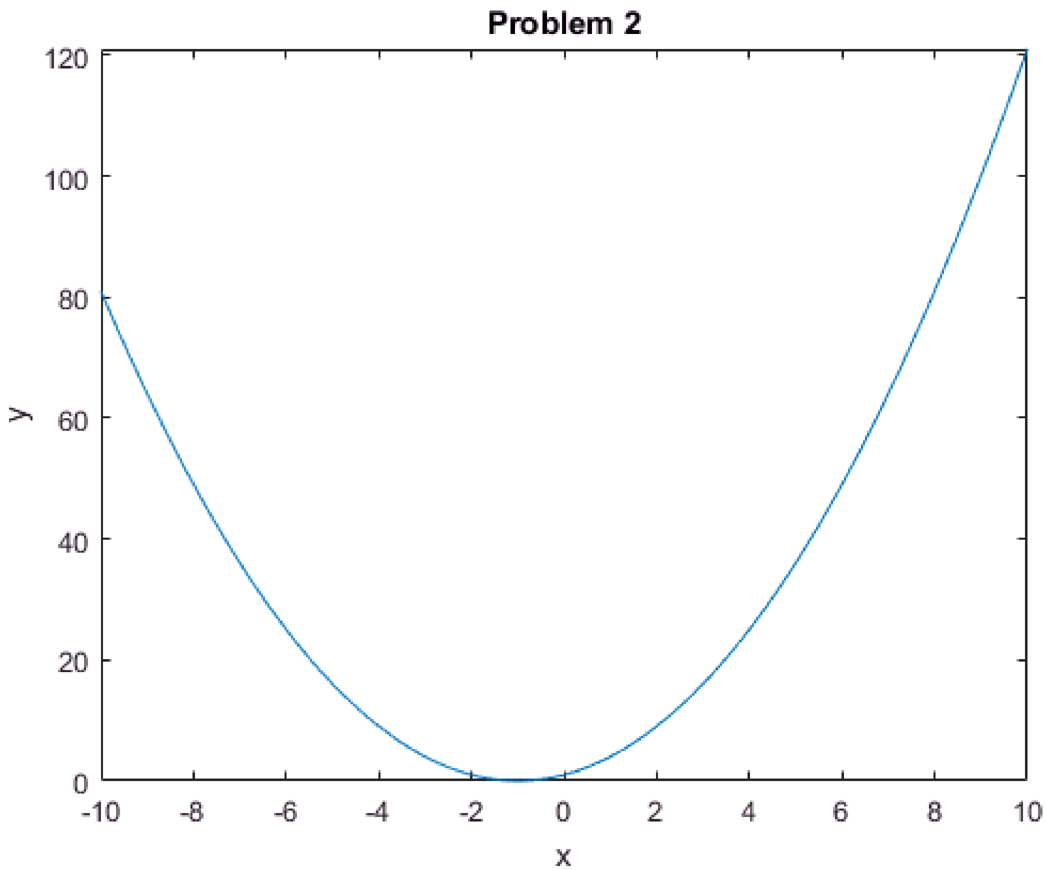
```
figure(1)
fplot(ex1,[-2*pi,2*pi])
title('Problem 1'),xlabel('x'),ylabel('y')
```



Exercise 12.6.2

Use `fplot` to plot `ex2` from −10 to +10.

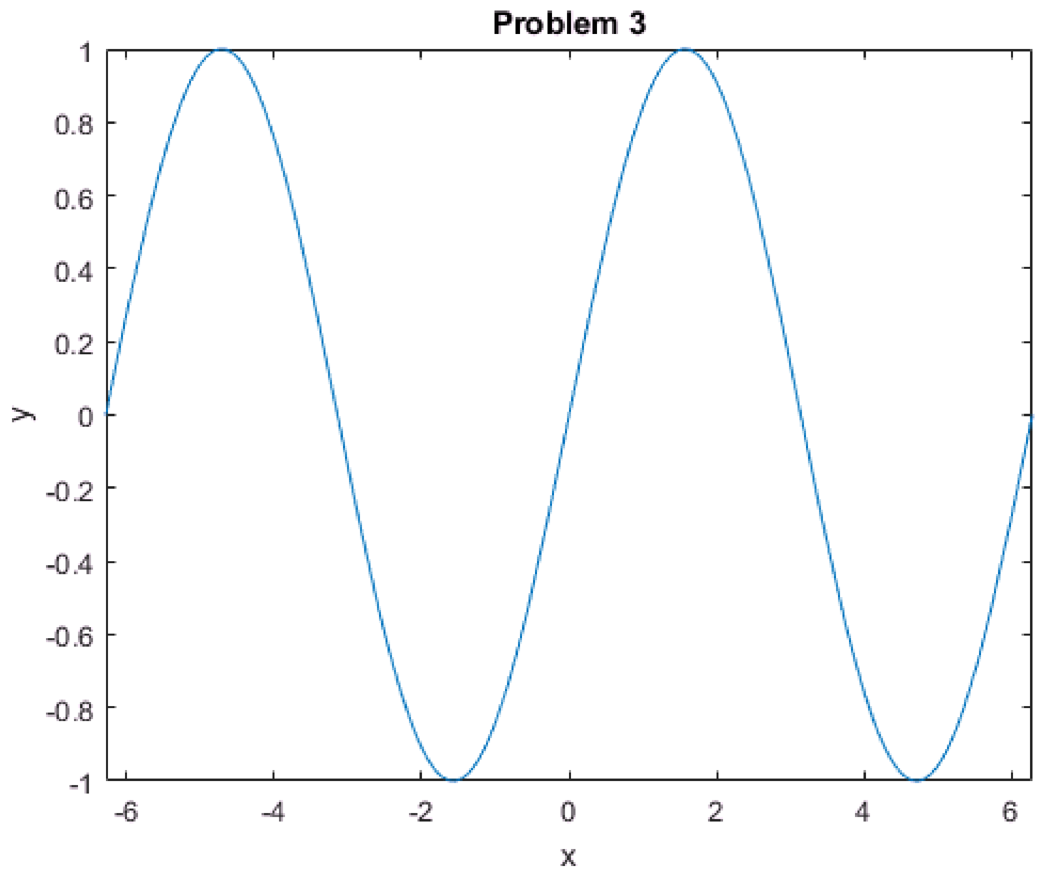
```
figure(2)
fplot(ex2,[-10,10])
title('Problem 2'),xlabel('x'),ylabel('y')
```



Exercise 12.6.3

Use `fplot` to plot `ex6` from -2π to $+2\pi$.

```
figure(3)
fplot(ex6,[-2*pi,2*pi])
title('Problem 3'),xlabel('x'),ylabel('y')
```

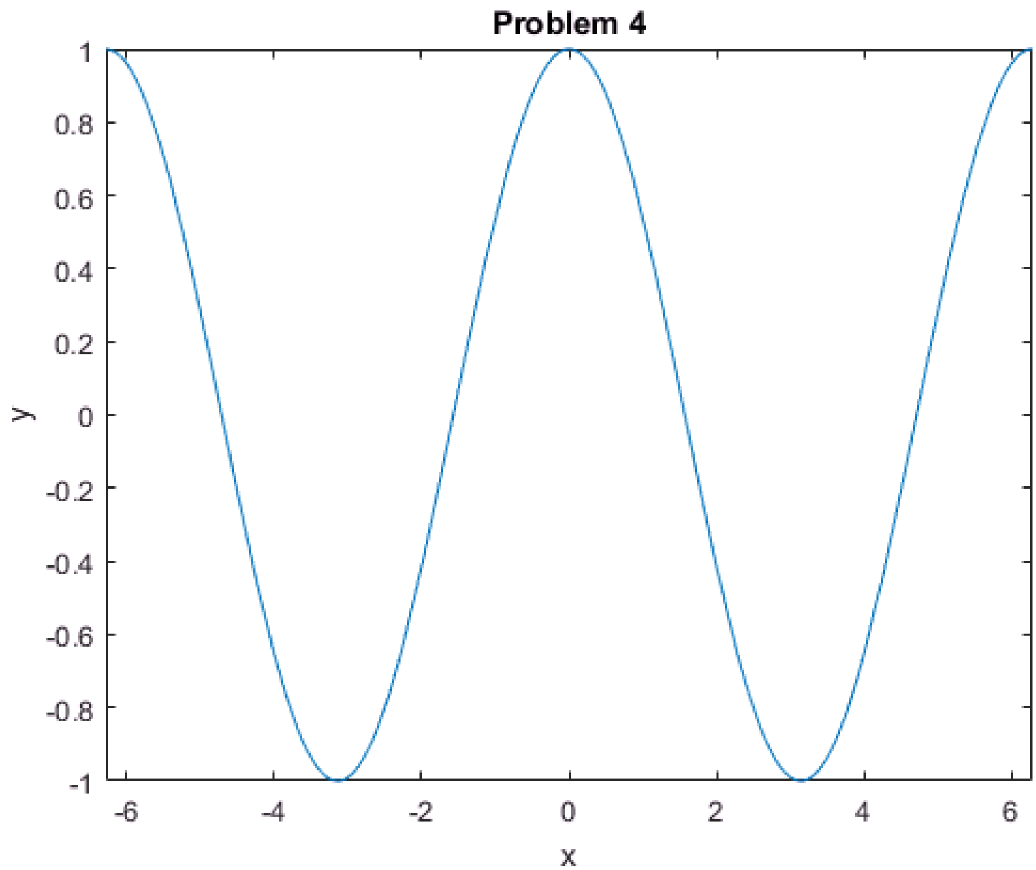


Exercise 12.6.4

Use `fplot` to plot cos(x) from -2π to $+2\pi$. Don't define an expression for cos(x); just enter it directly into `fplot` .

