Chapter Seven

Repetition Loops

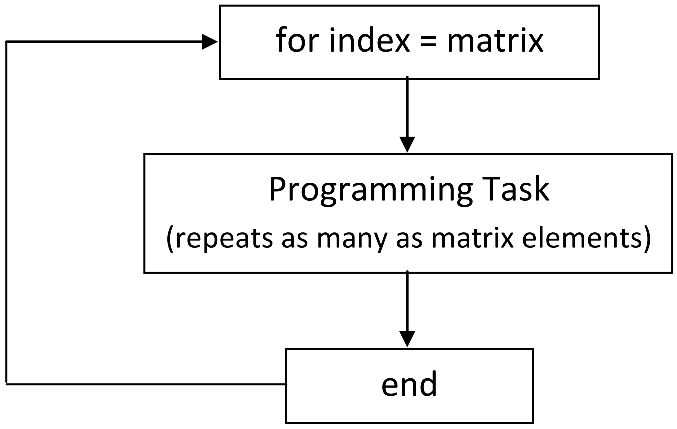
Repetition loops are used to repeat part of a program a specified number of times. The start and end of the repeated block must be identified. The program does not move to the next part unless finishing the loop. There are two methods in MATLAB to create loops:

- Using (**for**) loop.
- Using (**while**) loop.

Use of loops in MATLAB should always be avoided except for certain cases. The program becomes somewhat slow if it contains loops. The inherent repeating property in MATLAB which is called (vectorization) should always be used for efficient programming. In vectorization all elements in the matrix are automatically considered by the program without the need to generate a loop. Loops are restricted to cases when the matrix elements are needed be considered one by one not as a whole group in vectorization.

7.1 (for) loops

The loop using (for) command are generated as follows:



Ex. 7.1 Write MATLAB program to evaluate the function y=x^2 by vectorization once and by (for) loop once again. Enter x as the numbers from 1 to 10.

Sol. *Method (1): using vectorization*

```
clear,clc
x=1:10;
y=x.^2;
disp([x',y'])
```

Method (2): using (for) loop

```
 clear,clc
 for x=1:10
    y=x^2;
    disp([x,y])
 end
```

Ex. 7.2 Repeat the last program using five arbitrary numbers and a loop.

```
clear,clc
for x=[2 -3 7 5 -4]
    y=x^2;
    z=x+5;
    disp([x,y,z])
end
```

Note: The printing statement should always be placed inside the loop, before (end), to ensure the display of all numbers, not only the last number or last line in a table.

Ex. 7.3 Write Matlab code to evaluate the function $y = \frac{x-1}{x+1}$ for 7 values of x equally spaced from 1 to 10. Use (for) loop and display 1 decimal digit of x and 3 decimal digits of y.

Sol.

```
clear,clc
for x=linspace(1,10,7)
    y=(x-1)/(x+1);
    fprintf('    %5.1f    %5.3f \n',[x,y])
end
```

Run:

```
  1.0      0.000

   2.5      0.429

   4.0      0.600

   5.5      0.692

   7.0      0.750

   8.5      0.789

  10.0      0.818
```

Ex. 7.4 Write Matlab program to enter a group of arbitrary 10 numbers. If there is any negative number within the group, replace it with zero. Print the old and the modified matrices. Apply vectorization and loops methods.

Sol. *Method (1): Vectorization*

```
clear,clc
x=[-1 5 -2 -5 4 6 7 -6 -4 8];
disp(x)
neg=find(x<0);
x(neg)=0;
disp(x)
```

Method (2): Looping

```
clear,clc
x=[-1 5 -2 -5 4 6 7 -6 -4 8];
disp(x)
for i=1:length(x)
    if x(i)<0
        x(i)=0;
    end
end
disp(x)
```

Ex. 7.5 Write MATLAB program to evaluate the natural logarithm of ten arbitrary numbers using loops. Discard any negative or zero numbers.

Sol.

```
clear,clc
for x=[2 -3 4 -5 -6 8 -4 7 3 -9];
    if x>0
        y=log(x);
        fprintf('    %2.0f    %4.2f \n',[x,y])
    elseif x<=0
        fprintf('    %2.0f    not pssible \n',x)
    end
end
```

Ex. 7.6 Write MATLAB program to evaluate the following function for the given values of x. Use vectorization and (for) loop methods.

$$y = \begin{cases} \dfrac{\sqrt{x+1}}{x^2-1} & x > 1 \\ x^2 + x - 1 & x \le 1 \end{cases}$$

$$\mathrm{x} = \{-2\ \ 4\ \ 3\ \ -1\ \ -5\ \ 8\ \ 6\ \ 9\}$$

Sol.

Method (1): Using Vectorization

```
clear,clc
x=[ -2 4 3 -1 -5 8 6 9 ];
a=find(x>1);
x_pos=x(a);
b=find(x<=1);
x_neg=x(b);
y_pos = sqrt( x_pos + 1 )./( x_pos.^2 - 1 );
y_neg = x_neg.^ 2 + x_neg - 1;
fprintf('   %4.0f    %6.4f \n ',[x_pos;y_pos])
fprintf('   %4.0f    %6.4f \n ',[x_neg;y_neg])
```

Method (2): Using Loops

```
clear,clc
for x=[ -2 4 3 -1 -5 8 6 9 ];
  if x>1
     y = sqrt( x + 1 )/( x^2 - 1 );
  else
     y = x ^ 2 + x - 1;
  end
  fprintf('   %4.0f    %6.4f \n ',[x,y])
end
```