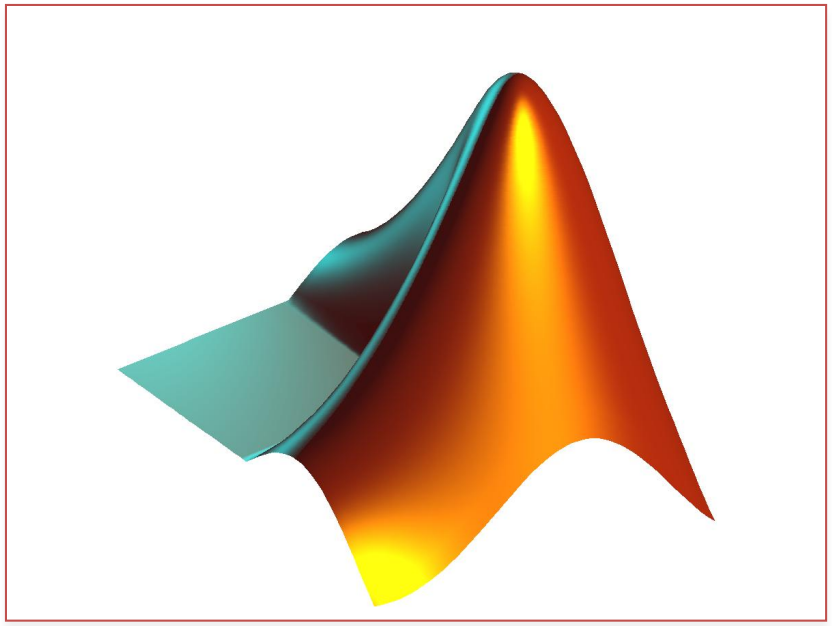
CPCS 212 NUMERICAL COMPUTATIONS



Chapter 1 & 2

Course Coordinator : Dr. Mohammad Husain

Introduction

What is MATLAB

- **MATLAB** is a numerical computing environment and programming language.
- **MATLAB**, which stands for MATrix LABoratory, is a powerful, general purpose system or environment for doing mathematics, scientific and engineering calculations.
- Created by *The MathWorks*, **MATLAB** allows easy
  - matrix manipulation,
  - plotting of functions and data,
  - implementation of algorithms,
  - creation of user interfaces, and
  - interfacing with programs in other languages.
- **MATLAB** is available for Windows, Macintosh and UNIX systems. It is used by more than one million people in industry and academia.

Why use Matlab?

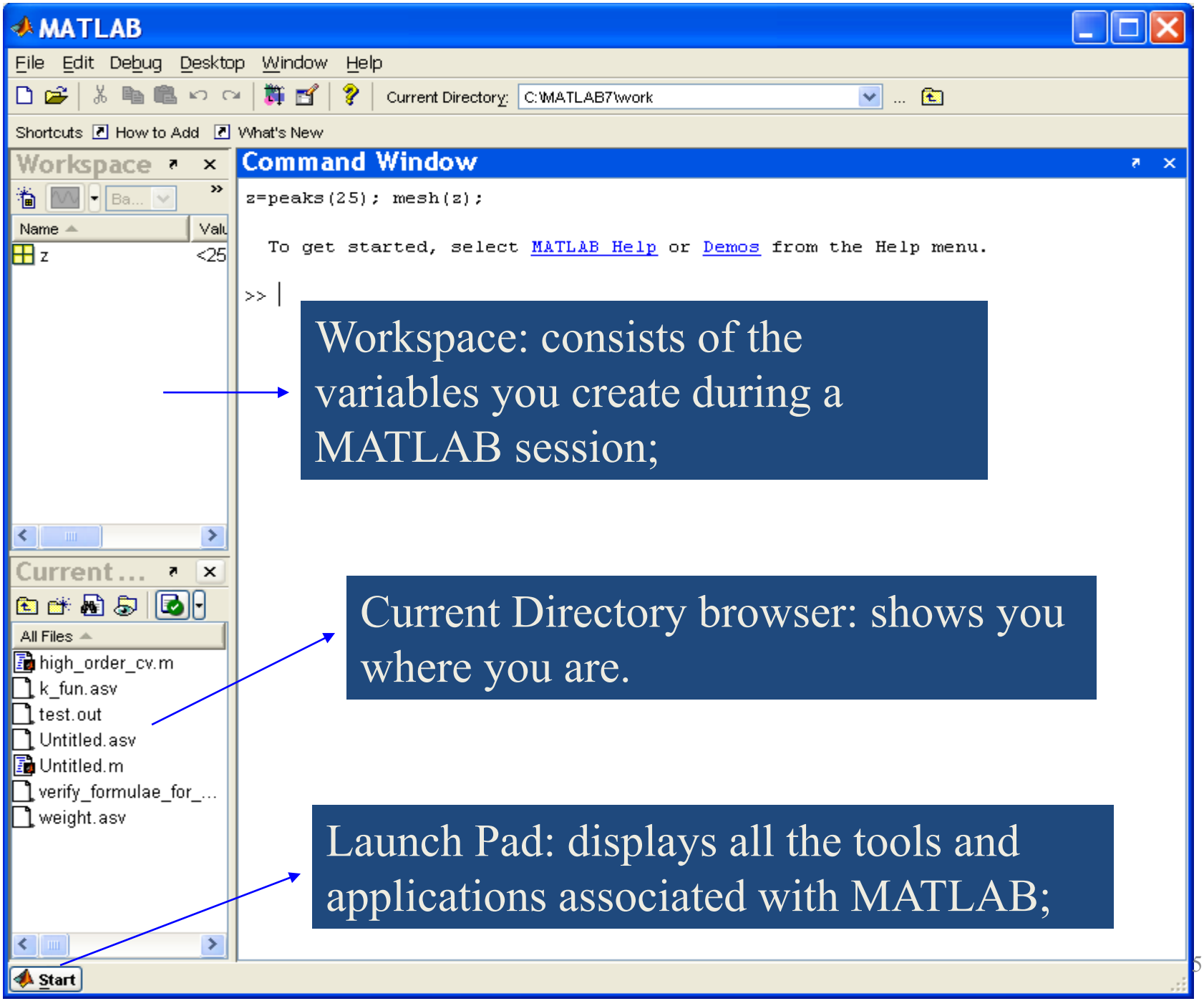
- Advantages:
  - It allows quick and easy coding in a very high-level language.
  - Rich data types: Complex number, Three dimensional matrix, structure, cell array, etc
  - Lots of nice functions and toolboxes: fminsearch, fsolve, normcdf, norminv, etc; garch, optimization, symbolic, …
  - Lots of users: economists, mathematicians, engineers, …
  - High-quality graphics and visualization facilities are available.
  - MATLAB M-files are completely portable across a wide range of platforms.
  - EXTENSIVE documentation (type 'helpwin')

