Contents

1 Introduction to MATLAB

MATLAB is an interactive, numerical computation program. It has powerful built-in routines for enabling a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. In some installations MATLAB will also have a Symbolic Toolbox which allows MATLAB to perform symbolic calculations as well as numerical calculations. In this chapter we will describe how MATLAB handles simple numerical expressions and mathematical formulas.

MATLAB is available on almost every computer system. Its interface is similar regardless of the system being used. In this Manual we are going to assume that the user has sufficient understanding of the computer he or she is using to start up MATLAB. At any rate we will assume that you have started up MATLAB, and that you are now faced with a window on your computer which contains the MATLAB prompt[1], `>>`, and a cursor waiting for you to do something. This is called the MATLAB Command Window, and it is time to begin.

1.1 Numerical Expressions

In its most elementary use, MATLAB is an extremely powerful calculator, with many built-in functions, and a very large and easily accessible memory. Let's start at the very beginning. Suppose you want to calculate a number such as $12.3(48.5 + \frac{342}{39})$. You can accomplish this using MATLAB by entering `12.3*(48.5+342/39)`. Try it. You should get the following:

```
>> 12.3*(48.5+342/39)
ans =
  704.4115
```

Notice that what you enter into MATLAB does not differ greatly from what you would write on a piece of paper. The only changes from the algebra that you use every day are the different symbols used for the algebraic operations. These are standard in the computer world, and are made necessary by the unavailability of the standard symbols on a keyboard. A partial list of MATLAB's algebraic operators is shown in Table 1.1.

Symbol	**Meaning**
$+$	addition
$-$	subtraction
$*$	multiplication
/	right division
\	left division
^	exponentiation

Table 1.1 MATLAB's basic operators.

While $+$ and $-$ have their standard meanings, $*$ is used to indicate multiplication. You will notice that division can be indicated in two ways. The fraction $\frac{2}{3}$ can be indicated in MATLAB as either `2/3` or as `3\2`. These are referred to as right division and left division, respectively.

[1] In the narrative that follows, readers are expected to enter the text that appears after the command prompt, `>>`. You must press the **Enter** or **Return** key to execute the command.

```
>> 2/3
ans =
    0.6667
>> 3\2
ans =
    0.6667
```

Exponentiation is quite different in MATLAB; it has to be, since MATLAB has no way of entering superscripts. Consequently, the power 4^3 must be entered as `4^3`

```
>> 4^3
ans =
    64
```

The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. Exponentiations are done first, followed by multiplications and divisions, and finally by additions and subtractions. The standard order of precedence of arithmetic operations can be changed by inserting parentheses. For example, the result of `12.3*(48.5+342)/39` is quite different than the similar expression we computed earlier, as you will discover if you try it.

MATLAB allows the assignment of numerical values to variable names. For example, if you enter

```
>> x=3
x =
     3
```

then MATLAB will remember that `x` stands for `3` in subsequent computations. Therefore, computing `2.5*x` will result in

```
>> 2.5*x
ans =
    7.5000
```

You can also assign names to the results of computations. For example,

```
>> y=(x+2)^3
y =
   125
```

will result in `y` being given the value $(3 + 2)^3 = 125$.

You will have noticed that if you do not assign a name for a computation, MATLAB will assign the default name **ans** to the result. This name can always be used to refer to the results of the previous computation. For example:

```
>> 2+3
ans =
     5

>> ans/5
ans =
     1
```

MATLAB has a number of preassigned variables or constants. The constant $\pi = 3.14159...$ is given the name `pi`.

```
>> pi
ans =
    3.1416
```

There is no symbol for e, the base of the natural logarithms, but this can be easily computed as `exp(1)`.

```
>> exp(1)
ans =
    2.7183
```

1.2 Mathematical Functions

There is a long list of mathematical functions that are built into MATLAB. Included are all of the functions that are standard in calculus courses.

Elementary Functions

`abs(x)`	The absolute value of x, i.e., $\|x\|$.
`sqrt(x)`	The square root of x, i.e., $\sqrt{x}$.
`sign(x)`	The signum of x, i.e., 0 if $x = 0$, -1 if $x < 0$, and $+1$ if $x > 0$.

