MATLAB

for Engineering Applications

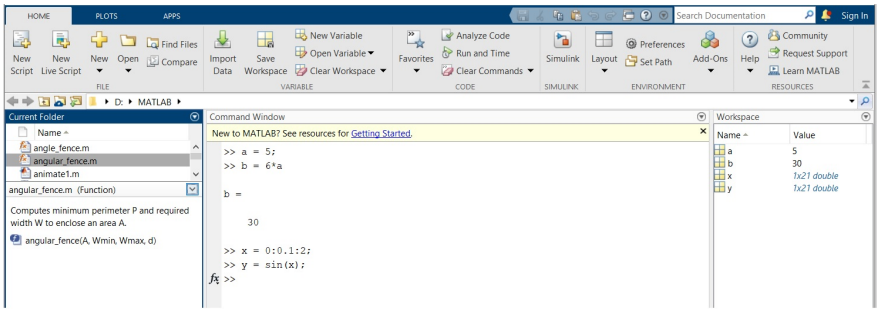
Fifth Edition

William J. Palm III

Chapter 01

An Overview of MATLAB

The Default MATLAB Desktop



Access the text alternative for slide images.

Source: MATLAB

Entering Commands and Expressions

MATLAB retains your previous keystrokes.

Use the up-arrow key to scroll back through the commands.

Press the key once to see the previous entry, and so on.

Use the down-arrow key to scroll forward. Edit a line using the left- and right-arrow keys the Backspace key, and the Delete key.

Press the Enter key to execute the command.

Scalar Arithmetic Operations

Symbol	Operation	MATLAB form
^	exponentiation: a^b	`a^b`
*	multiplication: ab	`a*b`
/	right division: $a/b = \frac{a}{b}$	`a/b`
\	left division: $a\backslash b = \frac{b}{a}$	`a\b`
+	addition: $a + b$	`a+b`
–	subtraction: $a - b$	`a-b`

An Example Session

```
>> 8/10
ans =
     0.8000
>> 5*ans
ans =
     4
>> r=8/10
r =
   0.8000
>> r
r =
   0.8000
>> s=20*r
s =
   16
```


Order of Precedence

Precedence	Operation
First	Parentheses, evaluated starting with the innermost pair.
Second	Exponentiation, evaluated from left to right.
Third	Multiplication and division with equal precedence, evaluated from left to right.
Fourth	Addition and subtraction with equal precedence, evaluated from left to right.

Examples of Precedence 1

```
>> 8 + 3*5
ans =
     23
>> 8 + (3*5)
ans =
     23
>>(8 + 3)*5
ans =
     55
>>4^2128/4*2
ans =
     0
>>4^212 8/(4*2)
ans =
     3
```

Examples of Precedence 2

```
>> 3*4^2 + 5
ans =
     53
>>(3*4)^2 + 5
ans =
     149
>>27^(1/3) + 32^(0.2)
ans =
     5
>>27^(1/3) + 32^0.2
ans =
     5
>>27^1/3 + 32^0.2
ans =
     11
```


Commands for Managing the Work Session 1

`clc`	Clears the Command window.
`clear`	Removes all variables from memory.
`clear x y`	Removes the variables x and y from memory.
`exist('abc')`	Determines if a file or variable exists having the name 'abc'.
`quit`	Stops MATLAB.

Commands for Managing the Work Session 2

`who`	Lists the variables currently in memory.
`whos`	Lists the current variables and sizes and indicate if they have imaginary parts.
`:`	Colon; generates an array having regularly spaced elements.
`,`	Comma; separates elements of an array.
`;`	Semicolon; suppresses screen printing; also denotes a new row in an array.
`...`	Ellipsis; continues a line.

Special Variables and Constants

`ans`	Temporary variable containing the most recent answer.
`eps`	Specifies the accuracy of floating-point precision.
`i, j`	The imaginary unit $\sqrt{-1}$.
`Inf`	Infinity.
`NaN`	Indicates an undefined numerical result.
`pi`	The number π.


```
Command Window
New to MATLAB? See resources for Getting Started.
>> i

ans =

   0.0000 + 1.0000i
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> pi

ans =

    3.1416
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> a=2+3*(pi)*i

a =

   2.0000 + 9.4248i
```


Complex Number Operations

The number $c_1 = 1 - 2i$ is entered as follows: `c1 = 12i`.

An asterisk is not needed between i or j and a number, although it is required with a variable, such as `c2 = 5i*c1`.

