University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 1: Getting Started with MatLab

Prepared by:

Abida Tahsin Hammoodat

2023-2024

Programming With Matlab :-

1. Inputs and Outputs :

1.1 Input Statement :

```
variable name = input ('any string')
```

This command is used when the variable has a numerical value.

```
variable name = input('any string', 's')
```

This command is used when the variable has a vector of string (Symbols, letters, --, etc).

A variable: Is a place in memory that information can be saved in, so that it can be retrieved by their names.

Rules to name a variable :-

1. The name must begin with a letter of alphabet and can contain digits and underscore character (_), but it can't have a space.

2. There is a limit to the length of the name (32 char)

3. Matlab is a case-sensitive, which means there is a difference between upper and lower case letters.

Programming With Matlab

Getting into Matlab

Matlab is a mathematical and graphical software packag with numerical, graphical and programming capabilities. Lik an other software working into windows OS, Matlab has man windows, like: من اهم نوافذ الماتلاب

1. Command window

In Command window, Matlab can be used interactivel At the prompt >>, any Matlab command or expression ca be intered, and Matlab will immediately respond with the result.

2. Command History Window

In this window, previous commands can be saved an retrieved at any time during Matlab session.

3. Work space Window

This window can keep all variables that are used during Matlab session.

Editing Matlab programs (Scripts)

لكتابة برامج بلغة الماتلاب، يجب فتح نافذة منفتح البرنامج

File → New → Script

أو من شريط الادوات ➕

وبعد الانتهاء من كتابة البرنامج يتم حفظه وتنفيذه من خلال F5 أو من شريط الادوات ▶

3. There are certain words called reserved words, or key words, that can't be used as variable names.

4. Names of built-in functions shouldn't be used as variabl names.

When executing input statement, the string is displaye on screen, waits for information to be input from key board. Input accesses variables in the current workspace. If y press the return key without entering anything, input retu an empty matrix.

```
variable name = input('any string', 's')
```

When executing the above commands, Matlab deals with the given variable as a vector of characters (tex without evaluating expressions.

Examples:

Write a Matlab program to calculate the area and the circumference of a rectangle:

```
length = input('length = ');
width = input('width = ');
area = length * width
circ = 2 * (length + width)
```

2. Output Statements :

In Matlab, there are many output statements, like disp, plot, fprintf, printf ... etc.

2.1 disp Statement :

disp(Variable name)

When this command is executed, the value stored in the variable is displayed without any additional information. eg :

```
>> x = 23;
>> disp (x)
```

الجواب سيكون 23 فقط

Variables that can be displayed in disp Command may be vectors, matrices or classes

eg :

```
>> x = 10;
>> y = -2;
>> z = 'Ahmed';
>> disp (x)        this returns
    10
>> disp ([x y])    this returns
    10    -2
```

متغير عادي

vector

2. Output Statements :

In Matlab, there are many output statements, like disp, plot, fprintf, printf .. etc.

2.1 disp statement :

disp(Variable name)

When this command is executed, the value stored in the variable is displayed without any additional information. eg:

```
>> x = 23;
>> disp (x)
```

الجواب سيكون 23 فقط

Variables that can be displayed in disp command may be vectors, matrices or classes

eg:

```
>> x = 10;
>> y = -2;
>> z = 'Ahmed';
>> disp (x)        this returns
    10
>> disp ([x y])    this returns
    10    -2
```

متغير حادي

vector

Class>> disp({x, Z}) this returns
{10} {Ali}

2.1 Format Statement:

This statement is used to control the format of results to be displayed.

format command does not affect how Matlab computations are done.

format may be used to switch between different output display formats

format	short	fixed point with 5 digits.
format	long	fixed point with 15 digits.
format	short e	floating point with 5 digits
format	long e	floating point with 15 digit
format	rat	Approximation by ratio of int
format	bank	fixed formats for dollars a cents.

