

Chapter 1

Starting with

MATLAB

This chapter begins by describing the characteristics and purpose of the different windows in MATLAB. Next, the Command Window is introduced in detail. The chapter shows how to use MATLAB for arithmetic operations with scalars in much the way that a calculator is used. This includes the use of elementary math functions with scalars. The chapter then shows how to define scalar variables (the assignment operator) and how to use these variables in arithmetic calculations. The last section in the chapter introduces script files. It shows how to write, save, and execute simple MATLAB programs.

1.1 STARTING MATLAB, MATLAB WINDOWS

It is assumed that the software is installed on the computer, and that the user can start the program. Once the program starts, the MATLAB desktop window opens with the default layout, Figure 1-1. The layout has a Toolstrip at the top, the Current Folder Toolbar below it, and four windows underneath. At the top of the Toolstrip there are three tabs: HOME, PLOTS, and APPS. 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 this chapter. The PLOTS tab can be used to create plots, as explained in Chapter 5 (Section 5.12), and the APPS tab can be used for opening additional applications and Toolboxes of MATLAB.

The default layout

The default layout (Figure 1-1) consists of the following four windows that are displayed under the Toolstrip: the Command Window (the larger window), the Current Folder Window (on the top left), the Details Window and the Workspace Window (on the bottom left). A list of several MATLAB windows and their purposes is given in Table 1-1.

Four of the windows—the Command Window, the Figure Window, the Editor Window, and the Help Window—are used extensively throughout the book

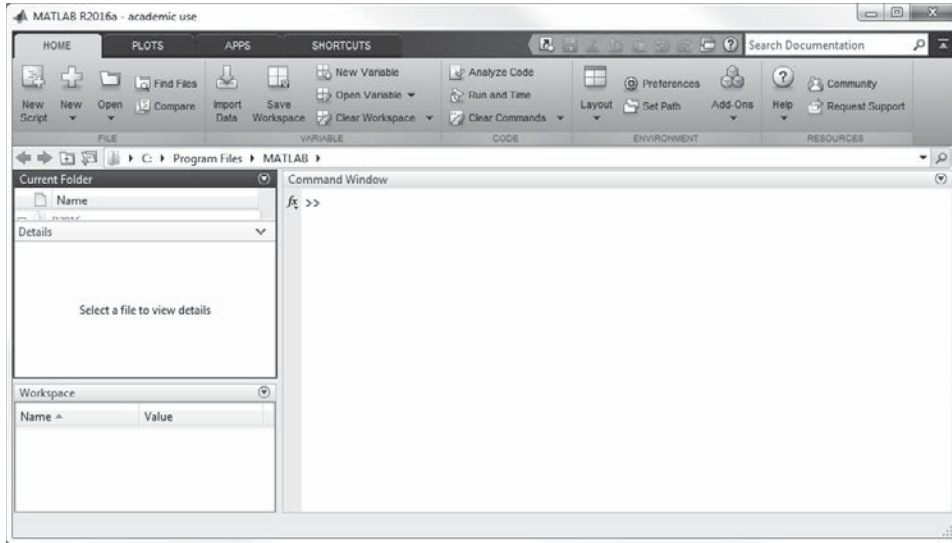


Figure 1-1: The default view of MATLAB desktop.

and are briefly described on the following pages. More detailed descriptions are included in the chapters where they are used. The Command History Window, Current Folder Window, and the Workspace Window are described in Sections 1.2, 1.8.4, and 4.1, respectively.

Command Window: The Command Window is MATLAB's main window and opens when MATLAB is started. It is convenient to have the Command Window as the only visible window. This can be done either by closing all the other windows, or by selecting **Command Window Only** in the menu that opens when the **Layout** icon on the Toolbar is selected. To close a window, click on the pull-down menu at the top right-hand side of the window and then select Close. Working in the Command Window is described in detail in Section 1.2.

Table 1-1: MATLAB windows

Window	Purpose
Command Window	Main window, enters variables, runs programs.
Figure Window	Contains output from graphic commands.
Editor Window	Creates and debugs script and function files.
Help Window	Provides help information.
Command History Window	Logs commands entered in the Command Window.

Table 1-1: MATLAB windows

Window	Purpose
Workspace Window	Provides information about the variables that are stored.
Current Folder Window	Shows the files in the current folder.

Figure Window: The Figure Window opens automatically when graphics commands are executed, and contains graphs created by these commands. An example of a Figure Window is shown in Figure 1-2. A more detailed description of this window is given in Chapter 5.

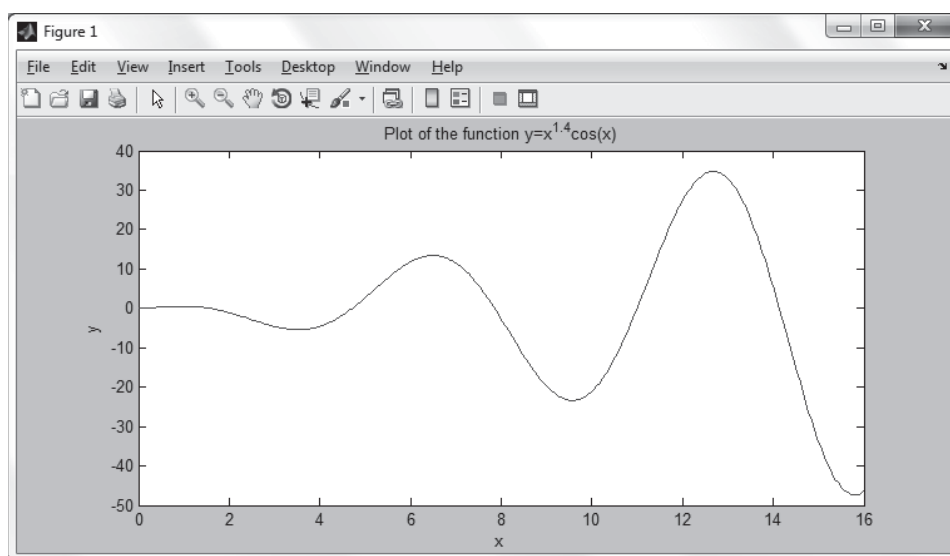


Figure 1-2: Example of a Figure Window.

Editor Window: The Editor Window is used for writing and editing programs. This window is opened by clicking on the **New Script** icon in the Toolstrip, or by clicking on the **New** icon and then selecting **Script** from the menu that opens. An example of an Editor Window is shown in Figure 1-3. More details on the Editor Window are given in Section 1.8.2, where it is used for writing script files, and in Chapter 7, where it is used to write function files.

