

1.5 ELEMENTARY MATH BUILT-IN FUNCTIONS

In addition to basic arithmetic operations, expressions in MATLAB can include functions. MATLAB has a very large library of built-in functions. A function has a name and an argument in parentheses. For example, the function that calculates the square root of a number is `sqrt(x)`. Its name is `sqrt`, and the argument is `x`. When the function is used, the argument can be a number, a variable that has been assigned a numerical value (explained in Section 1.6), or a computable expression that can be made up of numbers and/or variables. Functions can also be included in arguments, as well as in expressions. Tutorial 1-2 shows examples of using the function `sqrt(x)` when MATLAB is used as a calculator with scalars.

Tutorial 1-2: Using the `sqrt` built-in function.

```
>> sqrt(64)
ans =
     8
```

Argument is a number.

```
>> sqrt(50+14*3)
ans =
    9.5917
```

Argument is an expression.

```
>> sqrt(54+9*sqrt(100))
ans =
    12
```

Argument includes a function.

```
>> (15+600/4)/sqrt(121)
ans =
    15
>>
```

Function is included in an expression.

Some commonly used elementary MATLAB mathematical built-in functions are given in Tables 1-3 through 1-5. A complete list of functions organized by category can be found in the Help Window.

Table 1-3: Elementary math functions

Function	Description	Example
<code>sqrt(x)</code>	Square root.	<pre>>> sqrt(81) ans = 9</pre>
<code>nthroot(x,n)</code>	Real n th root of a real number x . (If x is negative n must be an odd integer.)	<pre>>> nthroot(80,5) ans = 2.4022</pre>
<code>exp(x)</code>	Exponential (e^x).	<pre>>> exp(5) ans = 148.4132</pre>

Table 1-3: Elementary math functions (Continued)

Function	Description	Example
<code>abs(x)</code>	Absolute value.	<pre>>> abs(-24) ans = 24</pre>
<code>log(x)</code>	Natural logarithm. Base e logarithm (\ln).	<pre>>> log(1000) ans = 6.9078</pre>
<code>log10(x)</code>	Base 10 logarithm.	<pre>>> log10(1000) ans = 3.0000</pre>
<code>factorial(x)</code>	The factorial function $x!$ (x must be a positive integer.)	<pre>>> factorial(5) ans = 120</pre>

Table 1-4: Trigonometric math functions

Function	Description	Example
<code>sin(x)</code> <code>sind(x)</code>	Sine of angle x (x in radians). Sine of angle x (x in degrees).	<pre>>> sin(pi/6) ans = 0.5000</pre>
<code>cos(x)</code> <code>cosd(x)</code>	Cosine of angle x (x in radians). Cosine of angle x (x in degrees).	<pre>>> cosd(30) ans = 0.8660</pre>
<code>tan(x)</code> <code>tand(x)</code>	Tangent of angle x (x in radians). Tangent of angle x (x in degrees).	<pre>>> tan(pi/6) ans = 0.5774</pre>
<code>cot(x)</code> <code>cotd(x)</code>	Cotangent of angle x (x in radians). Cotangent of angle x (x in degrees).	<pre>>> cotd(30) ans = 1.7321</pre>

The inverse trigonometric functions are `asin(x)`, `acos(x)`, `atan(x)`, `acot(x)` for the angle in radians; and `asind(x)`, `acosd(x)`, `atand(x)`, `acotd(x)` for the angle in degrees. The hyperbolic trigonometric functions are `sinh(x)`, `cosh(x)`, `tanh(x)`, and `coth(x)`. Table 1-4 uses `pi`, which is equal to π (see Section 1.6.3).

