BRAC UNIVERSITY
CSE 330: Numerical Methods (LAB)
Lab 3: Loops, Conditional Statements & Iterations

Like other program languages, MATLAB execute the program statements sequentially starting at the top of the code and moving down to the end. Because a strict sequence is highly limiting all computer languages include statements allowing programs to take non sequential paths. These can be classified as

- Decisions (or Selections): The branching of flow based on decision
- Loops (or Repetitions): The looping of flow to allow statements to be repeated.

Decisions in MATLAB

The if Structure: This structure allows you to execute a set of statements if a logical condition is true. Its general syntax is:

```
if condition
    statements
end
```

The if else Structure: This structure allows you to execute a set of statements if a logical statement is true and to execute a second set if the condition is false. Its general syntax is:

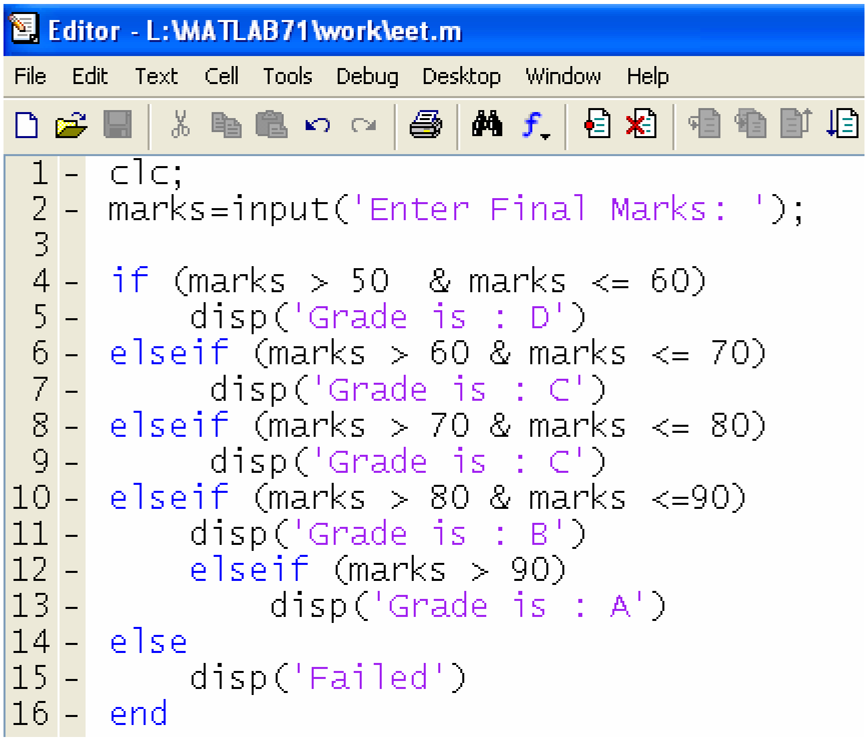
```
if condition
    statements
else
   statements
end
```

Relationship	**MATLAB Operator**
Equal	= =
Not equal	~=
Greater than	>
Less than	<
Not	~
Less than or equal	<=
Greater than or equal	>=
Or operator	|
And operator	&

The if…elseif Structure: It often happens that the false option of an if…else structure is another decision. This type of structure often occurs when we have more than two options for a particular problem setting. For such cases the if….elseif structure is used.

```
if  condition$_1$
    statements$_1$
elseif  condition$_2$
        statements$_2$
elseif  condition$_3$
        statements$_3$
.
.
.
else
    statement $_{else}$
end
```

Example 1: Write an M file that will take mark from the keyboard and display the corresponding grade.



```
clc;
marks=input('Enter Final Marks: ');

if (marks > 50  & marks <= 60)
    disp('Grade is : D')
elseif (marks > 60 & marks <= 70)
     disp('Grade is : C')
elseif (marks > 70 & marks <= 80)
     disp('Grade is : C')
elseif (marks > 80 & marks <=90)
    disp('Grade is : B')
    elseif (marks > 90)
        disp('Grade is : A')
else
    disp('Failed')
end
```

Figure 1: Code to take input from keyboard and execute conditional statements

The *switch* structure: The switch structure is similar to the if…elseif structure.However rather than testing individual conditions, the branching is based on the value of a single test expression. Depending on its value, different blocks of code are executed. In addition, an optional block is implemented if the expression takes on none of the provided values. Its general syntax is:

```
switch test_expression

case $value_1$
      $statements_1$
case $value_2$
       $statements_2$
.
.
otherwise
       $statements_{otherwise}$
end
```

