Contents

```
%These are the notes that we cover in the tutorial session. They ar
%meant to get you started with matrix operations and plotting in MA
%You can learn a lot more from MATLAB demos and examples.
close all
clear all
```

Defining matrices in matlab

```
%To define a matrix:
A=[1 0 1;-1 2 0]
%or
A=[1 0 1
    -1 2 0]
```

```
A =

     1     0     1
    -1     2     0


A =

     1     0     1
    -1     2     0
```

```
whos A
```

```
  Name      Size            Bytes  Class     Attributes

  A         2x3                48  double
```

```
%You can concatenate matrices in a similar fashion. For instace,
A=[A;0 0 1]
```

```
%adds one row to A.
```

```
A =

     1     0     1
    -1     2     0
     0     0     1
```

```
[m,n]=size(A) %returns the size of A. See also length and ndims.
```

```
m =

     3


n =

     3
```

```
e=ones(3,1) %a 5x1 vector of all ones. see also zeros.
```

```
e =

     1
     1
     1
```

```
I=eye(20) %the identity matrix
```

```
I =

     1     0     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
```

```
H=sparse(I) %creates a sparse matrix from I, i.e., stores only the
%elements of I and their corresponding indices.
whos %Note the difference between the memory usage of H and I.
```

```
H =

   (1,1)        1
   (2,2)        1
   (3,3)        1
   (4,4)        1
   (5,5)        1
   (6,6)        1
   (7,7)        1
   (8,8)        1
   (9,9)        1
  (10,10)       1
  (11,11)       1
  (12,12)       1
  (13,13)       1
  (14,14)       1
  (15,15)       1
  (16,16)       1
  (17,17)       1
  (18,18)       1
  (19,19)       1
  (20,20)       1

  Name        Size            Bytes  Class     Attributes

  A           3x3                72  double              
  H          20x20              488  double    sparse    
  I          20x20             3200  double              
  e           3x1                24  double              
  m           1x1                   8  double
  n           1x1                   8  double
```

```
I2=eye(3);
blkdiag(A,I2) %creates a block-diagonal matrix with A and eye(3) or
```

```
ans =

     1     0     1     0     0     0
    -1     2     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