Table 1-5: Rounding functions

Function	Description	Example
<code>round(x)</code>	Round to the nearest integer.	<pre>>> round(17/5) ans = 3</pre>
<code>fix(x)</code>	Round toward zero.	<pre>>> fix(13/5) ans = 2</pre>

Table 1-5: Rounding functions (Continued)

Function	Description	Example
<code>ceil(x)</code>	Round toward infinity.	<pre>>> ceil(11/5) ans = 3</pre>
<code>floor(x)</code>	Round toward minus infinity.	<pre>>> floor(-9/4) ans = -3</pre>
<code>rem(x,y)</code>	Returns the remainder after x is divided by y .	<pre>>> rem(13,5) ans = 3</pre>
<code>sign(x)</code>	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	<pre>>> sign(5) ans = 1</pre>

1.6 DEFINING SCALAR VARIABLES

A variable is a name made of a letter or a combination of several letters (and digits) that is assigned a numerical value. Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands. A variable is actually a name of a memory location. When a new variable is defined, MATLAB allocates an appropriate memory space where the variable's assignment is stored. When the variable is used the stored data is used. If the variable is assigned a new value the content of the memory location is replaced. (In Chapter 1 we consider only variables that are assigned numerical values that are scalars. Assigning and addressing variables that are arrays is discussed in Chapter 2.)

1.6.1 The Assignment Operator

In MATLAB the `=` sign is called the assignment operator. The assignment operator assigns a value to a variable.

Variable_name = A numerical value, or a computable expression

- The left-hand side of the assignment operator can include only one variable name. The right-hand side can be a number, or a computable expression that can include numbers and/or variables that were previously assigned numerical values. When the **Enter** key is pressed the numerical value of the right-hand side is assigned to the variable, and MATLAB displays the variable and its assigned value in the next two lines.

The following shows how the assignment operator works.

```
>> x=15
x =
    15

>> x=3*x-12
x =
    33
>>
```

The number 15 is assigned to the variable x.

MATLAB displays the variable name and its assigned value.

A new value is assigned to x. The new value is 3 times the previous value of x minus 12.

The last statement ($x = 3x - 12$) illustrates the difference between the assignment operator and the equal sign. If in this statement the = sign meant equal, the value of x would be 6 (solving the equation for x).

The use of previously defined variables to define a new variable is demonstrated next.

```
>> a=12
a =
    12

>> B=4
B =
     4

>> C=(a-B)+40-a/B*10
C =
    18
```

Assign 12 to a.

Assign 4 to B.

Assign the value of the expression on the right-hand side to the variable C.

- If a semicolon is typed at the end of the command, then when the **Enter** key is pressed, MATLAB does not display the variable with its assigned value (the variable still exists and is stored in memory).
- If a variable already exists, typing the variable's name and pressing the **Enter** key will display the variable and its value in the next two lines.

As an example, the last demonstration is repeated below using semicolons.

```
>> a=12;
>> B=4;
>> C=(a-B)+40-a/B*10;

>> C
C =
    18
```

The variables a, B, and C are defined but are not displayed, since a semicolon is typed at the end of each statement.

The value of the variable C is displayed by typing the name of the variable.

- Several assignments can be typed in the same line. The assignments must be separated with a comma (spaces can be added after the comma). When the **Enter** key is pressed, the assignments are executed from left to right and the variables and

their assignments are displayed. A variable is not displayed if a semicolon is typed instead of a comma. For example, the assignments of the variables a, B, and C above can all be done in the same line.

```
>> a=12, B=4; C=(a-B)+40-a/B*10
a =
    12
C =
    18
```

The variable B is not displayed because a semicolon is typed at the end of the assignment.

- A variable that already exists can be reassigned a new value. For example:

```
>> ABB=72;
>> ABB=9;
>> ABB
ABB =
     9
>>
```

A value of 72 is assigned to the variable ABB.

A new value of 9 is assigned to the variable ABB.

The current value of the variable is displayed when the name of the variable is typed and the Enter key is pressed.

- Once a variable is defined it can be used as an argument in functions. For example:

```
>> x=0.75;
>> E=sin(x)^2+cos(x)^2
E =
     1
>>
```

1.6.2 Rules About Variable Names

A variable can be named according to the following rules:

- Must begin with a letter.
- Can be up to 63 characters long.
- Can contain letters, digits, and the underscore character.
- Cannot contain punctuation characters (e.g., period, comma, semicolon).
- MATLAB is case-sensitive: it distinguishes between uppercase and lowercase letters. For example, AA, Aa, aA, and aa are the names of four different variables.
- No spaces are allowed between characters (use the underscore where a space is desired).
- Avoid using the name of a built-in function for a variable (i.e., avoid using cos, sin, exp, sqrt, etc.). Once a function name is used to for a variable name, the function cannot be used.