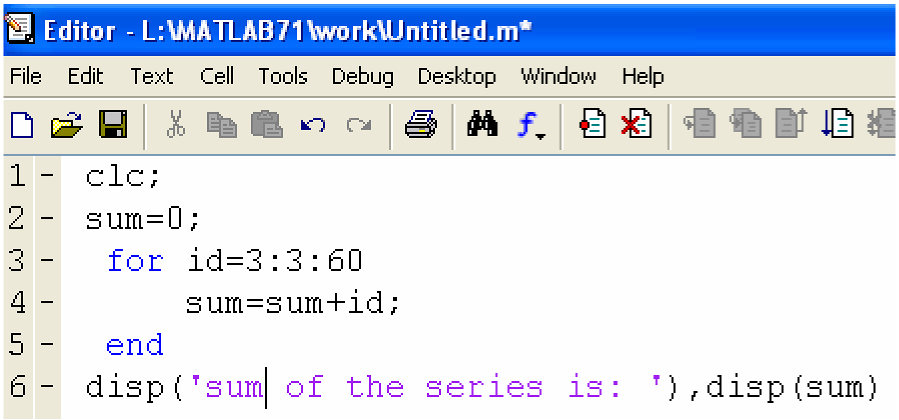
Loops

Loops perform operations repetitively. There are two types of loops depending on how the repetitions are terminated. A *for* loop ends after a specified number of repetitions. A *while* loop ends on the basis of logical condition.

The for….end structure: A for loop repeats statements a specific number of times. Its general syntax is

```
 for  index = start : step : finish
       statements
 end
```

Example 2: Use For Loop to find the sum of the series, S = 3+6+9+12+15+………+60



```
clc;
sum=0;
 for id=3:3:60
      sum=sum+id;
 end
disp('sum of the series is: '),disp(sum)
```

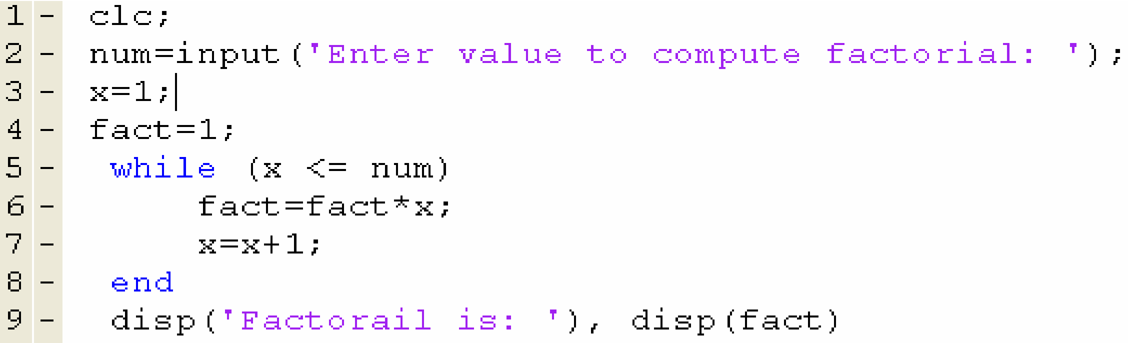
Figure 2: Use of for loop

Lab task 1: Use for loop to find the factorial of 10.

The while Structure: A while loop repeats as long as a logical condition is true.

```
while condition
        statements
end
```

Example 3: Compute the factorial of any value using while loop.



```
clc;
num=input('Enter value to compute factorial: ');
x=1;
fact=1;
 while (x <= num)
     fact=fact*x;
     x=x+1;
 end
 disp('Factorail is: '), disp(fact)
```