```
E=spdiags([e,-2*e,e],-1:1,length(e),length(e))'  %creates a sparse
```

```
E =

   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
```

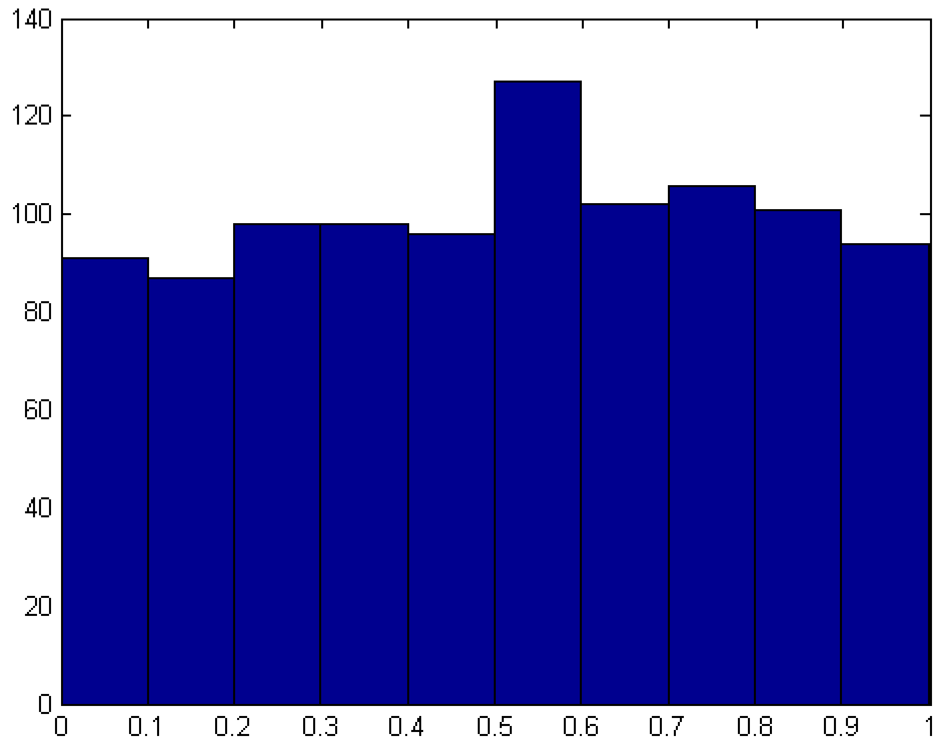
```
full(E) %prints E in its non-sparse form.
```

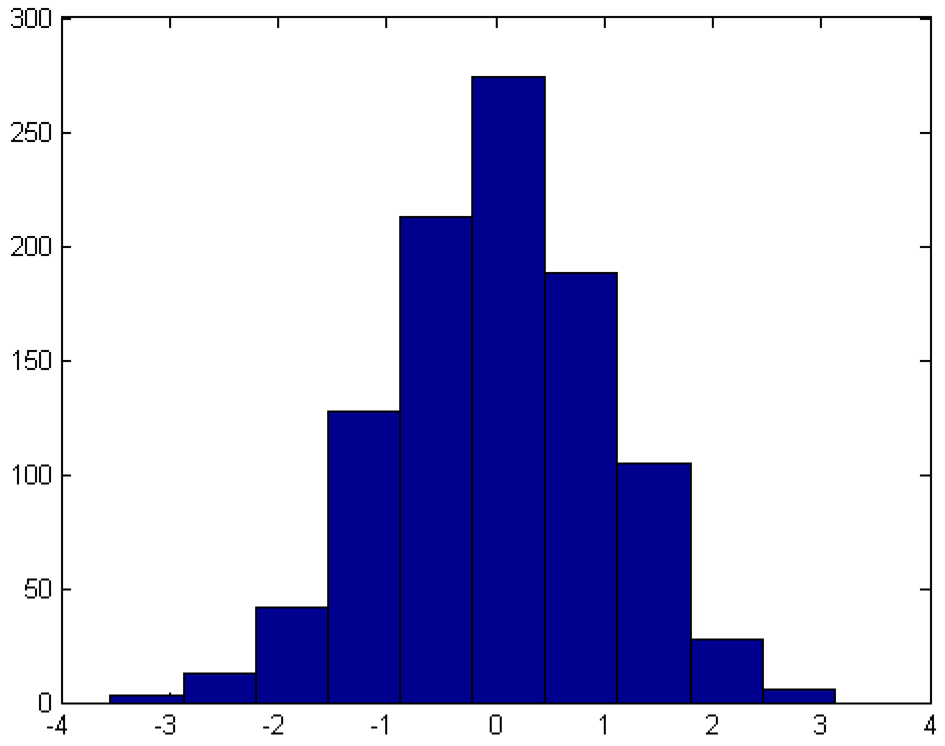
```
ans =

    -2     1     0
     1    -2     1
     0     1    -2
```

Generating random vectors/matrices

```
u=rand(1,1000); %generates a random row vector from the uniform dis
```

```
%Use semicolons if you don't want to see the output in
%the command window.
hist(u); %plots the histogram of u.
v=randn(1,1000); %random vector from the normal distribution.
figure
hist(v);%plots the histogram of v.
```





Some useful operations on matrices

```
%Arithmatic operations are easy in MATLAB. Use *,^+,- for regular m
% operation. Use .* and .^ for elementwise multiplication and power
% resepectively. Use A' to find the transpose of A
M=rand(20);
B=M.^2; %elementwise squared of M
C=(M+M')/2; %the symmetric half of M.
D=triu(M)% returns the upper triangle of M. Use help triu to learn
%about triu and its related commands.
% Exercise: Construct a random tridigonal matrix from M using only
%tril and triu.
```

```
D =

  Columns 1 through 13

    0.8911    0.3849    0.6039    0.4916    0.7215    0.5712    0.4
         0    0.6782    0.2885    0.7701    0.6473    0.4763    0.7
         0         0    0.8811    0.2520    0.8342    0.8529    0.6
         0         0         0    0.6437    0.5069    0.3534    0.5
         0         0         0         0    0.2027    0.1660    0.3
         0         0         0         0         0    0.5295    0.5
         0         0         0         0         0         0    0.2
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0