1.6.3 Predefined Variables and Keywords

There are 20 words, called keywords, that are reserved by MATLAB for various purposes and cannot be used as variable names. These words are:

`break` `case` `catch` `classdef` `continue` `else` `elseif`
`end` `for` `function` `global` `if` `otherwise` `parfor`
`persistent` `return` `spmd` `switch` `try` `while`

When typed, these words appear in blue. An error message is displayed if the user tries to use a keyword as a variable name. (The keywords can be displayed by typing the command `iskeyword`.)

A number of frequently used variables are already defined when MATLAB is started. Some of the predefined variables are:

ans A variable that has the value of the last expression that was not assigned to a specific variable (see Tutorial 1-1). If the user does not assign the value of an expression to a variable, MATLAB automatically stores the result in `ans`.

pi The number π .

eps The smallest difference between two numbers. Equal to $2^{(-52)}$, which is approximately $2.2204e-016$.

inf Used for infinity.

i Defined as $\sqrt{-1}$, which is: $0 + 1.0000i$.

j Same as `i`.

NaN Stands for Not-a-Number. Used when MATLAB cannot determine a valid numeric value. Example: `0/0`.

The predefined variables can be redefined to have any other value. The variables `pi`, `eps`, and `inf`, are usually not redefined since they are frequently used in many applications. Other predefined variables, such as `i` and `j`, are sometime redefined (commonly in association with loops) when complex numbers are not involved in the application.

1.7 USEFUL COMMANDS FOR MANAGING VARIABLES

The following are commands that can be used to eliminate variables or to obtain information about variables that have been created. When these commands are typed in the Command Window and the **Enter** key is pressed, either they provide information, or they perform a task as specified below.

<u>Command</u>	<u>Outcome</u>
<code>clear</code>	Removes all variables from the memory.

<u>Command</u>	<u>Outcome</u>
<code>clear x y z</code>	Removes only variables <code>x</code> , <code>y</code> , and <code>z</code> from the memory.
<code>who</code>	Displays a list of the variables currently in the memory.
<code>whos</code>	Displays a list of the variables currently in the memory and their sizes together with information about their bytes and class (see Section 4.1).

1.8 **SCRIPT FILES**

So far all the commands were typed in the Command Window and were executed when the **Enter** key was pressed. Although every MATLAB command can be executed in this way, using the Command Window to execute a series of commands—especially if they are related to each other (a program)—is not convenient and may be difficult or even impossible. The commands in the Command Window cannot be saved and executed again. In addition, the Command Window is not interactive. This means that every time the **Enter** key is pressed only the last command is executed, and everything executed before is unchanged. If a change or a correction is needed in a command that was previously executed and the result of this command is used in commands that follow, all the commands have to be entered and executed again.

A different (better) way of executing commands with MATLAB is first to create a file with a list of commands (program), save it, and then run (execute) the file. When the file runs, the commands it contains are executed in the order that they are listed. If needed, the commands in the file can be corrected or changed and the file can be saved and run again. Files that are used for this purpose are called script files.

IMPORTANT NOTE: This section covers only the minimum required in order to run simple programs. This will allow the student to use script files when practicing the material that is presented in this and the next two chapters (instead of typing repeatedly in the Command Window). Script files are considered again in Chapter 4, where many additional topics that are essential for understanding MATLAB and writing programs in script file are covered.

1.8.1 **Notes About Script Files**

- A script file is a sequence of MATLAB commands, also called a program.
- When a script file runs (is executed), MATLAB executes the commands in the order they are written, just as if they were typed in the Command Window.

- When a script file has a command that generates an output (e.g., assignment of a value to a variable without a semicolon at the end), the output is displayed in the Command Window.
- Using a script file is convenient because it can be edited (corrected or otherwise changed) and executed many times.
- Script files can be typed and edited in any text editor and then pasted into the MATLAB editor.
- Script files are also called M-files because the extension .m is used when they are saved.