The Trigonometric Functions

`sin(x)`	The sine of x, i.e., $\sin(x)$.
`cos(x)`	The cosine of x, i.e., $\cos(x)$.
`tan(x)`	The tangent of x, i.e., $\tan(x)$.
`cot(x)`	The cotangent of x, i.e., $\cot(x)$.
`sec(x)`	The secant of x, i.e., $\sec(x)$.
`csc(x)`	The cosecant of x, i.e., $\csc(x)$.

The Inverse Trigonometric Functions

`asin(x)`	The inverse sine of x, i.e., $\arcsin(x)$ or $\sin^{-1}(x)$.
`acos(x)`	The inverse cosine of x, i.e., $\arccos(x)$ or $\cos^{-1}(x)$.
`atan(x)`	The inverse tangent of x, i.e., $\arctan(x)$ or $\tan^{-1}(x)$.
`acot(x)`	The inverse cotangent of x, i.e., $\mathrm{arccot}(x)$ or $\cot^{-1}(x)$.
`asec(x)`	The inverse secant of x, i.e., $\mathrm{arcsec}(x)$ or $\sec^{-1}(x)$.
`acsc(x)`	The inverse cosecant of x, i.e., $\mathrm{arccsc}(x)$ or $\csc^{-1}(x)$.

The Exponential and Logarithm Functions

`exp(x)`	The exponential of x, i.e., e^x.
`log(x)`	The natural logarithm of x, i.e., $\ln(x)$
`log10(x)`	The logarithm of x to base 10, i.e., $\log_{10}(x)$.

The Hyperbolic Functions

`sinh(x)`	The hyperbolic sine of x, i.e., $\sinh(x)$.
`cosh(x)`	The hyperbolic cosine of x, i.e., $\cosh(x)$.

`tanh(x)`	The hyperbolic tangent of x, i.e., $\tanh(x)$.
`coth(x)`	The hyperbolic cotangent of x, i.e., $\coth(x)$.
`sech(x)`	The hyperbolic secant of x, i.e., $\operatorname{sech}(x)$.
`csch(x)`	The hyperbolic cosecant of x, i.e., $\operatorname{csch}(x)$.

The Inverse Hyperbolic Functions

`asinh(x)`	The inverse hyperbolic sine of x, i.e., $\sinh^{-1}(x)$.
`acosh(x)`	The inverse hyperbolic cosine of x, i.e., $\cosh^{-1}(x)$.
`atanh(x)`	The inverse hyperbolic tangent of x, i.e., $\tanh^{-1}(x)$.
`acoth(x)`	The inverse hyperbolic cotangent of x, i.e., $\coth^{-1}(x)$.
`asech(x)`	The inverse hyperbolic secant of x, i.e., $\operatorname{sech}^{-1}(x)$.
`acsch(x)`	The inverse hyperbolic cosecant of x, i.e., $\operatorname{csch}^{-1}(x)$.

For a more extensive list of the functions available, see the MATLAB *User's Guide,* or MATLAB's online documentation[2]. All of these functions can be entered at the MATLAB prompt either alone or in combination. For example, to calculate $\sin(x) - \ln(\cos(x))$, where $x = 6$, we simply enter

```
>> x=6
x =
     6

>> sin(x)-log(cos(x))
ans =
   -0.2388
```

Take special notice that $\ln(\cos(x))$ is entered as `log(cos(x))`. The function `log` is MATLAB's representation of the natural logarithm function.