`fplot(@(x) cos(x))`

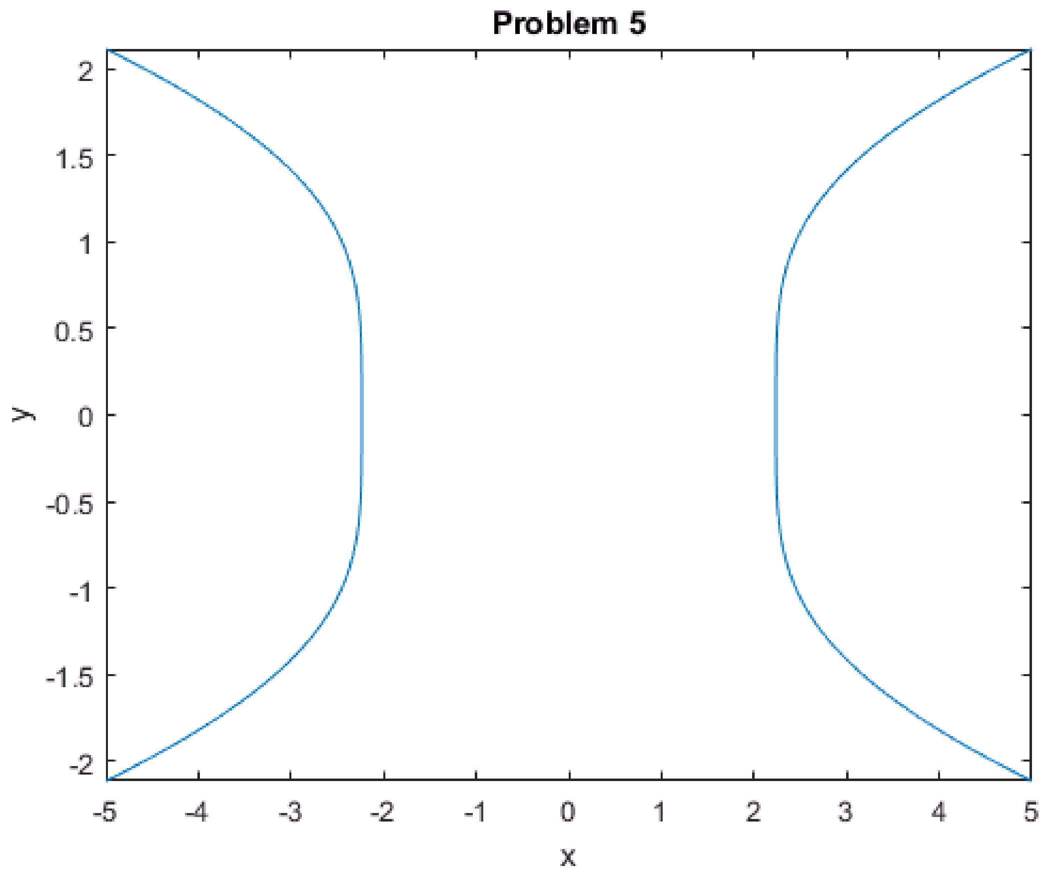
```
figure(4)
fplot(@(x)cos(x),[-2*pi,2*pi])
title('Problem 4'),xlabel('x'),ylabel('y')
```



Exercise 12.6.5

Use `fimplicit` to create an implicit plot of `x^2 - y^4 = 5`.

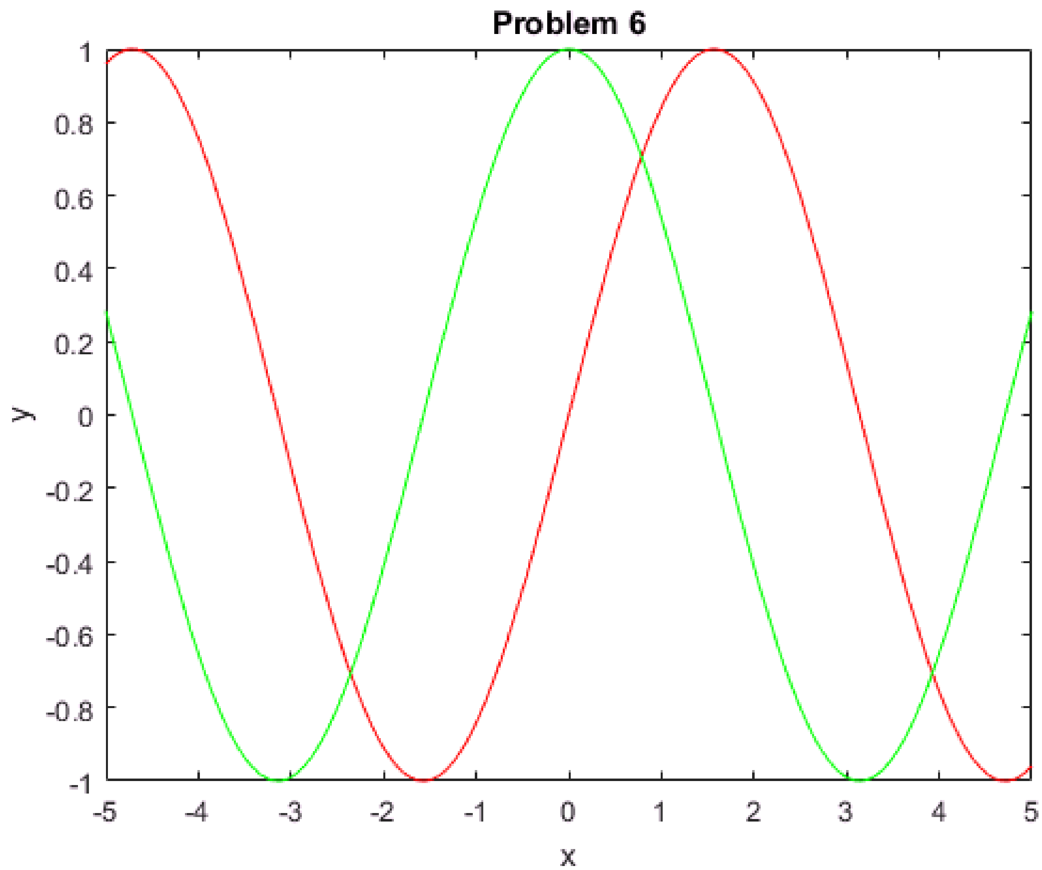
```
figure(5)
syms x y
fimplicit(x^2-y^4==5)
title('Problem 5'),xlabel('x'),ylabel('y')
```



Exercise 12.6.6

Use `fplot` to plot sin(*x*) and cos(*x*) on the same graph. Use the annotation tools to change the color of the sine graph from the default.

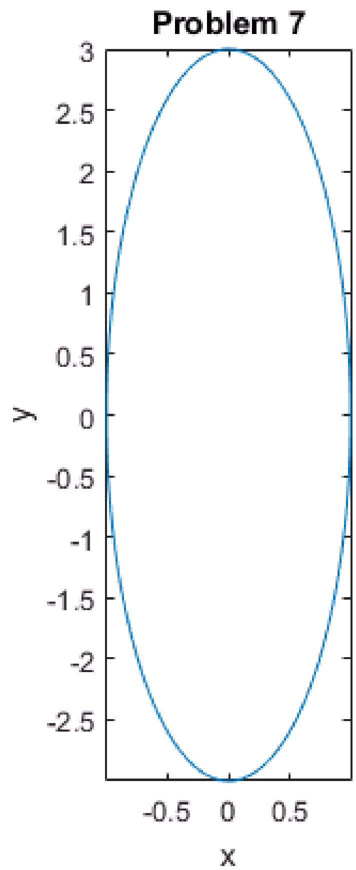
```
figure(6)
fplot(@(x) sin(x),'-r')
hold on
fplot(@(x) cos(x),'-g')
hold off
title('Problem 6'),xlabel('x'),ylabel('y')
```



Exercise 12.6.7

Use `fplot` to create a parametric plot of *x* = sin(*t*) and *y* = 3 cos(*t*).

```
figure(7)
fplot(@(t) sin(t), @(t) 3*cos(t))
axis equal
title('Problem 7'),xlabel('x'),ylabel('y')
```