1.8.2 **Creating and Saving a Script File**

In MATLAB script files are created and edited in the Editor/Debugger Window. This window is opened from the Command Window by clicking on the **New Script** icon in the Toolstrip, or by clicking **New** in the Toolstrip and then selecting **Script** from the menu that opens. An open Editor/Debugger Window is shown in Figure 1-6.

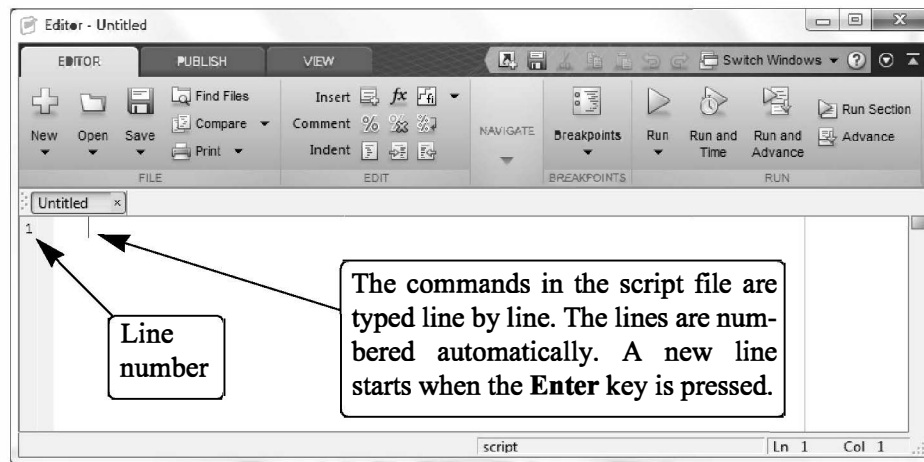


Figure 1-6: The Editor/Debugger Window.

The Editor/Debugger Window has a Toolstrip at the top and three tabs EDITOR, PUBLISH, and VIEW above it. Clicking on the tabs changes the icons in the Toolstrip. Commonly, MATLAB is used with the HOME tab selected. The associated icons are used for executing various commands, as explained later in the Chapter. Once the window is open, the commands of the script file are typed line by line. MATLAB automatically numbers a new line every time the **Enter** key is pressed. The commands can also be typed in any text editor or word processor program and then copied and pasted in the Editor/Debugger Window. An example of a short program typed in the Editor/Debugger Window is shown in Figure 1-7. The first few lines in a script file are typically comments (which are

not executed, since the first character in the line is %) that describe the program written in the script file.

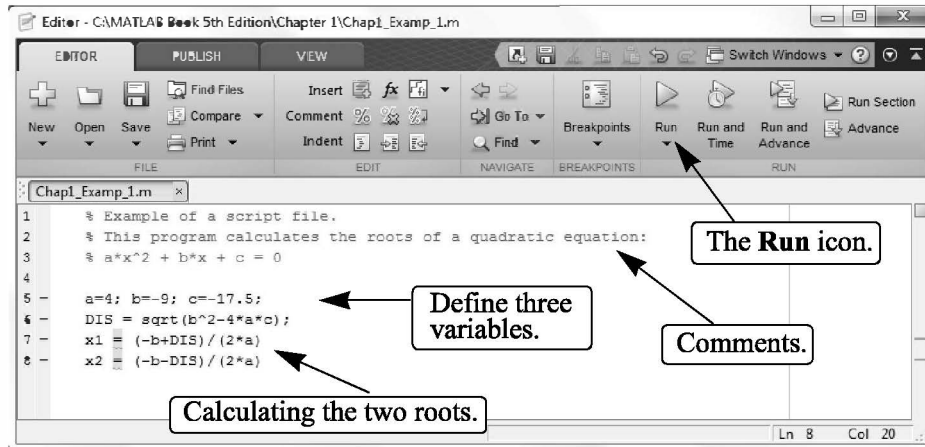


Figure 1-7: A program typed in the Editor/Debugger Window.