How to start and quit Matlab?

PC - a double click on the Matlab icon

On both system leave a Matlab session by typing quit or by typing exit at the Matlab prompt.

Use Control ^C to stop the running program. It may take Matlab a while to respond.


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work
Shortcuts How to Add What's New
Workspace
Name
z
<25
Command Window
z=peaks(25); mesh(z);
To get started, select MATLAB Help or Demos from the Help menu.
>>
Current ...
All Files
high_order_cv.m
k_fun.asv
test.out
Untitled.asv
Untitled.m
verify_formulae_for_...
weight.asv
Start
Workspace: consists of the variables you create during a MATLAB session;
Current Directory browser: shows you where you are.
Launch Pad: displays all the tools and applications associated with MATLAB;

Getting Started

MATLAB Desktop:

- Launch Pad: displays all the tools and applications associated with MATLAB;
- Workspace: consists of the variables you create during a MATLAB session;
- Command History: double click them to evaluate them;
- Current Directory browser: shows you where you are.
- Editor/Debugger: pops up when you create an M-files (click on “New” button to launch it.)

Using Help in Matlab

Online help is available from the Matlab prompt (>> a double arrow)

>> helpwin
[a long list of help topics follows]

>> helpwin fminsearch
[a help message on the fminsearch function follows].

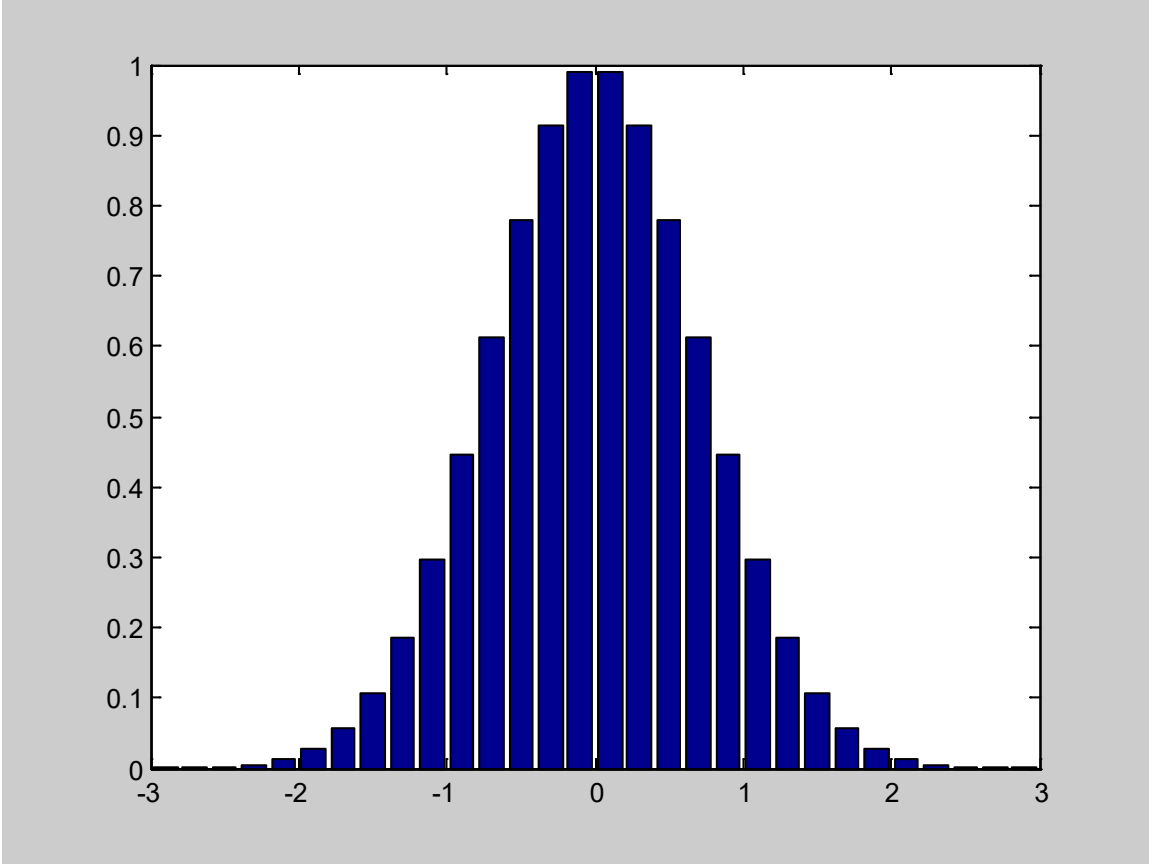
>>doc plot
[displays the HTML documentation for the MATLAB function plot].