Examples:

```
>> a = 0.3;
>> b = 1300i;
>> c = a/b;
>> format short
   c =
   2.3077e-4
```

حاسوبيا

رياضيا $\rightarrow 2.3077 \times 10^{-4}$

```
>> format long ↵
c =
   2.307692307692308 e-4

>> format rat ↵
c =
   3/13000
```

Examples:

1. Write a Matlab program to calculate the root of a first order equation $ax+b=0$.

Sol:

```
a = input ('a=');
b = input ('b=');
c = -b/a;
disp (c)
```

2. Write a Matlab program to calculate the average of a given student and display the name and the average:

Sol:

```
m1 = input ('mark1');
m2 = input ('mark 2');
n = input ('Student name', 's');
av = (m1+m2)/2;
disp ({n, av})
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 2: Flow Control Statements

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2. Flow Control Statements :

2.1 Selection.

2.2 Loops .

2.1 Selection

Selection means that some statements of the program would be executed or not according to a certain condition.

2.1.1 The if Statement :

There are many forms of if statements like:

2.1.1.1 The if ... end form :

```
if condition
    action
end
```

A condition is a relational expression that conceptually or logically is true or false.

The action is a statement or groups of statements, that will be executed if the condition is true. The action can be a number of statements until the reserved word end.

Relational Operators

> greater than	>= greater or equal to
< less than	<= less or equal to
== equality	~= inequality

Example:

Write a Matlab program to change any negative number to positive.

Sol:

```
n = input('number = ');
if n < 0
    n = abs(n);
    disp(n)
end
```

We can use more than one condition in the same if statements using logical operators:

&&	and
||	or
~	not

Example:

Write a Matlab program to calculate y where:

$y = \sqrt{x} + 2$ if x is positive and even.

Sol:

```
x = input('x = ');
if x > 0 && mod(x,2) == 0
    y = sqrt(x) + 2;
    disp(y)
end
```

توضيح: دالة الـ mod تستخدم لايجاد باقي القسمة، ففي المثال اعلاه استخدمت دالة الـ mod لمعرفة العدد زوجي اولا بقسمته على 2 وايجاد الباقي.

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 3: If Statements

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2.1.1.2. The if ... else ... end statement

```
if condition
    action t
else
    action f
end
```

When condition is satisfied, statements in (action t) would be executed. Otherwise, statements in (action f) would be executed.

Examples:

1. Write a Matlab program to enter a nominator and a denominator of a fraction. When the denominator equa zero, an error message 'Division by zero' should be displayed. Otherwise, the fraction should be converted its corresponding real number.

Sol:

```
n = input('nominator = ');
d = input('denominator = ');
if d == 0
    disp('Division by zero');
else
    s = n/d;
    disp(s)
end
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 4: If Statements- Continued

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2.1.1.3 If ... elseif ... else ... end Statement

```
if    condition 1
        action 1
elseif condition 2
        action 2
elseif condition 3
  :     action 3
else
        action e
end
```

When condition 1 is satisfied, statements in (action1) would be executed. If it is not satisfied, condition 2 is tested and when it is satisfied, statemen in (action 2) would be executed. When it is not satisfied, condition 3 would be tested, and when it is satisfied, statements in (action 3) would be executed ... etc. Otherwise, when all previous conditions are not satisfied, statements in (action e) would be executed

Examples :

1. Write a Matlab program to find the roots of a second order equation $ax^2 + bx + c = 0$. Verify that $a \neq 0$

sol:

```
a = input ('a = ');
b = input ('b = ');
c = input ('c = ');
if a ~= 0
    d = b^2 + 4*a*c;
    if d > 0
        x1 = (-b + sqrt(d)) / (2*a);
        x2 = (-b - sqrt(d)) / (2*a);
        disp([x1 x2])
    elseif d == 0
        x = -b / (2*a);
        disp(x)
    else
        disp('The equation has imaginary
              roots')
    end
else
    disp('It is not a second order equation')
end
```

2. Write a Matlab program to find f, where

$$f = \begin{cases} 4e^{x+2} & -6 < x < -2 \\ x^2 & -2 < x < 2 \\ \sqrt[3]{x+6} & 2 < x < 6 \end{cases}$$

Sol:

```
x = input('x=')
if x > -6 && x < -2
   f = 4*exp(x+2);
elseif x > -2 && x < 2
   f = x^2;
elseif x > 2 && x < 6
   f = (x+6)^(1/3);
else
   f = 'not in the given range';
end
disp(f)
```

3. Write a Matlab program to find the estimation of a student mark.

Sol:

```
m = input('mark=');
if m < 50
   disp('fail');
elseif m >= 50 && m < 60
   disp('Pass');
elseif m >= 60 && m < 70
   disp('fair');
```

```
elseif m >= 70 && m < 80
   disp('Good');
elseif m >= 80 && m < 90
   disp ('V. good');
elseif m >= 90 && m <= 100
   disp ('excellent');
else
      disp ('not accepted')
end
```