Be careful. The expressions

```
y = 7/2*i
```

and

```
x = 7/2i
```

give two different results:

`y = (7/2)i = 3.5i`

and

`x = 7/(2i) = -3.5i.`

```
Command Window
New to MATLAB? See resources for Getting Started.
>> c1=1-2i;
>> c2=5i*c1;
>> c2

c2 =

  10.0000 + 5.0000i
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> y=7/2*i

y =

   0.0000 + 3.5000i
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> 7/2i

ans =

   0.0000 - 3.5000i
```

Q1. What answers are produced by the following MATLAB expressions.

$$16\text{^}-1/2 \quad and \quad 16\ \text{^}(-1/2)$$

a) 0.03,0.25
b) 0.13,0.42
c) 0.02,0.22
d) 1.02,0.71

Uploaded Script

```
ans1=16 ^-1/2   ;
ans2=16 ^(-1/2) ;
```


Q2- The answer to the following expression $4\text{^}5 - 10 - 8/4 * 3$ using MATLAB is:

a) 1025
b) 2312
c) 1008
d) 2454

Uploaded Script

```
ans1 = 4^5-10-8/4*3 ;
```

Q3-Given x=-5+9i and y=6-2i, The answers to the following expressions are

$$x + 2 * y \; and \; x\text{^}2/y.$$

a) 4 + i , -9.3 -11.2i
b) 17- 5i , 3.9 +11.2i
c) 8 + 5i , 3.9 +16.3i
d) 7 + 5i , -9.3 -16.3i (correct)

Command Window

New to MATLAB? See resources for Getting Started.

```
>> x=-5+9i;
>> y=6-2i;
>> z=x+2*y

z =

   7.0000 + 5.0000i
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> x=-5+9i;
>> y=6-2i;
>> z=x^2/y

z =

  -3.9000 -16.3000i
```

 17

Numeric Display Formats 1

format short	Four decimal digits (the default); 13.6745.
format long	16 digits; 17.27484029463547.
format short e	Five digits (four decimals) plus exponent; 6.3792e+03.
format long e	16 digits (15 decimals) plus exponent; 6.379243784781294e−04.

Command Window

New to MATLAB? See resources for Getting Started.

```
>> format short
>> pi

ans =

    3.1416
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> format long
>> pi

ans =

   3.141592653589793
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> format short e
>> pi

ans =

   3.1416e+00
```

 18

Numeric Display Formats 2

format bank	Two decimal digits; 126.73.
format +	Positive, negative, or zero; +.
format rat	Rational approximation; 43/7.
format compact	Suppresses some blank lines.
format loose	Resets to less compact display mode.

Command Window

New to MATLAB? See resources for Getting Started.

```
>> format bank
>> pi

ans =

          3.14
```

Command Window

New to MATLAB? See resources for Getting Started.

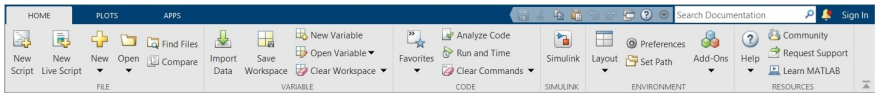
```
>> format +
>> pi

ans =

+
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> format rat
>> pi

ans =

     355/113
```


The Toolstrip after Selecting the HOME Tab.



Source: MATLAB

Arrays

The numbers 0, 0.1, 0.2, …, 10 can be assigned to the variable u by typing `u = 0:0.1:10.`

To compute $w = 5 \sin(u)$ for $u = 0, 0.1, 0.2, \ldots, 10$, the session is;

```
>>u = 0:0.1:10;
>>w = 5*sin(u);
```

The single line, `w = 5*sin(u),` computed the formula $w = 5 \sin(u)$ 101 times.

Array Index

```
>>u(7)
ans =
     0.6000
>>w(7)
ans =
     2.8232
```

Use the `length` function to determine how many values are in an array.

```
>>m = length(w)
m =
   101
```

Polynomial Roots

To find the roots of $x^3 - 7x^2 + 40x - 34 = 0$, the session is

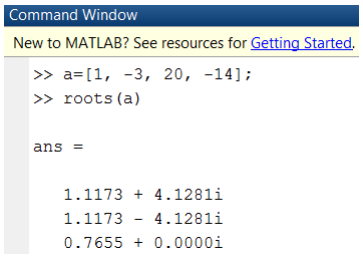
```
>>a = [1,-7,40,-34];
>>roots(a)
ans =
      3.0000 + 5.000i
      3.0000 - 5.000i
      1.0000
```

The roots are $x = 1$ and $x = 3 \pm 5i$.