>> type norminv
[display brief information of the non-built-in function norminv]

>> lookfor cdf

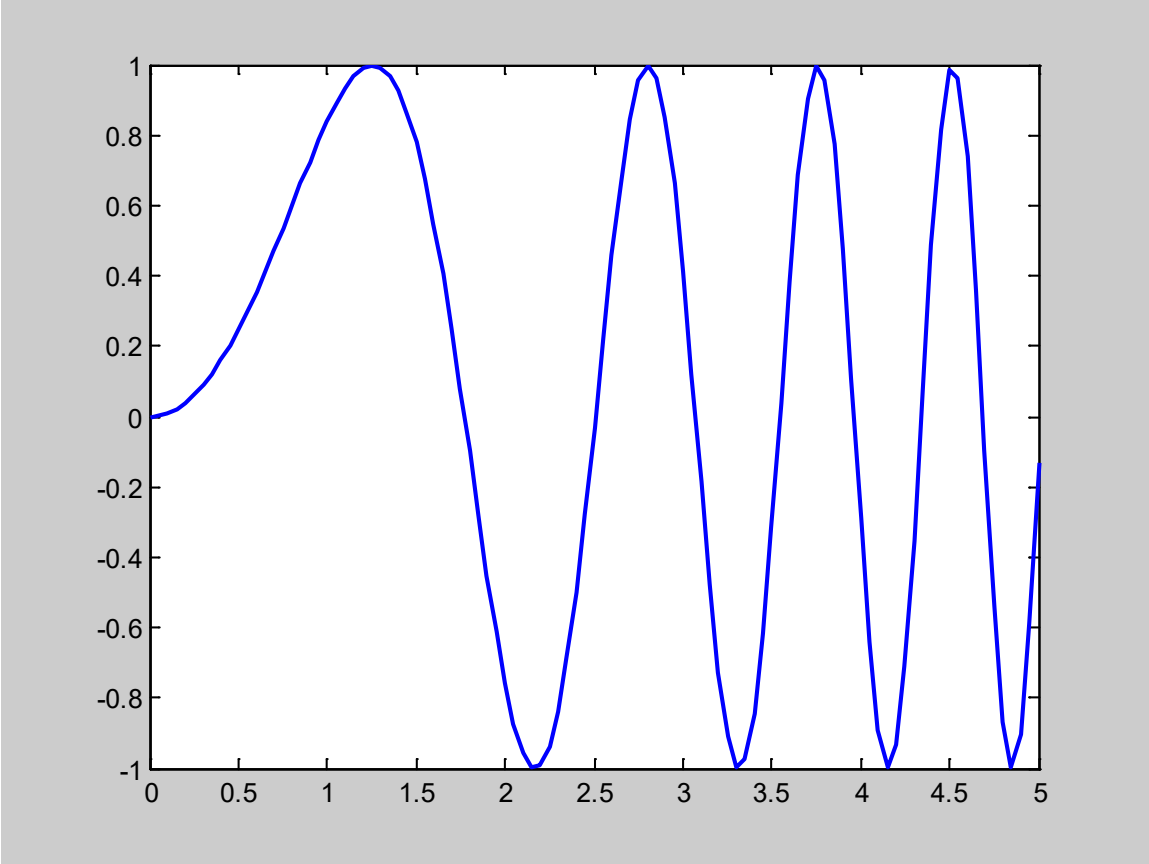
>> demo

What kind of graphics is possible in **Matlab**?




```
x = -2.9:0.2:2.9;
bar(x,exp(-x.*x));
```

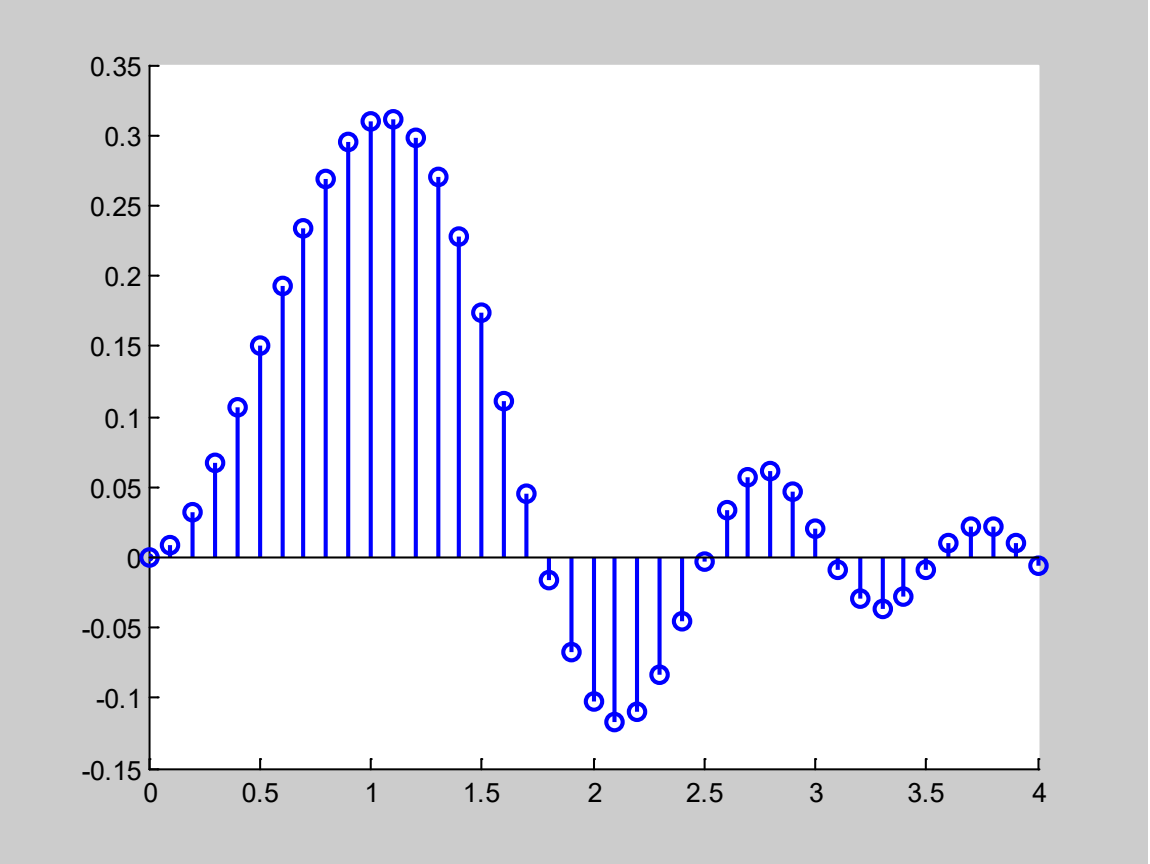
What kind of graphics is possible in **Matlab**?