3. Write a Matlab program to find z, where:

$z = 3x^2y + 4xy^2$ and:

$$y = \begin{cases} \frac{1}{\sqrt{x}} + 3 & \text{if } x \text{ is odd and positive} \\ \sqrt{x^2} - 5 & \text{if } x \text{ is even and negative} \\ 0 & \text{otherwise} \end{cases}$$

Sol:

```
x = input ('x = ');
if x > 0 && mod (x,2) == 1
   y = 1/sqrt(x) + 3 ;
elseif x < 0 && mod (x, 2) == 0
   y = sqrt(x^2) - 5 ;
else
   y = 0 ;
end
z = 3*x^2*y + 4*x*y^2;
disp( z)
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 5: Switch – case Statement

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2.1.2 Switch Statement:

```
Switch  switch expression
    Case  case expression 1
          action 1
    Case  Case expression 2
          action 2
    Case  Case expression 3
          action 3
    Otherwise
          action 0
end
```

We use Switch statement when we have multiple choices to select one of them.

The switch expression can be a scalar or character vector or class.

The switch statement executes groups of statements based on the value of a variable or case expression. In Matlab, only the first matching case is executed.

Note: There must be an end to match the switch.

Examples:

1. Write a Matlab program to calculate the surface area and the volume of :
   1. cube
   2. parallelogram
   3. sphere
   4. cone.

Sol:

```
s=input('shape=');
switch  s
   case  1
        l=input('length=');
        a=l^2*6
        v=l^3;
        disp([a v])
   case 2
        l=input('length=');
        w=input('width=');
        h=input('hieght=');
        a=2*(l+w)*h+2*l*w;
        v=l*w*h;
        disp([a v]);
   case 3
        r=input('r=');
        a=4*pi*r^2;
        v=4/3*pi*r^3;
        disp([a v])
   case 4
        r=input('radius=');
        h=input('hieght=');
        a=pi*r*h+pi*r^2
        v=1/3*pi*r^2*h
        disp([a v])
   otherwise
        disp('unaccepted shape')
end
```

In this example, and other programs, String variables can be used as switch expression and case expression :

```
S = input ('Shape :', 's');
switch lower(S)
    case 'cube'
    :
    case 'parllolegram'
    :
    case 'sphere'
    :
    case 'cone'
    :
    otherwise
    :
end
```

Function lower is used to convert its parameter to the lower case :

```
>> lower ( b3AKRt5 )  ↵
    b3akrt5
```

and the function upper is used to convert its parameter to the upper case.

```
>> upper (b3AkRt5) ↵
    B3AkRT5
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 6: Loops – counted Loops

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2.2. Loops

A loop means that some of the program's statements would be repeated many times according to a certain condition. There are two types of loops:

1. Counted Loops: are loops that repeats statements a specif number of times (i,e. it is known ahead of time how m Times the statements are to be repeated).

2. Conditional Loops: are loops that repeats statements, bu ahead of time, it is unknown how many times the statements will need to be executed (i,e. with this type of a loop, it might be said "repeat these statements until this condition is false").

2.2.1 Counted Loops : (For Loops)

```
for index variable= initial value : step : final value
   action
end
```

- The index variable which also known as loop variable should not be assigned to any value within the statemen of action.
- If the step is not mentioned in the for statement, it w be assigned to 1 by default.
- If the final value is less than the initial value , the step should be negative.

- Number of iterations the loop would execute could be calculated :

$$\text{no. of iterations} = int\left(\frac{\text{final value} - \text{initial value}}{\text{step}}\right) + 1$$

Examples:

1. Write a Matlab program to find y, where

$$y = \sum_{i=1}^{20} i$$

Sol:

```
y=0;
for i=1:20
    y=y+i;
end
disp(y)
```

2. A factorial of an integer number could be calculated:

$$n! = n(n-1)(n-2)\ldots \times 2 \times 1$$

Write a Matlab program to find n!

Sol:

```
f=1;
for i=n:-1:1
    f=f*i;
end
disp(f)
```

3. Given x = [-2 3 6 -1 2:3:12 7 9 -4]. Write a Matlab program to count number of even elements in

Sol:

```
n=0;
x=[-2 3 6 -1 2:3:12 7 9 -4];
l= length(x);
for i= 1: l
    if mod(x(i),2) == 0
        n= n+1;
    end
end
disp(n)
```

Nested Loops:

When the action of a loop is another loop, this is called nested loops.

```
for index variable 1 = initial value 1 : step 1 : final value 1
    for index variable 2 = initial value 2 : step 2: final value2
        action
    end
end
```

The first loop is called the outer loop, while the second loop is called the inner loop. The inner loop should be completed before the outer loop variable would be updated to its next value.

