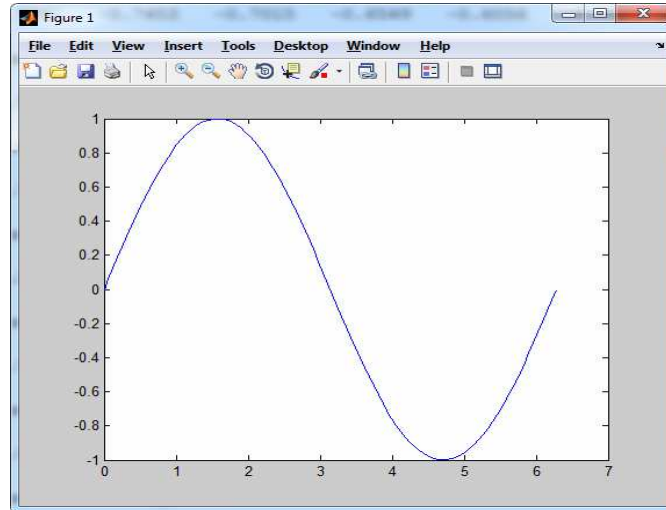


Plotting in MATLAB

- `linspace(a,b,n)`
 - ❖ Generates linearly spaced vectors between “a” and “b” with “n” number
- `x = linspace(0,2*pi,100)`
- `y = sin(x)`
- `plot(x,y)`
 - ❖ Plots functions with x and y

plot(x,y) result

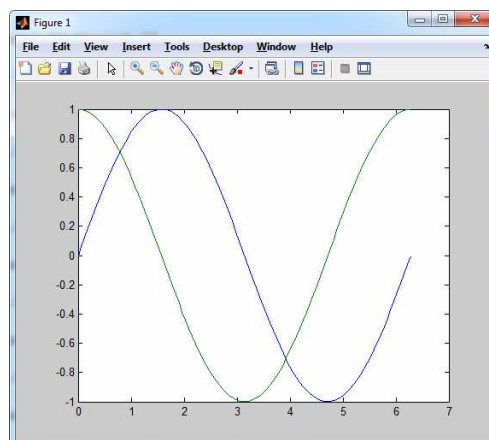


Özgür ZEYDAN

3

Plotting in MATLAB

- $z = \cos(x)$
- `plot(x,y,x,z)`

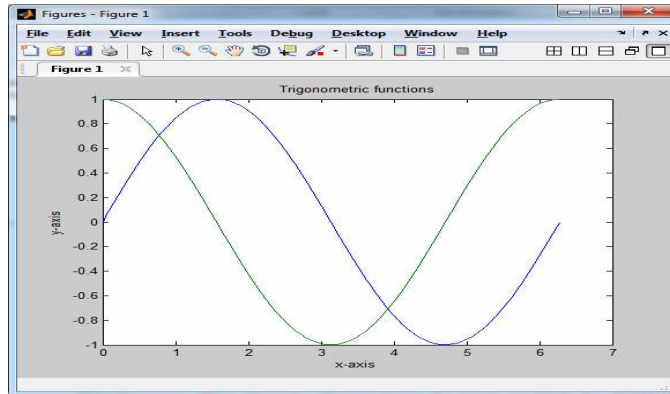


Özgür ZEYDAN

4

Title and Labels of Graph

- `title('Trigonometric functions');`
- `xlabel('x-axis');`
- `ylabel('y-axis');`



Özgür ZEYDAN

5

help plot command

```
>> help plot
plot Linear plot.
plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
then the vector is plotted versus the rows or columns of the matrix,
whichever line up. If X is a scalar and Y is a vector, disconnected
line objects are created and plotted as discrete points vertically at
X.

plot(Y) plots the columns of Y versus their index.
If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)).
In all other uses of plot, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with
plot(X,Y,S) where S is a character string made from one element
from any or all the following 3 columns:
```

Color	b	blue	.	point	-	solid	Line style
	g	green	o	circle	:	dotted	
	r	red	x	x-mark	-.	dashdot	
	c	cyan	+	plus	--	dashed	
	m	magenta	*	star	(none)	no line	
	y	yellow	s	square			
	k	black	d	diamond			
	w	white	v	triangle (down)			
			^	triangle (up)			
			<	triangle (left)			
		>	triangle (right)				
Point style		p	pentagram				
		h	hexagram				

