Fundamentals of MATLAB Usage

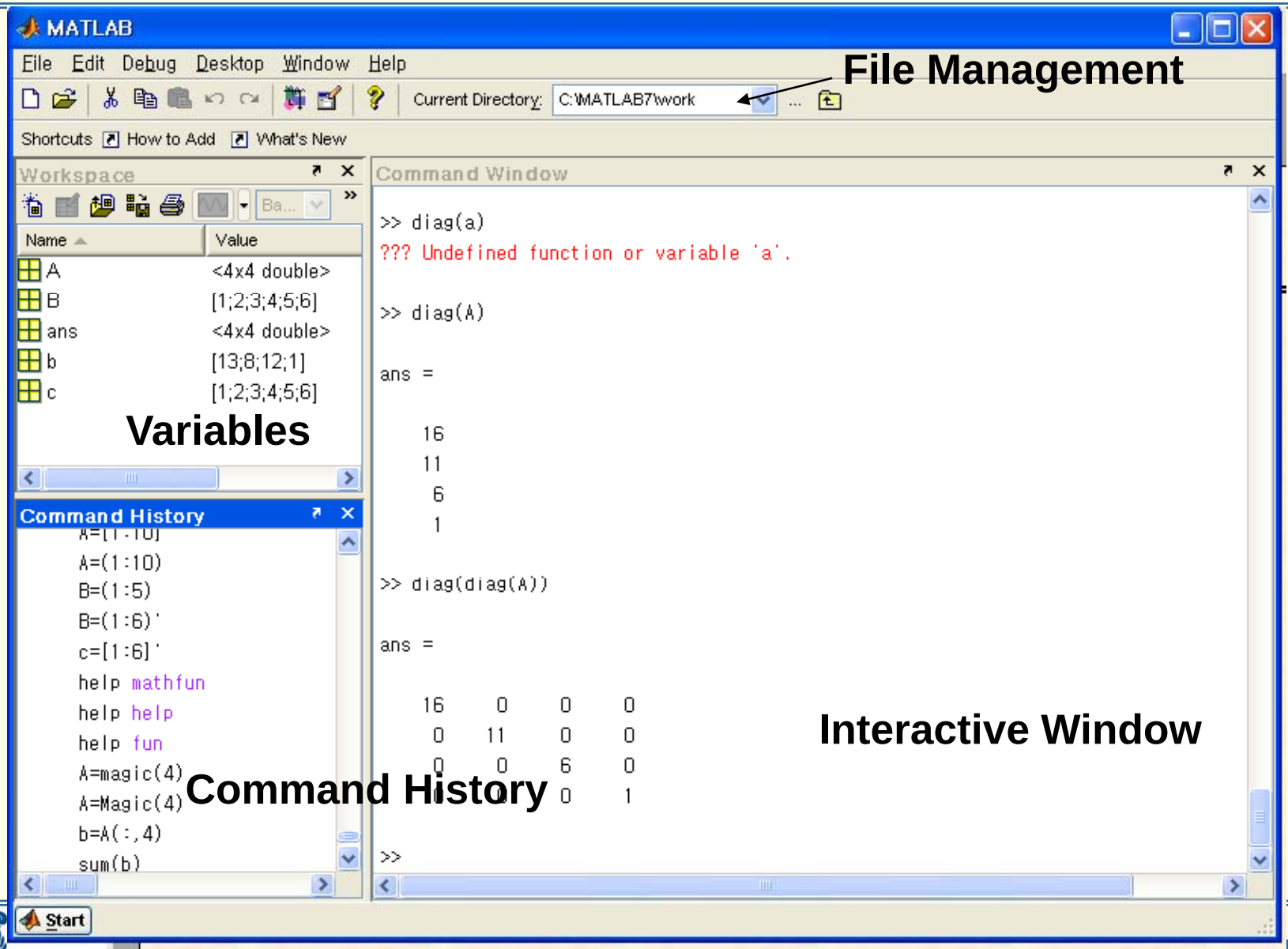
2008. 9

담당교수: 주 한 규
joohan@snu.ac.kr, x9241, Rm 32-205
원자핵공학과

MATLAB Features

- **MATLAB: Matrix Laboratory**
  - Process everything based on Matrix (array of numbers)
- **Math and Computation**
  - Linear System Solve, Eigenvalue, Matrix Manipulation, etc.
  - Integration, Root finding, Optimization (Minimization), etc.
- **Graphics and Visualization**
  - 2D Graphics: XY, Contour; 3D Graphics: Surface, Mesh
  - Lighting and Movie
- **Programming**
  - Simple FORTRAN or C typing programming with No Need for Explicit Variable Declarations (No consideration on memory usage)
- **Data Analysis**
  - Statistical Analysis
- **and a Lot More…**

MATLAB Screen



Entering Data and Basic Manipulation

To enter Dürer's matrix, simply type in the Command Window

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

Semicolon or 'next line' for row separation

```
sum(A)
```

MATLAB replies with

```
ans =
    34    34    34    34
```

```
diag(A)
```

produces

```
ans =
    16
    10
     7
     1
```

diag produces column vector

```
A'
```

produces

Transpose

```
ans =
    16     5     9     4
     3    10     6    15
     2    11     7    14
    13     8    12     1
```

Element Operation

```
A(1,4) + A(2,4) + A(3,4) + A(4,4)
```

This produces

```
ans =
    34
```

Colon Operator

```
1:10
```

is a row vector containing the integers from 1 to 10,

```
1     2     3     4     5     6     7     8     9     10
```

To obtain nonunit spacing, specify an increment. For example,

```
100:-7:50
```

Increment

is

```
100    93    86    79    72    65    58    51
```

```
0:pi/4:pi
```

is

```
0    0.7854    1.5708    2.3562    3.1416
```

Subscript expressions involving colons refer to portions of a matrix.

```
A(1:k,j)
```

is the first `k` elements of the `j`th column of `A`. So

```
sum(A(1:4,4))
```