  Columns 14 through 20

    0.4101    0.6199    0.0426    0.6259    0.1491    0.7147    0.3
    0.4093    0.4487    0.2920    0.6605    0.7008    0.2101    0.1
    0.6294    0.7251    0.1111    0.5750    0.1013    0.9524    0.9
    0.5383    0.0415    0.1072    0.7417    0.1957    0.5502    0.9
    0.5619    0.4451    0.4044    0.6733    0.3448    0.4142    0.1
    0.9571    0.7388    0.7333    0.3291    0.7022    0.5135    0.8
    0.7165    0.5263    0.3859    0.0838    0.2908    0.6675    0.0
    0.3175    0.3287    0.9675    0.9264    0.5251    0.0626    0.4
    0.1609    0.1064    0.8416    0.8418    0.6196    0.0228    0.5
    0.8128    0.1050    0.9664    0.4327    0.3917    0.2831    0.2
    0.3433    0.5447    0.8456    0.6380    0.8925    0.7514    0.8
    0.1986    0.7442    0.3588    0.2061    0.9489    0.6392    0.4
    0.0570    0.1380    0.4757    0.5914    0.0097    0.0645    0.4
    0.6561    0.6282    0.4613    0.5290    0.1828    0.2921    0.9
         0    0.0760    0.0264    0.5893    0.4150    0.1043    0.4
         0         0    0.0316    0.5036    0.6890    0.5842    0.6
         0         0         0    0.2472    0.1537    0.4100    0.9
         0         0         0         0    0.5136    0.7776    0.1
         0         0         0         0         0    0.1523    0.1
         0         0         0         0         0         0    0.4
```

```
reshape(M,2,[]) %reshapes M into a matrix with two rows.
```

```
ans =

  Columns 1 through 13
```

```
    0.8911    0.7729    0.9662    0.5891    0.4876    0.8683    0.7
    0.0688    0.6604    0.9854    0.9275    0.4662    0.4876    0.1

  Columns 14 through 26

    0.8054    0.6732    0.9858    0.2175    0.0276    0.8737    0.7
    0.9636    0.1134    0.8844    0.0965    0.9055    0.3537    0.7

  Columns 27 through 39

    0.6484    0.4835    0.3965    0.3111    0.4916    0.2520    0.5
    0.1686    0.1144    0.0574    0.8619    0.7701    0.6437    0.4

  Columns 40 through 52

    0.1308    0.7215    0.8342    0.2027    0.3783    0.4258    0.6
    0.0362    0.6473    0.5069    0.7319    0.8901    0.9131    0.9

  Columns 53 through 65

    0.1660    0.3438    0.0678    0.1460    0.6162    0.5526    0.3
    0.5295    0.4340    0.0728    0.0119    0.6226    0.3348    0.7

  Columns 66 through 78

    0.7867    0.2606    0.0521    0.1307    0.4175    0.7829    0.3
    0.3943    0.3352    0.4803    0.7913    0.1160    0.4995    0.7

  Columns 79 through 91

    0.2080    0.2829    0.1468    0.5590    0.6044    0.0318    0.2
    0.1853    0.4637    0.6811    0.7213    0.4157    0.3086    0.0

  Columns 92 through 104

    0.3048    0.3260    0.6154    0.1684    0.5159    0.0443    0.9
    0.6145    0.4222    0.4513    0.5153    0.9246    0.1775    0.6

  Columns 105 through 117

    0.1008    0.8248    0.7047    0.9116    0.3333    0.5015    0.0
    0.9967    0.1171    0.4454    0.6145    0.8344    0.1299    0.4

  Columns 118 through 130

    0.6912    0.3923    0.5756    0.5310    0.5829    0.8367    0.4
    0.8472    0.9937    0.9210    0.3122    0.6520    0.6669    0.4
  Columns 131 through 143

    0.4101    0.6294    0.5619    0.7165    0.1609    0.3433    0.0
    0.4093    0.5383    0.9571    0.3175    0.8128    0.1986    0.6

  Columns 144 through 156

    0.5263    0.1064    0.5447    0.1380    0.0760    0.7293    0.1
    0.3287    0.1050    0.7442    0.6282    0.5632    0.4692    0.8

  Columns 157 through 169

    0.4757    0.0264    0.6344    0.4412    0.6259    0.5750    0.6
    0.4613    0.0316    0.2657    0.3937    0.6605    0.7417    0.3

  Columns 170 through 182

    0.9953    0.1491    0.1013    0.3448    0.2908    0.6196    0.8
    0.8768    0.7008    0.1957    0.7022    0.5251    0.3917    0.9