Özgür ZEYDAN

6

Plotting with Styles

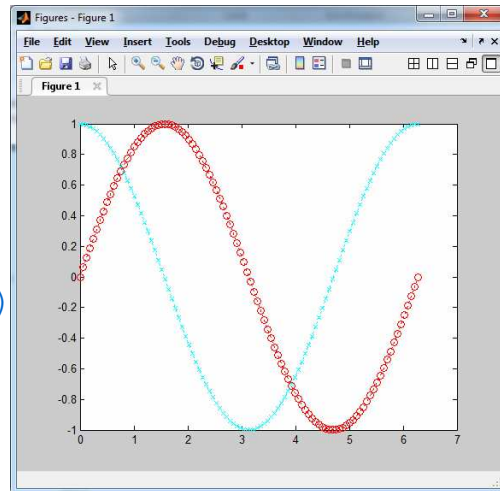
➤ `plot(x,y,'r:o',x,z,'c:x')`

➤ Or

➤ `plot(x,y,'r:o')`

➤ `hold`

➤ `plot(x,z,'c:x')`



Özgür ZEYDAN

7

Sub Plots

➤ `subplot(1,2,1);`

➤ `plot(x,y);`

➤ `xlabel('x-axis');`

➤ `ylabel('y-axis');`

➤ `legend('sinus');`

➤ `title('Sinus Function')`

➤ `subplot(1,2,2);`

➤ `plot(x,z,'r');`

➤ `ylabel('y-axis');`

➤ `xlabel('x-axis');`

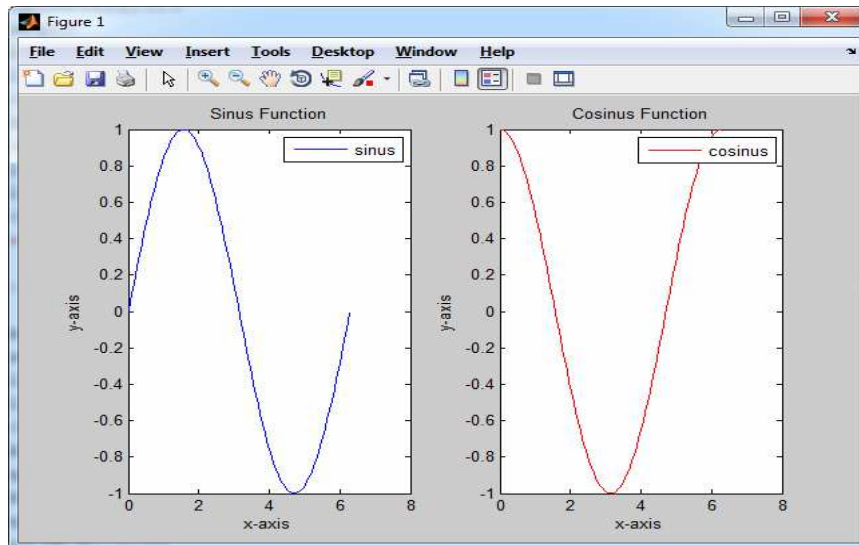
➤ `legend('cosinus');`

➤ `title('Cosinus Function')`

Özgür ZEYDAN

8

Sub Plots



Özgür ZEYDAN

9

Axis Commands

- `axis([xmin xmax ymin ymax])`
- `axis off`
 - ❖ Removes axis
- `axis on`
 - ❖ Shows axis

Özgür ZEYDAN

10

Systems of Linear Equations

- Suppose you have a system of linear equation

- ❖ $2x - y + z = 8$

- ❖ $-x + y - 2z = -9$

- ❖ $x + y - 3z = -8$

- $A = [2 \ -1 \ 1; \ -1 \ 1 \ -2; \ 1 \ 1 \ -3]$

- $b = [8; -9; -8]$

- System is defined as : $Ac=b$

- Solution is : $c=A \setminus b$

- ❖ $c = [x \ y \ z]$

```
>> A=[2 -1 1; -1 1 -2; 1 1 -3];
>> b=[8;-9;-8];
>> c=A\b
c =
    2.0000
   -1.0000
    3.0000
```

