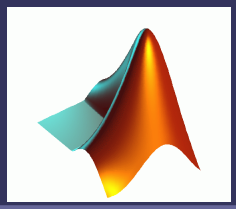
Introduction to MATLAB



--Dr. G. L. Prajapati,
IET-DAVV

MATLAB

What Is MATLAB?

MATLAB (MATrix LABoratory)
- high-performance language for technical computing
- computation, visualization, and programming in an easy-to-use environment

Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

Why MATLAB

A good choice for vision program development because:

- Easy to do very rapid prototyping
- Quick to learn, and good documentation
- A good library of image processing functions
- Excellent display capabilities
- Widely used for teaching and research in universities and industry
- Another language to impress your boss with !

Why not MATLAB

Has some drawbacks:

- Slow for some kinds of processes
- Not geared to the web
- Not designed for large-scale system development

MATLAB Components

MATLAB consists of:

- **The MATLAB language**
- a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features.

- **The MATLAB working environment**
- the set of tools and facilities that you work with as the MATLAB user or programmer, including tools for developing, managing, debugging, and profiling

- **Handle Graphics**
- the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics.

- …(cont'd)

MATLAB Components

…

- **The MATLAB function library.**
- a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms as well as special image processing related functions

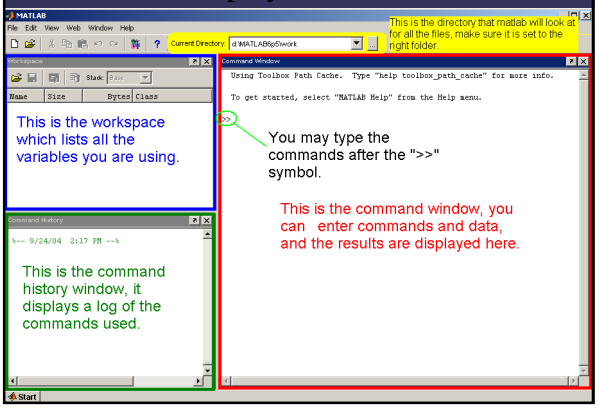
- **The MATLAB Application Program Interface (API)**
- a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB

Some facts for a first impression

- Everything in MATLAB is a matrix !
- MATLAB is an interpreted language, no compilation needed (but possible)
- MATLAB does not need any variable declarations, no dimension statements, has no packaging, no storage allocation, no pointers
- Programs can be run step by step, with full access to all variables, functions etc.

Display Windows



What does Matlab code look like?

A simple example:

```
a = 1
while length(a) < 10
a = [0 a] + [a 0]
end
```

which prints out Pascal's triangle:

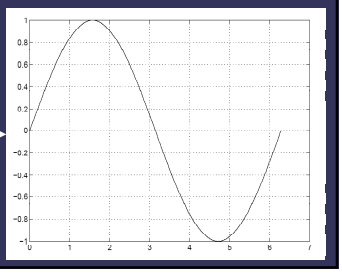
```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

(with "a=" before each line).

What does Matlab code look like?

Another simple example:

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```



What does Matlab code look like?

Another simple example:

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```

Remember:

EVERYTHING IN MATLAB IS A MATRIX !

creates 1 x 200 Matrix

Argument and result: 1 x 200 Matrix

Matrices

Matrices

Data in Matlab is always held as a **matrix**. Most are 2-D, in which individual elements are referenced by *row* and *column*. This is a 6×10 matrix..

	1	2	3	4	5	6	7	8	9	10
1	12	15	12	18	19	20	19	17	16	15
2	13	14	61	19	11	9	10	12	14	19
3	13	15	18	17	19	20	23	11	10	12
4	14	15	14	14	17	20	21	10	9	12
5	12	13	13	19	30	35	36	15	19	15
6	15	16	17	18	45	40	38	16	15	12

If this matrix is given the name `m`, then `m(5,4) = 19`, for example.

Remember: ***row*** then ***column***.

In Matlab, rows and columns are always numbered starting at 1.

Matrices

- Rows and columns are always numbered starting at 1
- Matlab matrices are of various types to hold different kinds of data (usually floats or integers)
- A single number is really a 1 x 1 matrix in Matlab!
- Matlab variables are not given a type, and do not need to be declared
- Any matrix can be assigned to any variable

Matrices

Building matrices with [ ]:

A = [2 7 4]

2	7	4

A = [2; 7; 4]

2
7
4

A = [2 7 4; 3 8 9]

2	7	4
3	8	9

B = [A A]

?

Matrices

Building matrices with [ ]:

A = [2 7 4]

2	7	4

A = [2; 7; 4]

2
7
4

A = [2 7 4; 3 8 9]

2	7	4
3	8	9

B = [A A]

2	7	4	2	7	4
3	8	9	3	8	9

Matrices

Operating on whole matrices

In most languages, you need to write loops to do things to all the elements of a data structure such as an array. In Matlab, many loops can be avoided.