Problem 7
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
y
-0.5
0
0.5
x

Practice Exercise 12.7

Create a symbolic expression for $Z = \sin\left(\sqrt{X^2 + Y^2}\right)$

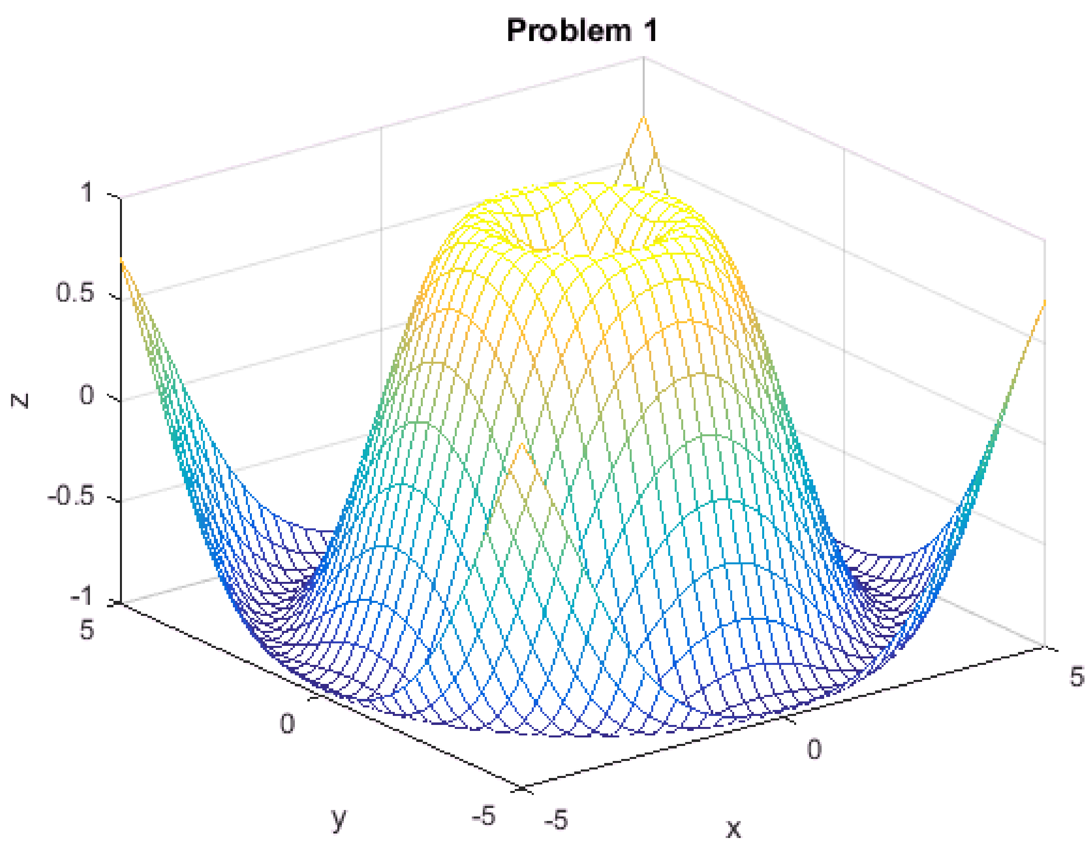
```
syms X Y
Z=sin(sqrt(X^2+Y^2))
```

Z $=\sin\left(\sqrt{X^2 + Y^2}\right)$

Exercise 12.7.1

Use `fmesh` to create a mesh plot of `Z`. Be sure to add a title and axis labels.

```
figure(1)
fmesh(Z)
title('Problem 1')
xlabel('x'), ylabel('y'),zlabel('z')
```

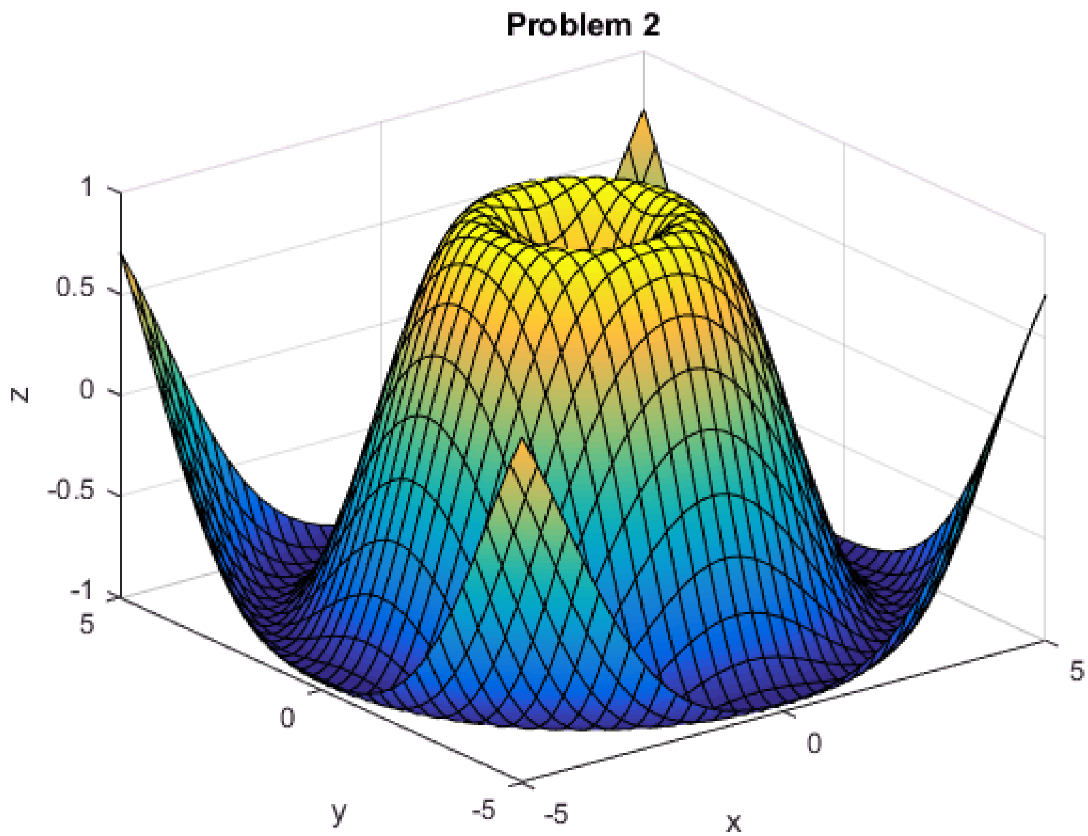


Exercise 12.7.2

Use `fsurf` to create a surface plot of `Z`. Be sure to add a title and axis labels.

```
figure(2)
```

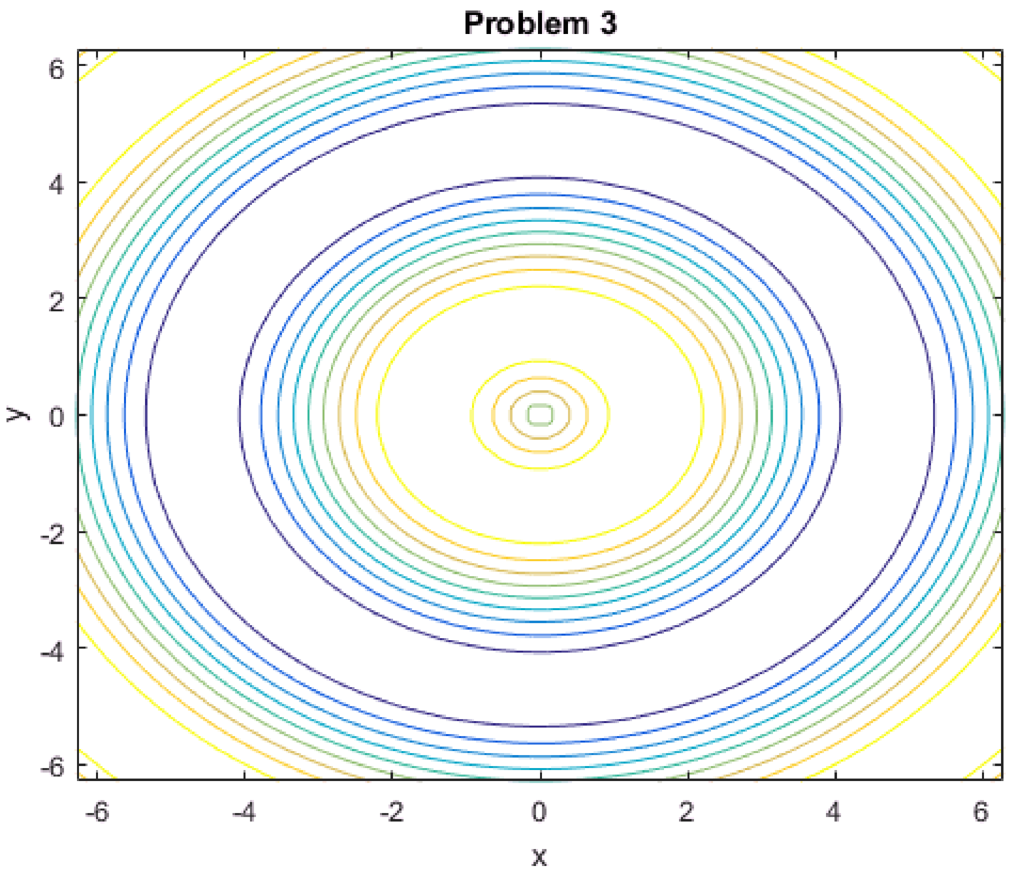
```
fsurf(Z)
title('Problem 2')
xlabel('x'), ylabel('y'), zlabel('z')
```



Exercise 12.7.3

Use `fcontour` to create a contour plot of `Z`. Be sure to add a title and axis labels.

```
figure(3)
ezcontour(Z)
title('Problem 3')
xlabel('x'), ylabel('y'), zlabel('z')
```



Exercise 12.7.4

The `fplot3` function requires us to define three variables as a function of a fourth. To do this, first define *t* as a symbolic variable, and then let

