Basic SCILAB Commands

Preliminaries

clear, clc // commands to clear all variable and clear command console

a=1 // initializing a variable, note that unlike C or C++ there is no need to define the variable type

b=1; // adding a semicolon at the end of a statement suppresses the output from appearing on the command window

c=a+b // addition of two variable and assignment of the value to another variable

t1=0:10 // creating a row vector and assigning it to a variable t1, i.e. t1=[0, 1, 2, ..., 10]

t1tc = t1' // creating transpose conjugate

t1t = t1.’ // creating transpose only

t2 = mtlb_t(t1) //conjugate of t1

t3 = 0:0.1:1 // creating a row vector of elements consisting of 0 to 1 with increments of 0.1

t4 = linspace(0,1,20) // creating a row vector of elements consisting of 0 to 1 with total of 20 elements

len = length(t4) // length of the vector t4 is stored in variable "len"

mat1 = [1,2,3;4,5,6] // initializing a matrix mat1; note SCILAB automatically determines the type of the variable

//note the brackets that are used

sz = size(mat1) // finding the dimensions of mat1 and storing in variable "sz"

Indexing

second_t1 = t1(2) // extracting the 2nd element of the vector t1; note the braces used

one_two_mat1 = mat1(1,2) // extracting the value at (1,2) position of the matrix mat1

// note that in SCILAB indexing starts from 1 and not from 0 as in C or C++

Creating ‘Zero’ Row Vector

- zero_row=zeros(1,10); // creating a row vector of zeros with 10 elements

‘For’ Loop

```
for j=1:10
    for_ex(j)=rand(1,1) // re-assigning each value of for_ex with a random number
end

indexes = [1,4,9,18];
for k=indexes // another way of using for loop
    disp(k)
end
```

‘While’ Loop

```
// while loop
a=[1,2,7,5,2,6,10,4,8];
j=0
break_while = 4
while break_while ~= 5
    j=j+1;
   break_while = a(j);
   printf('\nwhile loop has run %d time(s).', j )
   printf('\nvalue of break_while is %d.', break_while )

end
```

‘If’ Condition

```
// if condition
a=5
if a<3 then
    disp('a is less than three.')
elseif a==3 then
    disp('a is equal to three.')
else
    disp('a is greater than 3')
end
```

‘Switch-Case’

```
// switch case
n = input('Enter value of n where is a natural number: ')
switch(n)
case 1 then printf('This is case one')
    break;
case 2 then printf('This is case two')
    break
else printf('Any other case')
    break
end
```

Function Plotting

```
// for plotting a function
t=0:0.001:4*%pi; // in MATLAB % is used for comments
x=sin(t);
plot(t,x);
xlabel('Time(s)'), ylabel('Ampitude');
```

Matrix Multiplication

a = [1 2 3; 4 5 6] // initializing a matrix

b = [1; 2; 3] // initializing a column vector

prod_ab = a*b

Element Wise Multiplication

```
c = [1 2 3; 4 5 6]
d = [1 2 3; 4 5 6]
ele_mul = c.*d
```