Line plot:

```
x=0:0.05:5;y=sin(x.^2);plot(x,y);
```

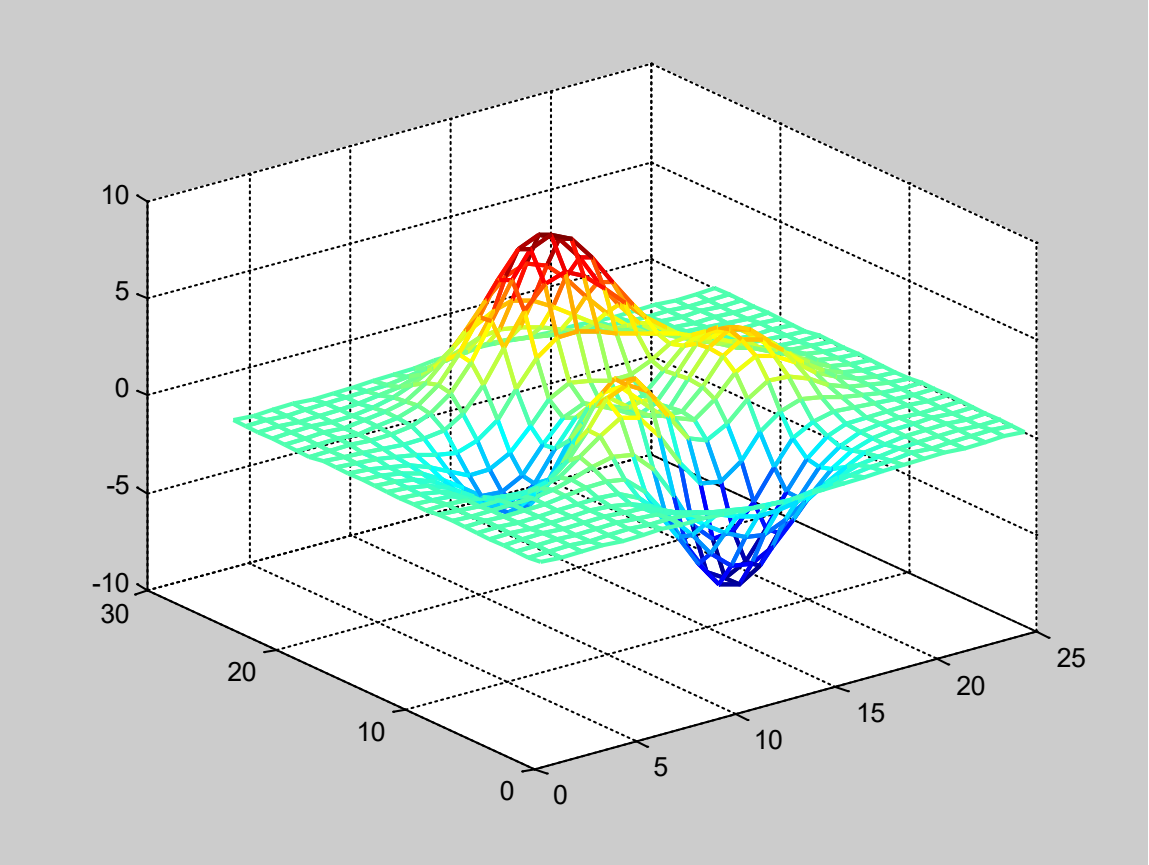
What kind of graphics is possible in **Matlab**?



Stem plot:

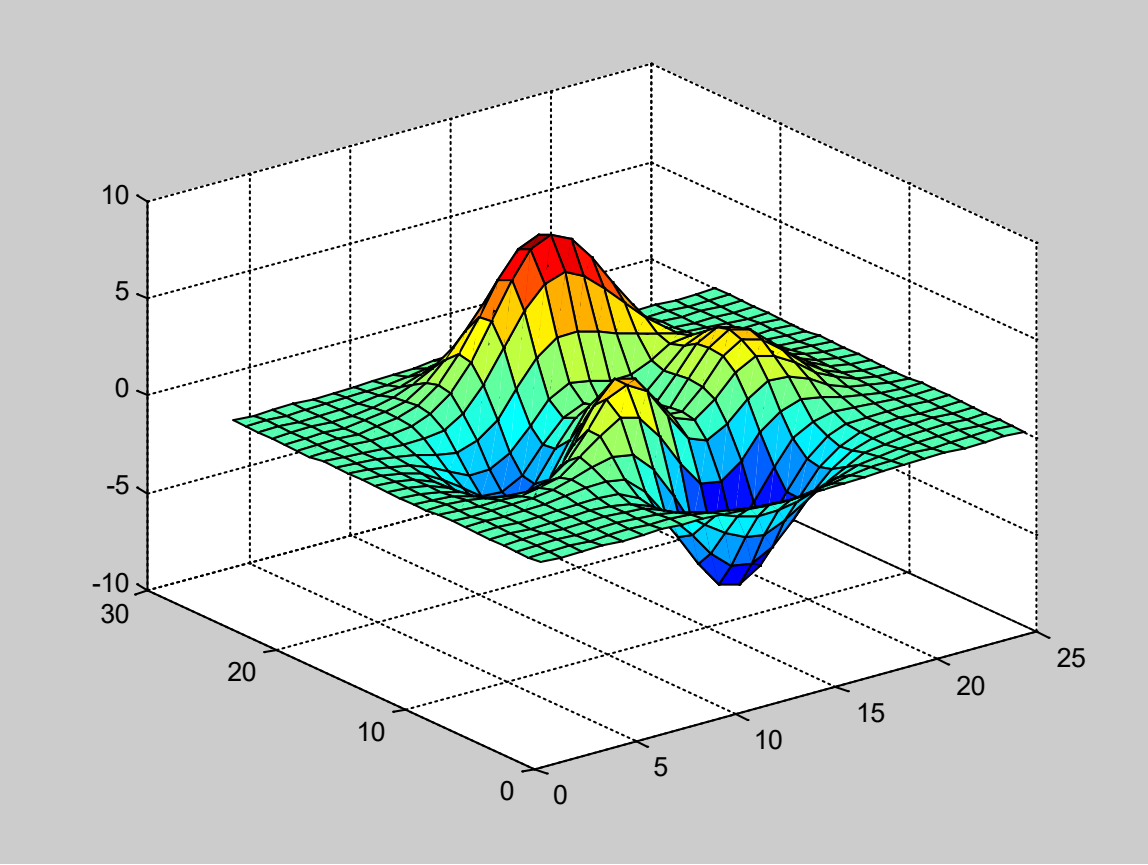
```
x = 0:0.1:4;, y = sin(x.^2).*exp(-x);
stem(x,y)
```

What kind of graphics is possible in **Matlab**?



Mesh plot: z=peaks(25); mesh(z);

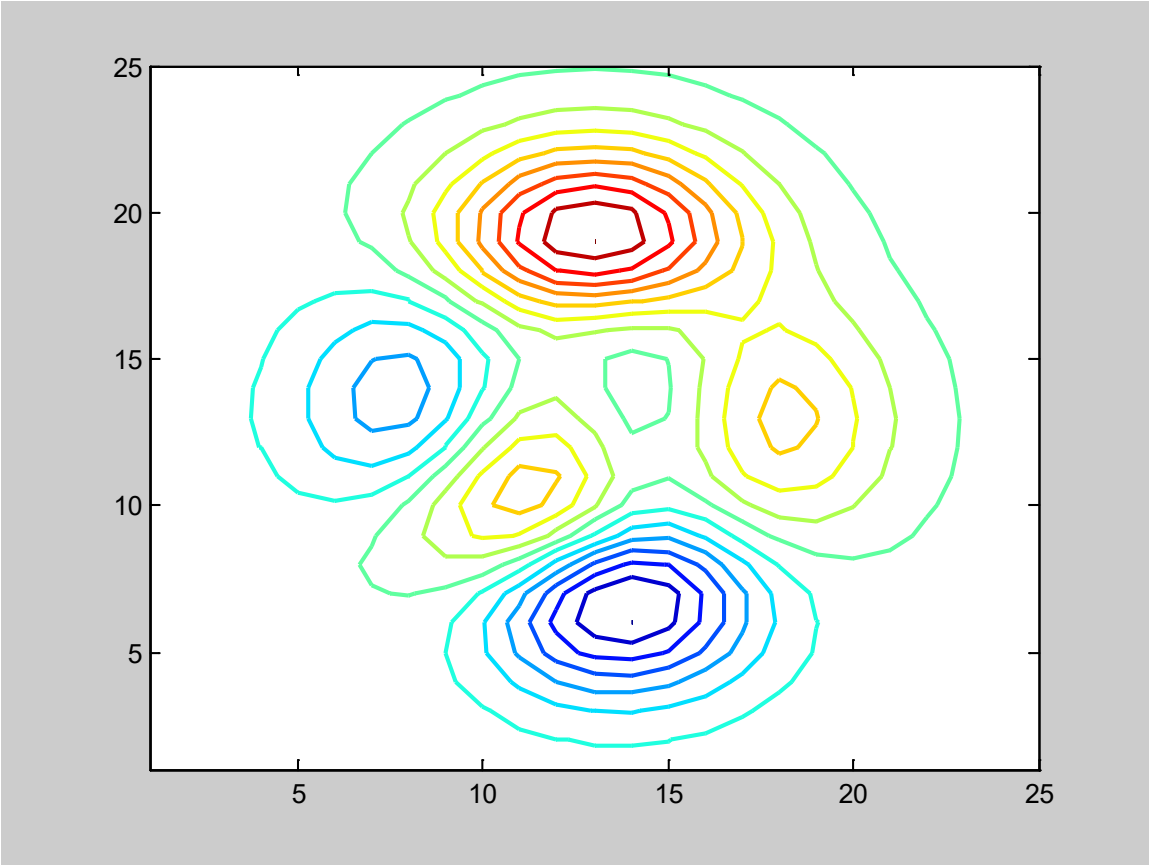
What kind of graphics is possible in **Matlab**?