To find the roots of $x^3 - 3x^2 + 20x - 14 = 0$, the session is

```
>>a = [1, -3, 20, -14];
>>roots(a)
ans =
   1.1173 + 4.1281i
   1.1173 – 4.1281i
   0.7655
```

The roots are $x = 0.7655$ and $x = 1.1173 \pm 4.1281i$.



Q4-Use MATLAB to determine how many elements are in the following array and what is the 35th element

$$cos(0):0.01:log10(125)$$

1. 110,1.34
2. 101,1.24
3. 51, 1.38
4. 101,1.24

Uploaded Script

```
A=cos(0):0.01:log10(125);
length(A);
ans1 = A(35);
```


Q5-Use MATLAB to find the roots of the polynomial

$$290 - 11x - 6x^2 - x^3.$$

a) 10,-1.3±5.6i
b) 4.7,-5.4±5.7i
c) 8.6,-1.3±5.6i
d) 8.6,-2.2±5.8i

Uploaded Script

```
a = [-1  -6  -11  290];
ans1 = roots(a);
```

Some Commonly used Mathematical Functions

Function	MATLAB syntax[1]
e^x	`exp(x)`
$\sqrt{x}$	`sqrt(x)`
$\ln x$	`log(x)`
$\log_{10} x$	`log10(x)`
$\cos x$	`cos(x)`
$\sin x$	`sin(x)`
$\tan x$	`tan(x)`
$\cos^{-1} x$	`acos(x)`
$\sin^{-1} x$	`asin(x)`
$\tan^{-1} x$	`atan(x)`

[1]The MATLAB trigonometric functions use radian measure.

```
>> exp(2)
ans =
    7.3891
```

```
>> sqrt(25)
ans =
     5
```

```
>> log(2.7182)
ans =
    1.0000
```

```
>> cos(pi/4)
ans =
    0.7071
```

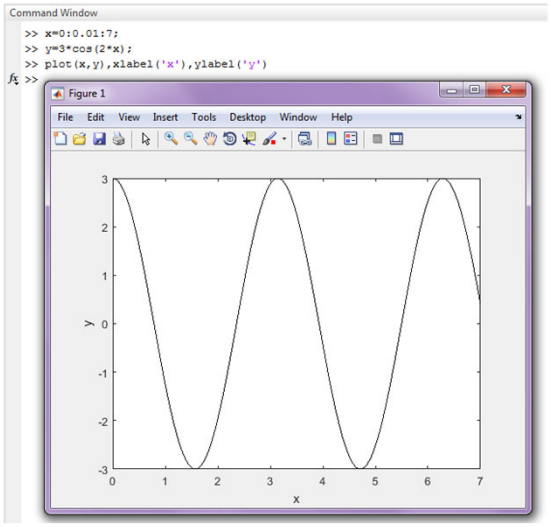
```
>> acos(0.7071)
ans =
    0.7854
>> pi/4
ans =
    0.7854
```


Some MATLAB Plotting Commands

Command	Description
`[x,y] = ginput(n)`	Enables the mouse to get n points from a plot, and returns the x and y coordinates in the vectors x and y.
`grid`	Puts grid lines on the plot.
`gtext('text')`	Enables placement of text with the mouse.
`plot(x,y)`	Generates a plot of the array y versus the array x on rectilinear axes.
`xlabel('text')`	Adds a text label to the horizontal axis (the abscissa).
`ylabel('text')`	Adds a text label to the vertical axis (the ordinate).

A Figure Window Showing a Plot.

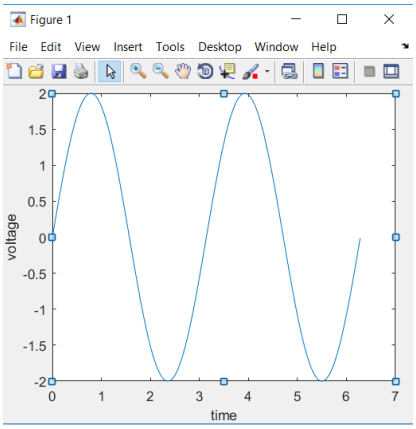
```
x = 0:0.01:7;
y = 3*cos(2*x);
plot(x,y),xlabel('x'),ylabel('y')
[x,y] = ginput(3)
```



Access the text alternative for slide images.

