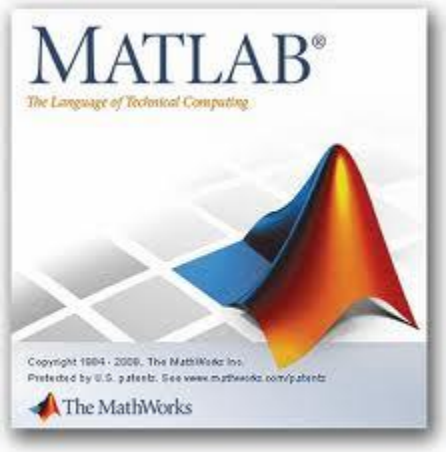
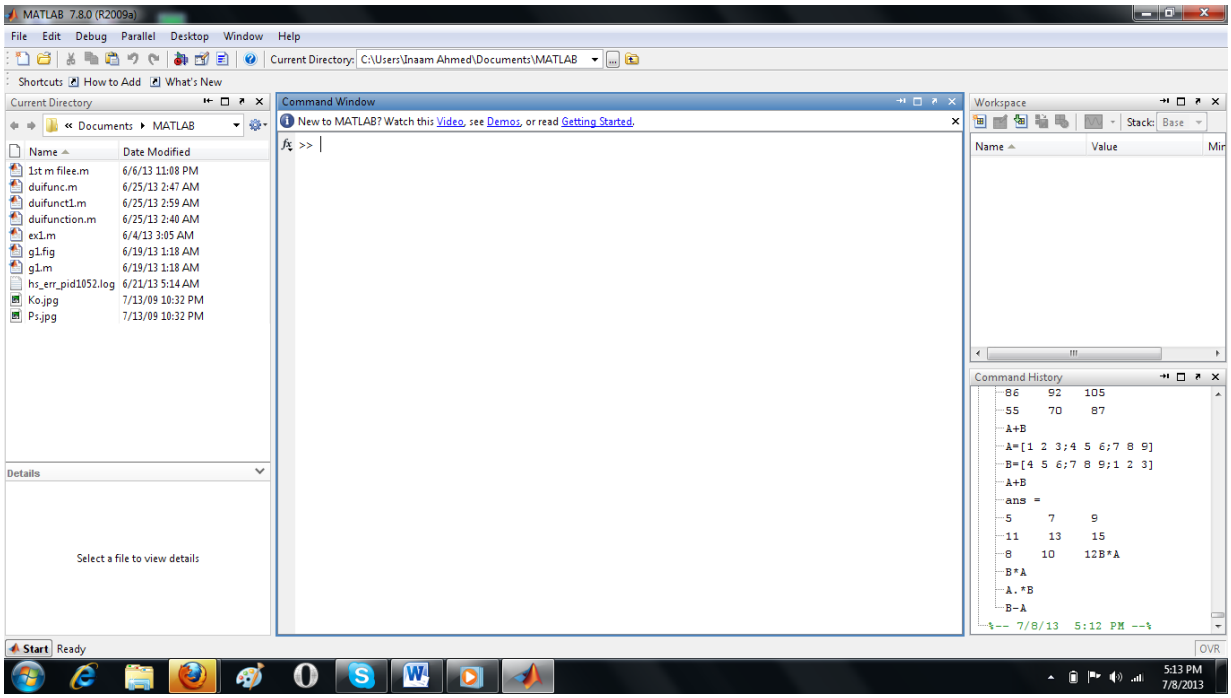
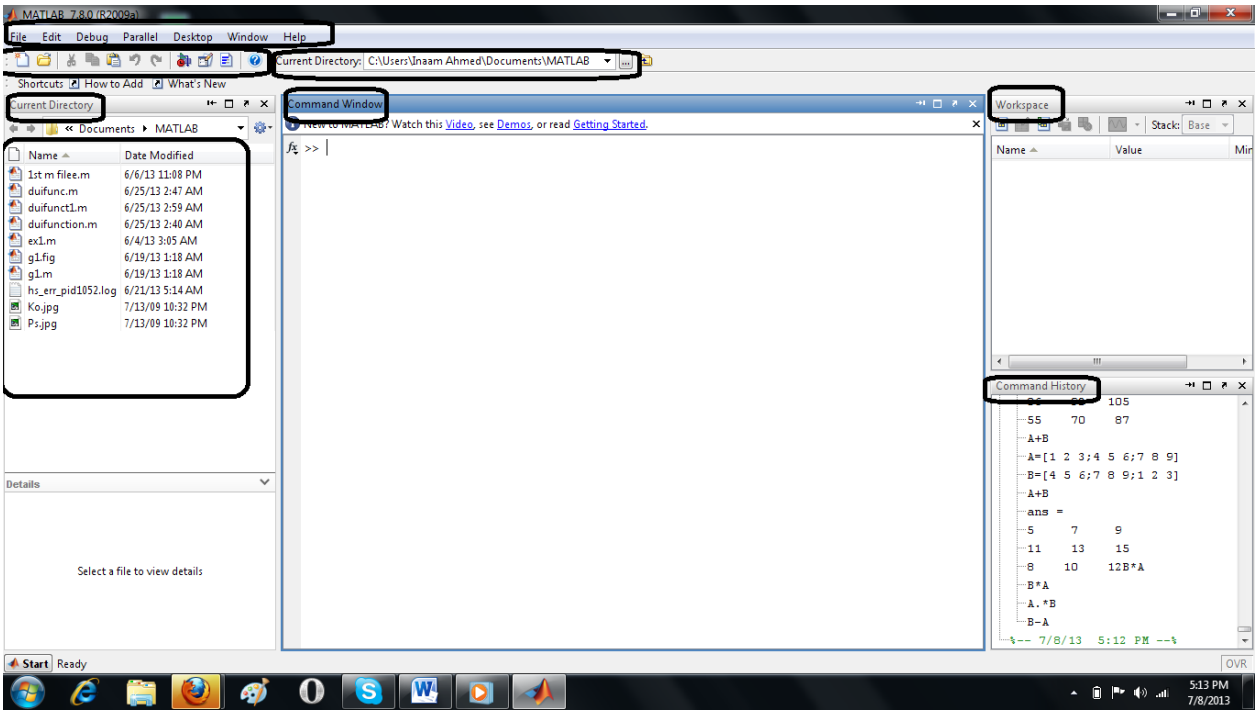
Signals and System



LAB work #01: Getting Started With MATLAB

Interface





MATLAB can be thought of a powerful graphic calculator in addition it is a programming Language