  Columns 183 through 195

    0.4142    0.6675    0.0228    0.7514    0.0645    0.1043    0.4
    0.5135    0.0626    0.2831    0.6392    0.2921    0.5842    0.7

  Columns 196 through 200

    0.8204    0.4331    0.4516    0.9567    0.1223
    0.4055    0.9701    0.6882    0.1985    0.4631
```

```
repmat(I2,3,1) %creates 3x1 blocks of I2
% Exercise: Let A=rand(10,2) and B=rand(2,2); Use reshape to calcul
% A*B as a vector matrix product. Reshape the result back into the
% form to see if you get the correct result.
```

```
ans =

     1     0     0
     0     1     0
     0     0     1
     1     0     0
     0     1     0
     0     0     1
     1     0     0
     0     1     0
     0     0     1
```

```
K=kron(E,I2); %the Kronecker product of A and I2
full(K)
```

```
ans =

    -2     0     0     1     0     0     0     0     0
     0    -2     0     0     1     0     0     0     0
     0     0    -2     0     0     1     0     0     0
     1     0     0    -2     0     0     1     0     0
     0     1     0     0    -2     0     0     1     0
     0     0     1     0     0    -2     0     0     1
     0     0     0     1     0     0    -2     0     0
     0     0     0     0     1     0     0    -2     0
     0     0     0     0     0     1     0     0    -2
```

```
[v,d]=eig(C); %finds the eigenvectors and eigenvalues of C.
e=diag(d) %extracts the diagonal of d (which contains the eigenvalu
```

```
e =

  -1.6833
  -1.4032
  -1.2183
  -1.0844
  -0.7114
  -0.5703
  -0.4741
  -0.3268
  -0.1237
  -0.0683
   0.1644
   0.2326
   0.3085
   0.4538
   0.7682
   0.8483
   1.3448
   1.4767
   1.7762
   9.9164
```

```
sum(e) %the sum of eigenvalues.
trace(C) %the sum of diagonal elements of C
[U,S,V]=svd(C); %returns the svd decomposition of C.
```