Before a script file can be executed it has to be saved. This is done by clicking **Save** in the **Toolstrip** and selecting **Save As...** from the menu that opens. When saved, MATLAB adds the extension `.m` to the name. The rules for naming a script file follow the rules of naming a variable (must begin with a letter, can include digits and underscore, *no spaces*, and up to 63 characters long). The names of user-defined variables, predefined variables, and MATLAB commands or functions should not be used as names of script files.

1.8.3 Running (Executing) a Script File

A script file can be executed either directly from the Editor Window by clicking on the **Run** icon (see Figure 1-7) or by typing the file name in the Command Window and then pressing the **Enter** key. For a file to be executed, MATLAB needs to know where the file is saved. The file will be executed if the folder where the file is saved is the current folder of MATLAB or if the folder is listed in the search path, as explained next.

1.8.4 Current Folder

The current folder is shown in the “Current Folder” field in the desktop toolbar of the Command Window, as shown in Figure 1-8. If an attempt is made to execute a script file by clicking on the **Run** icon (in the Editor Window) when the current folder is not the folder where the script file is saved, then the prompt shown in Figure 1-9 opens. The user can then change the current folder to the folder where the script file is saved, or add it to the search path. Once two or more different current folders are used in a session, it is possible to switch from one to another in the

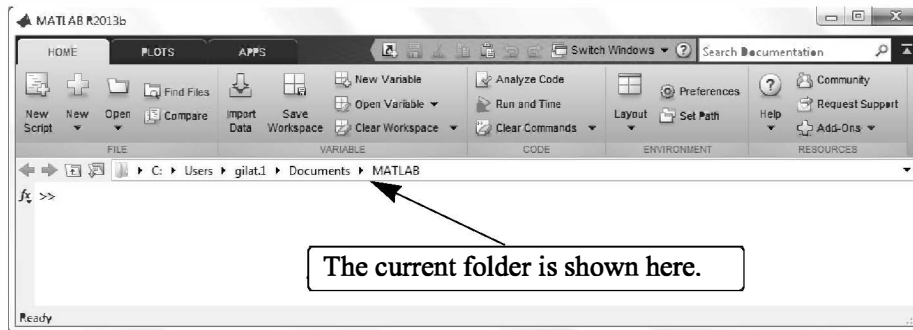


Figure 1-8: The Current folder field in the Command Window.

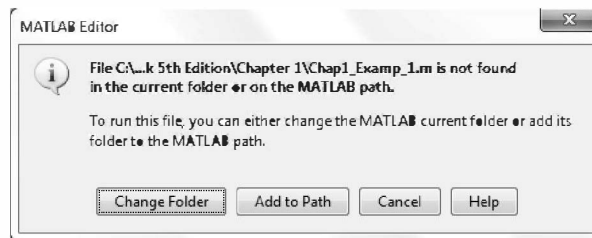


Figure 1-9: Changing the current directory.

Current Folder field in the Command Window. The current folder can also be changed in the Current Folder Window, shown in Figure 1-10, which can be opened from the **Desktop** menu. The Current Folder can be changed by choosing the drive and folder where the file is saved.

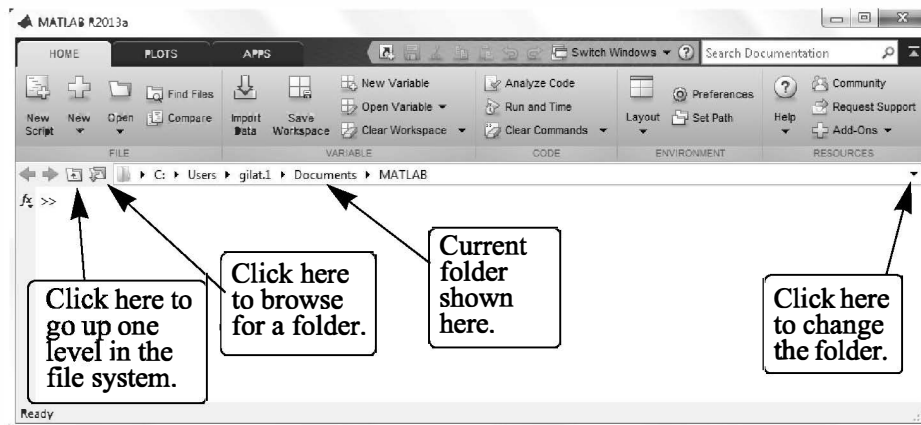


Figure 1-10: The Current Folder Window.