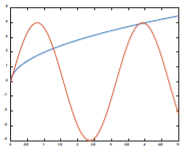
Source: MATLAB

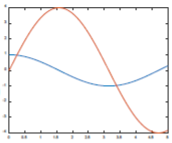

```
>> x=0:0.01:(2*pi);
>> y=2*sin(2*x);
>> plot(x,y),xlabel('time'),ylabel('voltage')
```

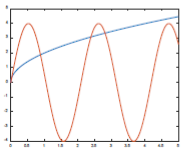


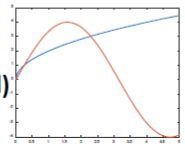

```
>> plot(x,y),xlabel('time'),ylabel('voltage')
>> [x,y]=ginput(3)

x =

    1.4698
    3.1265
    4.5557


y =

    0.3546
   -0.0222
    0.6094
```

Q6-Use MATLAB to plot the functions $y = 2\sqrt{x}$ and $z = 4sin2x$ for $0 \leq x \leq 5$ on the same plot. Which of the following plots is the answer?

(a) 

(c) 

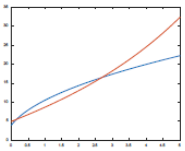
(b) 

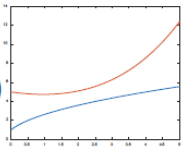
(d) 

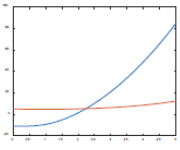
Uploaded Script

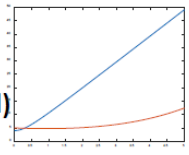
```
x = 0:0.01:5;
y = 2*sqrt(x);
z = 4*sin(2*x);
ans1=plot(x,y,x,z);
```


Q7-Which graph represents the plot for the following two functions $Y=4\sqrt{6x^2+1}$ *and* $z = 5e^{0.3x} - 2x$ *over* $0 \leq x \leq 1.5$

(a) 

(c) 

(b) 

(d) 

Uploaded Script

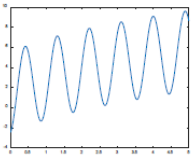
```
x = 0:.01:5;
y= 4*sqrt(6*(x.^2)+1);
z=5*exp(0.3*x)-2*x ;
ans1= plot(x,y,x,z);
```

Q8-Which graph represents the plot of the following function

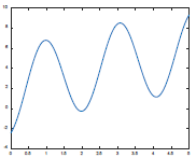
$$s=4sin\left(7t^2+5\right)+\sqrt{6t+2} \quad over \quad 0 \le t \le 5$$

```
t = 0:.01:5;
s= 4*sin(7*(t.^2)+5)+sqrt(6*t+2);
ans1=plot(t,s);
```

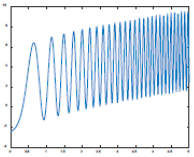
(a)



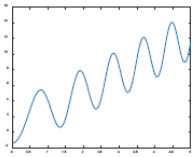
(c)



(b)



(d)



When you type `problem1`

1. MATLAB first checks to see if problem1 is a variable and if so, displays its value.
2. If not, MATLAB then checks to see if problem1 is one of its own commands and executes it if it is.
3. If not, MATLAB then looks in the current directory for a file named `problem1.m` and executes `problem1` if it finds it.
4. If not, MATLAB then searches the directories in its search path, in order, for `problem1.m` and then executes it if found.

System, Directory, and File Commands

`cd dirname`	Changes the current directory to `dirname`.
`dir`	Lists all files in the current directory.
`dir dirname`	Lists all the files in the directory `dirname`.
`path`	Displays the MATLAB search path.
`pwd`	Displays the current directory.
`which item`	Displays the path name of `item` if `item` is a function or file. Identifies `item` as a variable if so.

You can perform operations in MATLAB in two ways

1. In the interactive mode, in which all commands are entered directly in the Command window, or
2. By running a MATLAB program stored in *script* file. This type of file contains MATLAB commands, so running it is equivalent to typing all the commands—one at a time—at the Command window prompt. You can run the file by typing its name at the Command window prompt.