Help Window: The Help Window contains help information. This window can be opened from the **Help** icon in the Toolstrip of the Command Window or the toolbar of any MATLAB window. The Help Window is interactive and can be used to obtain information on any feature of MATLAB. Figure 1-4 shows an open Help Window.

When MATLAB is started for the first time, the screen looks like that shown in Figure 1-1. For most beginners it is probably more convenient to close

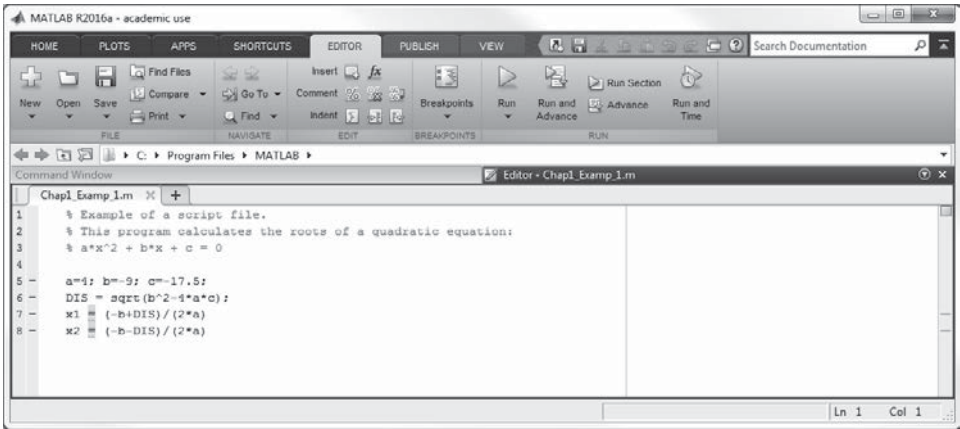


Figure 1-3: Example of an Editor Window.

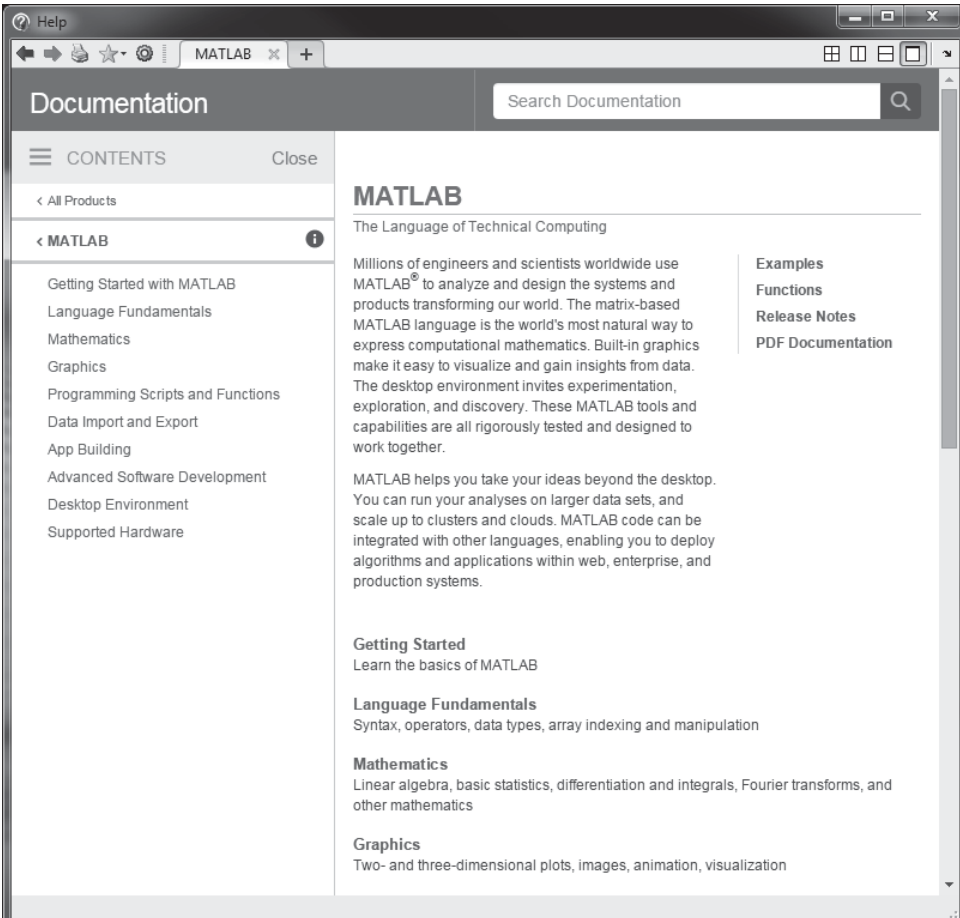


Figure 1-4: The Help Window.

all the windows except the Command Window. The closed windows can be reopened by selecting them from the **layout** icon in the Toolstrip. The windows shown in Figure 1-1 can be displayed by clicking on the **layout** icon and selecting **Default** in the menu that opens. The various windows in Figure 1-1 are docked to the desktop. A window can be undocked (become a separate, independent window) by dragging it out. An independent window can be redocked by clicking on the pull-down menu at the top right-hand side of the window and then selecting Dock.

1.2 WORKING IN THE COMMAND WINDOW

The Command Window is MATLAB's main window and can be used for executing commands, opening other windows, running programs written by the user, and managing the software. An example of the Command Window, with several simple commands that will be explained later in this chapter, is shown in Figure 1-5.

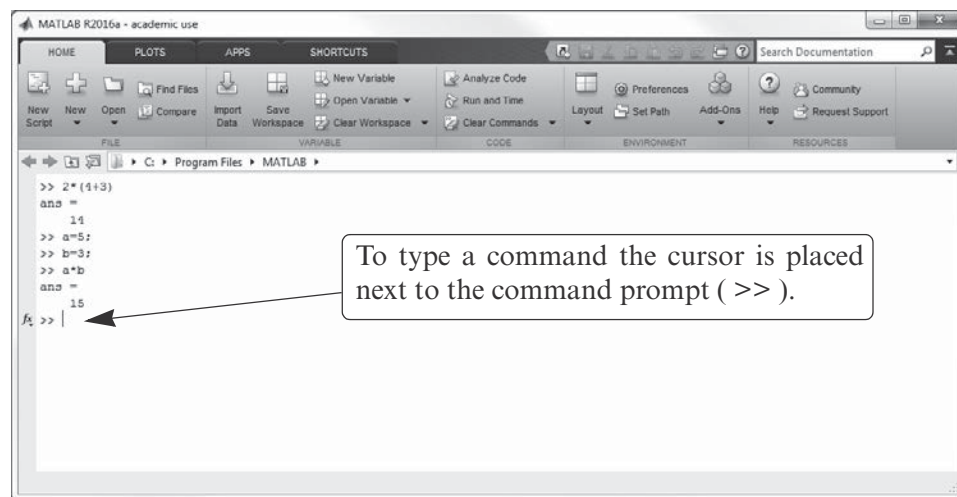


Figure 1-5: The Command Window.