1.3 Output Format

Up to now we have let MATLAB repeat everything that we enter at the prompt. Sometimes this is not useful, particularly when the output is pages in length. To prevent MATLAB from echoing what we tell it, simply enter a semicolon at the end of a command. For example, enter

```
>> q=7;
```

and then ask MATLAB what it thinks `q` is by entering

```
>> q
q =
     7
```

If you use MATLAB to compute $\cos(\pi)$, you get

[2] MATLAB 5 comes with extensive online help. Typing `helpdesk` at the MATLAB prompt should open MATLAB's helpdesk in your internet browser. You can also access MATLAB's standard help files by typing `help` at the MATLAB prompt. For a list of MATLAB's elementary functions, type `help elfun` at the MATLAB prompt. It might be a good idea to bookmark MATLAB's help file at this point in your browser. You are apt to return to this destination many times in the future.

```
>> cos(pi)
ans =
    -1
```

In this case MATLAB is smart enough to realize that the answer is an integer and it displays the answer in that form. However, cos(3) is not an integer, and MATLAB gives us `-0.9900` as its value. Thus, if MATLAB is not sure that a number is an integer, it displays five significant figures in its answer. As another example, 1.57 is very close to $\pi/2$, and $\cos(\pi/2) = 0$. MATLAB gives us

```
>> cos(1.57)
ans =
   7.9633e-004
```

This is an example of MATLAB's exponential, or scientific notation. It stands for 7.9633×10^{-4}, or 0.00079633. In this case MATLAB again displays five significant digits in its answer. All of these illustrate the default format, which is called the `short` format. It is important to realize that although MATLAB only displays five significant digits in the default format, it is computing the answer to an accuracy of sixteen significant figures.

There are several other formats.[3] We will discuss two of them. If it is necessary or desirable to have more significant digits displayed, enter `format long` at the MATLAB prompt. MATLAB will then display about sixteen significant digits. For example,

```
>> format long
>> cos(1.57)
ans =
    7.963267107332634e-004
```

There is another output format which we will find useful. If you enter `format rat`, then all numbers will be shown as rational numbers. This is called the rational format. If the numbers are actually irrational, MATLAB will find a very close rational approximation to the number.

```
>> cos(1.57)
ans =
    47/59021
```

The rational format is most useful when you are working with numbers you know to be rational.

After using a different format, you can return to the standard, short format by entering

```
>> format short
```

1.4 Functions

MATLAB provides a number of techniques for evaluating functions, one of which we will now share, called the *inline function*. Suppose, for example, that we wanted to evaluate the function $f(x) = x^2$ at $x = 1$. First create an inline function as follows.

---

[3] Type `help format` at the MATLAB prompt to get a complete description of the available formats.

```
>> f=inline('x^2','x')
f =
     Inline function:
     f(x) = x^2
```

Note the single apostrophes surrounding the formula, `'x^2'` and the independent variable, `'x'`. The single apostrophe is MATLAB's *string* delimiter.

One can now easily evaluate the function f at the required value.

```
>> f(1)
ans =
     1
```

In multivariable calculus, we frequently work with functions of more than one variable. For example, if $f(x, y, z) = x^2 + y^2 + z^2$, then $f(1, 2, 3) = 1^2 + 2^2 + 3^2 = 14$. MATLAB's inline function handles this with ease. Simply wrap each individual independent variable with string delimiters and separate them with commas.

```
>> f=inline('x^2+y^2+z^2','x','y','z')
f =
     Inline function:
     f(x,y,z) = x^2+y^2+z^2
>> f(1,2,3)
ans =
    14
```

1.5 Complex Arithmetic

MATLAB works as easily with complex numbers as it does with numbers that are real. We want to highlight those properties and laws of complex numbers that will allow us to work efficiently with MATLAB. With this thought in mind, we begin.

One of the solutions of $z^2 + 1 = 0$ is $z = i$. If you substitute this result back into the equation $z^2 + 1 = 0$, it is easily seen that $i^2 = -1$. It is now easily deduced that $i^3 = -i$ ($i^3 = i^2 i = -i$) and $i^4 = 1$. Indeed, $i^{4k} = 1$ when k is any integer.

```
>> i^16
ans =
     1
```

Define the complex numbers to be the set C, where

$$C = \{a + bi : a, b \in R\}.$$

That is, the set of complex numbers is all numbers of the form $a + bi$, where a and b are real numbers. If $z = a + bi$, then the *real* part of z is a, while the *imaginary part* of z is b. In symbols,

$$\text{Re}(z) = a \quad \text{and} \quad \text{Im}(z) = b.$$

MATLAB is well aware of the real and imaginary parts of complex numbers.

```
>> z=2-3i
z =
```

```
   2.0000 - 3.0000i
>> real(z)
ans =
     2
>> imag(z)
ans =
    -3
```