$x = t$

$y = \sin(t)$

$z = \cos(t)$

Use `fplot3` to plot this parametric function from 0 to 30.

```
syms t
x = t
```

x $= t$

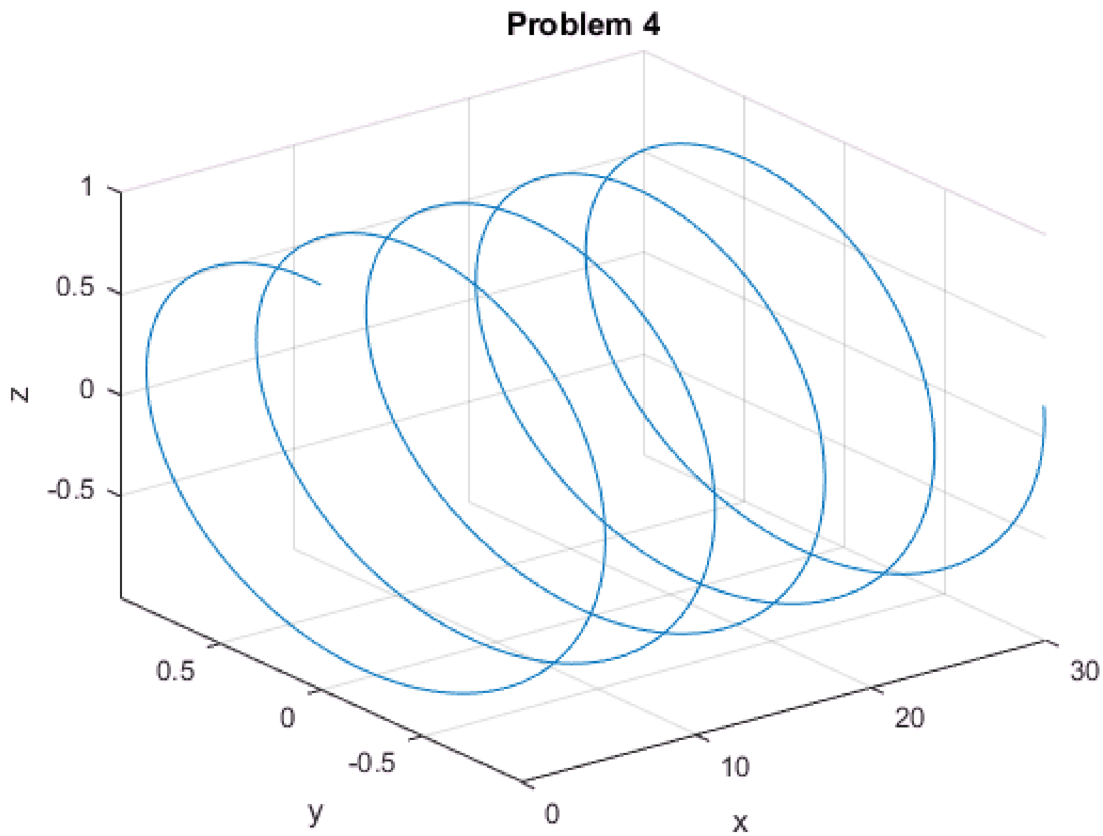
```
y = sin(t)
```

y $= \sin(t)$

```
z = cos(t)
```

z $= \cos(t)$

```
figure(4)
fplot3(x,y,z,[0,30])
title('Problem 4')
xlabel('x'), ylabel('y'), zlabel('z')
```



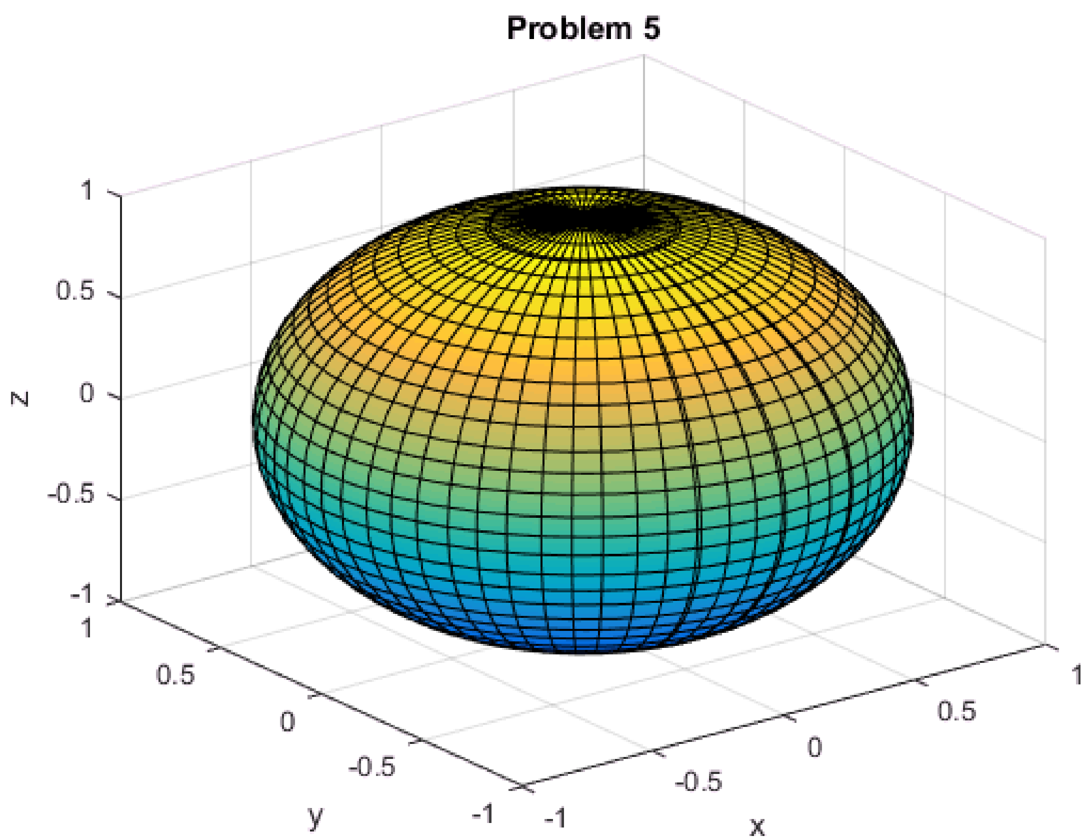
Exercise 12.7.5

A sphere can be defined parametrically as:

$$x = \cos(\text{phi}) * \cos(\text{theta})$$
$$y = \cos(\text{phi}) * \sin(\text{theta})$$
$$z = \sin(\text{phi})$$

for values of both phi and theta from -π to +π. Use fsurf create the corresponding plot.

```
figure(5)
syms phi theta
x = cos(phi)*cos(theta);
y = cos(phi)*sin(theta);
z = sin(phi);
fsurf(x,y,z)
title('Problem 5')
xlabel('x'), ylabel('y'), zlabel('z')
```


Problem 5
1
0.5
z
0
-0.5
-1
1
0.5
0
-0.5
y
-1
-1
-0.5
0
x
0.5
1

Practice Exercise 12.8

```
clear,clc
```

Exercise 12.8.1

Find the first derivative with respect to *x* of the following expressions:

x^2+x+1

sin (*x*)

tan (*x*)

ln (*x*)

```
syms x y a b c
diff(x^2+x+1)
```

ans $= 2x + 1$

```
diff(sin(x))
```

ans $= \cos(x)$

```
diff(tan(x))
```

ans $= \tan(x)^2 + 1$

```
diff(log(x))
```

ans =

$$\frac{1}{x}$$

Exercise 12.8.2

Find the first partial derivative with respect to *x* of the following expressions:

ax^2 + bx + c

$x^{0.5}$ – 3 y

tan (*x* + *y*)

3*x* + 4*y* – 3*xy*

```
diff(a*x^2 + b*x + c)
```

ans $= b + 2ax$

```
diff(x^0.5 - 3*y)
```

ans =

$$\frac{1}{2\ \sqrt{x}}$$

```
diff(tan(x+y))
```

ans $= \tan(x+y)^2 + 1$

```
diff(3*x + 4*y - 3*x*y)
```

ans $= 3 - 3\ y$

Exercise 12.8.3

Find the second derivative with respect to *x* for each of the expressions in Exercises 12.8.1 and 12.8.2.