":" alone in matrix element specification alone entire column or row

A(:,4) – 4th column of A , A(2,:) – 2nd row vector

Operators

Operators

Expressions use familiar arithmetic operators and precedence rules.

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division (described in "Matrices and Linear Algebra" in the MATLAB documentation)
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

X=A\b solves Ax=b where A is a Matrix and b is a column vector

```
rho = (1+sqrt(5))/2
rho =
    1.6180
```

Element-wise Operation

.*	Element-by-element multiplication
./	Element-by-element division
.\	Element-by-element left division

Convenient Matrix Generation

Generating Matrices

MATLAB provides four functions that generate basic matrices.

`zeros`	All zeros
`ones`	All ones
`rand`	Uniformly distributed random elements
`randn`	Normally distributed random elements

Here are some examples.

```
Z = zeros(2,4)
Z =
     0     0     0     0
     0     0     0     0

F = 5*ones(3,3)
F =
     5     5     5
     5     5     5
     5     5     5

N = fix(10*rand(1,10))
N =
     9     2     6     4     8     7     4     0     8     4

R = randn(4,4)
```

eye(n) – for nxn identity matrix

diag(v) – for diagonal matrix who's elements are given in vector v of dimension n

Block Matrices

Concatenation

Concatenation is the process of joining small matrices to make bigger ones. In fact, you made your first matrix by concatenating its individual elements. The pair of square brackets, [ ], is the concatenation operator. For an example, start with the 4-by-4 magic square, A, and form

```
B = [A  A+32; A+48  A+16]
```

scalar 16 added to all elements

The result is an 8-by-8 matrix, obtained by joining the four submatrices.

```
B =
```

16	3	2	13	48	35	34	45
5	10	11	8	37	42	43	40
9	6	7	12	41	38	39	44
4	15	14	1	36	47	46	33
64	51	50	61	32	19	18	29
53	58	59	56	21	26	27	24
57	54	55	60	25	22	23	28
52	63	62	49	20	31	30	17