- MATLAB is interpreted language
- Commands Executed Line by line

History of MATLAB: LAB work #02

Conversing with MATLAB

```
>> a=[1 2  3];
>> b=[4 5 6];
>> who

Your variables are:

a  b
>> what

M-files in the current directory C:\Users\Inaam Ahmed\Documents\MATLAB

duifunc       duifunct1      duifunction    ex1          g1
```

Naming variable Types

Create a variable, Simply assign a value to name

```
>> inaam=4444


inaam =


      4444


>> home=2.71


home =


   2.7100


>> place='Lahore'


place =
```

Lahore

Built in-variable Types

i and ***j*** *ca*n be used to indicate complex numbers

pi has the value 3.1415926

ans store the last unassigned value

Inf and ***–inf*** are positive and negate infinity

nan Represents *'Not a number'*

>> x=0+i

x =

0 + 1.0000i

>> a=pi

a =

3.1416

>> c=nan

c =

NaN

LAB work #03

Hello World

>> 'hello wolrd'

ans =

hello wolrd

>> disp('hello world')

hello world

>> a=1.83;

>> disp(sprintf('hello %g',a))

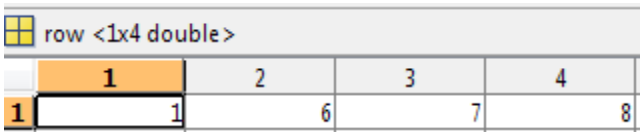
hello 1.83

Scalars

```
>> a=10

a =

    10

>> c=1.34*45-6*a

c =

    0.3000
```

Row Vectors

```
>> row=[1 6 7 8]

row =

    1     6     7     8
```

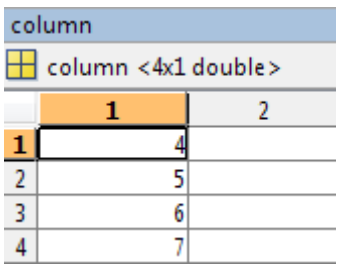


row <1x4 double>

	1	2	3	4
1	1	6	7	8

Row Vectors

```
>> column=[4;5;6;7]

column =

    4

    5

    6

    7
```



column

column <4x1 double>

	1	2
1	4	
2	5	
3	6	
4	7	

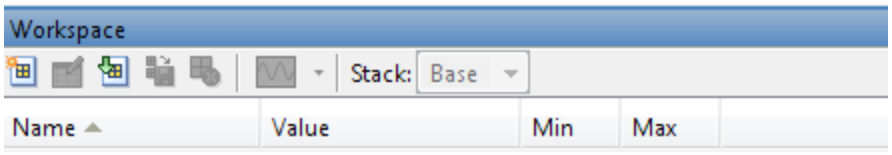
Matrices

```
>> a=[1 2;4 5]

a =

    1    2

    4    5

>> a=[1 2 3;4 5 6]

a =

    1    2    3

    4    5    6

>> a=[1 2]


a =

    1    2
```

Clear/clc

To remove the veriables from the enviconment

```
>> column=[4;5;6;7]

column =

    4

    5

    6

    7


>> clear

>>
```


Workspace
Stack:
Base
Name
Value
Min
Max

LAB work #04

Basic scalar operations

```
>> 7+2
ans =
     9
>> 0/0
ans =
   NaN
>> 1*0
ans =
     0
>> 2^8
ans =
   256
>> (2+i)*(4+j)
ans =
   7.0000 + 6.0000i
```

Built in Functions

```
>> sqrt(2)
ans =
    1.4142
>> log(2)
ans =
    0.6931
>> log10(0.53)
ans =
```

```
   -0.2757
>> cos(1.2)
ans =
    0.3624
>> atan(-.8)
ans =
   -0.6747
>> round(1.4)
ans =
     1


>> exp(2+5i)
ans =
   2.0960 - 7.0855i
>> angle(i)
ans =
    1.5708
>> abs(1+i)
ans =
    1.4142
```

Help

Get info on how to use a function.

```
>> help tan
 TAN    Tangent of argument in radians.
    TAN(X) is the tangent of the elements of X.
    See also atan, tand, atan2.
    Overloaded methods:
```

codistributed/tan

Reference page in Help browser

doc tan

LAB work #05

Size and Length

---

a =

1 2 3

1 25 6

>> size(a)

ans =

2 3

>> length(a)

ans =

3

Transpose

---

b =

45 6 2

3 23 67

>> transpose(b)

ans =

45 3

6 23

2 67

Addition And Subtraction

```
a =

     1     2     3

     1    25     6

b =

    45     6     2

     3    23    67

>> a+b

ans =

    46     8     5

     4    48    73

>> a-b

ans =

   -44    -4     1

    -2     2   -61
```

LAB work #06

Element wise function

```
>> t=[1 2 3];
>> f=exp(t)
f =
   2.7183    7.3891   20.0855
```

Element wise operators

```
>> a.*b
ans =
   45    12     6
    3   575   402
>> a./b
ans =
   0.0222    0.3333    1.5000
   0.3333    1.0870    0.0896
```

Exercise Mul 4×4 Matrices

```
>> a=[1 2 3 4;5 6 7 8;9 10 11 12;45 23 87 21];
>> b=[1 5 3 4;5 100 7 8;9 10 45 12;63 23 87 21];
>> a*b
ans =
        290         327         500         140
        602         879        1068         320
        914        1431        1636         500
       2266        3878        6038        1849
```

Line Space (Automatic Initialization)

```
>> a=linspace(0,10,5)
```

```
a =

        0    2.5000    5.0000    7.5000   10.0000

>> b=0:2:10

b =

    0     2     4     6     8    10
```

LAB work #07

Vector Indexing

```
>> x=[12 13 14 5 8];

>> a=x(1:2)

a =

   12    13

>> b=x(1:end-1)

b =

   12    13    14     5
```

Metrix Indexing

```
b=[1 2;5 6];

A=rand(5)

A =

   0.8147    0.0975    0.1576    0.1419    0.6557
   0.9058    0.2785    0.9706    0.4218    0.0357
   0.1270    0.5469    0.9572    0.9157    0.8491
   0.9134    0.9575    0.4854    0.7922    0.9340
   0.6324    0.9649    0.8003    0.9595    0.6787



A(1:3,1:2)

ans =
```

