2 Exercises

a) Use MATLAB as a calculator. Try the following:

```
pi*pi-10
sin(pi/4)
ans^2 %<---“ans” holds the last result
```

b) Do variable name assignment in MATLAB. Try the following:

```
x=sin(pi/5)
cos(pi/5) %<---assigned to what??
y=sqrt(1-x*x)
ans
```

c) Complex numbers are natural in MATLAB. The basic operations are supported. Try the following:

```
z=3+4i,w=-3+4j
real (z), imag (z)
abs ([z,w]) %<--- Vector constructor
conj(z+w)
angle(z)
exp(j*pi)
exp(j*[pi/4, 0, -pi/4])
```

d) Make sure that you understand the colon notation. In particular, explain in words what following MATLAB code will produce.

```
Jkl=0:6
Jkl=2:4:17
Jkl=99:-1:88
ttt=2:(1/9):4
tpi=pi*[0:0.1:2];
```

e) Extracting and/or inserting numbers into a vector is very easy to do. Consider the following definition of `xx`:

```
xx=[zeros(1,3), (0:0.25:1), ones(1,4)]
xx(4:6)
size(xx)
length(xx)
xx(2:2:length(xx))
```

Explain the results echoed from the last four lines of the above code.

f) Observe the result of the following assignments:

```
yy=xx; yy(4:6)=pi*(1:3)
```

Now write a statement that will take the vector `xx` defined in part (e) and replace the even indexed elements (i.e., `xx(2)`, `xx(4)`, etc.) with a constant. *Use a vector replacement, not a loop.*

g) Experiment with vectors in MATLAB. Think of the vector as a set of numbers. Try the following:

```
xk=cos(pi*(0:11)/4) %<---comment: compute cosines
```

What is `xk(1)`? Is `xk(0)` defined?

!!! The text following the % is a comment, it may be omitted.

h) (A taste of vectorization) Loops can be written in MATLAB, but they are NOT the most efficient way to get things done. It's better **always avoid loops** and use the colon notation instead. The following code has a loop that computes values of the cosine function (The index of `yy()` must start at 1). Rewrite this computation without using the loop (follow the style in the previous part).

```
yy=[]; %<---initialize the yy vector to be empty
for kk=-5:5
yy(k+6)=cos(k*pi/3)
end
yy
```

Why it is necessary to write `yy(k+6)`. What happens if you use `yy(k)` instead?

i) Plotting is easy in MATLAB for both real and complex numbers. The basic plot command will plot a vector `y` versus a vector `x` connecting successive points by straight lines. Try the following:

```
x= [-3 -1 0 1 3];
y=x.*x-3*x;
plot(x,y)
z=x+y*sqrt(-1)
plot (z) %<---complex values: plot imag vs. real
```

Use `help arith` to learn how the operation `xx.*xx` works when `xx` is a vector; compare to matrix multiplication.

When unsure about a command, use `help`.

j) Use the built-in MATLAB editor (on Windows 95/98/NT/XP), or an external one such as EMACS on UNIX/LINUX, to create a script file called mylab1.m containing the following lines:

```
tt=0:0.01:1;
xx=cos (5*pi*tt);
zz=1.4*exp(j*pi/2)*exp(j*5*pi*tt);
plot(tt, xx, 'b-', tt, real(zz), 'r—'), grid on
%<--- plot a sinusoid
title ('TEST PLOT of a SINUSOID')
xlabel ('TIME (sec)')
```

k) Run your script from MATLAB. To run the file `mylab1` that you created previously:

```
mylab1    %<--- will run the commands in the file
type mylab1    %<--- will type out the contents of
               % mylab1.m to the screen
```

l) Generate a time vector `(t)` to cover a range of t that will exhibit two cycles (from –T to T) of a sinusoid defined below with f=4000 Hz. We will use T to denote the period (cycle) of the sinusoids and define the starting time of the vector t to be equal to –T, and the ending time as +T. You should be using a definition for t similar to tt in part (2.j). Make sure that you have at least 25 samples per period of the sinusoidal wave. Make a plot of this signal.

$x(t)=A\cos(2\pi ft)$