It is important to note that the square root of a negative number is a purely imaginary number. That is, if $a > 0$, then $\sqrt{-a} = \sqrt{a}\,i$.

```
>> sqrt(-9)
ans =
        0 + 3.0000i
```

When adding, subtracting, and multiplying complex numbers, the usual laws of algebra apply. One need only remember that $i^2 = -1$. For example,

$$(2+3i)(3-4i) = 6 - 8i + 9i - 12i^2 = 18 + i.$$

```
>> z=2+3i;w=3-4i;
>> z*w
ans =
  18.0000 + 1.0000i
```

If $z = a + bi$, then the *complex conjugate* of z is $a - bi$. In symbols, $\overline{z} = a - bi$. For example, if $z = 1 - 2i$, then $\overline{z} = 1 + 2i$. MATLAB knows all about complex conjugates.

```
>> z=1-2i;
>> conj(z)
ans =
   1.0000 + 2.0000i
```

One can use the conjugate to help find the quotient of two complex numbers. For example,

$$\frac{1+i}{1-i} = \frac{1+i}{1-i} \cdot \frac{1+i}{1+i} = \frac{2i}{2} = i.$$

Of course, MATLAB performs this computation flawlessly.

```
>> (1+i)/(1-i)
ans =
        0 + 1.0000i
```

The conjugate possesses a number of important properties, some of which are given in Table 1.2.

Most of these properties will be investigated in the exercises, but we'd like to pay particular attention to property 5 of Table 1.2, as it gives us a nice way to check whether a number is real or complex. The test is simple: a number is real if and only if its conjugate is itself. This leads us to MATLAB's *relational operators* which are listed in Table 1.3, along with their interpretations.

If a relation is true, MATLAB returns a 1, otherwise it returns a 0. Consequently, if `conj(z)==z` is a true statement, MATLAB returns a 1 and we know that `z` is a real number.

1. $\overline{z+w} = \overline{z} + \overline{w}$
2. $\overline{z-w} = \overline{z} - \overline{w}$
3. $\overline{zw} = \overline{z}\,\overline{w}$
4. $\overline{z^n} = \overline{z}^n$
5. $\overline{r} = r$, if r is a real number.

Table 1.2 Some properties of the conjugate.

Operator	Interpretation
`a<b`	True if a is less than b
`a<=b`	True if a is less than or equal to b
`a>b`	True if a is greater than b
`a>=b`	True if a is greater than or equal to b
`a==b`	True if a is equal to b
`a~=b`	True if a is not equal to b

Table 1.3 MATLAB's relational operators.

```
>> z=2;
>> conj(z)==z
ans =
     1
```

For contrast, if `conj(z)==z` is a false statement, MATLAB returns a 0 and we know that `z` is a complex number.

```
>> z=3+2i;
>> conj(z)==z
ans =
     0
```

One final note. Engineers and physicists frequently use i to represent current, so they prefer to use j for the square root of -1. MATLAB is well aware of this preference.

```
>> j
ans =
        0 + 1.0000i
```

1.6 Euler's Formula

In your previous calculus classes, you saw that Taylor's theorem can be used to provide power series expansions of common functions. One of the most important series in mathematics is the expansion for the exponential function.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots \qquad (1.1)$$

Equally important are series expansions for the sine and cosine.

$$\cos x = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots \tag{1.2}$$

$$\sin x = 1 - x + \frac{x^3}{3!} - \frac{x^5}{5!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots \tag{1.3}$$

Now evaluate $e^{i\theta}$ by substituting $i\theta$ for x in equation 1.1.

$$\begin{aligned} e^{i\theta} &= 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \frac{(i\theta)^4}{4!} + \frac{(i\theta)^5}{5!} + \frac{(i\theta)^6}{6!} + \cdots \\ &= \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \cdots\right) + i\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots\right) \\ &= \cos\theta + i\,\sin\theta \end{aligned}$$