```
ans =

    9.6260


ans =

    9.6260
```

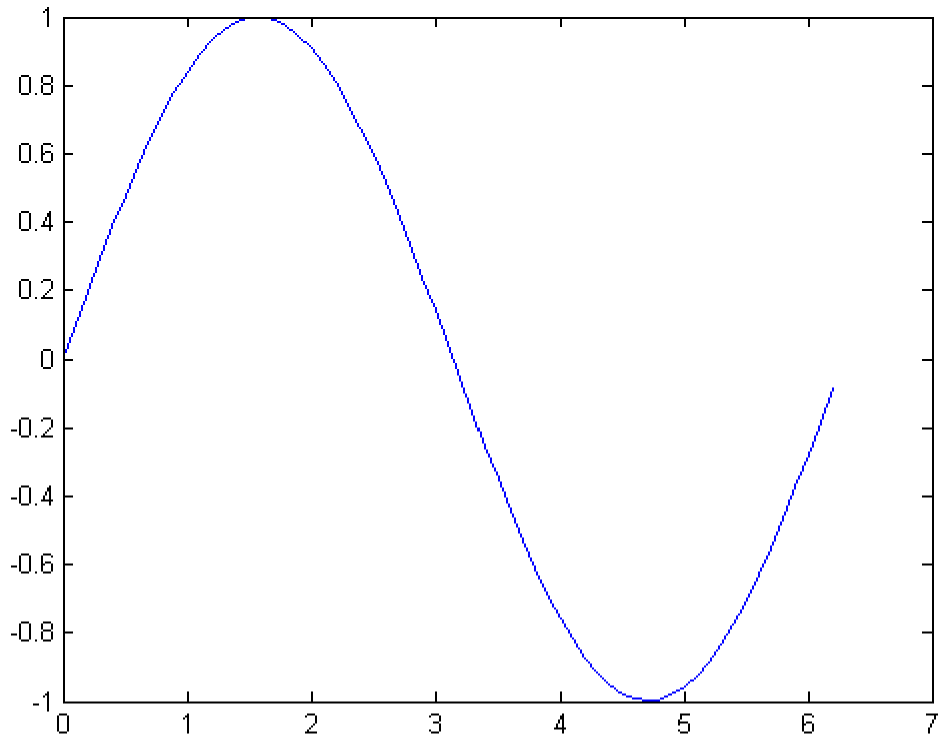
```
%Exercise: What is the expected sum of the eigenvalues of C? (hint:
%distribution of A and use the fact that the trace is the sum of
%eigenvalues).
```

Plotting in matlab

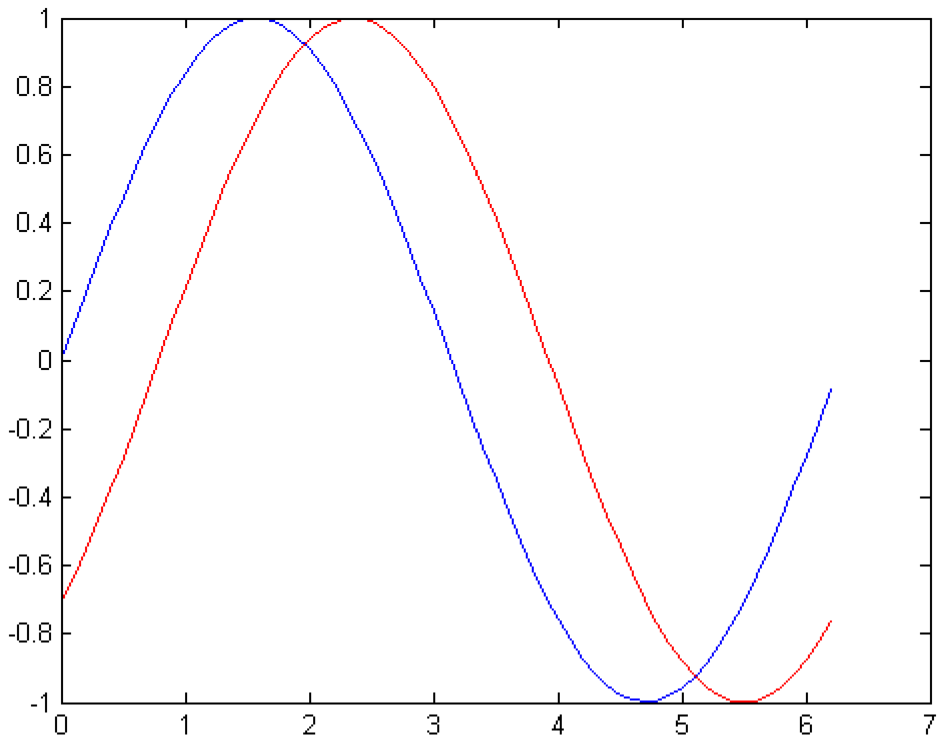
```
x=0:0.1:2*pi %a vector with elements between 0 and 2*pi and spacing
```

```
x =

  Columns 1 through 13

         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6

  Columns 14 through 26

    1.3000    1.4000    1.5000    1.6000    1.7000    1.8000    1.9

  Columns 27 through 39

    2.6000    2.7000    2.8000    2.9000    3.0000    3.1000    3.2

  Columns 40 through 52

    3.9000    4.0000    4.1000    4.2000    4.3000    4.4000    4.5

  Columns 53 through 63

    5.2000    5.3000    5.4000    5.5000    5.6000    5.7000    5.8
```

```
y=sin(x);
plot(x,y) %plots y vs x. Use "help plot" for more details.
```



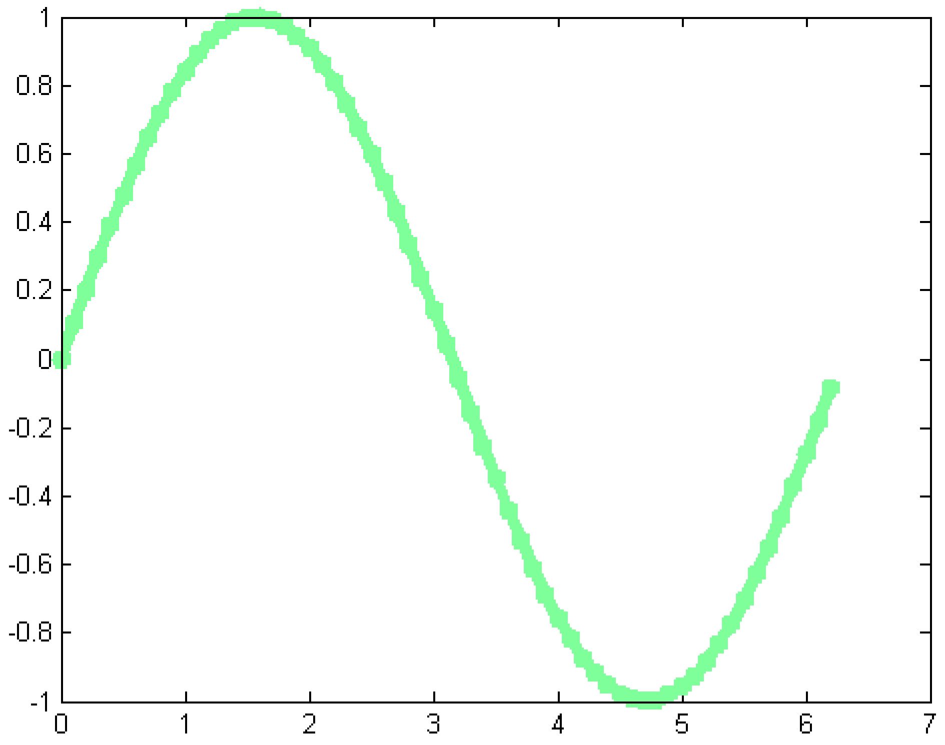
```
hold on %holds the current plot and axis
h=sin(x-pi/4);
plot(x,h,'r')
%One can define a plot object and use set() to modify its propertie
```