Notes for working in the Command Window:

- To type a command, the cursor must be placed next to the command prompt (>>).
- Once a command is typed and the **Enter** key is pressed, the command is executed. However, only the last command is executed. Everything executed previously (that might be still displayed) is unchanged.
- Several commands can be typed in the same line. This is done by typing a comma between the commands. When the **Enter** key is pressed, the commands are executed in order from left to right.
- It is not possible to go back to a previous line that is displayed in the Command

Window, make a correction, and then re-execute the command.

- A previously typed command can be recalled to the command prompt with the up-arrow key (\uparrow). When the command is displayed at the command prompt, it can be modified if needed and then executed. The down-arrow key (\downarrow) can be used to move down the list of previously typed commands.
- If a command is too long to fit in one line, it can be continued to the next line by typing three periods ... (called an ellipsis) and pressing the **Enter** key. The continuation of the command is then typed in the new line. The command can continue line after line up to a total of 4,096 characters.

The semicolon (;):

When a command is typed in the Command Window and the **Enter** key is pressed, the command is executed. Any output that the command generates is displayed in the Command Window. If a semicolon (;) is typed at the end of a command, the output of the command is not displayed. Typing a semicolon is useful when the result is obvious or known, or when the output is very large.

If several commands are typed in the same line, the output from any of the commands will not be displayed if a semicolon instead of a comma is typed between the commands.

Typing %:

When the symbol % (percent) is typed at the beginning of a line, the line is designated as a comment. This means that when the **Enter** key is pressed the line is not executed. The % character followed by text (comment) can also be typed after a command (in the same line). This has no effect on the execution of the command.

Usually there is no need for comments in the Command Window. Comments, however, are frequently used in a program to add descriptions or to explain the program (see Chapters 4 and 6).

The `clc` command:

The `clc` command (type `clc` and press **Enter**) clears the Command Window. After typing in the Command Window for a while, the display may become very long. Once the `clc` command is executed, a clear window is displayed. The command does not change anything that was done before. For example, if some variables were defined previously (see Section 1.6), they still exist and can be used. The up-arrow key can also be used to recall commands that were typed before.

The Command History Window:

The Command History Window lists the commands that have been entered in the Command Window. This includes commands from previous sessions. A command in the Command History Window can be used again in the Command Window. By double-clicking on the command, the command is reentered in the Command Window and executed. It is also possible to drag the command to the Command Window, make changes if needed, and then execute it. The list

in the Command History Window can be cleared by selecting the lines to be deleted and then right-clicking the mouse and selecting **Delete Selection**. The whole history can be deleted by right-clicking the mouse and selecting choose **Clear Command History** in the menu that opens.

1.3 ARITHMETIC OPERATIONS WITH SCALARS

In this chapter we discuss only arithmetic operations with scalars, which are numbers. As will be explained later in the chapter, numbers can be used in arithmetic calculations directly (as with a calculator) or they can be assigned to variables, which can subsequently be used in calculations. The symbols of arithmetic operations are:

<u>Operation</u>	<u>Symbol</u>	<u>Example</u>
Addition	+	$5 + 3$
Subtraction	–	$5 - 3$
Multiplication	*	$5 * 3$
Right division	/	$5 / 3$
Left division	\	$5 \setminus 3 = 3 / 5$
Exponentiation	^	$5 \wedge 3$ (means $5^3 = 125$)

It should be pointed out here that all the symbols except the left division are the same as in most calculators. For scalars, the left division is the inverse of the right division. The left division, however, is mostly used for operations with arrays, which are discussed in Chapter 3.

1.3.1 Order of Precedence

MATLAB executes the calculations according to the order of precedence displayed below. This order is the same as used in most calculators.

<u>Precedence</u>	<u>Mathematical Operation</u>
First	Parentheses. For nested parentheses, the innermost are executed first.
Second	Exponentiation.
Third	Multiplication, division (equal precedence).
Fourth	Addition and subtraction.

In an expression that has several operations, higher-precedence operations are executed before lower-precedence operations. If two or more operations have the same precedence, the expression is executed from left to right. As illustrated in the next section, parentheses can be used to change the order of calculations.

1.3.2 Using MATLAB as a Calculator

The simplest way to use MATLAB is as a calculator. This is done in the Command Window by typing a mathematical expression and pressing the **Enter** key. MATLAB calculates the expression and responds by displaying `ans =` followed by the numerical result of the expression in the next line. This is demonstrated in Tutorial 1-1.

Tutorial 1-1: Using MATLAB as a calculator.

```

>> 7+8/2
ans =
    11
>> (7+8)/2
ans =
    7.5000
>> 4+5/3+2
ans =
    7.6667
>> 5^3/2
ans =
    62.5000
>> 27^(1/3)+32^0.2
ans =
     5
>> 27^1/3+32^0.2
ans =
    11
>> 0.7854-(0.7854)^3/(1*2*3)+0.785^5/(1*2*3*4*5)...
- (0.785)^7/(1*2*3*4*5*6*7)
ans =
    0.7071
>>

```

Annotations:

- For `7+8/2`: Type and press **Enter**. `8/2` is executed first.
- For `(7+8)/2`: Type and press **Enter**. `7+8` is executed first.
- For `4+5/3+2`: `5/3` is executed first.
- For `5^3/2`: `5^3` is executed first, `/2` is executed next.
- For `27^(1/3)+32^0.2`: `1/3` is executed first, `27^(1/3)` and `32^0.2` are executed next, and `+` is executed last.
- For `27^1/3+32^0.2`: `27^1` and `32^0.2` are executed first, `/3` is executed next, and `+` is executed last.
- For the long expression: Type three periods ... (and press **Enter**) to continue the expression on the next line.
- For the final result: The last expression is the first four terms of the Taylor series for $\sin(\pi/4)$.