Basic Internal Functions

diag(A)	- vector consisting of diagonal elements of Matrix A
tril(A)	- lower triangular matrix of A
triu(A)	- upper triangular matrix of A
inv(A)	- inverse matrix of A
det(A)	- determinant of A
size(A)	- size of A, e.g mxn
rank(A)	- rank of A
rref(A)	- reduced row echelon form of A
length(v)	- dimension of vector v
[V,D]=eig(A)	- eigenvector (V) and eigenvalues (D) of A
[L,U]=lu(A)	- L, U Factors of A obtained after Gaussian Eli.
whos	- list all the current variables in memory
clear A B C	- remove A B C

Flow Control

Conditional

```
if rem(n,2) ~= 0
   M = odd_magic(n)
elseif rem(n,4) ~= 0
   M = single_even_magic(n)
else
   M = double_even_magic(n)
end
```

<= : less than or equal
~= : not equal
& : and
| : or
~ :not

Loop

```
for n = 3:32
   r(n) = rank(magic(n));
end
```

exit out of a loop →

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
   x = (a+b)/2;
   fx = x^3-2*x-5;
   if fx == 0
      break
   elseif sign(fx) == sign(fa)
      a = x; fa = fx;
   else
      b = x; fb = fx;
   end
end
```

Input/Output Syntax

```
A = magic(100);
```

Use Semicolon to suppress screen display of result!

```
s = 1 -1/2 + 1/3 -1/4 + 1/5 - 1/6 + 1/7 ...
      - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;
```

Continuation on next line

Save and Load

```
save magic.dat A B x y
load magic.dat
```

Formatted File I/O

```
fin=fopen('ri.inp','r');
yref=zeros(nbg,ntemp)
for ibg=1:nbg
  xbg(ibg)=fscanf(fin,'%g',1);
  yref(ibg,:)=fscanf(fin,'%g',ntemp);
end
```

Formart Specifieres
%10.3f : floating point with 3 digits after decimal
%10d : integer with a width of 10 colums
%15.3e: exponetial

```
fout=fopen('ri.out','w');
for ibg=1:nbg
  fprinft(fout,'%d %g\n',ibg,yref(ibg));
end
```

Output Display Patterns

```
x = [4/3 1.2345e-6]

format short

   1.3333    0.0000

format short e

   1.3333e+000   1.2345e-006
```

```
format long

   1.33333333333333    0.00000123450000

format long e

   1.333333333333333e+000    1.234500000000000e-006

format long g

   1.33333333333333                  1.2345e-006

format bank

   1.33           0.00

format rat

   4/3            1/810045

format hex

   3ff5555555555555    3eb4b6231abfd271
```

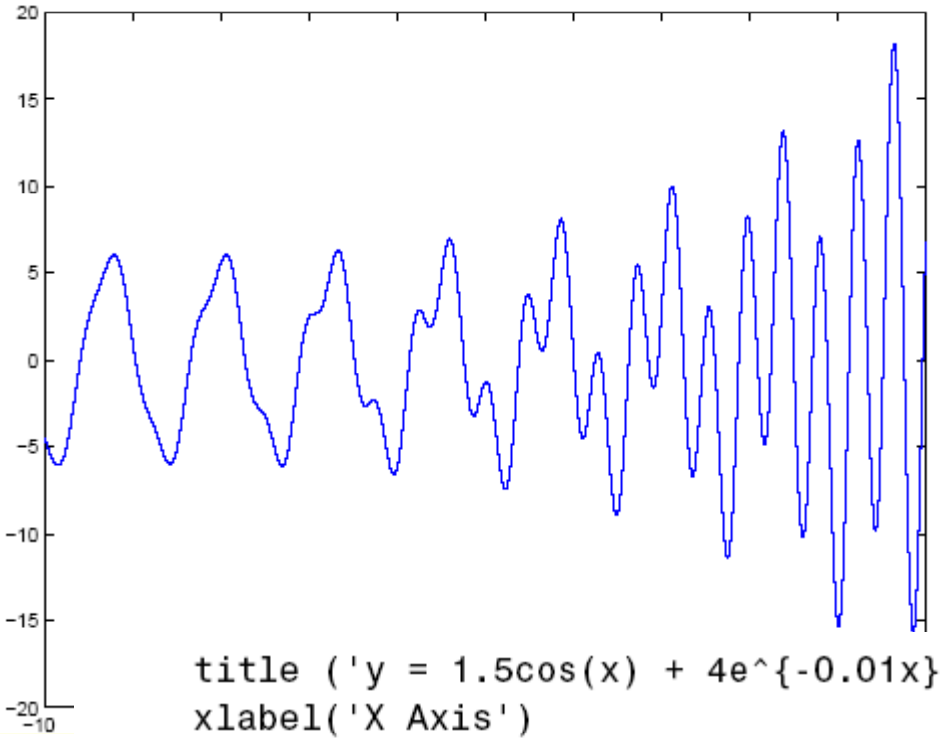
Generic Format

Rational Number →

2D Graph