The result, $e^{i\theta} = \cos\theta + i\,\sin\theta$, is known as *Euler's formula* and is one of the most powerful identities in mathematics. For example,

$$e^{i\pi/2} = \cos\frac{\pi}{2} + i\,\sin\frac{\pi}{2} = i.$$

```
>> exp(i*pi/2)
ans =
   0.0000 + 1.0000i
```

1.7 The Geometry of the Complex Numbers

Every complex number can be associated with a vector in the *complex plane*. In Figure 1.1, the horizontal axis is called the *real axis* and the vertical axis is called the *imaginary axis*. The complex number $z = a + bi$ is represented by a vector, with tail at $(0, 0)$ and the tip of its arrowhead at the point (a, b).

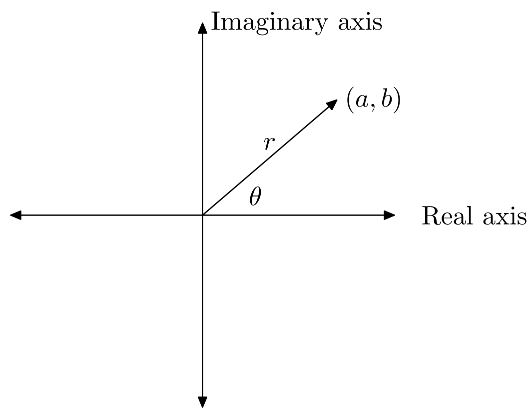

Figure 1.1 The complex plane.

If r represents the *magnitude* or length of the vector, then the Pythagorean theorem gives us $r = \sqrt{a^2 + b^2}$. Let θ represent the orientated angle from the positive real axis to the vector. Clearly, knowing the magnitude and angle uniquely determines the vector in the complex plane. One can easily determine both magnitude and angle with the following calculation.

$$\begin{aligned} a + bi &= \sqrt{a^2+b^2}\left(\frac{a}{\sqrt{a^2+b^2}} + i\,\frac{b}{\sqrt{a^2+b^2}}\right) \\ &= r(\cos\theta + i\,\sin\theta) \\ &= re^{i\theta} \end{aligned}$$

For example, the complex number $z = 1+i$ has magnitude $\sqrt{2}$ and angle $\pi/4$. Consequently,

$$1 + i = \sqrt{2}e^{i\pi/4}.$$

Matlab is well aware of magnitude and angle. Matlab's `angle` command returns an angle θ, with $-\pi < \theta \le \pi$.

```
>> z=1+i;
>> abs(z)
ans =
    1.4142
>> angle(z)
ans =
    0.7854
```

One can easily check that these numbers are equal to $\sqrt{2}$ and $\pi/4$, respectively.

The form $re^{i\theta}$ is called the *polar*, or *trigonometric form* of the complex number and is an extremely valuable geometric tool. For example, if we multiply the complex number $re^{i\alpha}$ by $e^{i\beta}$, then

$$re^{i\alpha}e^{i\beta} = re^{i(\alpha+\beta)},$$

a vector having magnitude r and angle $\alpha + \beta$. Consequently, multiplying any complex number by $e^{i\beta}$ simply rotates the complex number through an angle of β radians in the complex plane.

Taking roots is equally painless and also has an easy geometrical interpretation.

$$\left(re^{i\theta}\right)^{1/n} = r^{1/n}e^{i\theta/n}$$

Raising a complex number to the power $1/n$ is equivalent to taking the nth root of the magnitude and dividing the angle by n. For example, the complex number -1, or $-1 + 0i$, has magnitude 1 and angle π. Therefore, $(-1)^{1/3}$ should have magnitude 1 and angle $\pi/3$, i.e.,

$$(-1)^{1/3} = \left(1e^{i\pi}\right)^{1/3} = 1^{1/3}e^{i\pi/3} = \cos\frac{\pi}{3} + i\,\sin\frac{\pi}{3} = \frac{1}{2} + i\,\frac{\sqrt{3}}{2}.$$

This last computation is probably overkill, particularly if you are familiar with the unit circle (we hope you are). You should have a very good feel for the eventual location of the unit vector if you divide the angle by 3. Of course, Matlab knows all about this theory, so the computation

```
>> (-1)^(1/3)
ans =
   0.5000 + 0.8660i
```

yields an answer whose real part is $1/2$ and whose imaginary part is $\sqrt{3}/2$ (check this with Matlab).