0.8147 0.0975

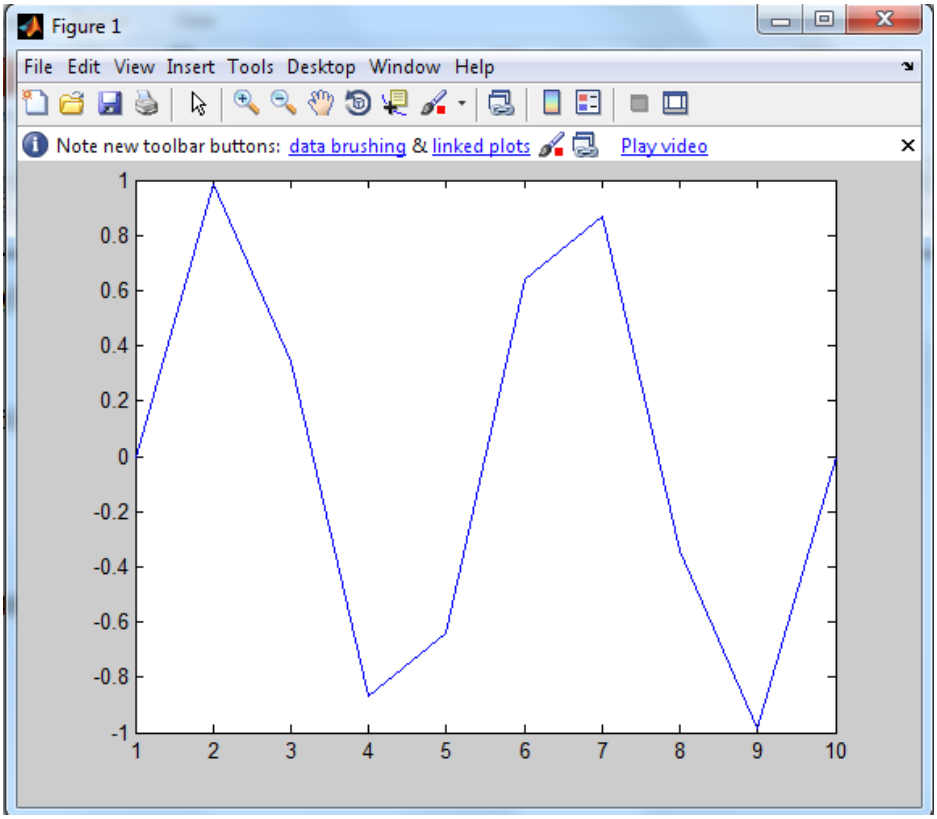
0.9058 0.2785

0.1270 0.5469

LAB work #08

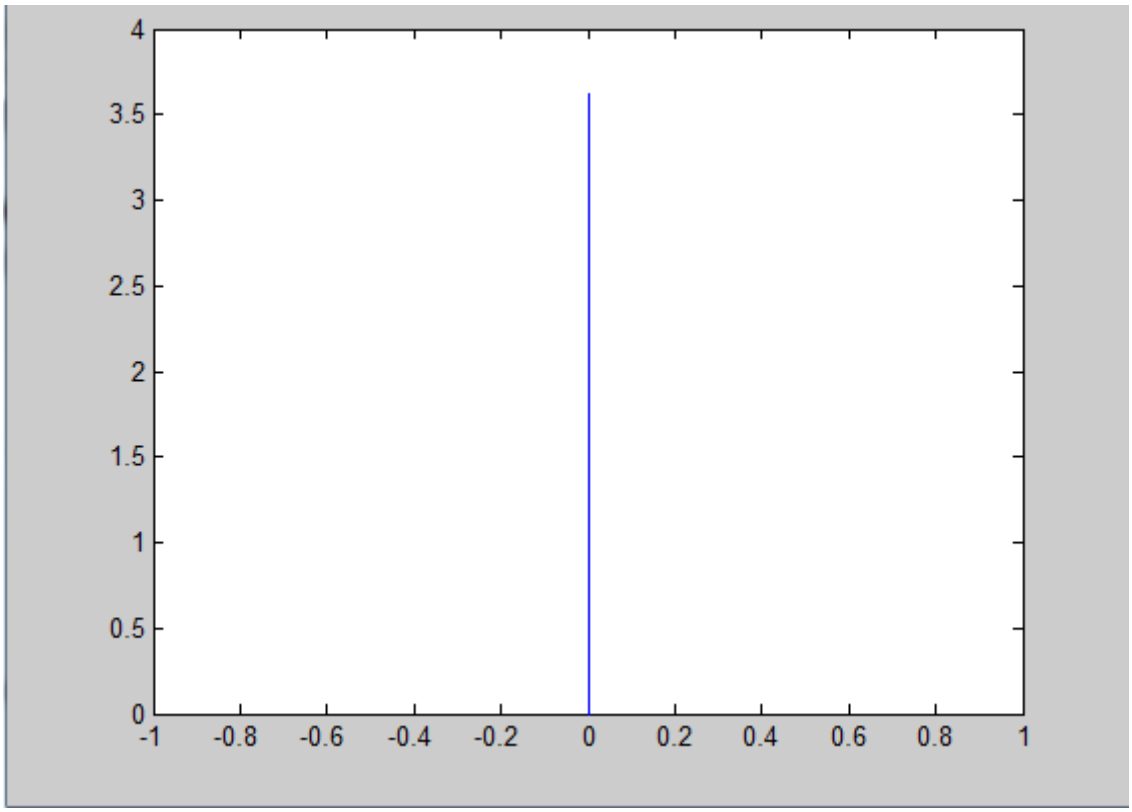
Plotting Vector

```
>> x=linspace(0,4*pi,10)

x =

    0    1.3963    2.7925    4.1888    5.5851    6.9813    8.3776    9.7738   11.1701   12.5664

>> y=sin(x)

y =

    0    0.9848    0.3420   -0.8660   -0.6428    0.6428    0.8660   -0.3420   -0.9848   -0.0000

>> plot(y)
```