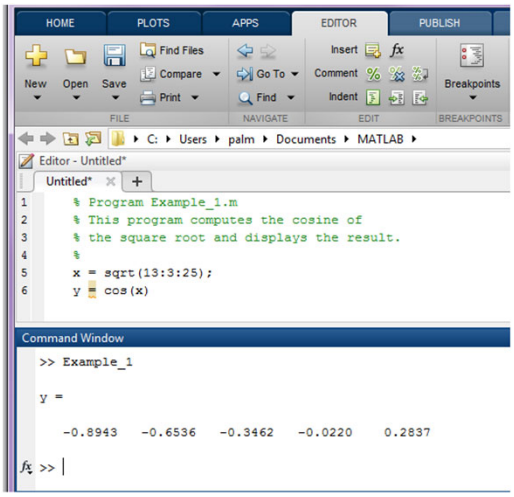
Inserting Comments

The comment symbol may be put anywhere in the line. MATLAB ignores everything to the right of the % symbol. For example,

```
>>% This is a comment.

>>x = 2+3 % So is this.

x =

    5
```

Note that the portion of the line before the % sign is executed to compute x.

The MATLAB Command Window with the Editor Open.



 Source: MATLAB

Keep in mind when using script files

1. The name of a script file must begin with a letter, and may include digits and the underscore character, and can much longer than you will ever need.
2. Do not give a script file the same name as a variable.
3. Do not give a script file the same name as a MATLAB command or function. You can check to see if a command, function or file name already exists by using the `exist` command.

Debugging Script Files

Program errors usually fall into one of the following categories.

1. Syntax errors such as omitting a parenthesis or comma, or spelling a command name incorrectly. MATLAB usually detects the more obvious errors and displays a message describing the error and its location.
2. Errors due to an incorrect mathematical procedure, called *runtime errors*. Their occurrence often depends on the particular input data. A common example is division by zero.

To locate program errors

Try the following:

1. Test your program with a simple version of the problem which can be checked by hand.
2. Display any intermediate calculations by removing semicolons at the end of statements.
3. Use the debugging features of the Editor/Debugger.

Programming Style 1

1. Comments section
   - The name of the program and any key words in the first line.
   - The date created, and the creators' names in the second line.
   - The definitions of the variable names for every input and output variable. Include definitions of variables used in the calculations and units of measurement for all input and all output variables!
   - The name of every user-defined function called by the program.

Programming Style 2

2. *Input section* Include input data and/or the input functions and comments for documentation.
3. *Calculation section*
4. *Output section* This section might contain functions for displaying the output on the screen.

Some Input/Output Commands

`disp(A)`	Displays the contents, but not the name, of the array `A`.
`disp('abc')`	Displays the text string enclosed within single quotes.
`x = input('abc')`	Displays the text in quotes, waits for user input from the keyboard, and stores the value in `x`.
`x = input('abc', 's')`	Displays the text in quotes, waits for user input from the keyboard, and stores the input as a string in `x`.

Example of a Script File 1

Problem:

- The speed v of a falling object dropped with no initial velocity is given as a function of time t by $v = gt$.
- Plot v as a function of t for $0 < t < t_{final}$, where t_{final} is the final time entered by the user.

Example of a Script File 2

```
% Program falling_speed.m:
% Plots speed of a falling object.
% Created on May 5, 2022
% Input Variable:
% tfinal = final time (in seconds)
% Output Variables:
% t = array of times at which speed is
% computed (in seconds)
% v = array of speeds (meters/second)
% Parameter Value:
g = 9.81; % Acceleration in SI units
% Input section:
tfinal = input('Enter final time in seconds:');
% Calculation section:
dt = tfinal/500;
% Create an array of 501 time values.
t = 0:dt:tfinal;
% Compute speed values.
v = g*t;
% Output section:
plot(t,v),xlabel('t (s)'),ylabel('v m/s)')
```

Getting Help From the Textbook

Throughout each chapter margin notes identify where key terms are introduced.