Many people find this result surprising, thinking that the cube root of -1 ought to be -1. Well, they're both right and wrong. There are actually *three* different cube roots of -1. There is -1, the recently calculated $1/2+i\sqrt{3}/2$, and there is another cube root of -1, namely $1/2-i\sqrt{3}/2$.

```
>> (1/2-i*sqrt(3)/2)^3
ans =
  -1.0000 - 0.0000i
```

MATLAB always selects the so-called *principal* cube root, the first that it encounters as it searches in a counterclockwise direction on the unit circle.

One final thought. Plot each of the cube roots of -1 on the unit circle and note the symmetry. Note also, that the real cube root of -1 is second in line, as you travel in a counterclockwise direction from the real axis.

1.8 Recording Your Work

It is frequently useful to be able to record what happens in a MATLAB session. For example, in the process of preparing a homework submission, it should not be necessary to copy all of the output from the computer screen. You ought to be able to do this automatically. The MATLAB `diary` command makes this possible.

For example, suppose you are doing your first homework assignment and you want to record what you are doing in MATLAB. To do this, choose a name, perhaps `hw1`, for the file in which you wish to record the output. Then enter `diary hw1` at the MATLAB prompt. From this point on, everything that appears in the Command Window will also be recorded in the file `hw1`. When you want to stop recording enter `diary off`. If you want to start recording again, enter `diary on`.

The file that is created is a simple text file. It can be opened by an editor or a word processing program and edited to remove extraneous material, or to add your comments. You can also print the file to get a hard copy.

1.9 Exercises

1. Start MATLAB and begin your session with the command `>> pwd`. MATLAB will respond with the *present working directory*. If this directory is not where you wish to store your work, use MATLAB's `cd` command to change to a directory or folder of choice. For example, on a PC the command `cd c:\mywork` will change the current working directory to `c:\mywork`. You can also click the pathtool icon on the toolbar or enter `pathtool` at the MATLAB prompt which opens the Path Browser dialog box, after which you change the current directory interactively. Start a diary session with `>> diary hmwk1`. Read Chapter One of this manual again, but this time enter each of the commands in the narrative at the MATLAB prompt as you read. When you are finished, enter the command `>> diary off`. Open the file `hmwk1` in your favorite editor or word processor. Edit and correct any mistakes that you made. Save and print the edited file and submit the result to your instructor.

In exercises 2–7, use MATLAB to evaluate the given expression.

2. $\sqrt{(1.2)^2+(2.3)^2+(-3.1)^2}$

3. $\dfrac{1.2+2(3.4)}{\sqrt{(1.2)^2+(-1.5)^3+(3.4)^2}}$

4. $\sqrt{1-e^2\cos^2(2\pi/3)}$

5. $2\pi\cos(\pi/3)\sqrt{1+\sin^2(\pi/3)}$

6. $1+2\sin(\pi/12)$

7. $\dfrac{6}{1+2\cos(\pi/3)}$

Use MATLAB's inline function to evaluate the functions in exercises 8–17 at each given input.

8. $f(x)=\sqrt{16-x^2}$, at $x=0$, 1, 2, 3, 4, and 5.

9. $g(x)=\dfrac{x}{\sqrt{x^2-4}}$ at $x=3$, 4, 5, and 6.

10. $r(\theta)=e^\theta$ at $\theta=0$, $\pi/2$, and π.

11. $y(x)=\frac{1}{5}\cosh 5x$ at $x=0$, 1, and 2.

12. $r(t)=t\cos^{-1}t-\sqrt{1-t^2}$ at $t=\pi/4$, $\pi/5$, and π.

13. $h(u)=\frac{1}{4}\cosh^{-1}\frac{u}{4}$ at $u=0$, 1, and 2.

14. $f(x,y)=\ln(x-y)$ at $(x,y)=(4,3)$, $(3,4)$, and $(5,5)$.