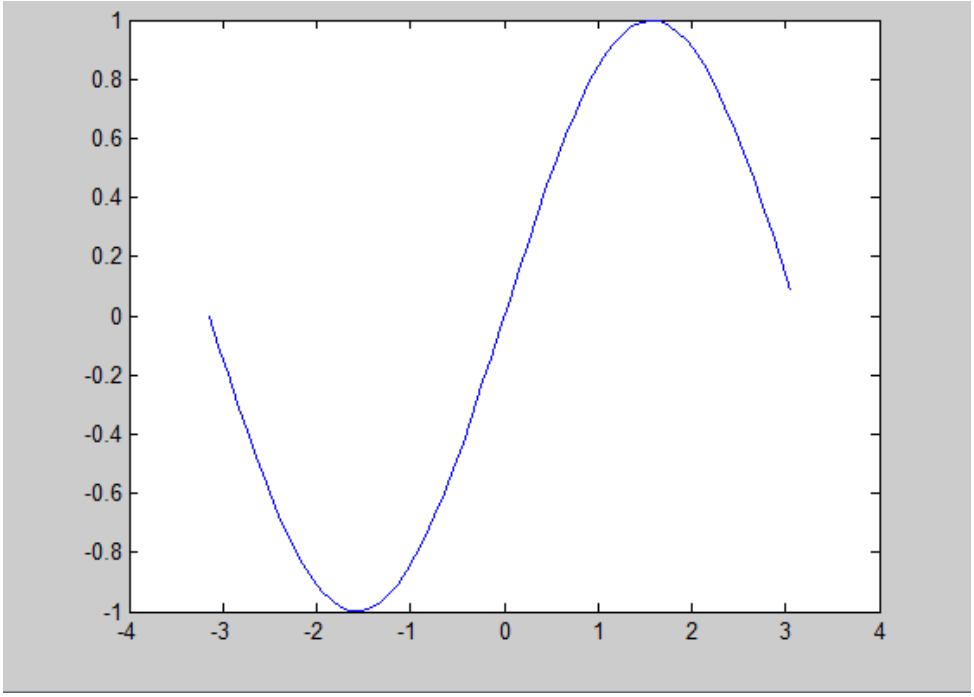

```
>> x=1:2:20

x =

    1     3     5     7     9    11    13    15    17    19
```

```
>> y=acos(x)

y =

  Columns 1 through 8

        0                0 + 1.7627i        0 + 2.2924i        0 + 2.6339i        0 + 2.8873i        0 + 3.0890i        0 + 3.2566i
0 + 3.4001i

  Columns 9 through 10

        0 + 3.5255i        0 + 3.6369i

>> plot(y)
```

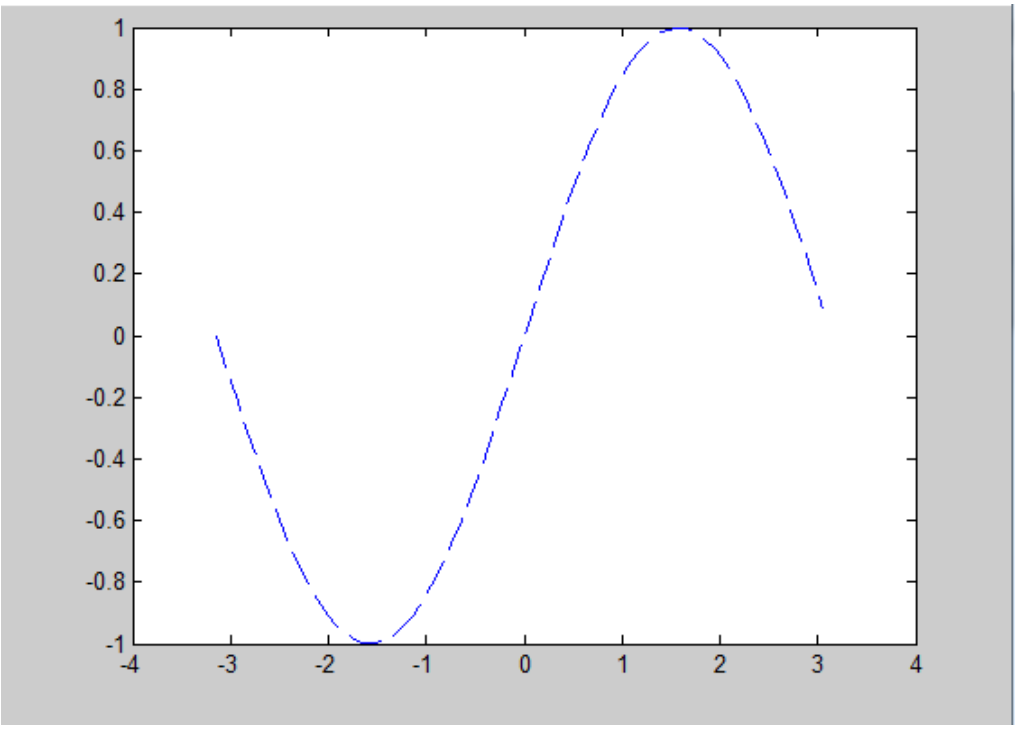


LAB work #09

```
>> x=-pi:.1:pi;

>> y=sin(x);

>> plot(x,y)
```

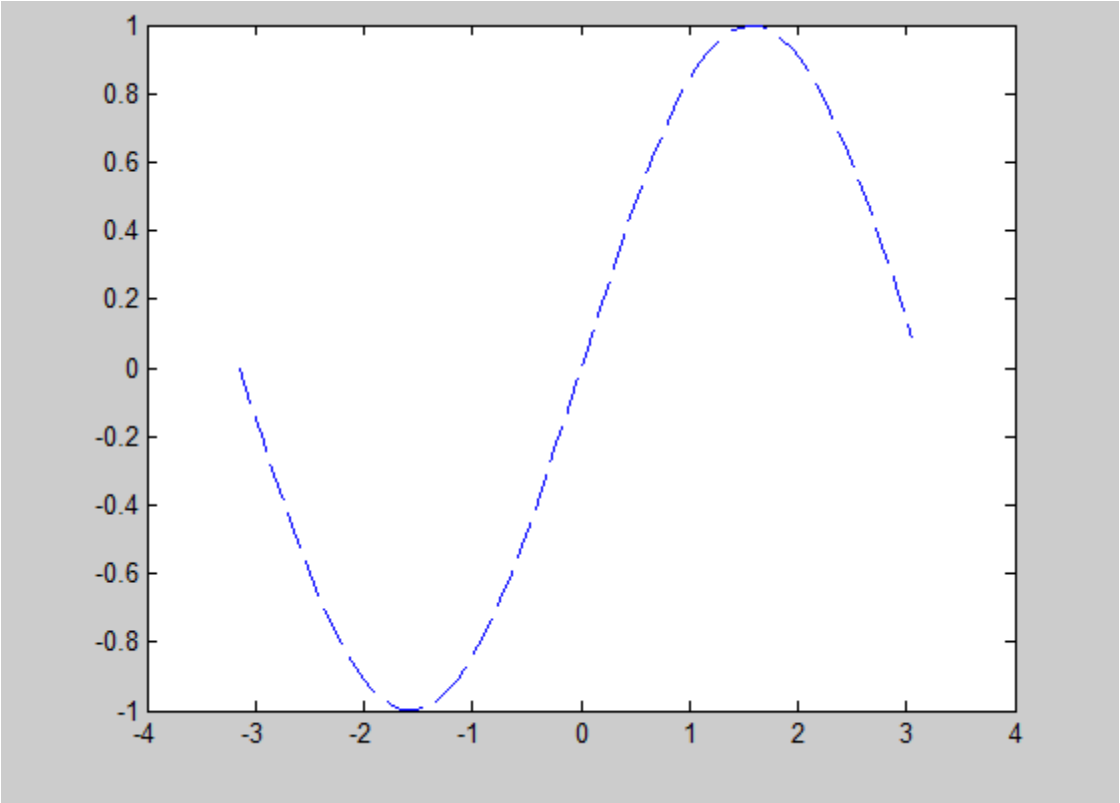



```
>> x=-pi:.1:pi;
>> y=sin(x);
>> plot(x,y,'--')
```