Surface plot:

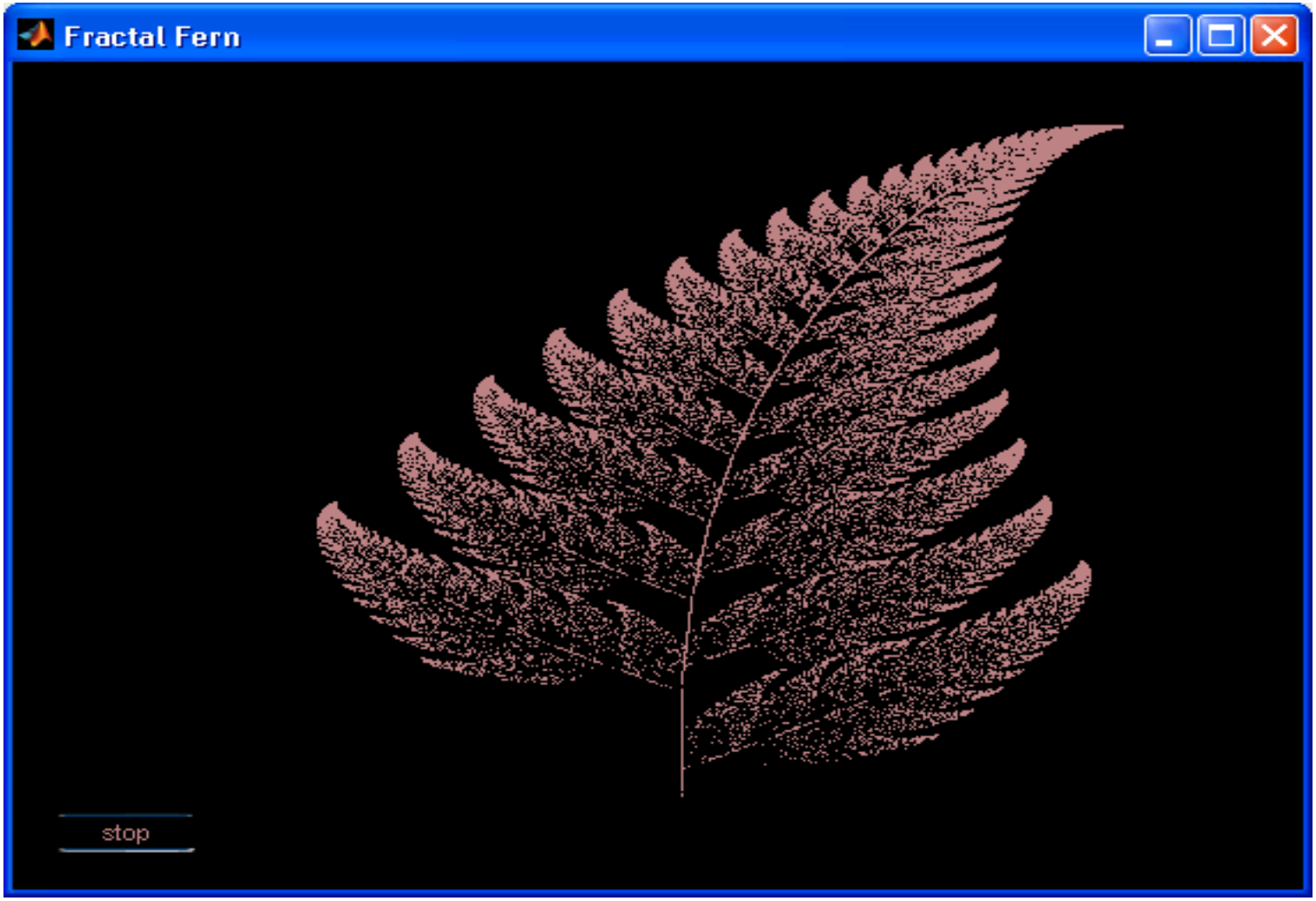
z=peaks(25);, surf(z);, colormap(jet);

What kind of graphics is possible in **Matlab**?



Contour plot: z=peaks(25); contour(z,16);

What kind of graphics is possible in **Matlab**?



Basic Features

- Execute commands either by
  - Write the code in command window
  - Write code in a script file
    - Execute script file by typing the file name in command window.
  - Write code in a function file
    - Execute function file by typing the file name in command window.

- Addition +
- Subtraction -
- Multiplication *
- Division /
- Exponentiation ^ e.g., 1.3^3.2 is 1.3 power 3.2
- Store results in variables
  - It is Case sensitive
  - Maximum of 31 characters
  - Must start with a letter
- Example
  - In command window

```
>> 5+5
ans =
   10
>> a=5+5
a =
   10
```

Variable assignment statements

- The equals sign is the MATLAB assignment statement. The command a=5 stores the value 5 in the variable named a.

```
>> a=5;                                  % class double
>> fname='Robert';                       % class char
>> temperature=101.2;                    % class double
>> isDone=true;                          % class logical
```

Example

```
>> a=5;
>> b=7;
>> c=a+b                    % uses current values of a and b
    c =
          12
>> a=0;
>> b=-2;
>> c
    c =12                   % kept same value despite a and b changing
```

Invalid Expressions

```
>> r=a=4;                          % not a valid MATLAB statement
>> a+1=press-2;                    % not a valid MATLAB statement
>> 4=a;                            % not a valid MATLAB statement
>> 'only the lonely'='how I feel'; % not a valid MATLAB statement
```

Variable names

Variable names are case-sensitive and must begin with a letter. The name must be composed of letters, numbers, and underscores; do not use other punctuation symbols. Only the first 31 characters of the variable name are significant. For example -

xinit	okay
VRightInitial	okay
4You2do	not okay
Start-up	not okay
vector%1	not okay
TargetOne	okay
ThisIsAVeryVeryLongVariableName	okay
x_temp	okay

Variable workspace

The currently defined variables exist in the MATLAB workspace. The workspace is part of the dynamic memory of the computer. Items in the workspace will vanish when the current MATLAB session is ended (i.e., when we quit MATLAB). The workspace can be saved to a file and reloaded later, although use of this feature will be rare.