```
x = -10:.005:40;
y = [1.5*cos(x)+4*exp(-.01*x).*cos(x)+exp(.07*x).*sin(3*x)];
plot(x,y)
```

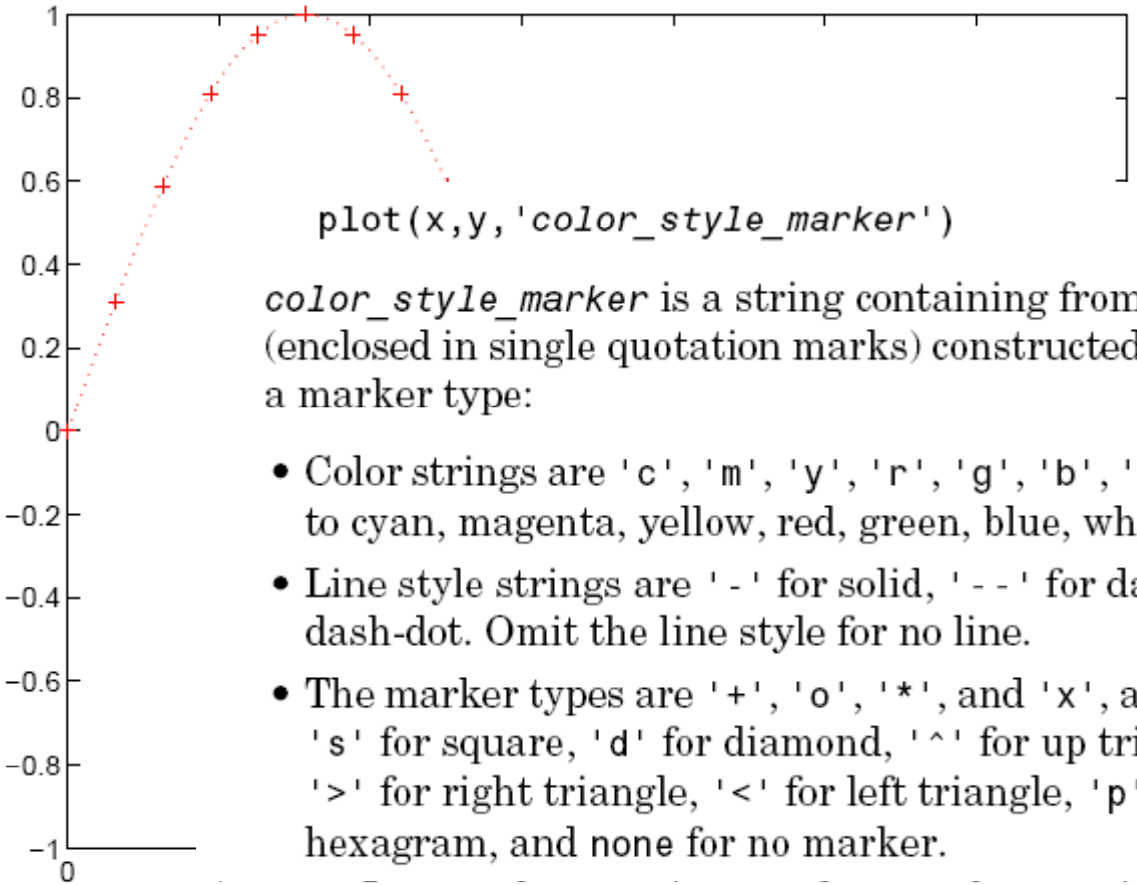
This picture shows the graph created by the code above.