```
figure
plotobj=plot(x,y)
get(plotobj) %A list of all the fields in plotobj
set(plotobj,'Marker','*','color',[0.5 1 0.6],'linewidth',5)
```

```
plotobj =

  524.0426

           DisplayName: ''
            Annotation: [1x1 hg.Annotation]
                 Color: [0 0 1]
             LineStyle: '-'
             LineWidth: 0.5000
                Marker: 'none'
            MarkerSize: 6
       MarkerEdgeColor: 'auto'
       MarkerFaceColor: 'none'
                 XData: [1x63 double]
                 YData: [1x63 double]
```

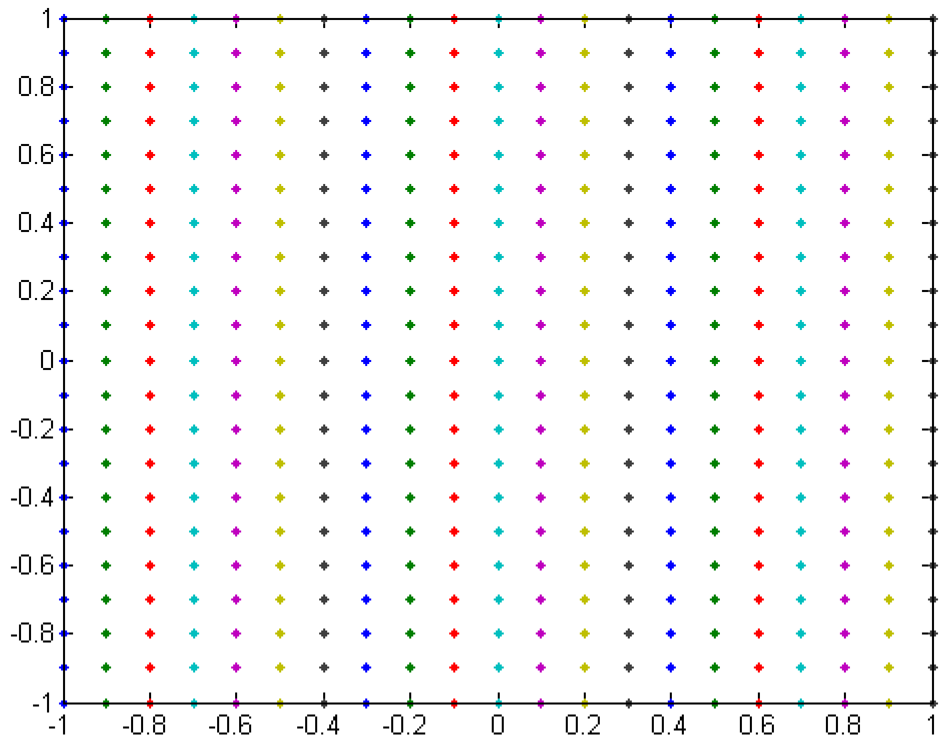
```
               ZData: [1x0 double]
        BeingDeleted: 'off'
       ButtonDownFcn: []
            Children: [0x1 double]
            Clipping: 'on'
           CreateFcn: []
           DeleteFcn: []
          BusyAction: 'queue'
    HandleVisibility: 'on'
             HitTest: 'on'
       Interruptible: 'on'
            Selected: 'off'
  SelectionHighlight: 'on'
                 Tag: ''
                Type: 'line'
       UIContextMenu: []
            UserData: []
             Visible: 'on'
              Parent: 523.0421
           XDataMode: 'manual'
         XDataSource: ''
         YDataSource: ''
         ZDataSource: ''
```