1.4 DISPLAY FORMATS

The user can control the format in which MATLAB displays output on the screen. In Tutorial 1-1, the output format is fixed-point with four decimal digits (called `short`), which is the default format for numerical values. The format can

be changed with the `format` command. Once the `format` command is entered, all the output that follows is displayed in the specified format. Several of the available formats are listed and described in Table 1-2.

MATLAB has several other formats for displaying numbers. Details of these formats can be obtained by typing `help format` in the Command Window. The format in which numbers are displayed does not affect how MATLAB computes and saves numbers.

Table 1-2: Display formats

Command	Description	Example
<code>format short</code>	Fixed-point with 4 decimal digits for: $0.001 \leq \text{number} \leq 1000$ Otherwise display format <code>short e</code> .	<pre>>> 290/7 ans = 41.4286</pre>
<code>format long</code>	Fixed-point with 15 decimal digits for: $0.001 \leq \text{number} \leq 100$ Otherwise display format <code>long e</code> .	<pre>>> 290/7 ans = 41.428571428571431</pre>
<code>format short e</code>	Scientific notation with 4 decimal digits.	<pre>>> 290/7 ans = 4.1429e+001</pre>
<code>format long e</code>	Scientific notation with 15 decimal digits.	<pre>>> 290/7 ans = 4.142857142857143e+001</pre>
<code>format short g</code>	Best of 5-digit fixed or floating point.	<pre>>> 290/7 ans = 41.429</pre>
<code>format long g</code>	Best of 15-digit fixed or floating point.	<pre>>> 290/7 ans = 41.4285714285714</pre>
<code>format bank</code>	Two decimal digits.	<pre>>> 290/7 ans = 41.43</pre>
<code>format compact</code>	Eliminates blank lines to allow more lines with information displayed on the screen.	
<code>format loose</code>	Adds blank lines (opposite of compact).	

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
>> sqrt(50+14*3)
ans =
    9.5917
>> sqrt(54+9*sqrt(100))
ans =
    12
>> (15+600/4)/sqrt(121)
ans =
    15
>>
```

Argument is a number.

Argument is an expression.

Argument includes a function.

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>

Table 1-5: Rounding functions (Continued)

Function	Description	Example
<code>fix(x)</code>	Round toward zero.	<pre>>> fix(13/5) ans = 2</pre>
<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;
```

A value of 72 is assigned to the variable `ABB`.

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

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

```
>> ABB
```

```
ABB =  
     9
```

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 $2.2204\text{e-}016$.

`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.

Command

`clear`

Outcome

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 open. An open Editor/Debugger Window is shown in Figure 1-6.

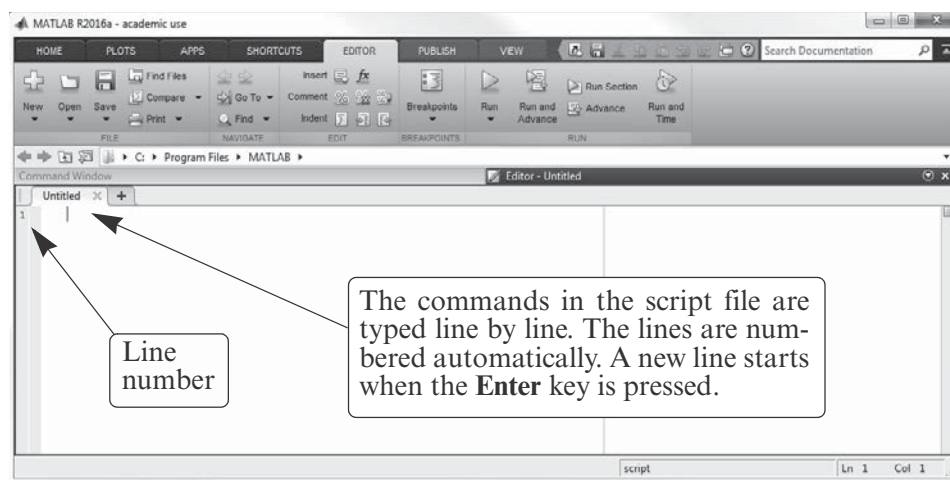


Figure 1-6: The Editor/Debugger Window.

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.

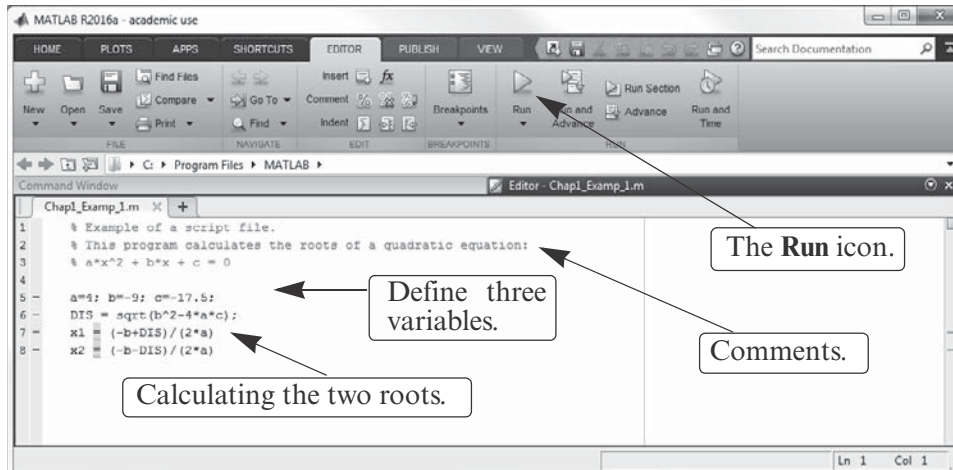


Figure 1-7: A program typed in the Editor/Debugger Window.

