****

**MATLAB**

**Stage2**

**Lec2**

**Matlab Interactive Sessions**

**MS.c Ola Ali**

**MS.c Ali Haider**

**Matlab Interactive Sessions**

**Using MATLAB as a calculator:**

As an example of a simple interactive calculation, just type the expression you want to evaluate.

Let’s start at the very beginning. For example, let’s suppose you want to

the expression, 1 + 2 × 3. You type it at the prompt command (>>) as follows,

>> 1+2\*3

ans =

7

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable ans, short for answer, to store the results of the current calculation.

that the variable ans is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name.

For example,

>> x = 1+2\*3

x =

7

will result in x being given the value 1 + 2 × 3 = 7. This variable name can always be used to refer to the results of the previous computations.

Therefore, computing 4x will

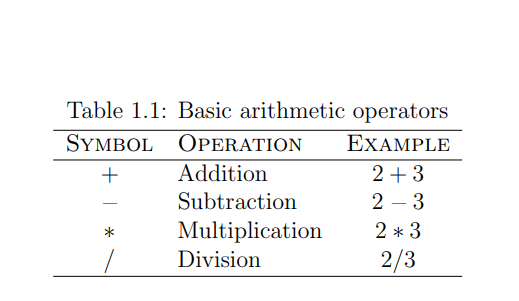
result in

>> 4\*x

ans =

28.0000

Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic operators.



**Quitting MATLAB:**

To end your MATLAB session, type quit in the Command Window, or select File −→ Exit MATLAB in the desktop main menu.

. **Getting started:**

After learning the minimum MATLAB session, we will now learn to use some additional operations.

**Creating MATLAB variables:**

MATLAB variables are created with an assignment statement. The syntax of variable assignment is variable name = a value (or an expression).

For example,

>> x = expression

where expression is a combination of numerical values, mathematical operators, variables,

and function calls. On other words, expression can involve:

• manual entry

• built-in functions

• user-defined functions

Overwriting variable

Once a variable has been created, it can be reassigned. In addition, if you do not wish to

see the intermediate results, you can suppress the numerical output by putting a semicolon

(;) at the end of the line. Then the sequence of commands looks like this:

>> t = 5;

>> t = t+1

t =

6

**Error messages:**

If we enter an expression incorrectly, MATLAB will return an error message. For example,

in the following, we left out the multiplication sign, \*, in the following expression

>> x = 10;

>> 5x

??? 5x

|

Error: Unexpected MATLAB expression.

**Making corrections :**

To make corrections, we can, of course retype the expressions. But if the expression is

lengthy, we make more mistakes by typing a second time. A previously typed command

can be recalled with the up-arrow key ↑. When the command is displayed at the command

prompt, it can be modified if needed and executed.

**Controlling the hierarchy of operations or precedence:**

Let’s consider the previous arithmetic operation, but now we will include parentheses. For

example, 1 + 2 × 3 will become (1 + 2) × 3

>> (1+2)\*3

ans =

9

and, >> 1+2\*3

ans =

7

By adding parentheses, these two expressions give different results: 9 and 7.

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. However, the standard order of precedence of arithmetic operations can be changed by inserting parentheses.