```
%Plotting a surface:
x=-1:0.1:1;
y=-1:0.1:1;
[X,Y]=meshgrid(x,y);
```

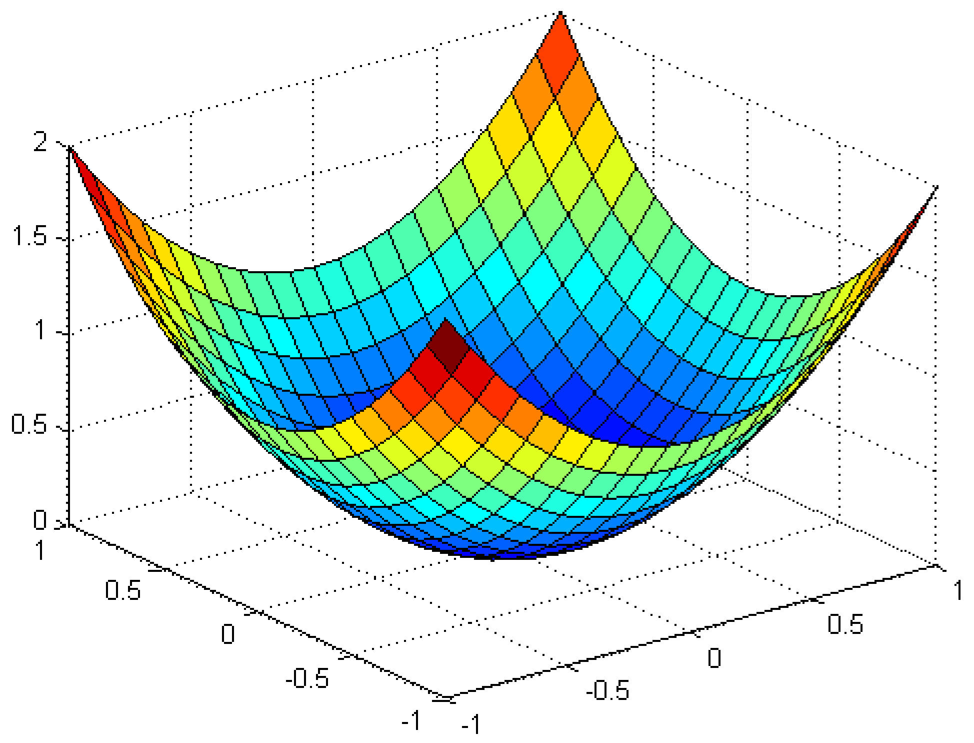
X and Y define a two dimensional grid.