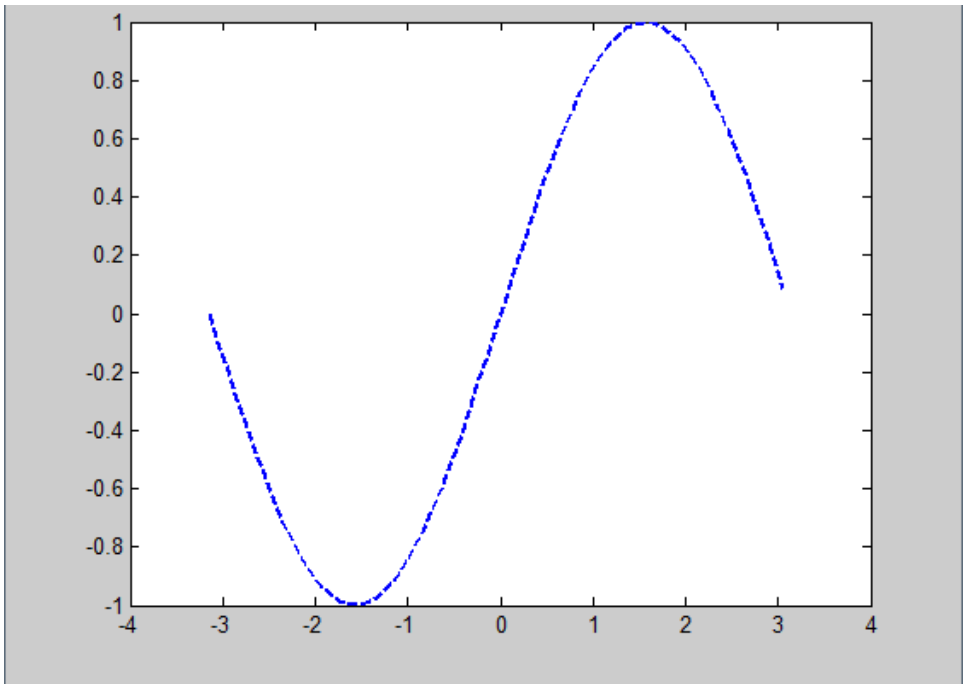

```
>> x=-pi:.1:pi;
>> y=sin(x);
```

```
>> plot(x,y,'.')
```