- Number of iterations in nested loops can be calculated:

no. of iterations = loop1 iterations x loop2 iterations x...

Examples:

1. Write a Matlab program to display product table for numbers 1-10.

Sol:

```
for i = 1:10
    for j = 1:10
        x(i,j) = i*j;
    end
end
disp(x)
```

2. Write a Matlab program to find the most minimum valu
in the matrix below:

$$\begin{bmatrix} -5 & 2 & 0 & 1 & -10 \\ 4 & 0 & -2 & -1 & 7 \\ -20 & -3 & 1 & 15 & 3 \end{bmatrix}$$

Sol:

```
X=[-5 2 0 1 -10; 4 0 -2 -1 7; -20 -3 1 15 3]
[r,c] = size(x);
m = X(1,1);
for i = 1:r
    for j = 1:c
        if X(i,j) < m
            m = x(i,j);
        end
    end
end
disp(m)
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 7: Loops – counted Loops

Nested Loops

Prepared by:

Abida Tahsin Hammoodat

2023-2024

Second Course:

Loops : More Examples:

1. Write a Matlab program to multiply two matrices using loops.

Sol:

```
x=[          ];
y=[          ];
[rx cx] = size(x);
[ry cy] = size(y);
if cx == ry
    for i = 1:rx
        for j = 1:cy
            z(i,j) = 0;
            for k = 1:cx
                z(i,j) = z(i,j) + x(i,k) * y(k,j);
            end
        end
    end
    disp(z)
else
    disp('not allowed to be multiplied')
end
```

Exiting loops:

To terminate execution of a loop before it is completed.

```
break
```

In nested loops, break exits from the innermost loop only.

Examples:

1. Write a Matlab program to find the least common multiplier between any two integers.

Sol:

```
m = input ('m = ');
n = input ('n = ');
if m <= n
    max = n;
else
    max = m;
end
f = 1;
for i = max : m*n
    if mod(i,m)==0 && mod(i,n)==0
        f = 0;
        break
    end
end
disp(i)
```

2. A prime number is known to be divisible by 1 and itself only. Write a Matlab program to verify that a given number is prime or not.

Sol

```
m=input('m=');
f=1;
for i=2:sqrt(m)
   if mod(m,i)==0
      f=0;
      break
   end
end
if f==1
   disp('The number is prime')
else
   disp('The number is not prime')
end.
```

3. Write a Matlab program to display all prime numbers with the range 3-100.

Sol:

```
for i=3:2:100
  f=1;
  for j=2:sqrt(j)
     if mod(i,j)==0
        f=0;
        break
     end
```

```
    end
    if f==1
        disp(i)
    end
end
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 8: Loops – conditional Loops

Prepared by:

Abida Tahsin Hammoodat

2023-2024

2.2.2 Conditional Loops:

A conditional loop is used to repeat an action when ahead of time it is not known how many times the action will be repeated. The general form is:

```
While condition
    action
end
```

The action, which consists of any number of statements, is executed as long as condition is true.

The way it works is that first the condition is evaluated. If it is logically true, the action is executed. So, the while statement is just like the if-statement. However, at that point the condition is evaluated again. Something in the action has to change something in the condition so that it becomes false in order to stop the loop execution.

Examples:

1. Write a Matlab program to find $m \div n$ using recursive subtraction:

Sol

```
m = input('m=');
n = input('n=');
q = 0;
while m >= 0
    m = m - n;
    q = q + 1;
end
disp([q m])
```

2. Write a Matlab program to find the greatest common divisor between any two integers using Eucledian Algorithm:

Sol

```
m = input('m=');
n = input('n=');
while n ~= 0
    r = mod(m,n);
    m = n;
    n = r;
end
disp(m).
```

3. Write a Matlab program to display Fibonacci sequence terms, where the terms are:

1 1 2 3 5 8 13 ...

such that the second term shouldn't exceed 250.