There are two different approaches - using diff twice or specifying the order of the derivative. You may also want to specify that the derivative is with respect to x

```
diff(diff(a*x^2 + b*x + c))
```

ans $= 2\ a$

```
diff(x^0.5 - 3*y,2)
```

ans =

$$-\frac{1}{4\ x^{3/2}}$$

```
diff(tan(x+y),x,2)
```

ans $= 2\tan(x+y)\ \left(\tan(x+y)^2 + 1\right)$

```
diff(diff(3*x + 4*y - 3*x*y, x))
```

ans $= -3$

Exercise 12.8.4

Find the first derivative with respect to *y* for the following expressions:

y^2-1

2 $y + 3x^2$

$ay + bx + cz$

```
diff(y^2-1,y)
```

ans = $2y$

```
% or , since there is only one variable
diff(y^2-1)
```

ans = $2y$

```
%
diff(2*y + 3*x^2,y)
```

ans = 2

```
diff(a*y + b*x + c*x,y)
```

ans = a

Exercise 12.8.5

Find the second derivative with respect to *y* for each of the expressions in Problem 12.8.4.

```
diff(y^2-1,y,2)
```

ans = 2

```
% or , since there is only one variable
diff(y^2-1,2)
```

ans = 2

```
%
diff(diff(2*y + 3*x^2,y),y)
```

ans = 0

```
diff(a*y + b*x + c*x,y,2)
```

ans = 0

Practice Exercise 12.9

```
clear,clc
syms x y z a b c
```

Exercise 12.9.1

Integrate the following expressions with respect to x:

x^2+x+1

sin (x)

tan (x)

ln (x)

```
int(x^2 + x + 1)
```

ans =

$$\frac{x\ (2\,x^2 + 3\,x + 6)}{6}$$

```
int(sin(x))
```

ans $= -\cos(x)$

```
int(tan(x))
```

ans $= -\log(\cos(x))$

```
int(log(x))
```

ans $= x\ (\log(x) - 1)$

Exercise 12.9.2

$a\,x^2 + bx + c$

$x^{0.5} - 3y$

tan ($x + y$)

$3x + 4y - 3xy$

you don't need to specify that integration is with respect to x, because it is the default

```
int(a*x^2 + b*x + c)
```

ans =

$$\frac{a\,x^3}{3} + \frac{b\,x^2}{2} + c\,x$$

```
int(x^0.5 - 3*y)
```

ans =

$$\frac{2\,x^{3/2}}{3} - 3\,x\,y$$

```
int(tan(x+y))
```

ans = $-\log\left(\cos\left(x + y\right)\right)$

```
int(3*x + 4*y -3*x*y)
```

ans =

$$4\,x\,y - x^2\,\left(\frac{3\,y}{2} - \frac{3}{2}\right)$$

Exercise 12.9.3

Perform a double integration with respect to *x* for each of the expressions in Exercises 1 and 2.

```
int(int(x^2 + x + 1))
```

ans =

$$\frac{x^2\,\left(x^2 + 2\,x + 6\right)}{12}$$

```
int(int(sin(x)))
```

ans = $-\sin\left(x\right)$

```
int(int(tan(x)))
```

ans =

$$-\frac{x\,\left(x + 2\log\left(\mathrm{e}^{2\,x\,\mathrm{i}} + 1\right)\,\mathrm{i}\right)\,\mathrm{i}}{2} - x\log\left(\cos\left(x\right)\right) - \frac{\mathrm{polylog}\left(2, -\mathrm{e}^{2\,x\,\mathrm{i}}\right)\,\mathrm{i}}{2}$$

```
int(int(log(x)))
```

ans =

$$\frac{x^2\,\left(2\log\left(x\right) - 3\right)}{4}$$

```
%
int(int(a*x^2 + b*x + c))
```

ans =

$$\frac{x^2\ \left(a\,x^2+2\,b\,x+6\,c\right)}{12}$$

```
int(int(x^0.5 - 3*y))
```

ans =

$$\frac{4\,x^{5/2}}{15}-\frac{3\,x^2\,y}{2}$$

```
int(int(tan(x+y)))
```

ans =

$$-\frac{\left(x+y\right)\ \left(x+y+2\log\left(\mathrm{e}^{2\,x\,\mathrm{i}+2\,y\,\mathrm{i}}+1\right)\mathrm{i}\right)\mathrm{i}}{2}-\log\left(\cos\left(x+y\right)\right)\ \left(x+y\right)-\frac{\mathrm{polylog}\left(2,-\mathrm{e}^{2\,x\,\mathrm{i}+2\,y\,\mathrm{i}}\right)\mathrm{i}}{2}$$

```
int(int(3*x + 4*y -3*x*y))
```

ans =

$$2\,x^2\,y-x^3\ \left(\frac{y}{2}-\frac{1}{2}\right)$$

Exercise 12.9.4

Integrate the following expressions with respect to *y*:

$y^2 - 1$

$2y + 3x^2$

$ay + bx + cz$

```
int(y^2-1)
```

ans =

$$\frac{y\ \left(y^2-3\right)}{3}$$

```
int(2*y+3*x^2,y)
```

ans = $y\ \left(3\,x^2+y\right)$

```
int(a*y + b*x + c*z,y)
```

ans =

$$\frac{a\,y^2}{2}+\left(b\,x+c\,z\right)\,y$$

Exercise 12.9.5

Perform a double integration with respect to *y* for each of the expressions in Exercise 12.4.

```
int(int(y^2-1))
```

ans =

$$\frac{y^2\ \left(y^2 - 6\right)}{12}$$

```
int(int(2*y+3*x^2,y),y)
```

ans =

$$\frac{y^2\ \left(9\,x^2 + 2\,y\right)}{6}$$

```
int(int(a*y + b*x + c*z,y),y)
```

ans =

$$\frac{a\,y^3}{6} + \left(\frac{b\,x}{2} + \frac{c\,z}{2}\right)\,y^2$$

Exercise 12.9.6

Integrate each of the expressions in Exercise 1 with respect to *x* from 0 to 5.

```
int(x^2 + x + 1,0,5)
```

ans =

$$\frac{355}{6}$$

```
int(sin(x),0,5)
```

ans = $1 - \cos\left(5\right)$

```
int(tan(x),0,5)
```

ans = NaN

```
int(log(x),0,5)
```

ans = $5\,\log\left(5\right) - 5$

Practice Exercise 13.1

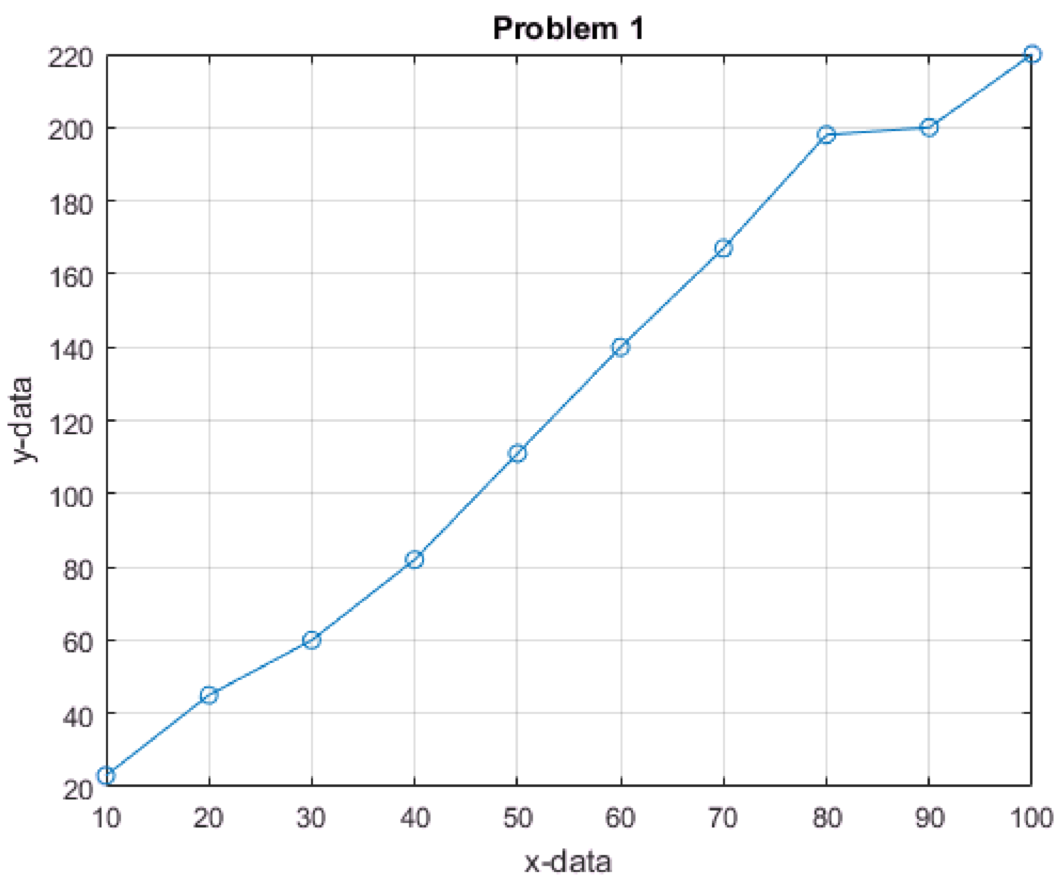
Create *x* and *y* vectors to represent the following data:

```
clear,clc
x=10:10:100;
y = [23 45 60 82 111 140 167 198 200 220];
```

Exercise 13.1.1

Plot the data on an *x*–*y* plot.

```
plot(x,y,'-o')
title('Problem 1')
xlabel('x-data'),ylabel('y-data'),grid on
```



Exercise 13.1.2

Use linear interpolation to approximate the value of *y* when *x* = 15.

```
interp1(x,y,15)
```

```
ans =      34
```

Exercise 13.1.3

Use cubic spline interpolation to approximate the value of *y* when *x* = 15.

```
interp1(x,y,15,'spline')
```

```
ans =        35.955
```

Note: the syntax is 'spline' NOT 'cubic', which is a common mistake

Exercise 13.1.4

Use linear interpolation to approximate the value of *x* when *y* = 80.

```
interp1(y,x,80)
```

```
ans =        39.091
```

Exercise 13.1.5

Use cubic spline interpolation to approximate the value of *x* when *y* = 80.

```
interp1(y,x,80,'spline')
```

```
ans =        39.224
```