Commands

clear a v g	clears the variablesa v gfrom the workspace
clear	clears all variables from the workspace
who	lists the currently defined variables
whos	displays a detailed list of defined variables
save	saves the current workspace to the file called matlab.mat
save foobar	saves the current workspace to the file called foobar.mat
load	loads variables saved in matlab.mat into the current workspace
load foobar	loads variables saved in foobar.mat into the current workspace

Some Standard Functions

sin(x)	returns the sine of x
sind(x)	returns the sine of x degrees
cos(x)	returns the cosine of x
cosd(x)	returns the cosine of x degrees
tan(x)	returns the tangent of x
tand(x)	returns the tangent of x degrees
atan(x)	returns the inverse tangent of x
atand(x)	returns the inverse tangent of x in degrees
acos(x)	returns the inverse cosine of x
acosd(x)	returns the inverse cosine of x in degrees
asin(x)	returns the inverse sine of x
asind(x)	returns the inverse sind of x in degrees
exp(x)	returns ex
log(x)	returns the natural logarithm of x
log10(x)	returns the log10(x)
sqrt(x)	returns the square root of x
abs(x)	returns the absolute value of x

Some Standard Functions

round(x)	returns the integer closest to x
ceil(x)	returns the smallest integer greater than or equal to x
floor(x)	returns the largest integer less than or equal to x
isprime(n)	returns true if n is prime
factor(k)	returns prime factors of k
sign(x)	returns the sign (1 or −1) of x; sign(0) is 0
rand	returns a pseudorandom number between 0 and 1
rand(m)	returns an m×m array of random numbers
rand(m,n)	returns an m×n array of random numbers

Matrix

Matlab works with essentially only one kind of object – a rectangular numerical matrix with possible complex entries.

Matrices can be

- Entered manually;
- Generated by built-in functions; (e.g., peaks(25))
- Loaded from external file (using "load" command)

Example:

```
A=[ 1, 2, 3, 4; 5, 6, 7, 8];  % Use ' ; ' to indicate the end of each row
```

Equivalently

```
A=[ 1, 2, 3, 4
    5, 6, 7, 8];
```

Or

```
A=[[ 1, 2, 3, 4]; [5, 6, 7, 8]];
```

Matrix

Examples

```
x = [ 3.5, 33.22, 24.5 ] ;
```

x is a row vector or 1 x 3 matrix

```
x1 = [ 2
       5
       3
      -1];
```

x1 is column vector or 4 x 1 matrix

The matrix name can be any group of letters and numbers up to 63, but always beginning with a letter. howaboutthisname how_about_this_name

Matlab is "case sensitive", that is, it treats the name 'C' and 'c' as two different variables.

Similarly, 'MID' and 'Mid' are treated as two different variables.

Syntax in Matlab

Colon operator: The colon operator ' : ' is understood by Matlab to perform special and useful operations.

For example, if two integer numbers are separated by a colon, Matlab will generate all of the integers between these two integers.

```
a = 1:8
```

generates the row vector, a = [1 2 3 4 5 6 7 8].

If three numbers, integer or non-integer, are separated by two colons, the middle number is interpreted to be a " step" and the first and third are interpreted to be "limits":

```
b = 0 :0 .2 : 1.0
```

generates the row vector b = [0.0 .2 .4 .6 .8 1.0]

Syntax in Matlab

The colon operator can be used to create a vector from a matrix. Thus if

```
x = [ 2  6  8
      0  1  7
     -2  5 -6 ]
```

The command `y = x(:,1)` creates the column vector

```
y = [  2
       0
      -2  ]
```

The command `z = x(1,:)` creates the row vector

```
z = [ 2  6  8 ]
```

What is `w = x(2:end,:)`?

Syntax in Matlab

The colon operator is useful in extracting smaller matrices from larger matrices. If the 4 x 3 matrix c is defined by

```
c = [ -1  0  0
       1  1  0
       1 -1  0
       0  0  2 ]
```

Then

```
d1 = c(:,2:3) or d1=c(:,[2, 3])
```

creates a matrix for which all elements of the rows from the 2nd and third columns are used. The result is a 4 x 2 matrix

```
d1 = [  0  0
        1  0
       -1  0
        0  2 ]
```

Matrix Operations

- + addition
- - subtraction
- * multiplication
- ^ power
- ‘ transpose for a real matrix and complex –conjugate transpose for a complex matrix (transpose for a complex matrix is .’)
- \ left division, / division

x = A \ b is the solution of A * x = b

x = b / A is the solution of x * A = b

Matrix math

- Dimensions must agree

Scalar math

- Same as usual

Scalar / matrix math

- Scalar + matrix = [scalar + matrix(i, j)]
- Scalar * matrix = [scalar * matrix(i, j)]

Matrix Operations

To make the '*' , '^', '\' and '/' entry-wise, we precede the operators by '.'

a .* b multiplies each element of a by the respective element of b

a ./ b divides each element of a by the respective element of b

a .\ b divides each element of b by the respective element of a

a .^ b raise each element of a by the respective b element

Example

- x = [1 2 3]; y = [4 5 6];
- x '* y = 32
- x .* y = [4 10 18]

Matrix Operations

- diag: Diagonal matrices or diagonals of a matrix.

diag(A) for matrix A is the diagonals of A.

A= 0.1531 0.4929 0.1478
0.0495 0.6107 0.4326
0.9340 0.1224 0.7917

Then diag(A)= [0.1531; 0.6107; 0.7917]

- blockdiag: Construct a block diagonal matrix

blockdiag(A,B,C)=

$$\begin{pmatrix} A & \dots & 0 \\ \dots & B & \dots \\ 0 & \dots & C \end{pmatrix}$$

Some Basic Statistics Functions

max(x) returns the maximum value of the elements in a vector or if x is a matrix, returns a row vector whose elements are the maximum values from each respective column of the matrix.

min (x) returns the minimum of x (see max(x) for details).

mean(x) returns the mean value of the elements of a vector or if x is a matrix, returns a row vector whose elements are the mean value of the elements from each column of the matrix.

median(x) same as mean(x), only returns the median value.

sum(x) returns the sum of the elements of a vector or if x is a matrix, returns the sum of the elements from each respective column of the matrix.

prod(x) same as sum(x), only returns the product of elements.