```
title ('y = 1.5cos(x) + 4e^{-0.01x}cos(x) + e^{0.07x}sin(3x)')
xlabel('X Axis')
ylabel('Y Axis')
```

2D Graphs

```
x1 = 0:pi/100:2*pi;
x2 = 0:pi/10:2*pi;
plot(x1,sin(x1),'r:',x2,sin(x2),'r+')
```



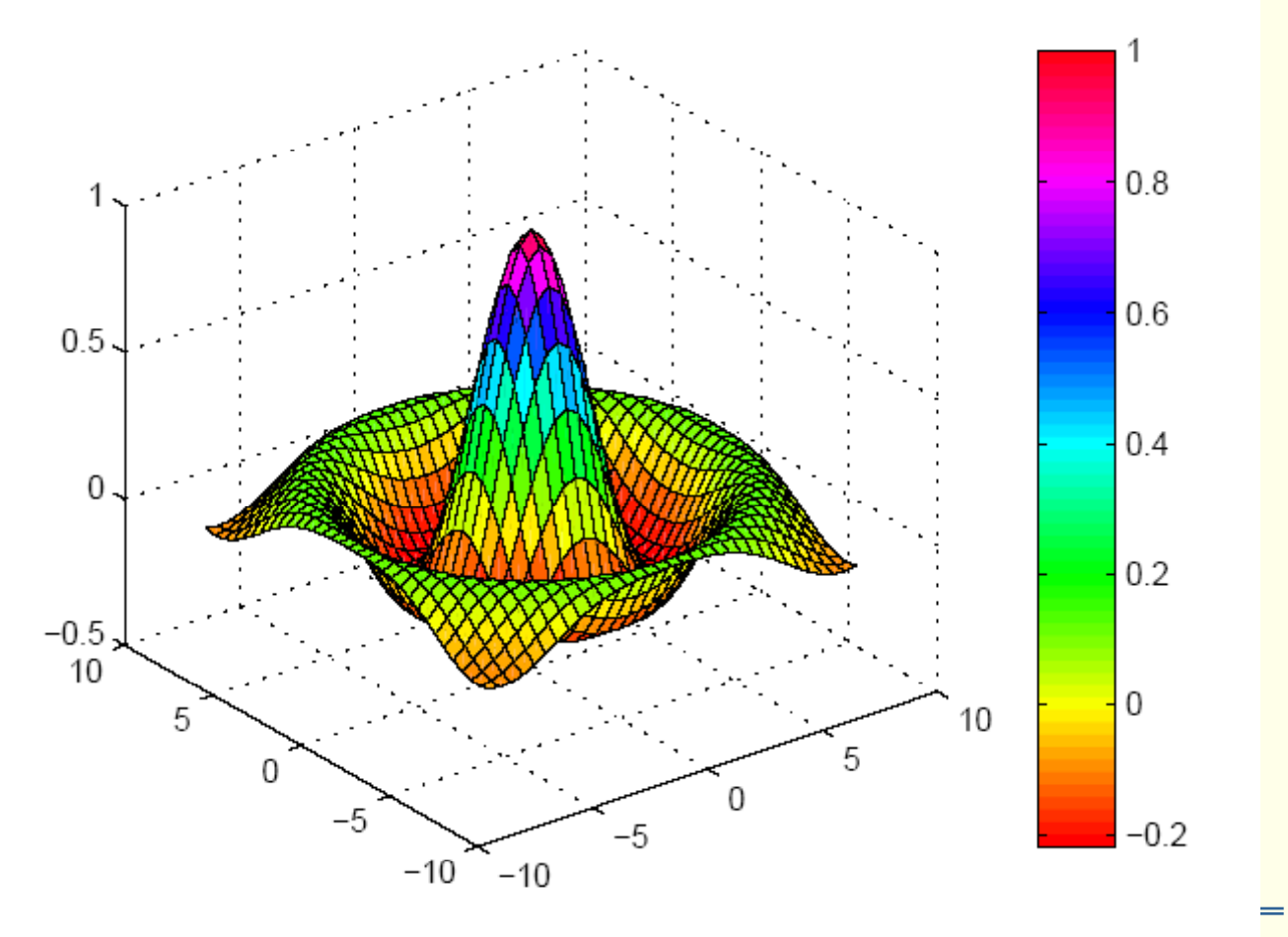
`plot(x,y,'color_style_marker')`

color_style_marker is a string containing from one to four characters (enclosed in single quotation marks) constructed from a color, a line style, and a marker type:

- Color strings are `'c'`, `'m'`, `'y'`, `'r'`, `'g'`, `'b'`, `'w'`, and `'k'`. These correspond to cyan, magenta, yellow, red, green, blue, white, and black.