Solving Polynomials

- Let $f(x) = ax^4 + bx^3 + cx^2 + dx + e = 0$

- ❖ N^{th} order polynomial

- Define $F=[a \ b \ c \ d \ e]$ matrix

- ❖ $N+1$ length matrix

- $y=\text{roots}(F)$ gives roots of $f(x)$

- Ex: solve $x^3 - 4x^2 + x + 6 = 0$

```
>> F=[1 -4 1 6];
>> y=roots(F)
y =
    3.0000
    2.0000
   -1.0000
```

Evaluating Polynomials

- Evaluate polynomial at a certain 'x0' point or points (x1, x2, x3, x4 etc)
- `y=polyval(F,x0)`

- `A=[x1, x2, x3, x4];`
- `Y=polyval(F,A)`
- Or
- `Y=polyval(F,[x1, x2, x3, x4])`

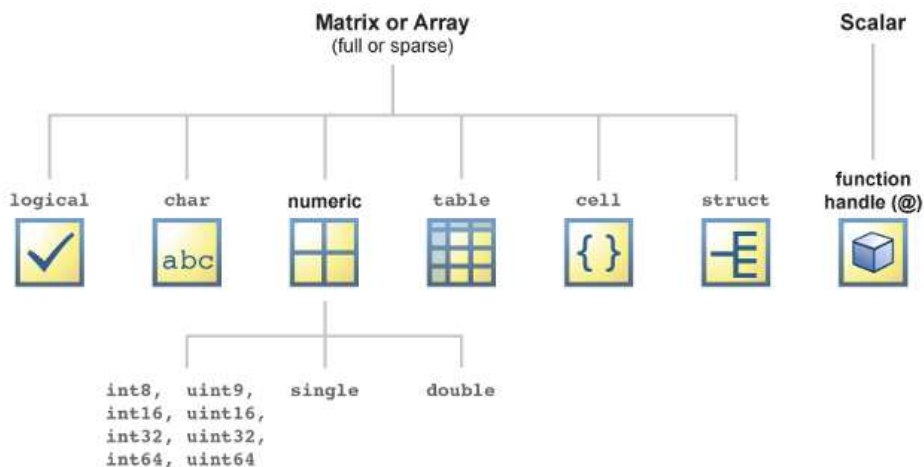
Programming in MATLAB

- Variables
- Input and Output Commands
- Decision (Selections) Structures
 - ❖ If
 - ❖ Switch
- Loop (Repetition) Structures
 - ❖ For
 - ❖ While

Variable Names

- A valid variable name starts with a letter, followed by letters, digits, or underscores.
- A valid variable names must be written with English alphabet.
- MATLAB is case sensitive
 - ❖ **name** and **NAME** are different variables!
- Valid variable examples:
 - ❖ Student
 - ❖ student_1
 - ❖ ENVE_student
- Invalid variable examples:
 - ❖ student-1
 - ❖ student.1
 - ❖ 1.student
 - ❖ Student first

Variable Types



Variable Limits

Class	Max value	Min Value	Bytes	Smallest difference
logical	1	0	1	1
int8	127	-128	1	1
int16	32767	-32768	2	1
int32	2.14e+09	-2.14e+09	4	1
int64	9.22e+18	-9.22e+18	8	1
uint8	255	0	1	1
uint16	65535	0	2	1
uint32	4.29e+09	0	4	1
uint64	1.84e+19	0	8	1
single	3.40e+038	-3.40e+038	4	1.1755e-38
double	1.79e+308	-1.79e+308	8	2.2251e-308

<http://www.matlabtips.com/you-must-choose/>

Özgür ZEYDAN

17

Input and Output Commands

- `your_number=input('Write a number:')`
 - ❖ The class of “your_number” variable is double
- `name=input('Write your name:', 's')`
 - ❖ The class of “name” variable is char

```
>> your_number=input('Write a number:')
Write a number:25


your_number =

    25

>> name=input('Write your name:', 's')
Write your name:Özgür

name =

Özgür
```