15. $x(\phi,\theta)=2\cos\phi\sin\theta$ at $(\phi,\theta)=(pi/3,2\pi/3)$, $(\pi/2,\pi/2)$, and $(-\pi,0)$.

16. $T(x,y,z)=\sqrt{x^2+y^2+z^2}$ at $(x,y,z)=(1,2,3)$, $(-3,4,0)$, and $(-1,0,-2)$.

17. $h(r,\theta,\phi)=r\cos\theta\sin\phi$ at $(r,\theta,\phi)=(1,2\pi,\pi)$, $(2,-\pi/2,\pi/2)$, and $(2,0,\pi/3)$.

Use MATLAB to evaluate the complex expressions in exercises 18–23.

18. $\overline{(1+i)^2}$

19. $(-1+\sqrt{3}\,i)^3$

20. $\left(\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}\,i\right)^2$

21. $\dfrac{1+i}{2-3i}$

22. $\mathrm{Re}\left(e^{2\pi\,i/3}\right)$

23. $\mathrm{Im}\left(i^i\right)$

Let $z=2+3i$ and $w=1-2i$. Use MATLAB to verify each of the properties of the conjugate given in exercises 24–27.

24. $\overline{z+w}=\overline{z}+\overline{w}$

25. $\overline{z-w}=\overline{z}-\overline{w}$

26. $\overline{zw}=\overline{zw}$

27. $\overline{\dfrac{z}{w}}=\dfrac{\overline{z}}{\overline{w}}$

In exercises 28–33, test the validity of the given relation by evaluating it in MATLAB.

28. $3<4$

29. $5\geq 6$

30. $\sin(0.2)<0.2$

31. $|3-2(2)|<3$

32. $\mathrm{Re}(\sqrt{1-(4.2)^2})=0$

33. $\mathrm{Re}(1-2i)=1-2i$

In exercises 34–36, increase the number of displayed digits with `format long`. *Hint: Evaluate* 4! *with* `factorial(4)`.

34. Evaluate e with `exp(1)`. Next, approximate e by evaluating the series 1.1 at $x = 1$. Experiment by increasing the number of terms used to increase the accuracy of the approximation.
35. Evaluate $\cos(\pi/4)$ with `cos(pi/4)`. Next, approximate $\cos(\pi/4)$ by evaluating the series 1.2 at $x = \pi/4$. Experiment by increasing the number of terms used to increase the accuracy of the approximation.
36. Evaluate $\sin(\pi/2)$ with `sin(pi/2)`. Next, approximate $\sin(\pi/2)$ by evaluating the series 1.3 at $x = \pi/2$. Experiment by increasing the number of terms used to increase the accuracy of the approximation.
37. On a sheet of graph paper, set up the complex plane and plot the vectors representing the complex numbers $z_1 = 3+4i$, $z_2 = -2-3i$, $z_3 = -5+6i$, and $z_4 = 1-4i$. Use a protractor and ruler to estimate the angle (restricted to $(-\pi, \pi]$) and magnitude of each vector. Compare these results with the output of MATLAB's `angle` and `abs` commands. For each z, verify $z = re^{i\theta}$ with `abs(z)*exp(i*angle(z))`.
38. Use Euler's Identity, $e^{i\,\theta} = \cos\theta + i\,\sin\theta$ to hand-calculate $e^{i\pi/2}$, $e^{-i\pi/3}$, $e^{i3\pi/4}$, and $e^{i\pi}$. Verify your hand-calculated results with MATLAB's `exp` command.
39. The complex number $z = -1$ has magnitude 1 and angle π. consequently,

$$z^{1/2} = (1e^{i\pi})^{1/2} = e^{i\pi/2} = i,$$

by Euler's indentity. Verify this result with `(-1)^(1/2)`. How does this differ from `-1^(1/2)`?
40. The complex number $z = i$ has magnitude 1 and angle $\pi/2$. consequently,

$$z^{1/2} = (1e^{i\pi/2})^{1/2} = e^{i\pi/4} = \frac{\sqrt{2}}{2} + i\,\frac{\sqrt{2}}{2},$$

by Euler's indentity. Verify this result with `i^(1/2)`.