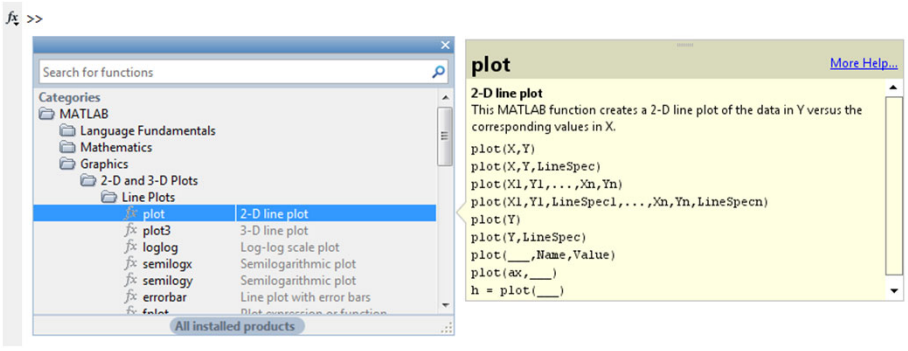
Each chapter contains tables summarizing the MATLAB commands introduced in that chapter.

At the end of each chapter is a summary guide to the commands covered in that chapter.

Appendix A contains tables of MATLAB commands, grouped by category, with the appropriate page references.

There are four indexes. The first lists MATLAB commands and symbols, the second lists Simulink blocks, the third lists commands for the Symbolic Math toolbox, and the fourth lists topics.

Getting Help From MATLAB: The Function Browser after `plot` has been selected



Access the text alternative for slide images.

 Source: MATLAB

MATLAB Help Functions

`help func_name`: Displays in the Command window a description of the specified function `func_name`.

`lookfor topic`: Displays in the Command window a brief description for all functions whose description includes the specified key word `topic`.

`doc func_name`: Opens the Help Browser to the reference page for the specified function `func_name`, providing a description, additional remarks, and examples.

Q9- The volume of a sphere can be calculated from: $V = \frac{4}{3}\pi r^3$, where r is the radius. Write a script file that prompts the user to enter a radius, computes the volume, and displays the result. For r=8, V is:

a) 3.5e+4
b) 4.8e+2
c) 2.1e+3 (correct)
d) 8.9e+4

Editor - C:\Users\t6588mt\Desktop\Volume.m

Volume.m

```
1 -    radius=input('radius in meters:');
2 -    V=(4/3)*pi*radius^3;
3 -    disp('The volume is:');
4 -    disp(V);
```

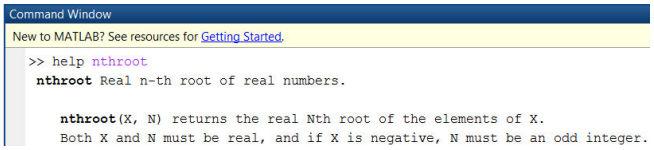
Command Window

New to MATLAB? See resources for Getting Started.

```
>> Volume
radius in meters:8
The volume is:
   2.1447e+03
```

Q10-Use the Help system to learn about the built-in function "nthroot".
(a) Use "nthroot" to calculate the cube root of 216.
(b) Use "lookfor" to find out how many "nthroot" functions are supported by MATLAB. The answers are:

a) 6 , 2 (correct)
b) 7, 9
c) 4, 5
d) 8, 14




```
>> nthroot (216,3)

ans =

     6
```

```
>> lookfor nthroot
nthroot                    - Real n-th root of real numbers.
sbionthroot                - Real n-th root of real numbers.
```


Steps in engineering problem solving 1

1. Understand the purpose of the problem.
2. Collect the known information. Realize that some of it might later found to be unnecessary.
3. Determine what information you must find.
4. Simplify the problem only enough to obtain the required information. State any assumptions you make.
5. Draw a sketch and label any necessary variables.
6. Determine which fundamental principles are applicable.

Steps in engineering problem solving 2

7. Think generally about your proposed solution approach and consider other approaches before proceeding with the details.
8. Label each step in the solution process.
9. If you solve the problem with a program, hand check the results using a simple version of the problem. Checking the dimensions and units and printing the results of intermediate steps in the calculation sequence can uncover mistakes.

Steps in engineering problem solving 3

10. Perform a "reality check" on your answer. Does it make sense? Estimate the range of the expected result and compare it with your answer. Do not state the answer with greater precision than is justified by any of the following:

    (a) The precision of the given information.

    (b) The simplifying assumptions.

    (c) The requirements of the problem.

11. Interpret the mathematics. If the mathematics produces multiple answers, do not discard them without considering what they mean. The mathematics might be trying to tell you something, and you might miss an opportunity to discover more about the problem.

Steps for developing a computer solution 1

1. State the problem concisely.
2. Specify the input data to be used by the program.
3. Specify the output information to be generated by the program.
4. Work through the solution steps by hand or with a calculator; use a simpler set of data if necessary.