- Line style strings are `'-'` for solid, `'--'` for dashed, `':'` for dotted, `'-.'` for dash-dot. Omit the line style for no line.
- The marker types are `'+'`, `'o'`, `'*'`, and `'x'`, and the filled marker types are `'s'` for square, `'d'` for diamond, `'^'` for up triangle, `'v'` for down triangle, `'>'` for right triangle, `'<'` for left triangle, `'p'` for pentagram, `'h'` for hexagram, and `none` for no marker.

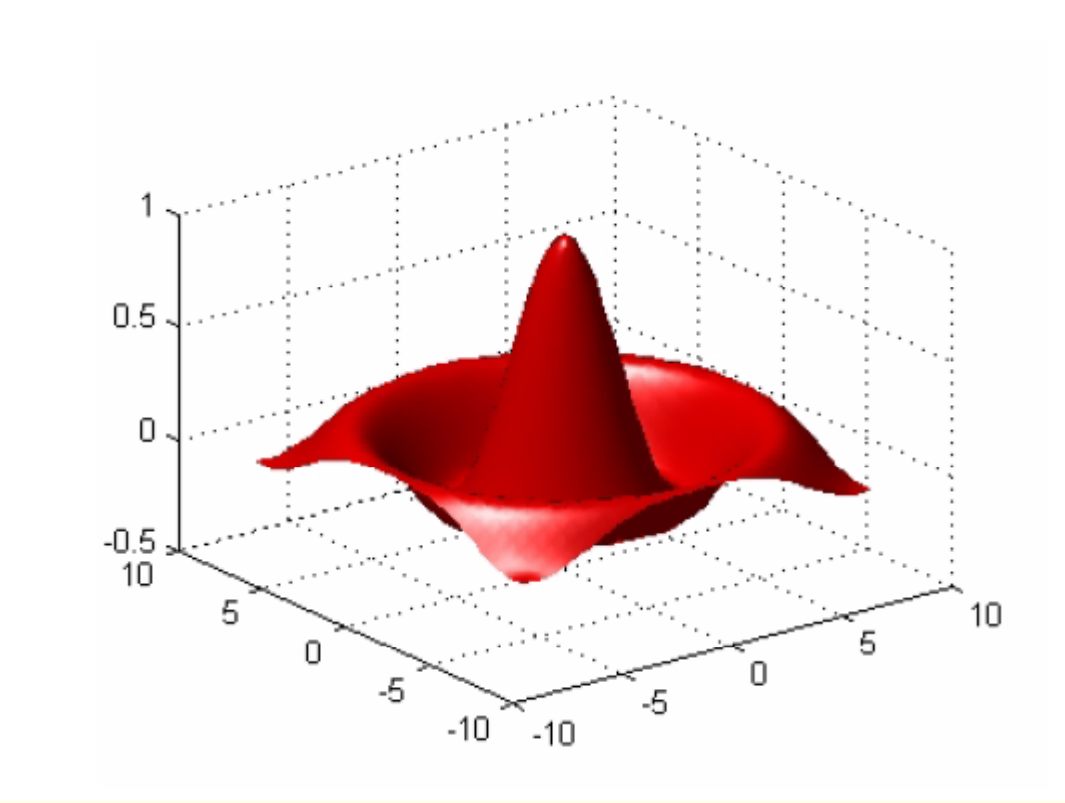
3-D Plots