```
plot(X,Y,'.')
```

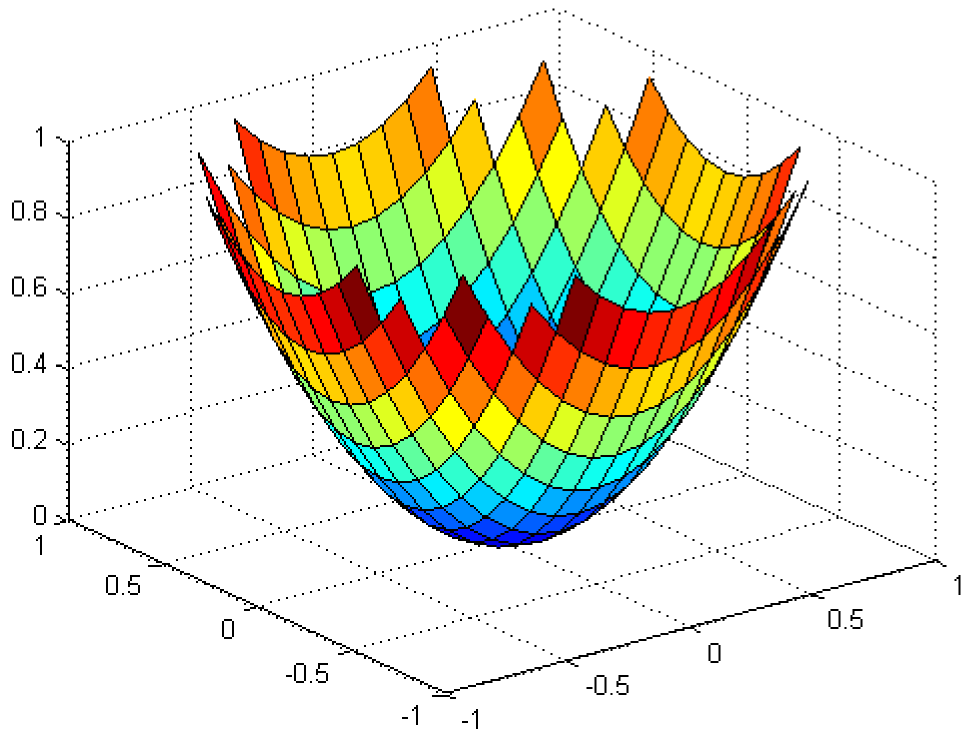


```
whos X Y x y
Z=X.^2+Y.^2;
surf(X,Y,Z)
```

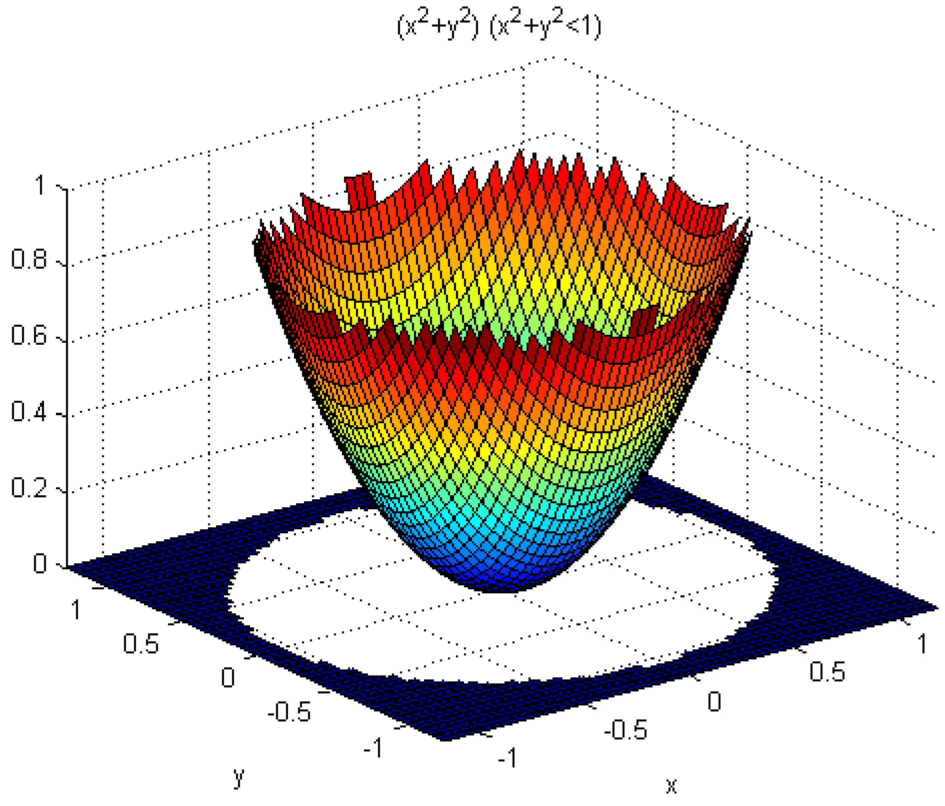
```
  Name         Size                Bytes  Class     Attributes

  X           21x21                 3528  double              
  Y           21x21                 3528  double              
  x            1x21                  168  double              
  y            1x21                  168  double              
```



```
Z(X.^2+Y.^2>=1)=nan;  %restricts the domain of Z to be inside the ur
surf(X,Y,Z)
```



```
%Alternatively, one can use the ezsurf command for plotting purpose
ezsurf('(x^2+y^2).*(x^2+y^2<1)' )
% this eliminates the need for griding, but sets
%the nonvalid entries to zero instead of nan.
```



$(x^2+y^2)\ (x^2+y^2<1)$
1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
y
-1
-0.5
0
0.5
1
x