Exercise 13.1.6

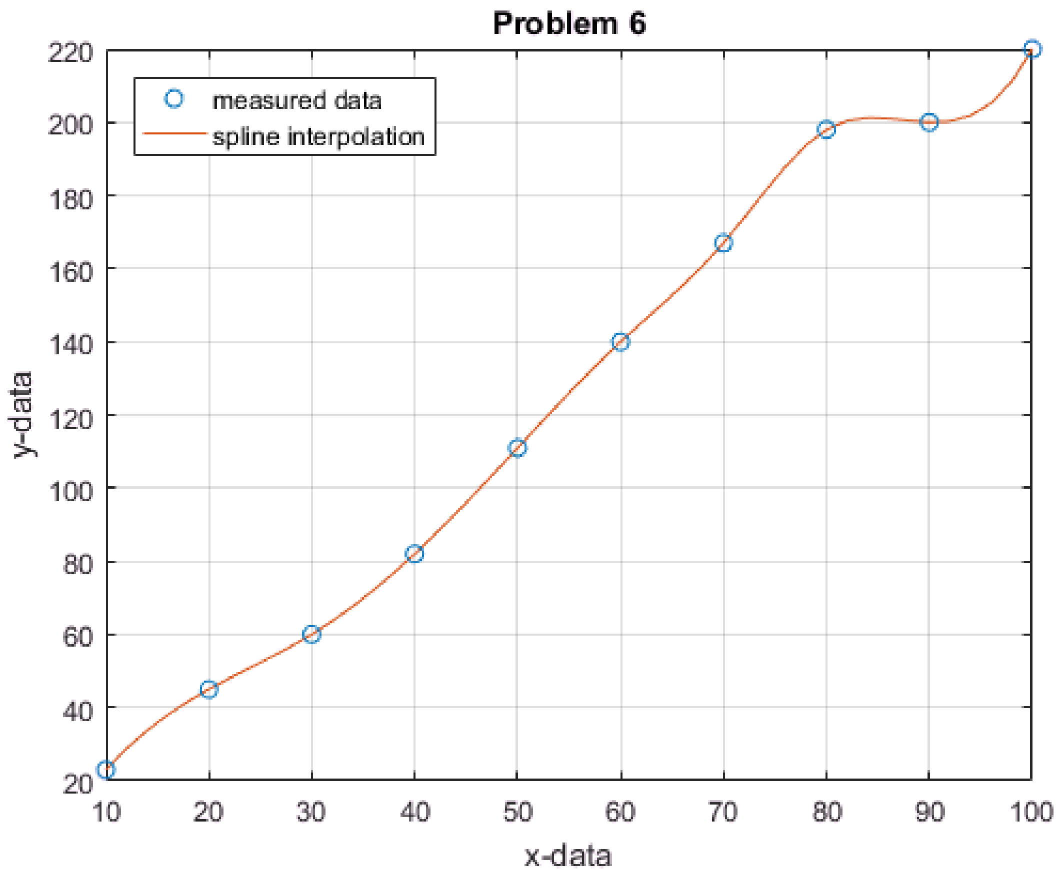
Use cubic spline interpolation to approximate *y*-values for *x*-values evenly spaced between 10 and 100 at intervals of 2.

```
new_x=10:2:100;
new_y=interp1(x,y,new_x,'spline');
```

Exercise 13.1.7

Plot the original data on an *x*–*y* plot as data points not connected by a line. Also, plot the values calculated in Exercise 6.

```
figure(2)
plot(x,y,'o',new_x,new_y)
legend('measured data','spline interpolation','Location','Northwest')
title('Problem 6'),xlabel('x-data'),ylabel('y-data'), grid on
```


Problem 6
measured data
spline interpolation
y-data
x-data
220
200
180
160
140
120
100
80
60
40
20
10
20
30
40
50
60
70
80
90
100

Practice Exercise 13.2

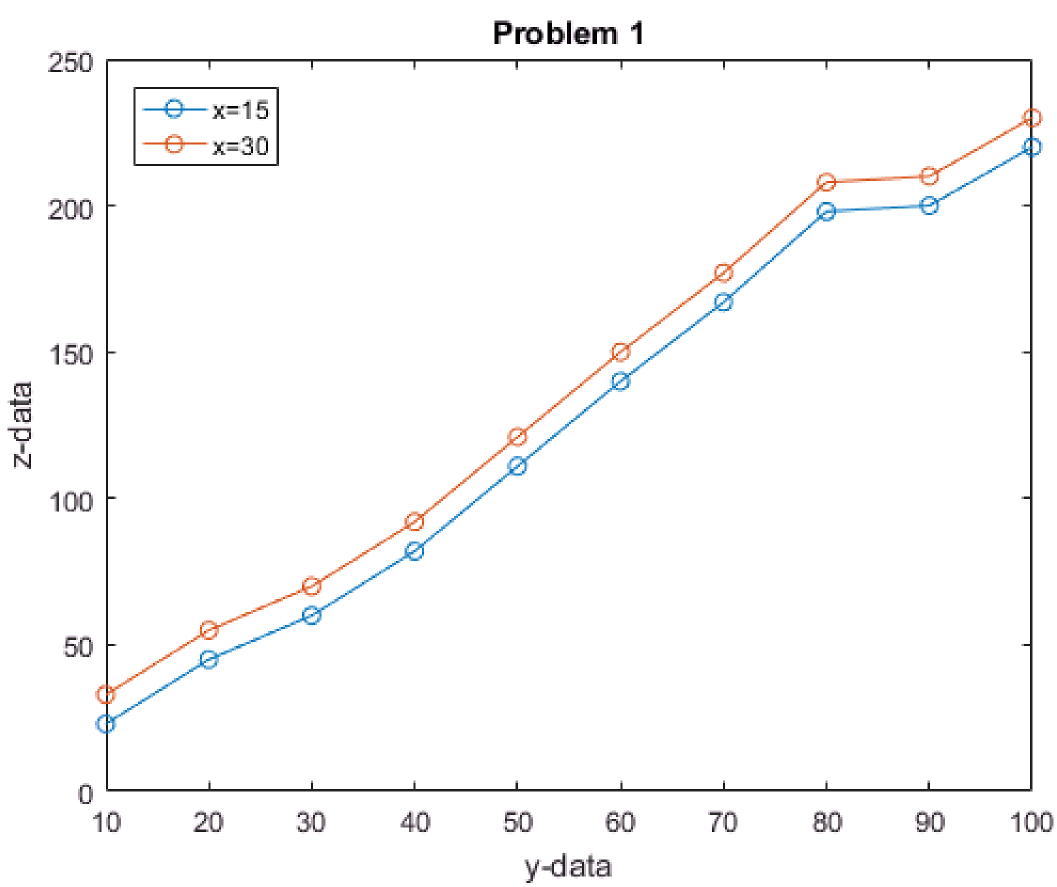
Create *x* and *y and z* vectors to represent the following data:

```
clear,clc
y=10:10:100';
x=[15, 30];
z=  [23        33
    45         55
    60         70
    82         92
    111        121
    140        150
    167        177
    198        208
    200        210
    220        230];
```

Exercise 13.2.1

Plot both sets of *y*–*z* data on the same plot. Add a legend identifying which value of *x* applies to each data set.

```
plot(y,z,'-o')
title('Problem 1'),xlabel('y-data'),ylabel('z-data')
legend('x=15','x=30','Location','Northwest')
```



Exercise 13.2.2

Use two-dimensional linear interpolation to approximate the value of *z* when *y* = 15 and

x = 20.

```
new_z=interp2(x,y,z,15,20)
```

```
new_z =     45
```

Exercise 13.2.3

Use two-dimensional cubic spline interpolation to approximate the value of *z* when *y* = 15 and *x* = 20.

```
new_z=interp2(x,y,z,15,20,'spline')
```

```
new_z =     45
```

Exercise 13.2.4

Use linear interpolation to create a new subtable for *x*= 20 and *x* = 25 for all the *y*-values.

```
new_z=interp2(x,y,z,[20,25],y')
```

```
new_z =
             26.333       29.667
             48.333       51.667
             63.333       66.667
             85.333       88.667
             114.33       117.67
             143.33       146.67
             170.33       173.67
             201.33       204.67
             203.33       206.67
             223.33       226.67
```

Practice Exercise 13.3

Create *x* and *y* vectors to represent the following data:

```
clear,clc
x=[10:10:100];
y=  [23 33
    45 55
    60 70
    82 92
    111 121
    140 150
    167 177
    198 198
    200 210
    220 230]';
```

Exercise 13.3.1

Use the `polyfit` function to fit the data for *z* = 15 to a first-order polynomial.

```
coef=polyfit(x,y(1,:),1)
```

```
coef =        2.3224      -3.1333
```