```
[X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
surf(X,Y,Z)
colormap hsv
colorbar
```



Graphs with Light

```
surf(X,Y,Z,'FaceColor','red','EdgeColor','none')
camlight left; lighting phong
```



M-file and Function Files

M-file

- Script file containing series of commands having extension ".m"
- Use "%" to put comments for descriptive program

Function File

- A file to define a function

```
function [ri,phi]=risg(nw,x,w,xbg)
ri=zeros(size(xbg,1),1);
for iw=1:nw
  phi=xbg./(x(iw)+xbg);
  ri=ri+x(iw)*phi*w(iw);
end
```

Basic Functions

```
help
matlab\general       -  General purpose commands.
matlab\elmat        -  Elementary matrices and matrix
     manipulation.
matlab\elfun        -  Elementary math functions.
matlab\specfun      -  Specialized math functions.
matlab\matfun       -  Matrix functions - numerical linear
     algebra.
matlab\datafun      -  Data analysis and Fourier transforms.
matlab\polyfun      -  Interpolation and polynomials.
matlab\datafun      -  Data analysis and Fourier
     transforms.solvers.
matlab\funfun       -  Function functions and ODE solvers.
matlab\sparfun      -  Sparse matrices.
matlab\graph2d      -  Two dimensional graphs.
matlab\graph3d      -  Three dimensional graphs.
matlab\timefun        -  File input and output.
matlab\iofun        -  File input and output.
matlab\demos        -  Examples and demonstrations.

matlab\general       -  General purpose commands.
  Managing the workspace.
    who           - List current variables.
    whos          - List current variables, long form.
    clear         - Clear variables and functions from memory.
    load          - Load workspace variables from disk.
    save          - Save workspace variables to disk.

  Operating system commands.
    cd            - Change current working directory.
    pwd           - Show (print) current working directory.
    ls            - List directory.
    !             - Execute operating system command (see PUNCT).
```

```
matlab\elmat        -  Elementary matrices and matrix
     manipulation.
  Elementary matrices.
    zeros       - Zeros array.
    ones        - Ones array.
    eye         - Identity matrix.
    rand        - Uniformly distributed random numbers.
    randn       - Normally distributed random numbers.

  Basic array information.
    size        - Size of array.
    length      - Length of vector.
    ndims       - Number of dimensions.
    disp        - Display matrix or text.

  Matrix manipulation.
    cat         - Concatenate arrays.
    reshape     - Change size.
    diag        - Diagonal matrices and diagonals of matrix.
    tril        - Extract lower triangular part.
    triu        - Extract upper triangular part.
    find        - Find indices of nonzero elements.
```

Basic Functions

```
matlab\elfun         -  Elementary math functions.
  Trigonometric.
    sin          - Sine.
    sinh         - Hyperbolic sine.
    cosd         - Cosine of argument in degrees.
    cosh         - Hyperbolic cosine.
    acos         - Inverse cosine.
    acoth        - Inverse hyperbolic cotangent.

  Exponential.
    exp          - Exponential.
    log          - Natural logarithm.
    log10        - Common (base 10) logarithm.
    sqrt         - Square root.
    nthroot      - Real n-th root of real numbers.