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 **Current Folder** field in the Command Window. The cur-

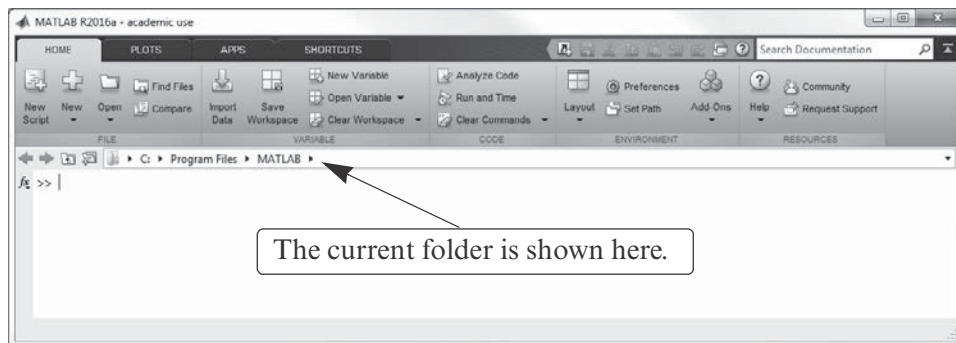


Figure 1-8: The Current folder field in the Command Window.

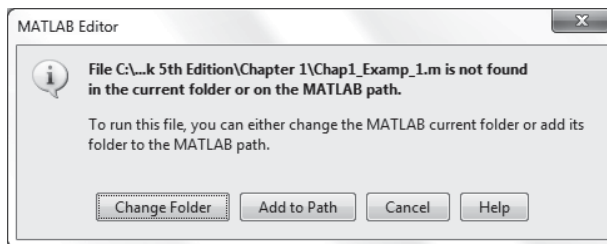


Figure 1-9: Changing the current directory.

rent 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.

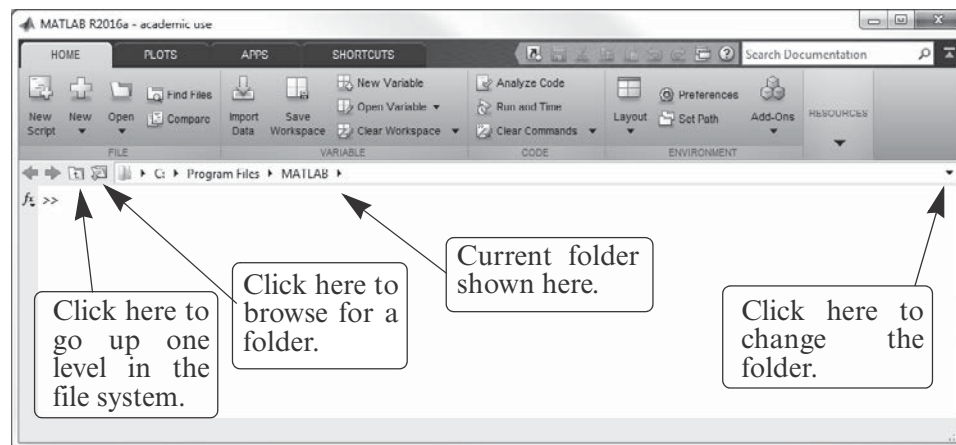


Figure 1-10: The Current Folder Window.

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_Example1
```

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

```
x1 =
    3.5000
x2 =
   -1.2500
```

← 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;
```

Define x.

```
>> LHS=cos(x/2)^2
```

Calculate the left-hand side.

```
LHS =
    0.9045
```

```
>> RHS=(tan(x)+sin(x))/(2*tan(x))
```

Calculate the right-hand side.

```
RHS =
    0.9045
```

Sample Problem 1-2: Geometry and trigonometry

Four circles are placed as shown in the figure. At each point where two circles are in contact, they are tangent to each other. Determine the distance between the centers C_2 and C_4 .

The radii of the circles are:

$$R_1 = 16 \text{ mm}, \quad R_2 = 6.5 \text{ mm}, \quad R_3 = 12 \text{ mm}, \quad \text{and} \\ R_4 = 9.5 \text{ mm}.$$

Solution

The lines that connect the centers of the circles create four triangles. In two of the triangles, $\Delta C_1 C_2 C_3$ and $\Delta C_1 C_3 C_4$, the lengths of all the sides are known. This information is used to calculate the angles γ_1 and γ_2 in these triangles by using the law of cosines. For example, γ_1 is calculated from:

$$(C_2 C_3)^2 = (C_1 C_2)^2 + (C_1 C_3)^2 - (C_1 C_2)(C_1 C_3) \cos \gamma_1$$

Next, the length of the side $C_2 C_4$ is calculated by considering the triangle $\Delta C_1 C_2 C_4$. This is done, again, by using the law of cosines (the lengths $C_1 C_2$ and $C_1 C_4$ are known and the angle γ_3 is the sum of the angles γ_1 and γ_2).

The problem is solved by writing the following program in a script file:

```
% Solution of Sample Problem 1-2
```

```
R1=16; R2=6.5; R3=12; R4=9.5;
```

```
C1C2=R1+R2; C1C3=R1+R3; C1C4=R1+R4;
```

```
C2C3=R2+R3; C3C4=R3+R4;
```

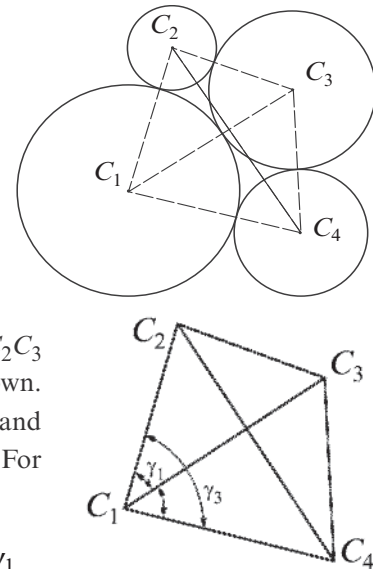
```
Gama1=acos((C1C2^2+C1C3^2-C2C3^2)/(2*C1C2*C1C3));
```

```
Gama2=acos((C1C3^2+C1C4^2-C3C4^2)/(2*C1C3*C1C4));
```

```
Gama3=Gama1+Gama2;
```

```
C2C4=sqrt(C1C2^2+C1C4^2-2*C1C2*C1C4*cos(Gama3))
```

Calculate the length of side $C_2 C_4$.



Define the R 's.

Calculate the lengths of the sides.

Calculate γ_1 , γ_2 , and γ_3 .

When the script file is executed, the following (the value of the variable $C2C4$) is displayed in the Command Window:

```
C2C4 =
    33.5051
```

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.

Sample Problem 1-4: Compounded interest

The balance B of a savings account after t years when a principal P is invested at an annual interest rate r and the interest is compounded n times a year is given by:

$$B = P \left(1 + \frac{r}{n} \right)^{nt} \quad (1)$$

If the interest is compounded yearly, the balance is given by:

$$B = P(1+r)^t \quad (2)$$

Suppose \$5,000 is invested for 17 years in one account for which the interest is compounded yearly. In addition, \$5,000 is invested in a second account in which the interest is compounded monthly. In both accounts the interest rate is 8.5%. Use MATLAB to determine how long (in years and months) it would take for the balance in the second account to be the same as the balance of the first account after 17 years.