Figure 3: Use of while loop

Termination of ‘while’ or ‘for loop’ in advance: The *break* command

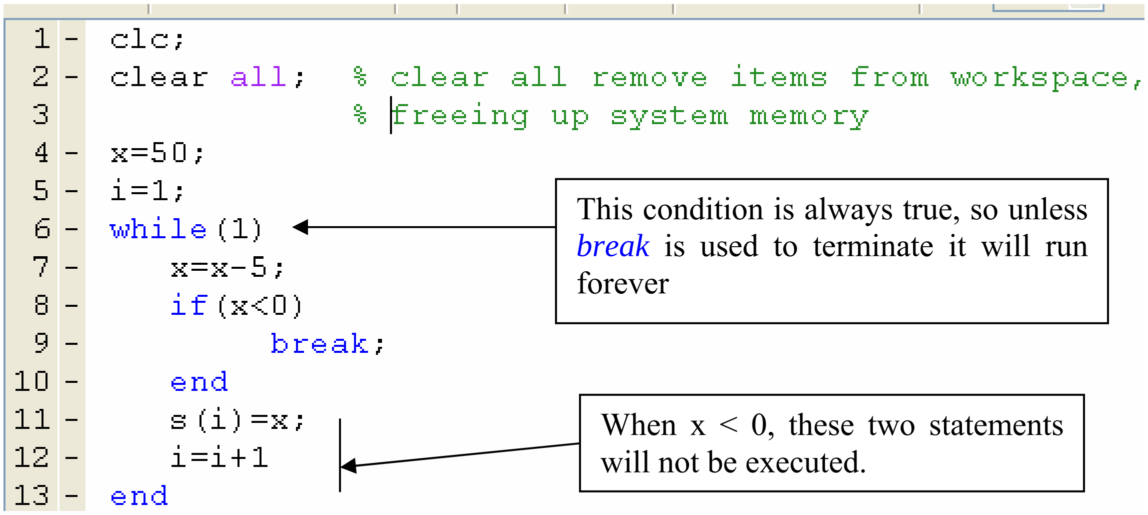
break terminates the execution of a for or while loop. Statements in the loop that appear after the *break* statement are not executed.

In nested loops, break exits only from the loop in which it occurs. Control passes to the statement that follows the end of that loop.

Example 4: Use while and break commands to generate the sequence

50 45 40 35 30…….0

See figure 4 for a possible solution.



```
clc;
clear all;  % clear all remove items from workspace,
            % freeing up system memory
x=50;
i=1;
while(1)
   x=x-5;
   if(x<0)
        break;
   end
   s(i)=x;
   i=i+1
end
```

This condition is always true, so unless *break* is used to terminate it will run forever

When x < 0, these two statements will not be executed.

Figure 4: Use of break command

Lab task 2: Write a M file to compute the variance of the array A= [-2 4 -6 0 2 8 9]

Formula for variance is given by

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \text{where , N= number of elements in the array}$$