An alternative simple way to change the current folder is to use the `cd` command in the Command Window. To change the current folder to a different drive, type `cd`, space, and then the name of the directory followed by a colon `:` and press the **Enter** key. For example, to change the current folder to drive E (e.g., the flash drive) type `cd E:`. If the script file is saved in a folder within a drive, the path to that folder has to be specified. This is done by typing the path as a string in the `cd` command. For example, `cd('E:\Chapter 1')` sets the path to the folder Chapter 1 in drive E. The following example shows how the current folder is changed to be drive E. Then the script file from Figure 1-7, which was saved in drive E as `ProgramExample.m`, is executed by typing the name of the file and pressing the **Enter** key.

```
>> cd('E:\Chapter 1')
```

The current directory is changed to drive E.

```
>> Chap1_Example
x1 =
    3.5000
x2 =
   -1.2500
```

The script file is executed by typing the name of the file and pressing the **Enter** key.

The output generated by the script file (the roots `x1` and `x2`) is displayed in the Command Window.

1.9 EXAMPLES OF MATLAB APPLICATIONS

Sample Problem 1-1: Trigonometric identity

A trigonometric identity is given by:

$$\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}$$

Verify that the identity is correct by calculating each side of the equation, substituting $x = \frac{\pi}{5}$.

Solution

The problem is solved by typing the following commands in the Command Window.

```
>> x=pi/5;
>> LHS=cos(x/2)^2
LHS =
    0.9045
>> RHS=(tan(x)+sin(x))/(2*tan(x))
RHS =
    0.9045
```

Define x.

Calculate the left-hand side.

Calculate the right-hand side.

Sample Problem 1-3: Heat transfer

An object with an initial temperature of T_0 that is placed at time $t = 0$ inside a chamber that has a constant temperature of T_s will experience a temperature change according to the equation

$$T = T_s + (T_0 - T_s)e^{-kt}$$

where T is the temperature of the object at time t , and k is a constant. A soda can at a temperature of 120° F (after being left in the car) is placed inside a refrigerator where the temperature is 38° F. Determine, to the nearest degree, the temperature of the can after three hours. Assume $k = 0.45$. First define all of the variables and then calculate the temperature using one MATLAB command.

Solution

The problem is solved by typing the following commands in the Command Window.

```
>> Ts=38; T0=120; k=0.45; t=3;
```

```
>> T=round(Ts+(T0-Ts)*exp(-k*t))
```

```
T =  
    59
```

Round to the nearest integer.

1.10 PROBLEMS

The following problems can be solved by writing commands in the Command Window, or by writing a program in a script file and then executing the file.

1. Calculate:

$$(a) \frac{22 + 5.1^2}{50 - 6.3^2}$$

$$(b) \frac{44}{7} + \frac{8^2}{5} - \frac{99}{3.9^2}$$

2. Calculate:

$$(a) \frac{\sqrt{41^2 - 5.2^2}}{e^5 - 100.53}$$

$$(b) \sqrt[3]{132} + \frac{\ln(500)}{8}$$

3. Calculate:

$$(a) \frac{14.8^3 - 6.3^2}{(\sqrt{13} + 5)^2}$$

$$(b) 45\left(\frac{288}{9.3} - 4.6^2\right) - 1065e^{-1.5}$$

4. Calculate:

$$(a) \frac{24.5 + 64/3.5^2 + 8.3 \cdot 12.5^3}{\sqrt{76.4} - 28/15}$$

$$(b) (5.9^2 - 2.4^2)/3 + \left(\frac{\log_{10} 12890}{e^{0.3}}\right)^2$$

5. Calculate:

$$(a) \cos\left(\frac{7\pi}{9}\right) + \tan\left(\frac{7}{15}\pi\right) \sin(15^\circ)$$