Steps for developing a computer solution 2

5. Write and run the program.
6. Check the output of the program with your hand solution.
7. Run the program with your input data and perform a "reality check" on the output. Does it make sense? Estimate the range of the expected result and compare it with your answer.
8. If you will use the program as a general tool in the future, test it by running it for a range of reasonable data values; perform a reality check on the results.

Q11 The *ideal gas law* relates the pressure P, volume V, absolute temperature T, and amount of gas n. The law is

$$P = \frac{nRT}{V}$$

where R is the gas constant. An engineer must design a large natural gas storage tank to be expandable to maintain the pressure constant at 2.2 atm. In December when the temperature is 4°F (-15°C), the volume of gas in the tank is 18 500 ft^3. What will the volume of the same quantity of gas be in July when the temperature is 88°F (31°C)? (*Hint*: Use the fact that n, R, and P are constant in this problem. Note also that K=C+273.2.)

a) 3.3577e+04
b) 2.1796e+04 (correct)
c) 4.1565e+01
d) 6.2377e+06

Command Window

New to MATLAB? See resources for Getting Started.

```
>> V1=18500;
>> T1=273.2-15;
>> T2=273.2+31;
>> V2=V1*T2/T1

V2 =

   2.1796e+04
```

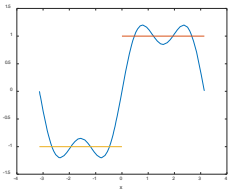

Q12- The Fourier series representation of the

Function $f(x) = \begin{cases} 1 & 0 < x < \pi \\ -1 & -\pi < x < 0 \end{cases}$

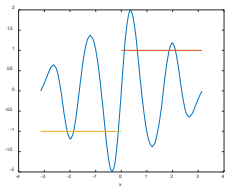
is $\frac{4}{\pi}\left(\frac{\sin x}{1} + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \frac{\sin 7x}{7} + \cdots\right)$

Plot on the same graph, the function $f(x)$ and its series representation, using the four terms shown.

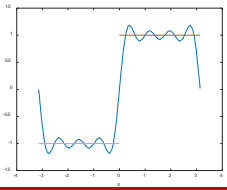
(a)



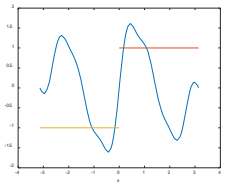
(c)



(b)



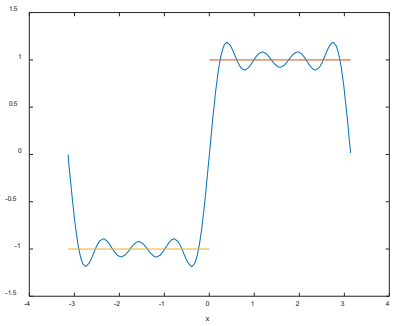
(d)



Command Window

New to MATLAB? See resources for Getting Started.

```
>> x1=[0,pi];
x2=[-pi,0];
x3=-pi:0.01:pi;
f1=[1,1];
f2=[-1,-1];
f3=(4/pi)*(sin(x3)+((1/3)*sin(3*x3))+((1/5)*sin(5*x3))+((1/7)*sin(7*x3)));
>> plot(x1,f1,x2,f2,x3,f3)
```



Q13-Write a script file to compute the three roots of the cubic equation $x^3 + ax^2 + bx + c = 0$. The roots for x if the parameters are a = 4, b = -8, c = -12.

a) -5.1, 2.1, -1.1
b) -4.2, 1.8, -5.1
c) 1.1, 4.2, -7.7
d) 3.2, -2.2, -2.2

```
disp('To compute the roots of x^3+ax^2+bx+c=0, enter a, b, and c.') a
= input('Enter a: ');
b = input('Enter b: ');
c = input('Enter c: ');
roots([1,a,b,c]);
```

```
ans =

  -5.1064
   2.1830
  -1.0765
```

Q14-Suppose x takes on the values x = 1, 1.2, 1.4, 1.6, … , 5. Use MATLAB to compute the array y that results from the function of y = 4 sin(3x). Find the number of elments in the array y and the value of its third element.

a) 17, -4.355
b) 14, -3.234
c) 19, -5.237
d) 26, -3.486

```
x=1:0.2:6;
y = 4*sin(3*x);
ans1=length(y)
; ans2=y(3);
```

 61

Because learning changes everything.®

www.mheducation.com