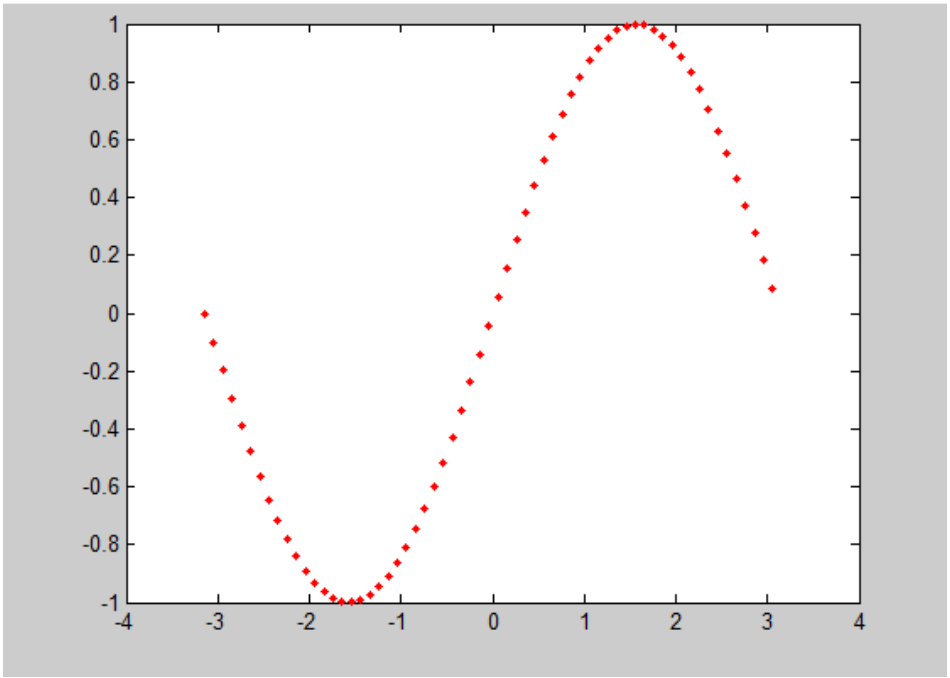


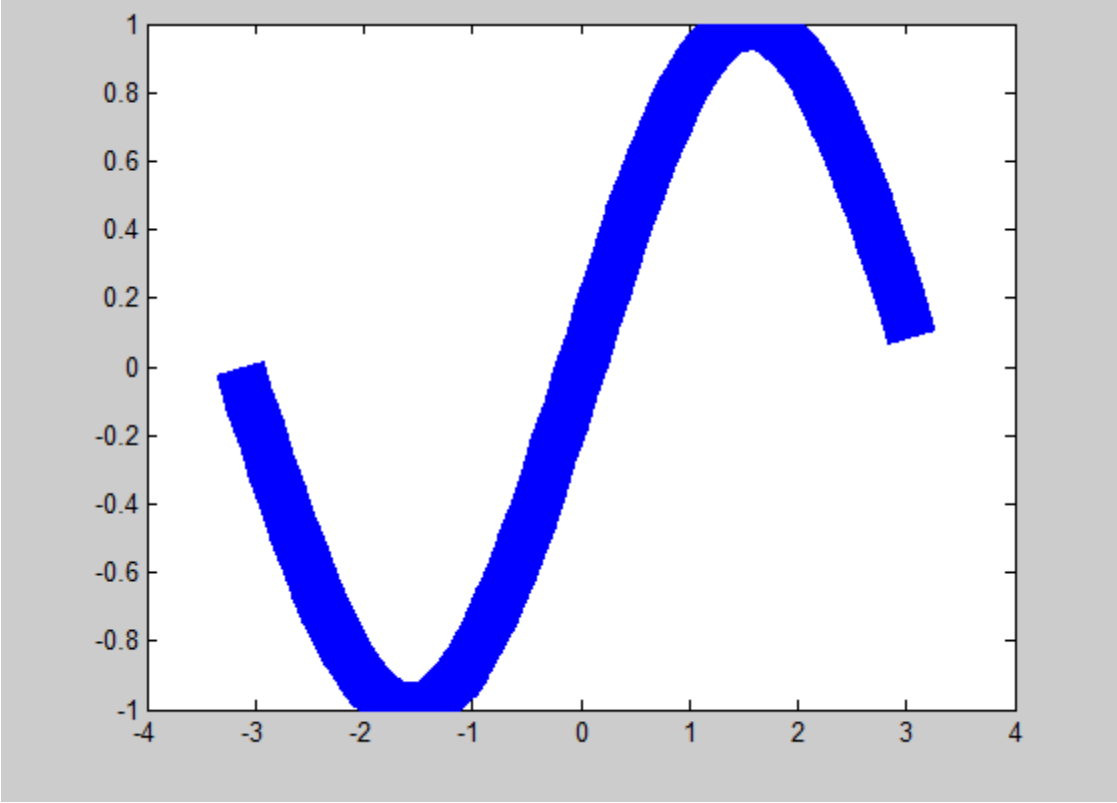
Line and Marker

```
>> x=-pi:.1:pi;

>> y=sin(x);

>> plot(x,y,'--','linewidth',2)
```




```
>> plot(x,y,'.','linewidth',2,'MarkerEdgeColor','r')
```




1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-3
-2
-1
0
1
2
3
4

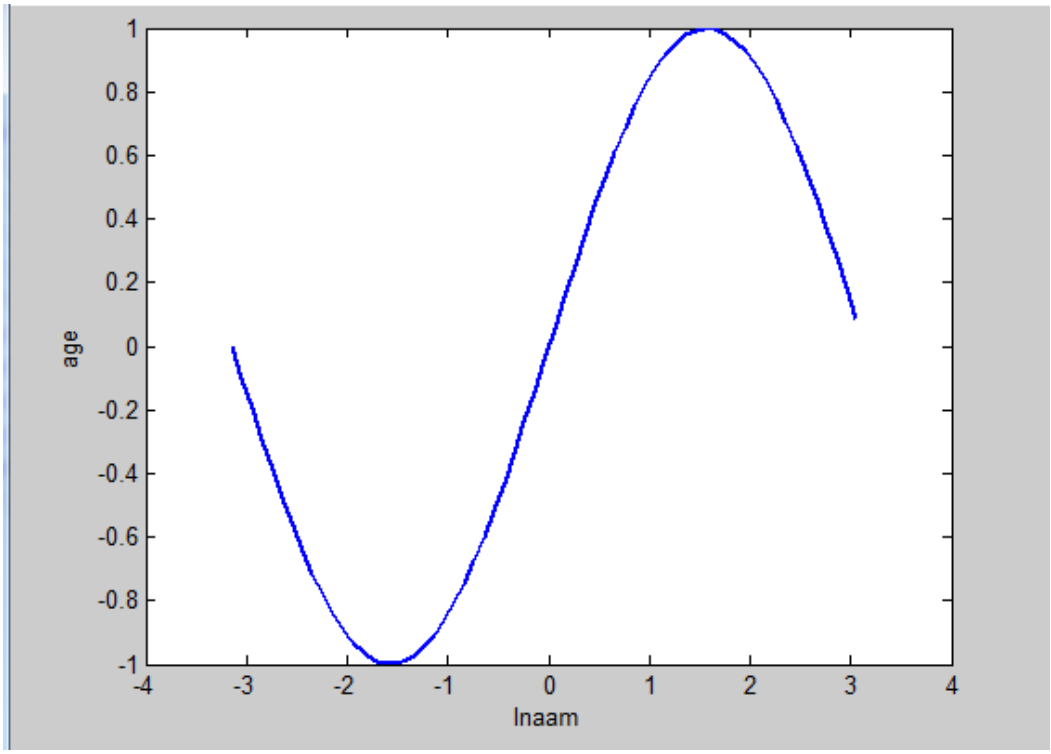
LAB work #10

plot(x,y,'linewidth',20,'MarkerFaceColor','r')

>> plot(x,y,'-','linewidth',2)

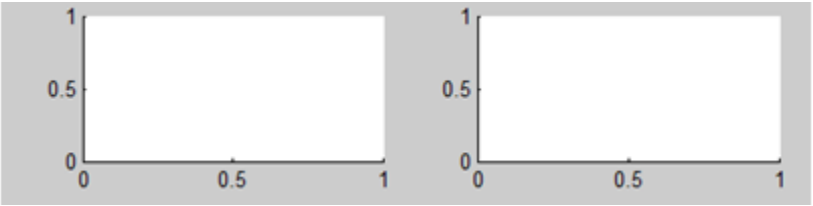
>> ylabel('age')

>> xlabel('Inaam')



>> subplot(3,2,1)