$$(b) \sin^2 80^\circ - \frac{(\cos 14^\circ \sin 80^\circ)^2}{\sqrt[3]{0.18}}$$

6. Define the variable x as $x = 6.7$, then evaluate:

$$(a) 0.01x^5 - 1.4x^3 + 80x + 16.7$$

$$(b) \sqrt{x^3 + e^x - 51/x}$$

7. Define the variable t as $t = 3.2$, then evaluate:

$$(a) 56t - 9.81 \frac{t^2}{2}$$

$$(b) 14e^{-0.1t} \sin(2\pi t)$$

8. Define the variables x and y as $x = 5.1$ and $y = 4.2$, then evaluate:

$$(a) \frac{3}{4}xy - \frac{7x}{y^2} + \sqrt{xy}$$

$$(b) (xy)^2 - \frac{x+y}{(x-y)^2} + \sqrt{\frac{x+y}{2x-y}}$$

9. Define the variables a , b , c , and d as:

$$a = 12, b = 5.6, c = \frac{3a}{b^2}, \text{ and } d = \frac{(a-b)^c}{c}, \text{ then evaluate:}$$

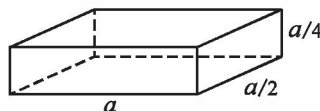
$$(a) \frac{a}{b} + \frac{d-c}{d+c} - (d-b)^2$$

$$(b) e^{\frac{d-c}{a-2b}} + \ln\left(c - d + \frac{b}{a}\right)$$

10. A sphere has a radius of 24 cm. A rectangular prism has sides of a , $a/2$, and $a/4$.

(a) Determine a of a prism that has the same volume as the sphere.

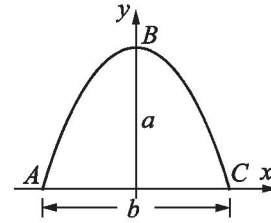
(b) Determine a of a prism that has the same surface area as the sphere.



11. The arc length of a segment of a parabola ABC of an ellipse with semi-minor axes a and b is given approximately by:

$$L_{ABC} = \frac{1}{2}\sqrt{b^2 + 16a^2} + \frac{b^2}{8a} \ln\left(\frac{4a + \sqrt{b^2 + 16a^2}}{b}\right).$$

- (a) Determine L_{ABC} if $a = 11$ in. and $b = 9$ in.



12. Two trigonometric identities are given by:

$$(a) \sin 5x = 5 \sin x - 20 \sin^3 x + 16 \sin^5 x \quad (b) \sin^2 x \cos^2 x = \frac{1 - \cos 4x}{8}$$

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = \frac{\pi}{12}$.

13. Two trigonometric identities are given by:

$$(a) \tan 3x = \frac{3 \tan x - \tan^3 x}{1 - 3 \tan^2 x} \quad (b) \cos 4x = 8(\cos^4 x - \cos^2 x) + 1$$

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = 24^\circ$.

14. Define two variables: $\alpha = \pi/6$, and $\beta = 3\pi/8$. Using these variables, show that the following trigonometric identity is correct by calculating the values of the left and right sides of the equation.

$$\sin \alpha + \sin \beta = 2 \sin\left(\frac{\alpha + \beta}{2}\right) \cos\left(\frac{\alpha - \beta}{2}\right)$$

15. Given: $\int x \sin ax dx = \frac{\sin ax}{a^2} - \frac{x \cos ax}{a}$. Use MATLAB to calculate the following definite integral:

$$\int_{\frac{\pi}{3}}^{\frac{3\pi}{2}} x \sin(0.6x) dx.$$

16. In the triangle shown $a = 5.3$ in., $\gamma = 42^\circ$, and $b = 6$ in. Define a , γ , and b as variables, and then:

- (a) Calculate the length b by using the Law of Cosines.

$$\text{(Law of Cosines: } c^2 = a^2 + b^2 - 2ab \cos \gamma)$$

- (b) Calculate the angles β and γ (in degrees) using the Law of Cosines.

- (c) Check that the sum of the angles is 180° .