Some Basic Statistics Functions

std(x) returns the standard deviation of the elements of a vector or if x is a matrix, a row vector whose elements are the standard deviations of each column of the matrix

sort(x) sorts the values in the vector x or the columns of a matrix and places them in ascending order.

hist(x) plots a histogram of the elements of vector, x. The bins are scaled based on the max and min values

hist(x,n) plots a histogram with 'n' bins scaled between the max and min values of the elements

hist((x(:,2)) plots a histogram of the elements of the 2nd column from the matrix x

corr(x) returns a pairwise correlation coefficient between the columns in X

Some Basic Mathematical Functions

- log(x) is the natural log. There is no ln(x) in Matlab
- exp(x), sqrt(x), sin(x), cos(x), …
- floor(x): Round towards minus infinity.

  floor(5.4) =5; floor(-5.4)=-6
- ceil(x): Round towards plus infinity.
- Try 'helpwin elfun' to get the list of elementary math functions.
- Try 'helpwin specfun' to get the list of special math functions, such as gamma(x), beta(x),
- For a matrix A, fun(A) in general operates on each element of A.

Useful Constants

NaN the arithmetic representation for Not-a-Number, a NaN is obtained as a result of mathematically undefined operations like 0.0/0.0

Inf the arithmetic representation for positive infinity, a infinity is also produced by operations like dividing by zero, e.g. 1.0/0.0, or from overflow, e.g. exp(1000).

```
>> exp(1000)

ans =  Inf
```

eps	machine epsilon
ans	most recent unassigned answer
pi	3.14159….
i and j	Matlab supports imaginary numbers!

Some Basic Commands

pwd prints working directory

demo demonstrates what is possible in **Matlab**

who lists all of the variables in your matlab workspace

whos list the variables and describes their matrix size

clear erases variables and functions from memory

clear x erases the matrix 'x' from your workspace

; (semicolon) prevent commands from outputting results

% (percent sign) comments line

```
%{comments1
   comments2
%}
```

Block comments

… A line can be terminated with three periods (...), which causes the next line to be a continuation line

1. Problem Definition:

The first steps in problem solving include:

- Recognize and define the problem precisely by exploring it thoroughly (may be the most difficult step).
- Determine what question is to be answered and what output or results are to be produced.
- Determine what theoretical and experimental knowledge can be applied.
- Determine what input information or data is available

After defining the problem:

- Collect all data and information about the problem.
- Verify the accuracy of this data and information.
- Determine what information you must find: intermediate results or data may need to be found before the required answer or results can be found.

2. Mathematical Model:

To create a mathematical model of the problem to be solved:

- Determine what fundamental principles are applicable.
- Draw sketches or block diagrams to better understand the problem.
- Define necessary variables and assign notation.
- Reduce the problem as originally stated into one expressed in purely mathematical terms.
- Apply mathematical expertise to extract the essentials from the underlying physical description of the problem.
- Simplify the problem only enough to allow the required information and results to be obtained.
- Identify and justify the assumptions and constraints inherent in this model.

3. Computational Method:

A computational method for solving the problem is to be developed, based on the mathematical model.

- Derive a set of equations that allow the calculation of the desired parameters and variables.
- Develop an algorithm, or step-by-step method of evaluating the equations involved in the solution.
- Describe the algorithm in mathematical terms and then implement as a computer program.
- Carefully review the proposed solution, with thought given to alternative approaches.

4. Implementation of Computational Method:

Once a computational method has been identified, the next step is to carry out the method with a computer, whether human or silicon.

Some things to consider in this implementation:

- Assess the computational power needed, as an acceptable implementation may be hand calculation with a pocket calculator.
- If a computer program is required, a variety of programming languages, each with different properties, are available.
- A variety of computers, ranging from the most basic home computers to the fastest parallel supercomputers, are available.
- The ability to choose the proper combination of programming language and computer, and use them to create and execute a correct and efficient implementation of the method, requires both knowledge and experience.

5. Test and Assess the Solution:

The final step is to test and assess the solution. In many aspects, assessment is the most open-ended and difficult of the five steps involved in solving computational problems.

The numerical solution must be checked carefully:

- A simple version of the problem should be hand checked.
- The program should be executed on obtained or computed test data for which the answer or solution is either known or which can be obtained by independent means, such as hand or calculator computation.
- Intermediate values should be compared with expected results and estimated variations. When values deviate from expected results more than was estimated, the source of the deviation should be determined and the program modified as needed.
- A “reality check” should be performed on the solution to determine if it makes sense.
- The assumptions made in creating the mathematical model of the problem should be checked against the solution.

Definitions of Computing Terminology:

Command: A user-written statement in a computer language that provides instructions to the computer.

Variable: The name given to a quantity that can assume a value,.

Default: The action taken or value chosen if none has been specified.

Toggle: To change the value of a variable that can have one of two states or values. For example, if a variable may be “on” or “off” and the current value is “on,” to toggle would change the value to “off.”

Arguments: The values provided as inputs to a command.

Returns: The results provided by the computer in response to a command.

Execute: To run a program or carry out the instructions specified in a command.

Display: Provide a listing of text information on the computer monitor or screen.

Echo: To display commands or other input typed by the user.

Print: To output information on a computer printer (often confused with “display” in the textbook).

Thank You !!!