Exercise 13.3.2

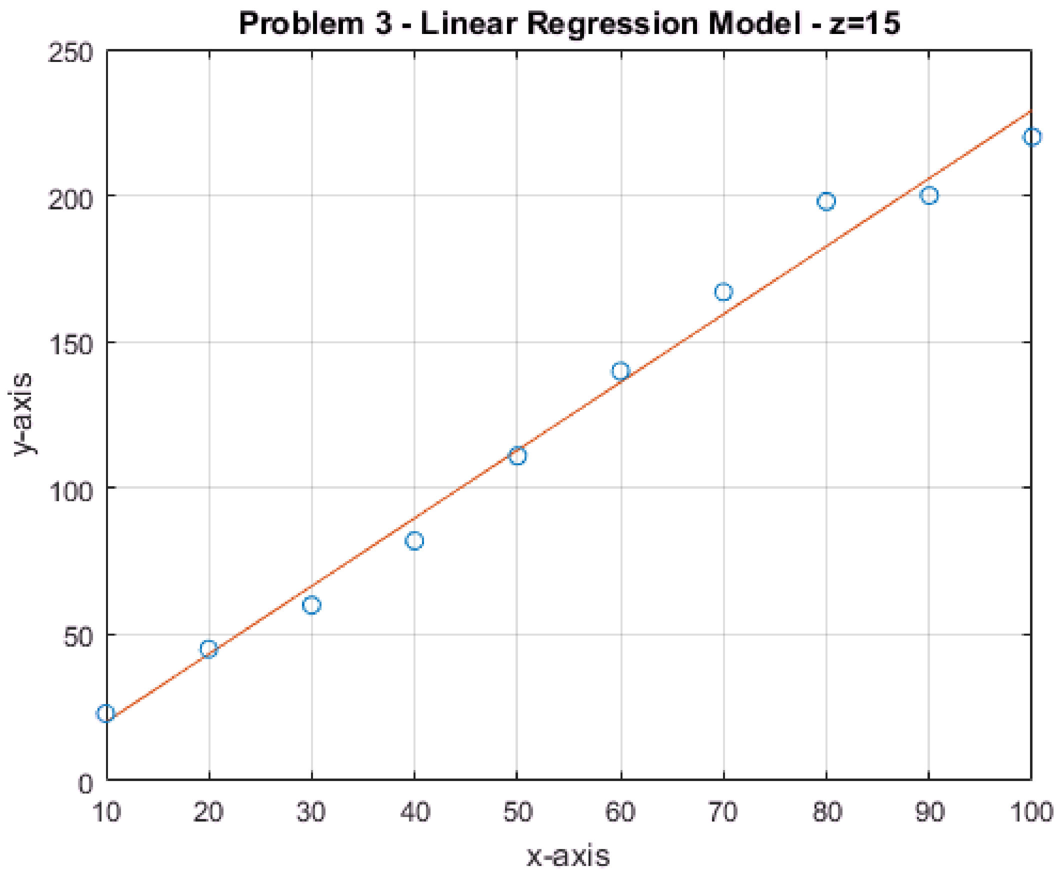
Create a vector of new *x* values from 10 to 100 in intervals of 2. Use your new vector in the `polyval` function together with the coefficient values found in Exercise 1 to create a new *y* vector.

```
new_x=10:2:100;
new_y=polyval(coef,new_x);
```

Exercise 13.3.3

Plot the original data as circles without a connecting line and the calculated data as a solid line on the same graph. How well do you think your model fits the data?

```
figure(1)
plot(x,y(1,:),'o',new_x,new_y)
title('Problem 3 - Linear Regression Model - z=15')
xlabel('x-axis'), ylabel('y-axis'), grid on
```



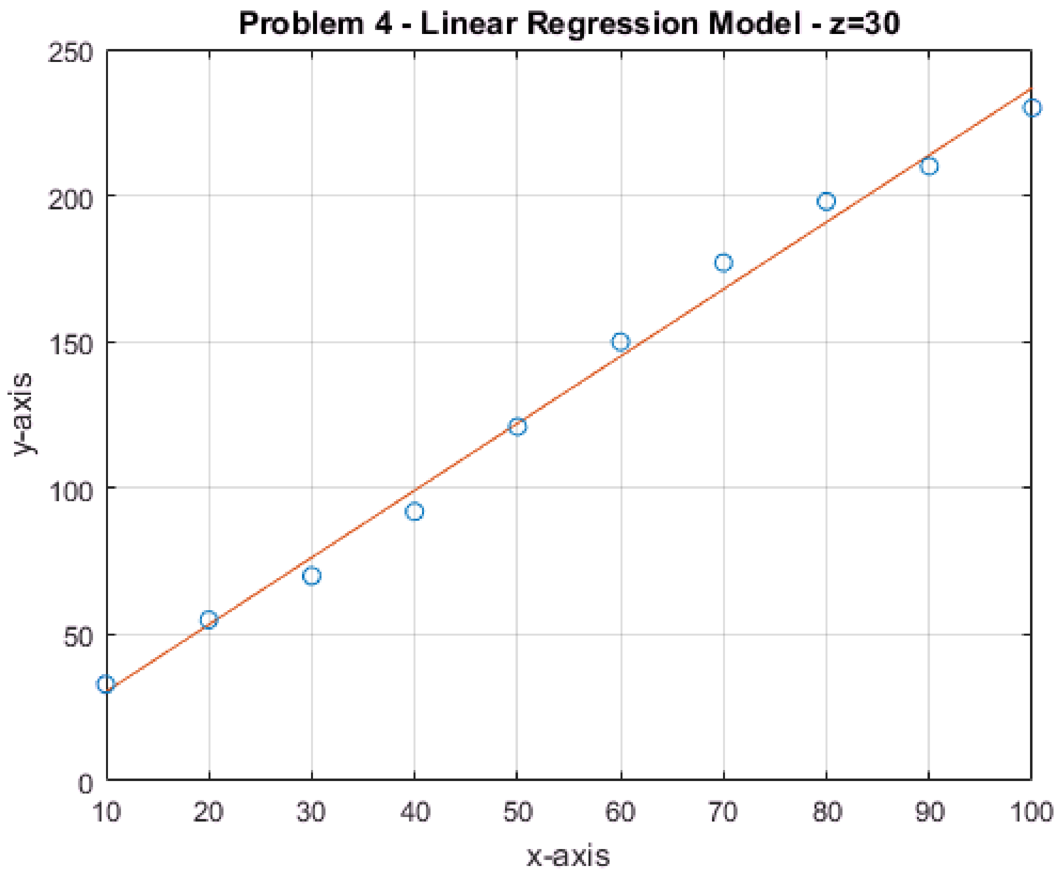
Exercise 13.3.4

Repeat Exercises 1 through 3 for the *x* and *y* data corresponding to *z* = 30.

```
figure(2)
coef2=polyfit(x,y(2,:),1)
```

```
coef2 =        2.2921       7.5333
```

```
new_y2=polyval(coef2,new_x);
plot(x,y(2,:),'o',new_x,new_y2)
title('Problem 4 - Linear Regression Model - z=30')
xlabel('x-axis'),ylabel('y-axis'), grid on
```


Problem 4 - Linear Regression Model - z=30
250
200
150
100
50
0
y-axis
10
20
30
40
50
60
70
80
90
100
x-axis

Practice Exercise 13.4

```
clear,clc
```

Exercise 13.4.1

Consider the following equation:

$$y = x^3 + 2x^2 - x + 3$$

Define an **x** vector from −5 to +5, and use it together with the `diff` function to approximate the derivative of *y* with respect to *x*, using the forward difference approach.

Found analytically, the derivative is

$$\frac{\text{dy}}{\text{dx}} = \text{y}' = 3x^2 + 4x - 1$$

Evaluate this function, using your previously defined **x** vector. How do your results differ?

```
x=-5:1:5;
y1=x.^3 + 2.*x.^2 - x + 3;
dy_dx1=diff(y1)./diff(x)
```

```
dy_dx1 =      42     22      8      0     -2      2     12     28     50     78
```

```
dy_dx_analytical1=3*x.^2 + 4*x -1
```

```
dy_dx_analytical1 =      54     31     14      3     -2     -1      6     19     38     63     94
```

```
result=[[dy_dx1,NaN]',dy_dx_analytical1']
```

```
result =
    42    54
    22    31
     8    14
     0     3
    -2    -2
     2    -1
    12     6
    28    19
    50    38
    78    63
   NaN    94
```

```
% We added NaN to the dy_dx vector so that the length of each vector would
% be the same
```

Exercise 13.4.2a

Repeat Exercise 1 for the following functions and their derivatives:

$$y = \sin(x)$$

$$\frac{\mathrm{dy}}{\mathrm{dx}} = \cos(x)$$

```
x=-5:1:5;
y2a=sin(x);
dy_dx2a=diff(y2a)./diff(x);
dy_dx_analytical2a=cos(x);
result=[[dy_dx2a,NaN]',dy_dx_analytical2a']
```

```
result =
    -0.20212      0.28366
    -0.89792     -0.65364
    -0.76818     -0.98999
    0.067826     -0.41615
     0.84147       0.5403
     0.84147            1
    0.067826       0.5403
    -0.76818     -0.41615
    -0.89792     -0.98999
    -0.20212     -0.65364
         NaN      0.28366
```

Exercise 13.4.2b

$$y = x^5 - 1$$

$$\frac{\mathrm{dy}}{\mathrm{dx}} = 5x^4$$

```
x=-5:1:5;
y2b=x.^5-1;
dy_dx2b=diff(y2b)./diff(x);
dy_dx_analytical2b=5*x.^4;
result=[[dy_dx2b,NaN]',dy_dx_analytical2b']
```

```
result =
        2101        3125
         781        1280
         211         405
          31          80
           1           5
           1           0
          31           5
         211          80
         781         405
        2101        1280
         NaN        3125
```

Exercise 13.4.2c

$$y = 5xe^x$$

$$\frac{\mathrm{dy}}{\mathrm{dx}} = 5e^x + 5xe^x$$

```
x=-5:1:5;
y2c=5*x.*exp(x);
dy_dx2c=diff(y2c)./diff(x);
```

```
dy_dx_analytical2c=5*exp(x) + 5*x.*exp(x);
result=[[dy_dx2c,NaN]',dy_dx_analytical2c']
```

```
result =
     -0.19786     -0.13476
     -0.38049     -0.27473
     -0.60655     -0.49787
     -0.48604     -0.67668
       1.8394            0
       13.591            5
       60.299       27.183
       227.39       110.84
       790.68       401.71
       2618.4         1365
          NaN       4452.4
```

Exercise 13.4.3

Use the `gradient` function to find the value of the derivatives in the previous problems.