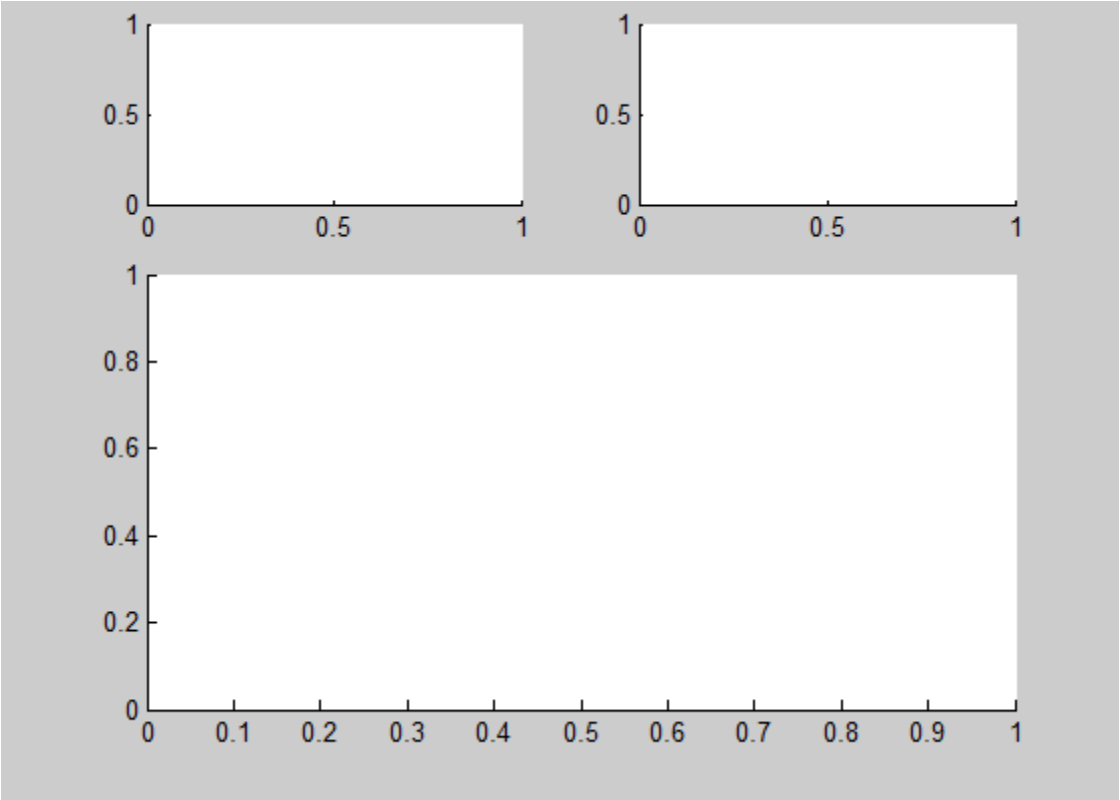
>> subplot(3,2,2)



>> subplot(3,2,1)

>> subplot(3,2,2)

>> subplot(3,2,4:6)



LAB work #11

Create a vector

"A" of the even whole numbers between 31 and 75.

```
>> A=32:2:74;
>> disp(A)
  Columns 1 through 15

    32    34    36    38    40    42    44    46    48    50    52    54    56    58    60

  Columns 16 through 22

    62    64    66    68    70    72    74
```

"B" of the odd whole numbers between 75 and 131.

```
B=77:2:129

>> B=77:2:129

disp(B)

B =

  Columns 1 through 15

    77    79    81    83    85    87    89    91    93    95    97    99   101   103   105

  Columns 16 through 27

   107   109   111   113   115   117   119   121   123   125   127   129

  Columns 1 through 15

    77    79    81    83    85    87    89    91    93    95    97    99   101   103   105

  Columns 16 through 27

   107   109   111   113   115   117   119   121   123   125   127   129
```

"C" of the whole numbers between 1 and 100.

```
B=2:1:99
disp(B)
>> B=2:1:99
disp(B)

B =

  Columns 1 through 14

     2     3     4     5     6     7     8     9    10    11    12    13    14    15

  Columns 15 through 28
```

16 17 18 19 20 21 22 23 24 25 26 27 28 29

Columns 29 through 42

30 31 32 33 34 35 36 37 38 39 40 41 42 43

Columns 43 through 56

44 45 46 47 48 49 50 51 52 53 54 55 56 57

Columns 57 through 70

58 59 60 61 62 63 64 65 66 67 68 69 70 71

Columns 71 through 84

72 73 74 75 76 77 78 79 80 81 82 83 84 85

Columns 85 through 98

86 87 88 89 90 91 92 93 94 95 96 97 98 99

Columns 1 through 14

2 3 4 5 6 7 8 9 10 11 12 13 14 15

Columns 15 through 28

16 17 18 19 20 21 22 23 24 25 26 27 28 29

Columns 29 through 42

30 31 32 33 34 35 36 37 38 39 40 41 42 43

Columns 43 through 56

44 45 46 47 48 49 50 51 52 53 54 55 56 57

Columns 57 through 70

58 59 60 61 62 63 64 65 66 67 68 69 70 71

Columns 71 through 84

72 73 74 75 76 77 78 79 80 81 82 83 84 85

Columns 85 through 98

86 87 88 89 90 91 92 93 94 95 96 97 98 99

LAB work #12

Let x = [2 5 1 6]

Add 16 to each element

```
>> x=[2 5 1 6];
y=x+16;
disp(y);
   18    21    17    22
```

Add 3 to just the odd-index elements

```
>> x=[2 5 1 6];

y(1)=x(1)+3;

y(2)=x(2);

y(3)=x(3)+3;

y(4)=x(4);

disp(y);

    5     5     4     6
```

Compute the square root of each element

```
>> x=[2 5 1 6];

y=sqrt(x);

disp(y);

   1.4142    2.2361    1.0000    2.4495
```

Compute the square of each element

```
>> x=[2 5 1 6];

y=(x).^2;

disp(y);

    4    25     1    36
```

Let x = [3 2 6 8]' and y = [4 1 3 5]' (NB. x and y should be column vectors).

Add the sum of the elements in x to y

```
>> x = [3; 2; 6; 8];
y = [4; 1; 3; 5];
z=sum(x);
d=y+z;
disp(d);
   23
   20
   22
   24
```

Raise each element of x to the power specified by the corresponding element in y.

```
>> x = [3; 2; 6; 8];

y = [4; 1; 3; 5];

for n=1:4

z(n)=(x(n))^y(n);

end

disp(z);

        81          2         216       32768
```

Divide each element of y by the corresponding element in x

```
>> x = [3; 2; 6; 8];

y = [4; 1; 3; 5];

for n=1:4

z(n)=y(n)./x(n);

end

disp(z);

   1.3333    0.5000    0.5000    0.6250
```

Multiply each element in x by the corresponding element in y, calling the result "z".

```
>> x = [3; 2; 6; 8];

y = [4; 1; 3; 5];

for n=1:4

z(n)=x(n).*y(n);

end

disp(z);

   12     2    18    40
```

Add up the elements in z and assign the result to a variable called "w".

```
for n=1:4

z(n)=x(n).*y(n);

end

w=sum(z);
```

```
disp(w);

   72
```

Compute x'*y - w and interpret the result.

```
>> x = [3; 2; 6; 8];

y = [4; 1; 3; 5];

for n=1:4

z(n)=x(n).*y(n);

end

w=sum(z);

for m=1:4

s(m)=z(m)-w;

end

disp (s);

  -60   -70   -54   -32
```

LAB work #13

Evaluate the following MATLAB expressions by hand and use MATLAB to check the answers.