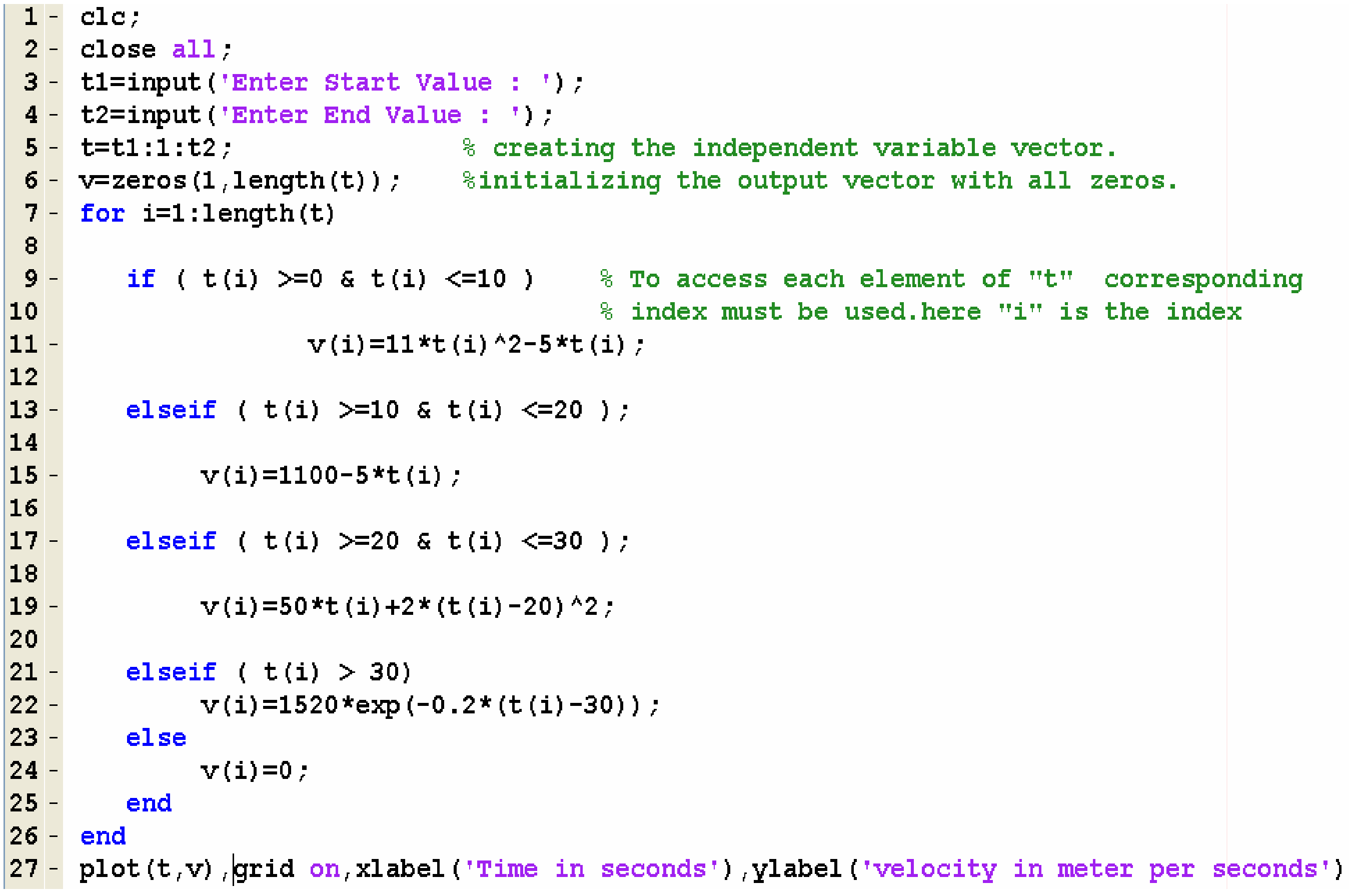
$$\bar{x} = \text{average of the array}$$

After you write and run your code, verify your answer by comparing your result with the one calculated using MATLAB built in function, by writing var(A) in command window.

Example 5: The velocity of a Rocket can be modeled by the following equations:

$$v(t) = \begin{cases} 11t^2 - 5t & 0 \le t \le 10 \\ 1100\text{ - 5t} & 10 \le t \le 20 \\ 50t + 2(t-20)^2 & 20 \le t \le 30 \\ 1520\,\mathrm{e}^{-0.2(t-30)} & t > 30 \\ 0 & \textit{otherwise} \end{cases}$$

Write a M file to compute v as a function of t. Then use this function to plot v versus t for t= -5 to 100



```
clc;
close all;
t1=input('Enter Start Value : ');
t2=input('Enter End Value : ');
t=t1:1:t2;               % creating the independent variable vector.
v=zeros(1,length(t));    %initializing the output vector with all zeros.
for i=1:length(t)

   if ( t(i) >=0 & t(i) <=10 )    % To access each element of "t"  corresponding
                                  % index must be used.here "i" is the index
               v(i)=11*t(i)^2-5*t(i);

   elseif ( t(i) >=10 & t(i) <=20 );

        v(i)=1100-5*t(i);

   elseif ( t(i) >=20 & t(i) <=30 );

        v(i)=50*t(i)+2*(t(i)-20)^2;

   elseif ( t(i) > 30)
        v(i)=1520*exp(-0.2*(t(i)-30));
   else
        v(i)=0;
   end
end
plot(t,v),grid on,xlabel('Time in seconds'),ylabel('velocity in meter per seconds')
```

Figure 5: M file for example 4

Fixed Point Iteration

To approximate the solution of the equation f(x) =0 the equation is rearranged in the form x = g(x) so that x is on the left side of the equation. This transformation can be accomplished by simply adding x to both sides of the original equation. We start with an initial "wise" guess of x_0 as the solution and then compute a new estimate x_1=g (x_0). This is called iteration .After nth iteration x_{n+1}=g (x_n).