Exercise 13.4.3.1

```
dy_dx31=gradient(y1)
```

```
dy_dx31 =     42    32    15     4    -1     0     7    20    39    64    78
```

Exercise 13.4.3a

```
dy_dx3a=gradient(y2a)
```

```
dy_dx3a =      -0.20212     -0.55002     -0.83305     -0.35018      0.45465      0.84147      0.45465     -0.35018     -0.83305     -0.55002     -0.20212
```

Exercise 13.4.3b

```
dy_dx3b=gradient(y2b)
```

```
dy_dx3b =         2101         1441          496          121           16            1           16          121          496         1441         2101
```

Exercise 13.4.3c

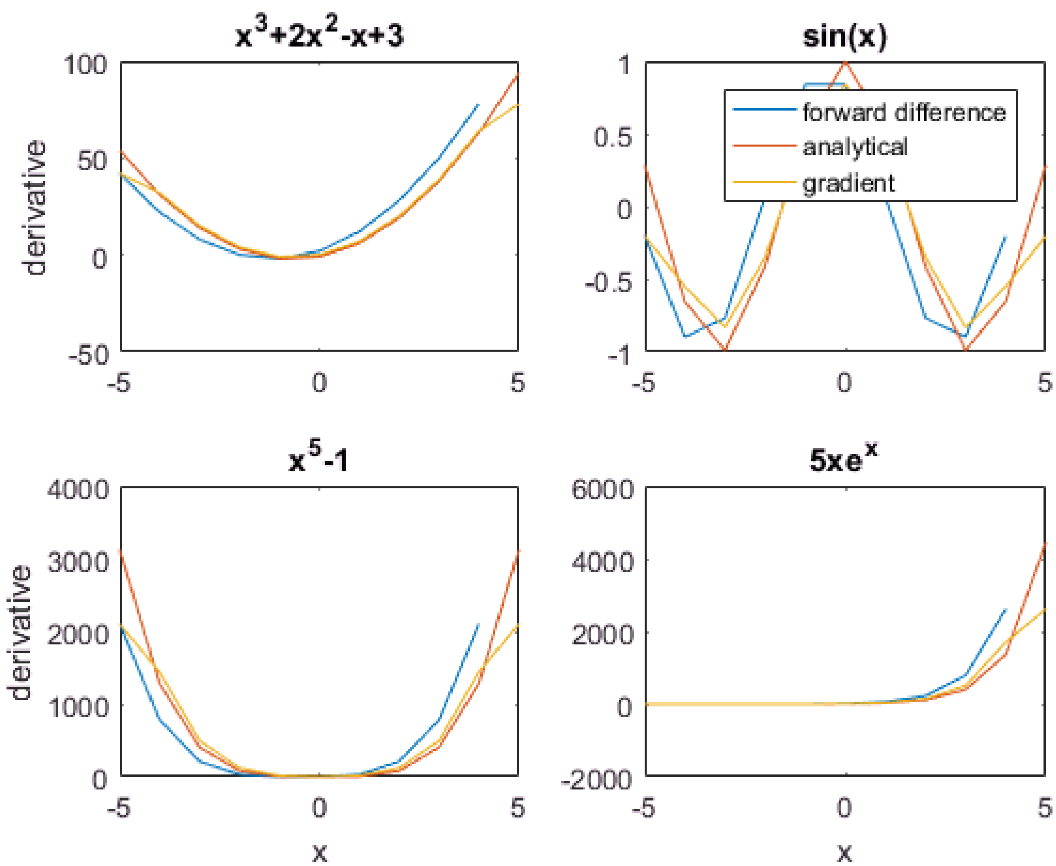
```
dy_dx3c=gradient(y2c)
```

```
dy_dx3c =      -0.19786     -0.28918     -0.49352      -0.5463      0.67668       7.7154       36.945       143.85       509.04       1704.5       2618.4
```

Exercise 13.4.4

Plot your results and compare the two approaches. Recall that the forward difference approach will provide one fewer values than the length of the *x* array. Be sure to pad the result array with a final value of `NaN` to make plotting easier.

```
subplot(2,2,1)
plot(x',[[dy_dx1,NaN]',dy_dx_analytical1',dy_dx31'])
title('x^3+2x^2-x+3')
ylabel('derivative')
subplot(2,2,2)
plot(x',[[dy_dx2a,NaN]',dy_dx_analytical2a',dy_dx3a'])
title('sin(x)')
legend('forward difference','analytical','gradient')
subplot(2,2,3)
plot(x',[[dy_dx2b,NaN]',dy_dx_analytical2b',dy_dx3b'])
title('x^5-1')
xlabel('x')
ylabel('derivative')
subplot(2,2,4)
plot(x',[[dy_dx2c,NaN]',dy_dx_analytical2c',dy_dx3c'])
title('5xe^x')
xlabel('x')
```



Practice Exercise 13.5

```
clear,clc
```

Exercise 13.5.1

Consider the following equation:

$$y = x + 2x^2 - x + 3$$

(a) Use the `trapz` function to estimate the integral of *y* with respect to *x*, evaluated from –1 to 1. Use 11 values of *x*, and calculate the corresponding values of *y* as input to the `trapz` function.

```
x=linspace(-1,1,11);
y = x.^3 + 2*x.^2 - x + 3;
trapz(x,y)
```

```
ans =          7.36
```

(b) Use the `quad` and `quadl` functions to find the integral of *y* with respect to *x*, evaluated from –1 to 1.

```
quad('x.^3+2*x.^2 -x + 3',-1,1)
```

```
ans =        7.3333
```

```
quadl('x.^3+2*x.^2 -x + 3',-1,1)
```

```
ans =        7.3333
```

(c) Compare your results with the values found by using the symbolic toolbox function `int` and the following analytical solution (remember that the `quad` and `quadl` functions take input expressed with array operators such as **.*** or **.^**, but that the `int` function takes a symbolic representation that does not use these operators):

Use X instead of x to keep symbolic and numeric variables separate

$$\int_a^b (x^3 + 2x^2 - x + 3)\,dx =$$
$$\left(\frac{x^4}{4} + \frac{2x^3}{3} - \frac{x^2}{2} + 3x\right)\Bigg|_a^b =$$
$$\frac{1}{4}(b^4 - a^4) + \frac{2}{3}(b^3 - a^3) - \frac{1}{2}(b^2 - a^2) + 3(b - a)$$

```
syms X
double(int(X^3+2*X^2 -X + 3,-1,1))
```

```
ans =        7.3333
```

Analytical Solution

```
a=-1;
b=1;
1/4*(b^4-a^4)+2/3*(b^3-a^3)-1/2*(b^2-a^2)+3*(b-a)
```

```
ans =        7.3333
```

Exercise 13.5.2

Repeat Exercise 1 for the following functions:

Exercise 13.5.2.1

function $y = \sin(x)$

integral $\int_a^b \sin(x)\mathrm{dx} = \cos(x)\,|_a^b = \cos(b) - \cos(a)$

a

```
y = sin(x);
trapz(x,y)
```

```
ans =    2.7756e-17
```

b

```
quad('sin(x)',-1,1)
```

```
ans =      0
```

```
quadl('sin(x)',-1,1)
```

```
ans =      0
```

c Symbolic Solution

```
double(int(sin(X),-1,1))
```

```
ans =      0
```

Analytical Solution

```
a=-1;
b=1;
cos(b)-cos(a)
```

```
ans =      0
```

Exercise 13.5.2.2

function $y = x^5 - 1$

integral $\int_a^b \left(x^5 - 1\right)\mathrm{dx} = \left(\frac{x^6}{6} - x\right)\Big|_a^b = \left(\frac{b^6 - a^6}{6} - \left(b - a\right)\right)$

a

```
y= x.^5-1;
trapz(x,y)
```

```
ans =     -2
```

b

```
quad('x.^5-1',-1,1)
```

```
ans =     -2
```

```
quadl('x.^5-1',-1,1)
```

```
ans =     -2
```

c Symbolic Solution

```
double(int(X^5-1,-1,1))
```

```
ans =     -2
```

Analytical Solution

```
a=-1;
b=1;
(b^6-a^6)/6-(b-a)
```

```
ans =     -2
```

Exercise 13.5.2.3

function $y = 5x * e^x$

integral $\int_a^b 5xe^x \mathrm{dx} = \left(-5e^x + 5xe^x\right)\Big|_a^b = \left(-5\left(e^b - e^a\right) + 5\left(be^b - ae^a\right)\right)$

a

```
y = 5*x.*exp(x);
trapz(x,y)
```

```
ans =         3.7693
```

b

```
quad('5*x.*exp(x)',-1,1)
```

```
ans =         3.6788
```

```
quadl('5*x.*exp(x)',-1,1)
```

```
ans =         3.6788
```

c Symbolic Solution

```
double(int(5*X*exp(X),-1,1))
```

```
ans =         3.6788
```

Analytical Solution

```
a=-1;
b=1;
-5*(exp(b)-exp(a)) + 5*(b*exp(b)-a*exp(a))
```

```
ans =         3.6788
```