  Rounding and remainder.
    fix          - Round towards zero.
    floor        - Round towards minus infinity.
    ceil         - Round towards plus infinity.
    round        - Round towards nearest integer.
    mod          - Modulus (signed remainder after
                   division).
    rem          - Remainder after division.
    sign         - Signum.
```

```
matlab\matfun        -  Matrix functions - numerical
    linear algebra.
  Matrix analysis.
    norm         - Matrix or vector norm.
    rank         - Matrix rank.
    det          - Determinant.
    trace        - Sum of diagonal elements.
    rref         - Reduced row echelon form.

  Linear equations.
    \ and /      - Linear equation solution; use "help
                   slash".
    inv          - Matrix inverse.
    cond         - Condition number with respect to
                   inversion.
    lu           - LU factorization.

  Eigenvalues and singular values.
    eig          - Eigenvalues and eigenvectors.
    poly         - Characteristic polynomial.

matlab\datafun       -  Data analysis and Fourier
    transforms.
  Basic operations.
    max          - Largest component.
    min          - Smallest component.
    mean         - Average or mean value.
    std          - Standard deviation.
    var          - Variance.
    sort         - Sort in ascending order.
    sum          - Sum of elements.
    prod         - Product of elements.
```

Basic Functions

```
matlab\funfun   -  Function functions and ODE solvers.
  Optimization and root finding.
    fminsearch - Multidimensional unconstrained nonlinear
    minimization,
                  by Nelder-Mead direct search method.
    fzero      - Scalar nonlinear zero finding.

  Numerical integration (quadrature).
    quad       - Numerically evaluate integral, low order
    method.
    quadl      - Numerically evaluate integral, higher
    order method.

matlab\sparfun      -  Sparse matrices.
  Full to sparse conversion.
    sparse     - Create sparse matrix.
    full       - Convert sparse matrix to full matrix.
    find       - Find indices of nonzero elements.
    spy        - Visualize sparsity pattern.
```

```
matlab\graph2d      -  Two dimensional graphs.
  Elementary X-Y graphs.
    plot       - Linear plot.
    loglog     - Log-log scale plot.
    semilogx   - Semi-log scale plot.
    semilogy   - Semi-log scale plot.

  Axis control.
    axis       - Control axis scaling and appearance.
    grid       - Grid lines.
    box        - Axis box.
    subplot    - Create axes in tiled positions.

  Graph annotation.
    title      - Graph title.
    xlabel     - X-axis label.
    ylabel     - Y-axis label.
    text       - Text annotation.
    gtext      - Place text with mouse.

  Hardcopy and printing.
    print      - Print graph or Simulink system; or
    save graph to M-file.
```

Basic Functions

```
matlab\graph3d       -  Three dimensional
   graphs.
    plot3       - Plot lines and points in 3-
   D space.
    mesh        - 3-D mesh surface.
    surf        - 3-D colored surface.
    fill3       - Filled 3-D polygons.

  Color control.
    colormap    - Color look-up table.
    caxis       - Pseudocolor axis scaling.

  Lighting.
    surfl       - 3-D shaded surface with
   lighting.
    lighting    - Lighting mode.

  Viewpoint control.
    view        - 3-D graph viewpoint
   specification.
    rotate3d    - Interactively rotate view
   of 3-D plot.

  Hardcopy and printing.
    vrml        - Save graphics to VRML 2.0
   file.
```

```
matlab\iofun         -  File input and
   output.

  File opening and closing.
    fopen         - Open file.
    fclose        - Close file.

  Binary file I/O.
    fread         - Read binary data from
   file.
    fwrite        - Write binary data to
   file.

matlab\timefun       -  Time and dates.
  Current date and time.
    now           - Current date and time as
   date number.
    date          - Current date as date
   string.
    clock         - Current date and time as
   date vector.

  Timing functions.
    cputime       - CPU time in seconds.
    etime         - Elapsed time.
    pause         - Wait in seconds.
```