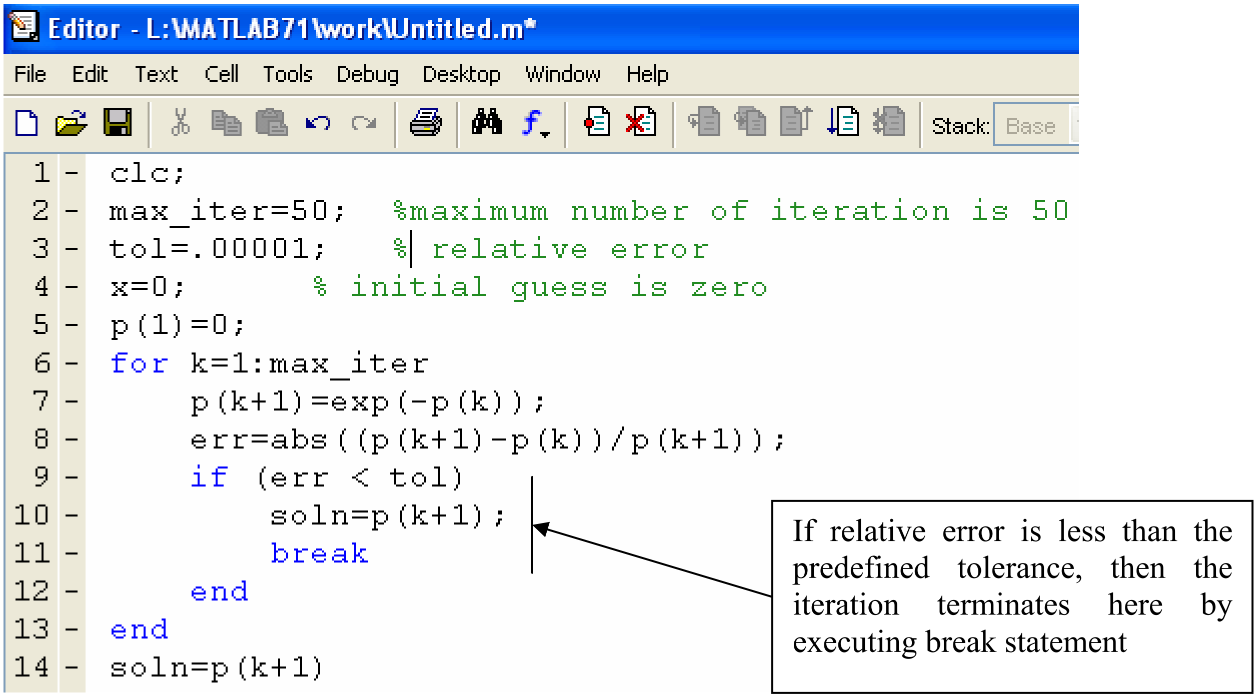
After each iteration the relative approximate error is calculated, $\varepsilon = \left|\frac{x_{n+1} - x_n}{x_{n+1}}\right| * 100\%$ If the relative error is within a predefined tolerance, the iteration can be stopped and X_{n+1} can be taken as the solution.

<u>Remember:</u> Fixed point iteration may not always converge to the solution. Whether fixed point iteration will converge to the desired solution depends on the initial guess and type of the function.

Example 6: Find the solution of $f(x) = e^{-x} - x$ by fixed point iteration. Maximum number of iteration allowed is 50. Also if the relative error is less than 0.001 percent the iteration can be terminated.

Solution: Rearranging: $x = e^{-x}$

As an initial guess assume the solution might be 0.



```
clc;
max_iter=50;  %maximum number of iteration is 50
tol=.00001;   % relative error
x=0;        % initial guess is zero
p(1)=0;
for k=1:max_iter
    p(k+1)=exp(-p(k));
    err=abs((p(k+1)-p(k))/p(k+1));
    if (err < tol)
        soln=p(k+1);
        break
    end
end
soln=p(k+1)
```

If relative error is less than the predefined tolerance, then the iteration terminates here by executing break statement

Figure 6: M file for fixed point iteration to find the root.

Home work

For the following, submit the printed M files.

[1]. Write a M file to compute the product of two Matrices. $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

$Y = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$.Compare your answer with actual one. Write A*B in command window

Hints: Here you need to multiply each row of first matrix with each column of second matrix. You will need nested for loops here. A total of three for loops will be required along with one if else command.

[2]. Use fixed point iteration to solve $f(x) = x^2 - \sin(x + 0.15)$. Use initial guess of x=0 as a potential solution. Maximum number of iteration allowed is 100. Tolerance =.00001. What happens if you change your initial guess to x = 5 ? Explain.