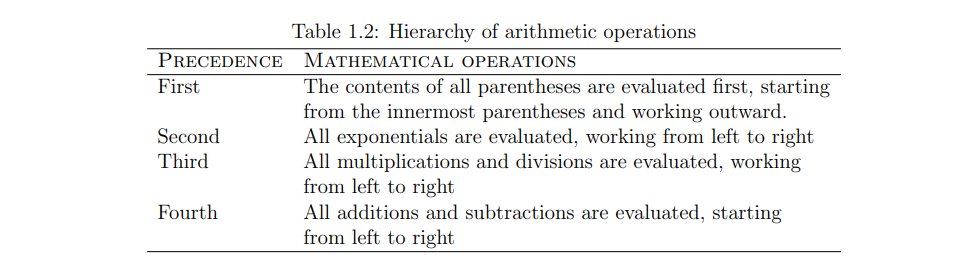
For example,

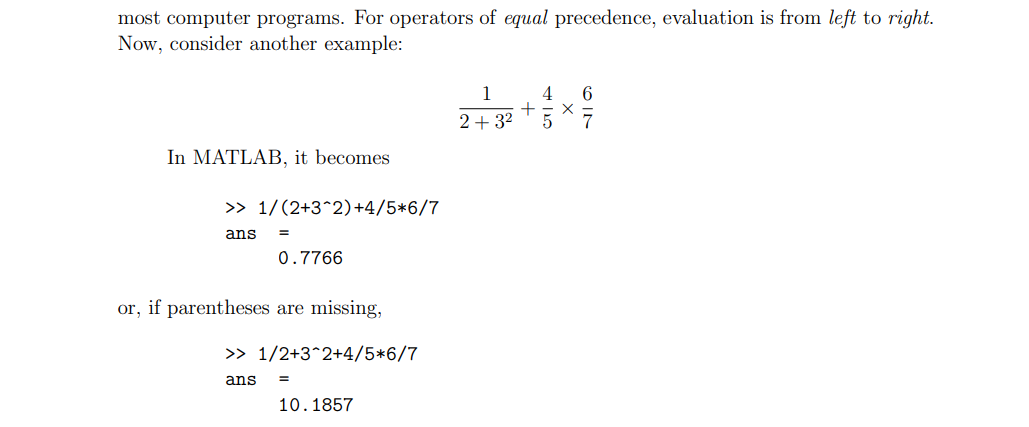
the result of 1+2×3 is quite different than the similar expression with parentheses (1+2)×3.

The results are 7 and 9 respectively. Parentheses can always be used to overrule priority, and their use is recommended in some complex expressions to avoid ambiguity.

Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules. The order in which the arithmetic operations are evaluated is given

in Table 1.2. MATLAB arithmetic operators obey the same precedence rules as those i from previous example.





Controlling the appearance of floating point number MATLAB by default displays only 4 decimals in the result of the calculations.

**for example:**

−163.6667, as shown in above examples.

MATLAB does numerical calculations in double precision, which is 15 digits.

The command format controls how the results of computations are displayed. Here are some examples of the different formats together with

the resulting outputs.

>> format short

>> x=-163.6667

If we want to see all 15 digits, we use the command format long

>> format long

>> x= -1.636666666666667e+002

To return to the standard format, enter format short, or simply format.

There are several other formats. For more details, see the MATLAB documentation, or type help format.

.

**Managing the workspace:**

The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a clear command at the start of each new independent calculation.

8

>> clear

The command clear or clear all removes all variables from the workspace. This frees up system memory.

In order to display a list of the variables currently in the memory,

type

>> who

while, whos will give more details which include size, space allocation, and class of the

variables.

**Keeping track of your work session:**

It is possible to keep track of everything done during a MATLAB session with the diary

command.

>> diary

or give a name to a created file,

>> diary FileName

where FileName could be any arbitrary name you choose.

The function diary is useful if you want to save a complete MATLAB session. They save all input and output as they appear in the MATLAB window.

When you want to stop the 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 use the function type to view the diary file or you can edit in a text editor or print. This command is useful,

for example in the process of preparing a homework or lab submission.

**Entering multiple statements per line:**

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to

enter more than one statement at once. Commas (,) allow multiple statements per line

without suppressing output.

>> a=7; b=cos(a), c=cosh(a)

b =

0.6570

c =

548.3170

9

**Miscellaneous commands:**

Here are few additional useful commands:

• To clear the Command Window, type clc

• To abort a MATLAB computation, type ctrl-c

• To continue a line, type . . .

**Getting help:**

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help

directly in the Command Window. The preferred method is to use the Help Browser. The

Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other

hand, information about any command is available by typing

>> help Command

Another way to get help is to use the lookfor command. The lookfor command differs

from the help command. The help command searches for an exact function name match,

while the lookfor command searches the quick summary information in each function for

a match. For example, suppose that we were looking for a function to take the inverse of

a matrix. Since MATLAB does not have a function named inverse, the command help

inverse will produce nothing. On the other hand, the command lookfor inverse will

produce detailed information, which includes the function of interest, inv.

>> lookfor inverse

Note - At this particular time of our study, it is important to emphasize one main point.

Because MATLAB is a huge program; it is impossible to cover all the details of each function

one by one. However, we will give you information how to get help. Here are some examples:

• Use on-line help to request info on a specific function

>> help sqrt

• In the current version (MATLAB version 7), the doc function opens the on-line version

of the help manual. This is very helpful for more complex commands.

>> doc plot

10

• Use lookfor to find functions by keywords. The general form is

>> lookfor FunctionName.