```
a = [4 5 1; 3 6 8] + 1
```

prints

```
a =
   5 6 2
   4 7 9
```

That is, 1 has been added to every element of the matrix. If + is applied to two matrices of the same dimensions, corresponding elements are added:

```
b = a + a
```

prints

```
b=
   10   12    4
    8   14   18
```

Matrices

Some operators must be handled with care:

A = [1 2 ; 4 5]

B = A * A prints

9	12
24	33

B = A .* A prints

1	4
16	25

Element by element multiplication

Submatrices

A matrix can be indexed using another matrix, to produce a subset of its elements:

a = [100 200 300 400 500 600 700] b = [3 5 6]

c = a(b):

300 500 600

Submatrices

To get a subsection of a matrix, we can produce the index matrix with the colon operator:

```
a(2:5)
```

prints

```
ans = 200 300 400 500
```

- This works in 2-D as well, e.g. c(2:3, 1:2) produces a 2 x 2 submatrix.
- The rows and columns of the submatrix are renumbered.

loops

'for' loops in MATLAB iterate over matrix elements:

```
b = 0
for  i = [ 3  9  17]
        b = b + i;
end
```

Result: 29

Note:
The MATLAB way to write that program would have been:
b = sum([3 9 17]);

Avoid loops if possible !

loops

The typical 'for' loop looks like:

```
for i = 1:6
        …
end
```

Which is the same as:

```
for i = [1 2 3 4 5 6]
        …
end
```

loops

Once again:

AVOID LOOPS

Images

So why MATLAB and IMAGE PROCESSING ?

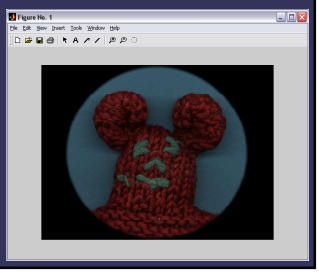
Images

Images can be treated as matrices !

Images

Loading an image:

a = imread('picture.jpg');
imshow(a);



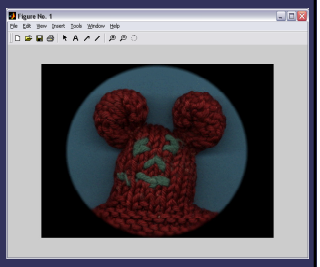
Images

Image (=matrix) size:
size(a): 384 512 3

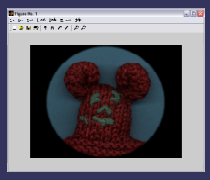
R G B

384

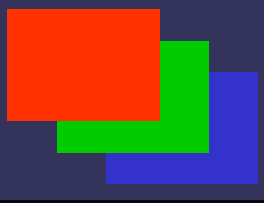
512



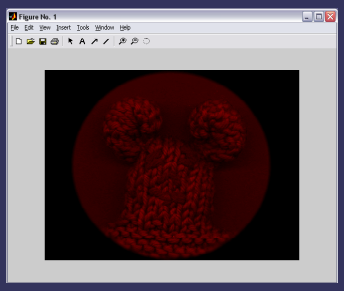
Images



Color image:
3D Matrix of RGB planes



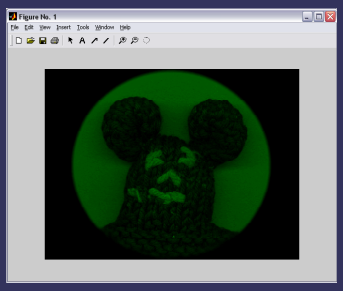
Images



Show RED plane:

a(:,:,2:3) = 0;
imshow(a);

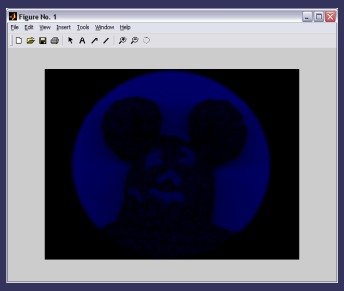
Images



Show GREEN plane:

a(:,:,[1 3]) = 0;
imshow(a);

Images



Show BLUE plane:

a(:,:,1:2) = 0;
imshow(a);

Plotting

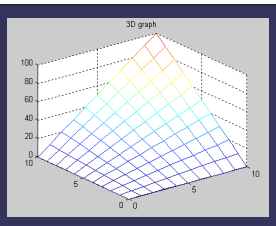
- Commands covered: plot, xlabel, ylabel, title grid, axis, stem, subplot
- *xlabel('time (sec)'); ylabel('step response'); title('My Plot');*

Eg:To plot more than one graph on the screen, use the command subplot(mnp) which partitions the screen into an mxn grid where p determines the position of the particular graph counting the upper left corner as p=1. For example,

- subplot(211),semilogx(w,magdb);
- subplot(212),semilogx(w,phase);

3D - Plotting example

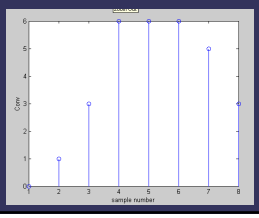
- x=[0:10]; y=[0:10]; z=x'*y;
- mesh(x,y,z); title('3-D Graph');



Convolution

- For example,

x = [1 1 1 1 1]; → [1 1 1 1 1]
h = [0 1 2 3]; → [3 2 1 0]
conv(x,h) yields y = [0 1 3 6 6 6 5 3].
stem*(y);*



ylabel('Conv');
xlabel('sample number');

By the way…

MATLAB can also handle

- Movies
- 3D objects
- …

Conclusion

MATLAB is a mighty tool to manipulate matrices

Images can be treated as matrices

MATLAB is a mighty tool to manipulate images

In my opinion…

MATLAB should be used to code software prototypes

Research is mostly about prototypes, not runtime-optimized software

MATLAB should be used in research

In my opinion…

- MATLAB prototypes must be re-coded (e.g. in C++) if there's need for speed
- Algorithm development time is drastically shorter in MATLAB