Name	Value	Size	Bytes	Class
name	'Özgür'	1x5	10	char
your_number	25	1x1	8	double

Özgür ZEYDAN

18

Input and Output Commands

- `disp` function is used as output command in MATLAB
- `disp('MATLAB Programming')`

```
>> disp('MATLAB Programming')
MATLAB Programming
```

If Command

```
if condition
    commands
end
```

```
>> x=9;
>> if (x<10) disp('X is smaller than 10')
end
X is smaller than 10
```

```
if condition
    commands-1
else
    commands-2
end
```

```
>> x=9;
>> if x>10 disp('X is bigger than 10')
else disp('X is smaller than 10')
end
X is smaller than 10
```

```
if condition-1
    commands1
elseif condition-2
    commands2
else
    commands3
end
```

```
>> x=10;
>> if (x>10) disp('X is bigger than 10')
elseif (x<10) disp('X is smaller than 10')
else disp('x is equal to 10')
end
x is equal to 10
```

Relational Operators

equal	==
not equal	~=
greater than	>
less than	<
greater or equal	>=
less or equal	<=

Logical Operators

And	&
Or	
Not	~

If Example

- Check whether the given number is odd or even?
- `number=23;`
- `if (rem(number,2) == 1) disp('number is odd')`
- `else disp('number is even')`
- `end`
- or
- `if (rem(number,2) ~= 1) disp('number is even')`
- `else disp('number is odd')`
- `end`

If Example

- Using If structure in MATLAB
- Display the **letter** of a student with respect to his/her **grade**

Letter	Grade range
A	90 – 100
B	80 – 89
C	70 – 79
D	60 – 69
F	0 – 59

Solution

```
>> grade = 85;
if grade >= 90, letter = 'A'
elseif grade >= 80, letter = 'B'
elseif grade >= 70, letter = 'C'
elseif grade >= 60, letter = 'D'
else letter = 'F'
end

letter =

B
```

Switch

```
switch condition
case x1, commands-1
case x2, commands-2
case x3, commands-3
...
case xn, commands-n
otherwise commands-n+1
end
```

```
>> day=input('Write the number of today?');
switch day
case 1, disp('Today is Monday')
case 2, disp('Today is Tuesday')
case 3, disp('Today is Wednesday')
case 4, disp('Today is Thursday')
case 5, disp('Today is Friday')
case 6, disp('Today is Saturday')
case 7, disp('Today is Sunday')
otherwise disp('Wrong day number!')
end
Write the number of today?1
Today is Monday
```

Switch Example

- Dice game



- Roll 2 dices by using **rand** and **floor** functions in Matlab.
- `d1 = floor(6*rand) + 1;`
- `d2 = floor(6*rand) + 1;`
- You will win if sum of 2 dices are 7, lose otherwise.
- Use **switch** to check win condition.

Switch Example

```
>> d1 = floor(6*rand)+1;
d2 = floor(6*rand)+1;
switch d1+d2
case 7
disp( '7 - You win' );
otherwise
disp( 'You lose' );
end
You lose
```

For

- In for loop, number of iterations are known.

```
for loop variable  
    commands  
end
```

- Loop variable is a vector.

```
>> for i=1:5  
    disp('Matlab Programming')  
end  
Matlab Programming  
Matlab Programming  
Matlab Programming  
Matlab Programming  
Matlab Programming
```

For Example

- Calculate

$$\sum_{i=1}^{10} i^2$$

- using for loop

```
>> sumA=0;  
>> for i=1:10  
    sumA=sumA+i*i;  
end  
>> sumA  
  
sumA =  
  
    385
```

While

- Loop is executed as long as condition gives true.
- Programmer should check condition to avoid infinite loops.

```
while condition
    commands
end
```

```
>> j=5;
while j>0
disp(j)
j=j-1;
end
5
4
3
2
1
```

While Example

- Create $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$ graphs by using **subplot** and **while** commands.
- `x=linspace(-2*pi,2*pi,100);`
- `i=1;`
- `while i<=4`
- `subplot(2,2,i), plot(x,sin(i*x)),`
`title(['Sinus ',num2str(i),'x']),`
`i=i+1;`
- `end`

While Example

