Practice 2

In this note, we will practice inputing variables and plotting in Matlab.

1. Matlab considers every variable as a matrix. For example, the variable \( x \) in the following command

\[
\text{>> } x = 2
\]

is understood as a \( 1 \times 1 \) matrix.

2. The command

\[
\text{>> } x = 1 : 0.3 : 5
\]

gives a row vector of real numbers starting with 1, equally spaced by 0.3, and not exceeding 5. In this case, \( x \) is a vector of 14 numbers, or a matrix of size \( 1 \times 14 \). To check the length of \( x \), use the command

\[
\text{>> length(x)}
\]

3. Most built-in functions in Matlab take matrix as input. For example, with vector \( x \) as above, try the following:

\[
\text{>> sin(x)} \\
\text{>> sqrt(x)} \\
\text{>> log(x)} \\
\text{>> exp(x)} \\
\text{>> x - 1} \\
\text{>> 2*x}
\]

Matlab will compute sin, sqrt, logarithm, exponentiation, subtraction by 1, multiplication by 2, at each entry of the vector \( x \).

4. However, to raise each entry of vector \( x \) to a power, one has to use the \(^.\) operator. For example,

\[
\text{>> x.^2} \\
\text{>> x.^(-2)} \\
\text{>> 1./x} \\
\text{>> x^2}
\]

The last command gives an error because Matlab understands it as \( x \ast x \). Because \( x \) is a \( 1 \times 14 \) matrix, it cannot be multiplied by itself (incompatibility of dimension). It would be correct to multiply \( x \) by the its transpose, which has dimension \( 14 \times 1 \).

\[
\text{>> x*transpose(x)}
\]

5. The entries of vector \( x \) are indexed from 1 to 14 (not from 0 to 13). To access the 9th entry of \( x \), for example, write

\[
\text{>> x(9)}
\]
Because $x$ is a matrix of size $1 \times 14$, one can also write

\[
>> x(1,9)
\]

6. The basis syntax of the ‘plot’ command is ‘plot(x,y)’ where $x$ and $y$ are vectors of the same length. Mallab will plot the following points $(x(1), y(1)), (x(2), y(2)), \ldots, (x(n), y(n))$, where $n$ is the common length of $x$ and $y$, and then connect two consecutive points by a straight line segment. Try the following:

\[
>> y = x.^2 \\
>> plot(x,y)
\]

Sometimes, we want to customize the appearance of the plot by, for example, removing the line segments. Try the following commands:

\[
>> plot(x,y,'.') \\
>> plot(x,y,'o') \\
>> plot(x,y,'-o') \\
>> plot(x,y,'.r')
\]

To graph functions $y = x^2$ and $z = 1/x$ on the same plot, one can do as follows:

\[
>> z = 1./x \\
>> plot(x,y,'b',x,z,'r')
\]

To learn more options of the ‘plot’ command, type

\[
>> help plot
\]

7. The ‘while’ loop is used to repeat certain commands until a condition is false. The basic syntax is:

```
while (condition) 
    commands 
end
```

The condition in the ‘while’ loop is a logical statement, having value 1 if true, 0 if false. Try the following:

\[
>> a = 1 \\
>> b = 2 \\
>> c = 3 \\
>> a == b \\
>> a+b == c \\
>> a ~= b \\
>> (a ~= b) && (c ~= b) \\
>> (a ~= b) || (c == b)
\]

Note that ~= denotes ‘not equal to’, && denotes ‘and’, || denotes ‘or’.