

## Entering formulas in MATLAB

In order to start a MATLAB session, click on the MATLAB icon. A screen with three windows should appear. One of these windows is called the “Command Window.” This is the one you will be working in. If for some reason you don’t see a command window, go to “View” on the menu bar, choose “Desktop Layout,” and click on “Default.”

In the Command Window, you see the symbol

```
>>
```

at the beginning of a line. This is your prompt. All commands will be entered at the prompt. You enter a command by typing it, and then hitting ENTER.

The simplest type of command you can enter is a number. Try:

```
>> 7.2
```

After hitting ENTER, MATLAB will show you an “answer” of 7.2000. This is not very exciting. Here is something more interesting to try: Enter

```
>> pi
```

MATLAB gives you the numerical value of  $\pi$  with a precision of four places after the decimal point. How can you find the value of  $\pi$  with a better precision? Enter

```
>> format long
```

```
>> pi
```

As you can see, MATLAB leaves a lot of blank lines in the output. This can be annoying, especially if you want to print your output. In order to suppress the blank lines, enter

```
>> format compact
```

MATLAB also can handle symbolic expressions, but a little more care needs to be taken with those. Try entering

```
>> x
```

MATLAB will give you an error message, since no numerical value has yet been assigned to `x`. You can assign a numerical value to a variable by an *assignment statement*. Enter

```
>> x = 6.34;
```

```
>> x
```

Now MATLAB tells you that the value of the variable `x` is 6.34, which makes sense, since this is the value we assigned to this variable. The semicolon at the end of the assignment statement (or any other statement) has the effect of suppressing output. Note that MATLAB did not give you any answer after you entered the assignment statement, it just gave you the next prompt. See what happens if you leave out the semicolon:

```
>> x = - .8
```

By now you may find it annoying that MATLAB gives you all these zeros at the end, and you may want to return to the format where the answers are given just with a precision of four digits after the decimal point. How can you do this? You may have guessed by now that you should enter

```
>> format 'something'
```

You may even have guessed what the “something” should be, but how can you make sure that your guess was right? MATLAB has two neat ways of telling you how to use its commands. One way is to use the “Help” feature on the menu bar. This one is rather self-explanatory, but it may be cumbersome at times. An often faster way is this:

```
>> help format
```

This outputs all the options for the “format” command, and you will notice that your first guess about the “something” was correct after all!

The symbols for algebraic operations in MATLAB are:

- + for addition,
- - for subtraction,
- \* for multiplication,
- / for division,
- ^ for exponentiation.

For example, if you want to compute  $2\pi$ , then you should enter

```
>> 2*pi
```

*Under no circumstances are you allowed to omit the \* symbol for multiplication!* Try entering

```
>> 2pi
```

You will get an error message.

For entering more complicated expressions, you need to know how MATLAB prioritizes execution of operations. This works as follows:

- exponentiation has the highest priority
- multiplication and division have the second highest priority
- addition and subtraction have the lowest priority

For example, if you enter

```
>> 2*3^2/3+5
```

MATLAB will evaluate this as  $(2 \cdot 3^2)/3 + 5 = 11$ . If you want MATLAB to evaluate  $2 \cdot 3^{\frac{2}{3}+5}$  instead, you need to enter:

```
>> 2*3^(2/3+5)
```

If you want MATLAB to evaluate  $2 \cdot (3^{\frac{2}{3}} + 5)$  instead, you need to enter:

```
>> 2*(3^(2/3)+5)
```

If several operations have the same priority, MATLAB executes them from left to right. For example, the expression

```
>> 6/2*3
```

will be evaluated as  $\frac{6}{2} \cdot 3 = 9$ ; if you want MATLAB to evaluate  $\frac{6}{2 \cdot 3}$  instead, you need to enter:

```
>> 6/(2*3)
```

Invariably, when entering long formulas, you will make occasional mistakes. Suppose you want to calculate  $\frac{\pi}{5+2^3} - 13$  and you inadvertently entered

```
>> pi/(5 + 2^3 - 13)
```

MATLAB will give you a warning: you should never divide by zero. It is easy enough to correct your mistake without retyping the whole formula: Hitting the  $\uparrow$  key will restore the formula, and then you can easily move the closing bracket into the right place. In general, you can recover all commands you entered in a given session by using the  $\uparrow$  and  $\downarrow$  keys.

Please correct the mistake in the formula right now and record the correct answer for possible submission with your first quiz.

If you don't exactly remember MATLAB's rules for prioritizing operations, there is a simple way to avoid possible problems: Use brackets liberally. A well-placed pair of brackets never hurts.

You can use arithmetic operations on variables as well as numbers. Try entering:

```
>> 3/x
```

```
>> 2*y
```

While MATLAB gives you a numerical answer to the former, the latter results in an error message. What is going on here? Recall that earlier in this session, we assigned the value  $-0.8$  to the variable `x`, and the answer that MATLAB returned to your first command is simply  $\frac{3}{-0.8}$ . No value has been assigned to the variable `y` yet, and this is why the second command generates an error message. You can find out which variables are currently in use by entering

```
>> who
```

Note that one of the variables in use is called `ans`; it is the default variable used by MATLAB when no other variable is specified. You can remove all assignments to any of these variables by entering

`>> clear`

It is highly recommended that you always begin a MATLAB session by using this command so that no previous work will interfere with the current session.