Solution

Follow these steps:

- Calculate B for \$5,000 invested in a yearly compounded interest account after 17 years using Equation (2).
- Calculate t for the B calculated in part (a), from the monthly compounded

interest formula, Equation (1).

(c) Determine the number of years and months that correspond to t .

The problem is solved by writing the following program in a script file:

```
% Solution of Sample Problem 1-4
```

```
P=5000; r=0.085; ta=17; n=12;
```

```
B=P*(1+r)^ta
```

Step (a): Calculate B from Eq. (2).

```
t=log(B/P)/(n*log(1+r/n))
```

Step (b): Solve Eq. (1)
for t , and calculate t .

```
years=fix(t)
```

Step (c): Determine the number of years.

```
months=ceil((t-years)*12)
```

Determine the number of months.

When the script file is executed, the following (the values of the variables B, t , years, and months) is displayed in the Command Window:

```
>> format short g
```

```
B =
```

```
20011
```

```
t =
```

```
16.374
```

```
years =
```

```
16
```

```
months =
```

```
5
```

The values of the variables B, t , years, and months are displayed (since a semicolon was not typed at the end of any of the commands that calculate the values).

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:

$$(a) \left(5 - \frac{19}{7} + 2.5^3\right)^2$$

$$(b) 7 \times 3.1 + \frac{\sqrt{120}}{5} - 15^{5/3}$$

2. Calculate:

$$(a) \sqrt[3]{8 + \frac{80}{2.6}} + e^{3.5}$$

$$(b) \left(\frac{1}{\sqrt{75}} + \frac{73}{3.1^3}\right)^{1/4} + 55 \times 0.41$$

3. Calculate:

$$(a) \frac{23 + \sqrt[3]{45}}{16 \times 0.7} + \log_{10} 589006 \quad (b) (36.1 - 2.25\pi)(e^{2.3} + \sqrt{20})$$

4. Calculate:

$$(a) \frac{3.8^2}{2.75 - 41 \times 25} + \frac{5.2 + 1.8^5}{\sqrt{3.5}} \quad (b) \frac{2.1 \times 10^6 - 15.2 \times 10^5}{3 \cdot \sqrt[3]{6 \times 10^{11}}}$$

5. Calculate:

$$(a) \frac{\sin(0.2\pi)}{\cos(\pi/6)} + \tan 72^\circ \quad (b) (\tan 64^\circ \cos 15^\circ)^2 + \frac{\sin^2 37^\circ}{\cos^2 20^\circ}$$

6. Define the variable z as $z = 4.5$; then evaluate:

$$(a) 0.4z^4 + 3.1z^2 - 162.3z - 80.7 \quad (b) (z^3 - 23) / \left(\sqrt[3]{z^2 + 17.5} \right)$$

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

$$(a) \frac{1}{2}e^{2t} - 3.81t^3 \quad (b) \frac{6t^2 + 6t - 2}{t^2 - 1}$$

8. Define the variables x and y as $x = 6.5$ and $y = 3.8$; then evaluate:

$$(a) (x^2 + y^2)^{2/3} + \frac{xy}{y-x} \quad (b) \frac{\sqrt{x+y}}{(x-y)^2} + 2x^2 - xy^2$$

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

$$c = 4.6, d = 1.7, a = cd^2, \text{ and } b = \frac{c+a}{c-d}; \text{ then evaluate:}$$

$$(a) e^{d-b} + \sqrt[3]{c+a} - (ca)^d \quad (b) \frac{d}{c} + \left(\frac{ct}{b} \right)^2 - c^d - \frac{a}{b}$$

10. Two trigonometric identities are given by:

$$(a) \cos^2 x - \sin^2 x = 1 - 2\sin^2 x \quad (b) \frac{\tan x}{\sin x - 2 \tan x} = \frac{1}{\cos x - 2}$$

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = \pi / 10$.

11. Two trigonometric identities are given by:

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

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = 20^\circ$.

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

$$\tan(\alpha + \beta) = \frac{\tan \alpha + \tan \beta}{1 - \tan \alpha \tan \beta}$$

13. Given: $\int x^2 \cos x \, dx = 2x \cos x + (x^2 - 2) \sin x$. Use MATLAB to calculate the following definite integral: $\int_{\pi/6}^{\pi/3} x^2 \cos x \, dx$.

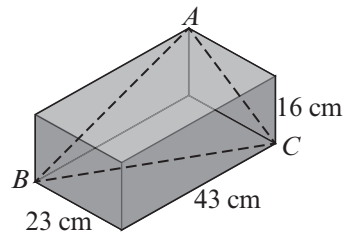
14. A rectangular box has the dimensions shown.
(a) Determine the angle BAC to the nearest degree.

- (b) Determine the area of the triangle ABC to the nearest tenth of a centimeter.

Law of cosines: $c^2 = a^2 + b^2 - 2ab \cos \gamma$

Heron's formula for triangular area:

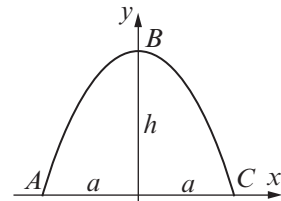
$$A = \sqrt{p(p-a)(p-b)(p-c)}, \text{ where } p = (a+b+c)/2.$$



15. The arc length of a segment of a parabola ABC is given by:

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

Determine L_{ABC} if $a=8$ in. and $h=13$ in.



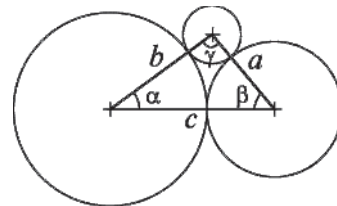
16. The three shown circles, with radius 15 in., 10.5 in., and 4.5 in., are tangent to each other.

- (a) Calculate the angle γ (in degrees) by using the law of cosines.

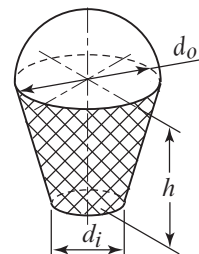
(Law of cosines: $c^2 = a^2 + b^2 - 2ab \cos \gamma$)

- (b) Calculate the angles γ and α (in degrees) using the law of sines.