2 / 2 * 3

```
The answer will be
ans =

    3
```

Because 2 cancel 2 and then we get 3.

6 - 2 / 5 + 7 ^ 2 – 1

```
The answer will be
ans =

  53.6000
```

10 / 2 \ 5 - 3 + 2 * 4

```
The answer will be
ans =

    6
```

3 ^ 2 / 4

```
The answer will be
ans =

   2.2500
```

3 ^ 2 ^ 2

The answer will be
ans =

81

2 + round (6 / 9 + 3 * 2) / 2 – 3

The result of term (6 / 9 + 3 * 2) is 6.6667. We use a command "**round**" with it.

2 + floor (6 / 9 + 3 * 2) / 2 – 3

"**floor**" command is used for rounding towards negative infinity. If we have a number 1.9, and we use floor command with it, the result will be 1. Similarly for 2.1, the result will be 2.
The result of term (6 / 9 + 3 * 2) is 6.6667. As we are use "floor" command,, the result will be 6.
When we solve the whole term we get
ans =

2

2 + ceil (6 / 9 + 3 * 2) / 2 – 3

"**ceil**" command is used for rounding towards positive infinity. If we have a number 1.9, and we use ceil command with it, the result will be 2. Similarly for 2.1, the result will be 3.
The result of term (6 / 9 + 3 * 2) is 6.6667. As we are use "ceil" command,, the result will be 7.
When we solve the whole term we get
ans =

2.5000

LAB work #14

Create a vector x with the elements...

a. **2, 4, 6, 8…**
b. **10, 8, 6, 4, 2, 0, -2, -4**
c. **1, 1/2, 1/3, 1/4, 1/5...**
d. **0, 1/2, 2/3, 3/4, 4/5...**

2, 4, 6, 8…

```
x=2:2:10000;
disp(x);
```

10, 8, 6, 4, 2, 0, -2, -4

```
>> x=10:-2:-4;
disp(x);
   10    8    6    4    2    0   -2   -4
```

1, 1/2, 1/3, 1/4, 1/5...

```
>> for n=1:100
   d=1/n;
   e=rats(d);
   disp(e);
end
      1
     1/2
     1/3
     1/4
     1/5
     1/6
     1/7
     1/8
     .
     .
     .
     1/95
     1/96
     1/97
     1/98
     1/99
     1/100
```

0, 1/2, 2/3, 3/4, 4/5...

```
>> for n=1:100
   d=n/(n+1);
   e=rats(d);
   disp(e);
```

```
end
    1/2
    2/3
    3/4
    4/5
    .
    .
    .
    94/95
    95/96
    96/97
    97/98
    98/99
    99/100
   100/101
```

LAB work #15

Create a vector x with the elements,

$$Xn = (-1)^{n+1}/(2n-1)$$

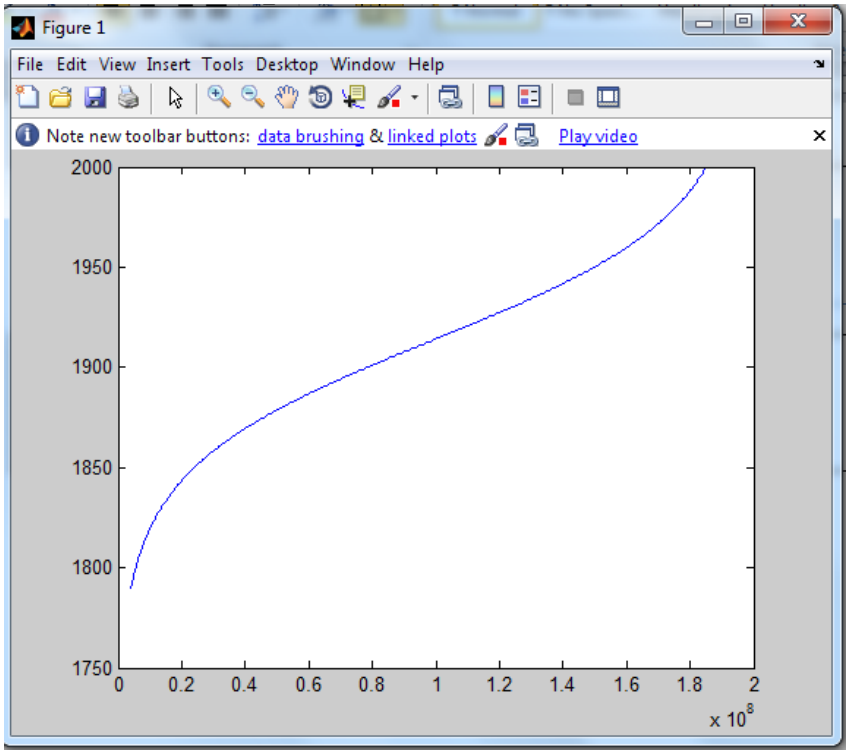
Add up the elements of the version of this vector that has 100 elements.

```
>> for n=1:100
X(n)= ((-1)^(n+1))/((2*n)-1);
%disp(X);    %Uncomment this in order to display result at every value of n
end
Y=sum(X);
disp(Y);
   0.7829
```

```
>> x=input('Perpendicular of Triangle =');
y=input('Base of Triangle =');
z=sqrt(x^2 +  y^2);
fprintf('Hypotenues of triangle =');
disp(z);
Perpendicular of Triangle =3
Base of Triangle =4
Hypotenues of triangle =     5
>> a=input('Length of first side =');
b=input('Base of second side =');
t=input('Angle Between the Given Sides =');
fprintf('Length of third side');
c = sqrt(a^2 + b^2 - 2*a*b*cos(t));
disp(c);
Length of first side =3
```

```
Base of second side =7

Angle Between the Given Sides =3

Length of third side    9.9790
```

```
>> t=1790:0.5:2000;

p=197273000./(1 + exp(-0.0313*(t - 1913.25)));

plot(p,t);
```




```
>> t=2020;

p=197273000./(1 + exp(-0.0313*(t - 1913.25)));

disp(p);

  1.9053e+008

  1.9053e+008

ans =

   190530000
```

LAB work #16

Write the following matrix in Matlab and perform the given function.

A= [12 15 16; 23 26 27; 31 40 41]
B= [22 35 56; 63 66 78; 24 30 46]

```
>> A+B

ans =

    5     7     9

   11    13    15

    8    10    12
```

```
B*A

ans =

   66    81    96

  102   126   150

>> A.*B

ans =

    4    10    18

   28    40    54

    7    16    27

>> B-A

ans =


    3     3     3

    3     3     3

   -6    -6    -6
```