Sol:

```
f = 1;
s = 1;
while s <= 250
    c = f + s;
    f = s;
    s = c;
end
```

4. Write a Matlab program to create a vector of Fibonacci sequence terms

Sol:

```
x(1) = 1;  i = 3;
x(2) = 1;
while x(i-1) <= 250
    x(i) = x(i-2) + x(i-1);
    i = i + 1;
end
disp (x)
```

5. Write a Matlab program to find y using conditional loop:

$$y = x^2 - x^4 + x^6 - \dots \pm x^n$$

Sol:

```
x = input ('x = ');
n = input ('n = ');
i = 2;
m = 1;
y = 0;
while i <= n
    y = y + m * x^i;
    i = i + 2;
    m = -m;
end
disp (y)
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 10: Functions in MatLab

Prepared by:

Abida Tahsin Hammoodat

2023-2024

3. Functions in Matlab:

A function is a group of statements that together perform a task. In Matlab, functions are defined in seperate files. The name of the file and the function name should be the same.

Functions operate variables within their own workspace, which is also called local workspace, seperate from workspace you access at the Matlab command window which is called the base workspace.

Functions can accept more than one input arguments and may return more than one output arguments.

3.1 Built-in Functions:

There are many functions within Matlab package that are defined and saved within it. So they are called built-in functions. Some of these functions are

1. round(X) - rounds each element of X to the nearest integer.

eg:

```
round(3.76)        ;   round(-3.76)
ans:  4                ans:  -4
round(3.49)        ;   round(-3.49)
ans:  3                ans:  -3
```

2. fix(X) - rounds towards zero, regardless the fraction part

eg:

fix (7.95) ; fix (-7.95)

ans: 7 ans -7

3. ceil (x) - rounds each element of x towards + ∞

eg:

ceil (4.33) ; ceil (-7.39)

ans: 5 ans: -7.

4. floor (x) - rounds each element of x towards - ∞

eg:

floor (5.76) ; floor (-5.76)

ans: 5 ans: -6

5. rand (x) - generates an x * x matrix uniformly distributed psuedo random numbers within the range 0-1.

6. magic (x) - creates an x * x matrix constructed of integers 1- x^2 with equal row, column and diagonal sums.

eg:

magic (3)

ans:

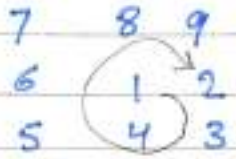
8	1	6
3	5	7
4	9	2

7. Spiral (X) - is an X*X matrix with elements ranging 1-X in a rectangular spiral patterns.

eg:

Spiral (3)

ans:



8. Linspace (initial value, final value, no. of elements) :
creates linearly spaced vector.

* if no. of elements is not mentioned, the generated vecto would be of 100 elements.

* if initial value = final value. All the vector's elements would be the same.

Examples :-

1. Write a Matlab program to generate 3 integers within the range 0-9, verify if:
   1. the three numbers are equal to 7, then display (Big winner)
   2. One or more numbers equal to 7, then display (winne
   3. if none of them equals 7, then display (Big looser).

Sol:

```
x = fix ( rand (1,3) *10);
if x(1) == 7 && x(2) == 7 && x(3) == 7
    disp ('Big Winner')
elseif x(1) == 7 || x(2) == 7 || x(3)
```

```
    disp ('Winner')
else
    disp ('Big Looser')
end
```

3.2. User_Defined Functions:

A user_defined function is a piece of code or a program that we can create and use later as any built_in function.

All we need to do is to save our code as a text file and ensuring that the name of our function and the name of the file are the same.

Steps to write a user_defined function:

1. Open a new script to write the function in it.
2. Write your function in the opened file.
3. Save and run your function. and as mentioned above, the function must be saved within the name it would be called with.

Syntax of a User_defined function:

```
function [a1 ... an] = function name (x1, x2, ... xn)
    action
```

University of Mosul

College of Science

Department of Physics

2nd Class

Computer 2 & 3

Programming with MatLab

Lecture 12: User-Defined Functions

Prepared by:

Abida Tahsin Hammoodat

2023-2024

$a_1 \ldots a_n$: are the outputs that are resulted by the function
$x_1, x_2, -x_n$: are the inputs to be used by the function.

Examples:

1. Write a user-defined function to find the remainder of $m \div n$.

Sol:

```
function r = remain(m,n)
a = fix(m/n)*n;
r = a;
```

2. Write a user-defined function to find the average of a matrix in a row-wise direction:

Sol:

```
function av = rowav(X)
[r c] = size(X);
for i = 1:r
    s(i) = 0;
    for j = 1:c
        s(i) = s(i) + X(i,j)
    end
    av(i) = s(i)/c;
end
```

3. Write a user defined function to count the elements within a matrix:

Sol:

```
function n = matcount(X);
[r c] = size(X);
n=0;
for i=1:r
    for j=1:c
        n=n+1;
    end
end
```

4. Write a User-defined function to verify if an integer is prime or not. Use this function to create a vector of all prime integers within the range 3-100:

Sol:

```
function f = prim(m);
f=1;
for i=2:sqrt(m)
    if mod(m,i)==0
        f=0;
        break
    end
end
```

والبرنامج الرئيسي يكون كالآتي

```
j=0;
for k=3:2:100
  p=prim(k);
  if p==1
    j=j+1;
    y(j)=k;
  end
end
```

5. Write a user-defined function called fact to find m!. Use this function in another user defined function called term to calculate the $\frac{z^k}{k!}$. Then write a Matlab program to find y

$$y = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \pm \frac{x^n}{n!}$$

Sol:

حساب m!

```
function f=fact(m)
f=1;
for i=m:-1:1
    f=f*i;
end
```

حساب $\frac{z^k}{k!}$

```
function t=term(z,k);
t= z^k / fact(k);
```

والبرنامج الرئيسي يكون كما يلي:

```
x = input ('x = ');
n = input ('n = ');
y = 0;
 m = 1;
 for j = 0 : 2 : n
    y = y + m * term (x, j);
 end
 disp (j)
```

5. Write a user-defined function called matmath to find :
   1. Addition of two matrices when the operation '+'
   2. Subtraction of two matrices when the operation '-'
   3. Dot production when the operation is '.*'
   4. Matrix multiplication when the operation is '*'

Sol:

```
function z = mathmat ( x, y, o)
[ rx  cx ] = size(x);
[ ry  cy ] = size(y);
switch  o
   case '+'
      if  rx == ry && cx = cy
          for i = 1 : rx
              for j = 1 : cx
                  z (i,j) = x (i,j) + y (i,j);
              end
          end
```

```
        end
    case '-'
        if cx == cy && rx == ry
            for i = 1 : rx
                for j = 1 : cx
                    z(i,j) = x(i,j) - y(i,j);
                end
            end
        end
    case '.*'
        if rx == ry && cx == cy
            for i = 1 : rx
                for j = cx
                    z(i,j) = x(i,j) * y(i,j);
                end
            end
        end
    case '*'
        if cx == ry
            for i = 1 : rx
                for j = 1 : cy
                    z(i,j) = 0
                    for k = 1 : cx
                        z(i,j) = z(i,j) + x(i,k) * y(k,j
                    end
                end
            end
        end
    end
```

The function can have one or more input variables and can return one or more output variables, so we can find in Matlab the following built-in variables:

- nargin : no. of input arguments.
- nargout : no. of output arguments.

With these variables we can control no. of inputs or outputs when calling a function:

Examples:

1- Write a user-defined function to find the area of a quadrelateral:

Sol:

```
function a = garea(l,w)
  switch nargin
      case 1
          a = l*l;
      case 2
          a = l*w;
  end
```

When calling this function with one input variable, the area of a square will be found. Otherwise the area of a rectangle will be calculated.

2. Write a Matlab program to find the goutient of $m \div n$, using recursive subtraction:

Sol:

```
function [q, r] = divd(m,n)
q=0;
while m >= n
    m = m-n;
    q = q+1;
end
if nargout > 1
   r = m;
end
```

Calling this function with one output variable, only the goutient will be returned. When calling this function with two outputs the goutient and the remainder will be displayed.

1. Getting Started with Matlab

https://www.youtube.com/watch?v=83S48Fs9WhY

2. Flow Control Statements :

2.1 Selection

2.1.1 If Statement

https://www.youtube.com/watch?v=6YbnCRYHQ4Y

2.1.2 Switch Statement

https://www.youtube.com/@MATLAB

2.2 Loops

2.2.1 counted Loops

https://www.youtube.com/watch?v=1sMOOceThTc

2.2.2 conditional Loops

https://www.youtube.com/watch?v=Xb9UTsZjIlw&list=PLPhi9dVsEVfZFAo5GSQ7vWX5UbLwacWsv&index=6&pp=iAQB

2.2.3 Exiting Loops

https://www.youtube.com/watch?v=VO3vg9DSHxo

3. Functions in Matlab

https://www.youtube.com/@MATLAB

3.1 Built in Functions

https://www.youtube.com/@LaplaceAcademy/shorts

3.2 User Defined Functions

https://www.youtube.com/@SpartanProfessor