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



17. A frustum of cone is filled with ice cream such that the portion above the cone is a hemisphere. Define the variables $d_i=1.25$ in., $d_o=2.25$ in., $h=2$ in., and determine the volume of the ice cream.



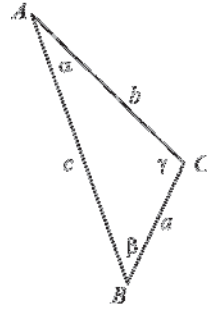
18. In the triangle shown $a = 27$ in., $b = 43$ in., and $c = 57$ in. Define a , b , and c as variables, and then:

(a) Calculate the angles α , β , and γ by substituting the variables in the law of cosines.

(Law of cosines: $c^2 = a^2 + b^2 - 2ab \cos \gamma$)

(b) Verify the law of tangents by substituting the results into the right and left sides of:

$$\text{law of tangents: } \frac{b-c}{b+c} = \frac{\tan \left[\frac{1}{2}(\beta-\gamma) \right]}{\tan \left[\frac{1}{2}(\beta+\gamma) \right]}$$



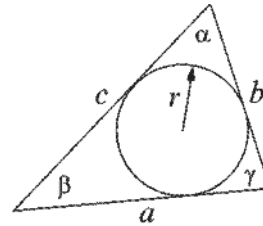
19. For the triangle shown, $\alpha = 72^\circ$, $\beta = 43^\circ$, and its perimeter is $p = 114$ mm. Define α , β , and p , as variables, and then:

(a) Calculate the triangle sides (Use the law of sines).

(b) Calculate the radius r of the circle inscribed in the triangle using the formula:

$$r = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}}$$

where $s = (a + b + c) / 2$.



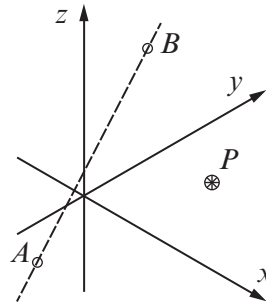
20. The distance d from a point $P(x_P, y_P, z_P)$ to the line that passes through the two points $A(x_A, y_A, z_A)$ and $B(x_B, y_B, z_B)$ can be calculated by $d = 2S / r$ where r is the distance between the points A and B , given by

$r = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$ and S is the area of the triangle defined by the three points calculated by $S = \sqrt{s_1^2 + s_2^2 + s_3^2} / 2$ where

$$s_1 = x_P y_A + x_A y_B + x_B y_P - (y_P x_A + y_A x_B + y_B x_P)$$

$$s_2 = y_P z_A + y_A z_B + y_B z_P - (z_P y_A + z_A y_B + z_B y_P)$$

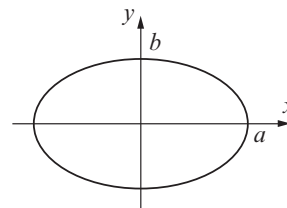
$s_3 = x_P z_A + x_A z_B + x_B z_P - (z_P x_A + z_A x_B + z_B x_P)$. Determine the distance of point $P(2, 6, -1)$ from the line that passes through point $A(-2, -1.5, -3)$ and point $B(-2.5, 6, 4)$. First define the variables $x_P, y_P, z_P, x_A, y_A, z_A, x_B, y_B$, and z_B , and then use the variable to calculate s_1, s_2, s_3 , and r . Finally calculate S and d .



21. The perimeter of an ellipse can be approximated by:

$$P = \pi(a+b) \left(3 - \frac{\sqrt{(3a+b)(a+3b)}}{a+b} \right)$$

Calculate the perimeter of an ellipse with $a = 18$ in. and $b = 7$ in.

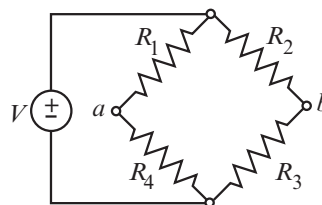


22. A total of 4217 eggs have to be packed in boxes that can hold 36 eggs each. By typing one line (command) in the Command Window, calculate how many eggs will remain unpacked if every box that is used has to be full. (Hint: Use MATLAB built-in function `fix`.)
23. A total of 777 people have to be transported using buses that have 46 seats and vans that have 12 seats. Calculate how many buses are needed if all the buses have to be full, and how many seats will remain empty in the vans if enough vans are used to transport all the people that did not fit into the buses. (Hint: Use MATLAB built-in functions `fix` and `ceil`.)
24. Change the display to `format long g`. Assign the number 7E8/13 to a variable, and then use the variable in a mathematical expression to calculate the following by typing one command:
- Round the number to the nearest tenth.
 - Round the number to the nearest million.

25. The voltage difference V_{ab} between points a and b in the Wheatstone bridge circuit is given by:

$$V_{ab} = V \left(\frac{c-d}{(c+1)(d+1)} \right)$$

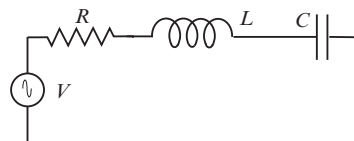
where $c = R_2 / R_1$ and $d = R_3 / R_4$. Calculate the V_{ab} if $V = 15$ V, $R_1 = 119.8 \, \Omega$, $R_2 = 120.5 \, \Omega$, $R_3 = 121.2 \, \Omega$ and $R_4 = 119.3 \, \Omega$



26. The current in a series RCL circuit is given by:

$$I = \frac{V}{\sqrt{R^2 + \left(\omega L - \frac{1}{\omega C} \right)^2}}$$

where $\omega = 2\pi f$. Calculate I for the circuit shown if the supply voltage is 80 V, $f = 50$ Hz, $R = 6 \, \Omega$, $L = 400 \times 10^{-3}$ H, and $C = 40 \times 10^{-6}$ F.



27. The monthly payment M of a mortgage P for n years with a fixed annual interest rate r can be calculated by the formula:

$$M = P \frac{\frac{r}{12} \left(1 + \frac{r}{12}\right)^{12n}}{\left(1 + \frac{r}{12}\right)^{12n} - 1}$$

Determine the monthly payment of a 30-year \$450,000 mortgage with interest rate of 4.2% ($r = 0.042$). Define the variables P , r , and n and then use them in the formula to calculate M .

28. The number of permutations ${}_nP_r$ of taking r objects out of n objects without repetition is given by:

$${}_nP_r = \frac{n!}{(n-r)!}$$

- (a) Determine how many six-letter passwords can be formed from the 26 letters in the English alphabet if a letter can only be used once.
 (b) How many passwords can be formed if the digits 0, 1, 2, ..., 9 can be used in addition to the letters.

29. The number of combinations $C_{n,r}$ of taking r objects out of n objects is given by:

$$C_{n,r} = \frac{n!}{r!(n-r)!}$$

In the Powerball lottery game the player chooses five numbers from 1 through 59, and then the Powerball number from 1 through 35.

Determine how many combinations are possible by calculating $C_{59,5} C_{35,1}$. (Use the built-in function `factorial`.)

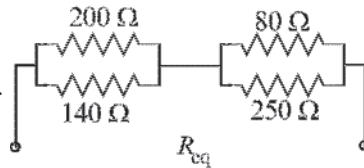
30. The equivalent resistance of two resistors R_1 and R_2 connected in parallel is given by

$$R_{eq} = \frac{R_1 R_2}{R_1 + R_2}$$

The equivalent resistance of two resistors R_1 and R_2 connected in series

is given by $R_{eq} = R_1 + R_2$. Determine the

equivalent resistance of the four resistors in the circuit shown in the figure.

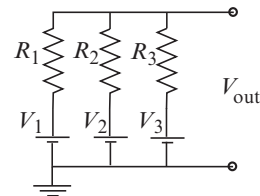


31. The output voltage V_{out} in the circuit shown is given by (Millman's theorem):

$$V_{out} = \frac{\frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3}}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Calculate V_{out} given $V_1 = 36\text{ V}$, $V_2 = 28\text{ V}$, $V_3 = 24\text{ V}$,

$R_1 = 400\ \Omega$, $R_2 = 200\ \Omega$, $R_3 = 600\ \Omega$.



32. Radioactive decay of carbon-14 is used for estimating the age of organic material. The decay is modeled with the exponential function $f(t) = f(0)e^{kt}$, where t is time, $f(0)$ is the amount of material at $t = 0$, $f(t)$ is the amount of material at time t , and k is a constant. Carbon-14 has a half-life of approximately 5,730 years. A sample taken from the ancient footprints of Acahualinca in Nicaragua shows that 77.45% of the initial ($t = 0$) carbon-14 is present. Determine the estimated age of the footprint. Solve the problem by writing a program in a script file. The program first determines the constant k , then calculates t for $f(t) = 0.7745f(0)$, and finally rounds the answer to the nearest year.
33. The greatest common divisor is the largest positive integer that divides the numbers without a remainder. For example, the greatest common divisor of 8 and 12 is 4. Use the MATLAB Help Window to find a MATLAB built-in function that determines the greatest common divisor of two numbers. Then use the function to show that the greatest common divisor of:
- (a) 91 and 147 is 7.
 - (b) 555 and 962 is 37.

34. The amount of energy E (in joules) that is released by an earthquake is given by:

$$E = 1.74 \times 10^{19} \times 10^{1.44M}$$

where M is the magnitude of the earthquake on the Richter scale.

- (a) Determine the energy that was released from the Anchorage earthquake (1964, Alaska, USA), magnitude 9.2.
 - (b) The energy released in Lisbon earthquake (Portugal) in 1755 was one-half the energy released in the Anchorage earthquake. Determine the magnitude of the earthquake in Lisbon on the Richter scale.
35. According to the Doppler effect of light, the perceived wavelength λ_p of a light source with a wavelength of λ_s is given by:

$$\lambda_p = \lambda_s \sqrt{\frac{1 - \frac{v}{c}}{1 + \frac{v}{c}}}$$

where c is the speed of light (about 300×10^6 m/s) and v is the speed the observer moves toward the light source. Calculate the speed the observer has to move in order to see a red light as green. Green wavelength is 530 nm and red wavelength is 630 nm.

36. Newton's law of cooling gives the temperature $T(t)$ of an object at time t in terms of T_0 , its temperature at $t = 0$, and T_s , the temperature of the surroundings.

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

A police officer arrives at a crime scene in a hotel room at 9:18 PM, where he finds a dead body. He immediately measures the body's temperature and finds it to be 79.5°F. Exactly one hour later he measures the temperature again and finds it to be 78.0°F. Determine the time of death, assuming that victim body temperature was normal (98.6°F) prior to death and that the room temperature was constant at 69°F.

37. The velocity v and the falling distance d as a function of time of a skydiver that experience the air resistance can be approximated by:

$$v(t) = \sqrt{\frac{mg}{k}} \tanh\left(\sqrt{\frac{kg}{m}} t\right) \quad \text{and} \quad d(t) = \frac{m}{k} \ln \left[\cosh\left(\sqrt{\frac{kg}{m}} t\right) \right]$$

where $k = 0.24 \text{ kg/m}$ is a constant, m is the skydiver mass, $g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity, and t is the time in seconds since the skydiver starts falling. Determine the velocity and the falling distance at $t = 8 \text{ s}$ for a 95-kg skydiver

38. Use the Help Window to find a display format that displays the output as a ratio of integers. For example, the number 3.125 will be displayed as 25/8. Change the display to this format and execute the following operations:

$$(a) \quad 5/8 + 16/6 \quad (b) \quad 1/3 - 11/13 + 2.7^2$$

39. Gosper's approximation for factorials is given by:

$$n! = \sqrt{\left(2n + \frac{1}{3}\right)} \pi n^n e^{-n}$$

Use the formula for calculating 19!. Compare the result with the true value obtained with MATLAB's built-in function `factorial` by calculating the error ($\text{Error} = (\text{True Val} - \text{Approx Val}) / \text{True Val}$).

40. According to Newton's law of universal gravitation, the attraction force between two bodies is given by:

$$F = G \frac{m_1 m_2}{r^2}$$

where m_1 and m_2 are the masses of the bodies, r is the distance between the bodies, and $G = 6.67 \times 10^{-11} \text{ N-m}^2/\text{kg}^2$ is the universal gravitational constant. Determine how many times the attraction force between the sun and the Earth is larger than the attraction force between the Earth and the moon. The distance between the sun and Earth is $149.6 \times 10^9 \text{ m}$, the distance between the moon and Earth is $384.4 \times 10^6 \text{ m}$, $m_{\text{Earth}} = 5.98 \times 10^{28} \text{ kg}$, $m_{\text{sun}} = 2.0 \times 10^{30} \text{ kg}$, and $m_{\text{moon}} = 7.36 \times 10^{22} \text{ kg}$.