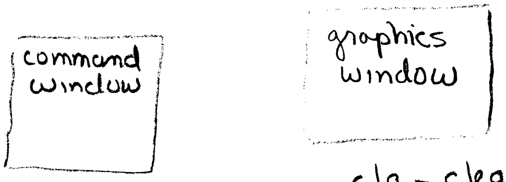
MATLAB primer (Ref: D.M. Etter, Engineering Problem Solving w/ MATLAB)

very simple syntax Fortran, Basic
basic

matlab prompt >>

basic commands demo - list of demos to run
quit
exit
save - save variables in workspace

limits of student edition
vector or matrix limited to 1024 elements.
graphics post processing not available (i.e. Post Script)



clc - clear command window

clg - clear graphics window

clear - clear all variables

^c - abort

MATLAB is case sensitive casesen off
casesen

who - list defined variables

whos - gives more information

% precedes comments

help - list of help topics

m-file <name>.m MATLAB program file
also called script file

to run an m-file enter the name of the m-file without the extension in the command window

echo cause m-files to be viewed as they execute

what lists all m-files on your computer

type <name> lists the contents of <name>.m

2.3 Matrices, Vectors and Scalars

Explicitly defining matrices:

A = [3.5] ;   suppresses printing of matrix

B = [1.5, 3.1] ;

C = [-1, 0, 0 ; 1, 1, 0 ; 1, -1, 0 ; 0, 0, 2]

end of row

can also do

```
C = [-1, 0, 0
      1, 1, 0
      1, -1, 0
      0, 0, 2] ;
```

<u>continue</u> for large matrices

F = [1, 52, 64, 197, 42, -42, 55, 82, 22, 109]

or F = [1, 52, 64, 197, 42, -42, ...   indicates continue on next line

55, 82, 22, 109] ;

using other matrices

B = [1.5 3.1]

S = [3.0 B]

gives S = [3.0 1.5 3.1].

S(2) references the 1.5

<u>All MATLAB subscripts begin with 1.</u>

Saving/loading matrices

easiest for images

save data1 x y ;   saves matrices x and y in binary format

load data1 ;   restores matrices

can also read/write ASCII files

save data1.dat z /ascii ;   matrix row by row colon operator

- when used in a matrix it represents all the rows or all the columns.

x = data1 (: , 1) ;

all rows column 1

data1 = (0 , 0
.01 , .1255
.02 , .2507);

y= data1 (:, 2);

new matrices all rows column 2.

x & y will be column vectors.

- can also be used to generate numbers.

H = 1:8 generates [1, 2, 3, 4, 5, 6, 7, 8]

TIME = 0.0:0.5 : 5.0 generates numbers from 0.0 to 5.0 in increments of 0.5

- can be used to select submatrices

$$C = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

c_1 = C (:,2:3); where. $c.1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 2 \end{bmatrix}$

all the rows and column

columns 2 to 3.

c_2= C (3:4,1:2); $c\_2 = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$

rows 3 & 4 columns 1 & 2.

Simple program

```
% Powers of a complex number
clear, clg                       % clears all variables & graphics
j = sqrt(-1)                     % define j
z1 = 1.1*exp(j*2*pi/16);         % assign complex points z1 & z2
z2 = 0.9*exp(j*2*pi/16);

z1powers = z1.^[1:32];   % raises point z1 to powers 1 thru 32
         element by element % i.e. z1powers = [z1**1 z1**2 ... z1**32]
x = [1:32]  % creates vector x = [1, 2, ... 32].
z2powers = z2.^x   % creates z2**1 z2**2, etc.

axis('normal')   % (opt) 1:1 plot aspect ratio
plot (z1powers,'or')  % plots each point of z1powers w/red circles
hold on                  % put more stuff on same plot
plot (z2powers,'.g')  % plots each point of z2powers w/green dot
grid                     % put a grid on graph.
hold off
```

The dot operator is for element by element operations

for example

```
>> A.*B   %computes AB_ij = a_ij b_ij
>> A_2 = A.^2   %squares each element of A
>> A^2          % will compute A*A.
>> 2.^A   raises 2 to a matrix power.
>> 2_.^A  raises 2 to the power of each element in A.
```

if $A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

$$2.\hat{}A = 1.0\times10^4 \begin{bmatrix} .7962 & 1.8029 & 2.8097 \\ .9782 & 2.2154 & 3.4523 \\ 1.1603 & 2.6276 & 4.0950 \end{bmatrix}$$

$$2\_.\hat{}A = \begin{bmatrix} 2 & 16 & 128 \\ 4 & 32 & 256 \\ 8 & 64 & 512 \end{bmatrix}.$$

Images in MATLAB.

indexed images — uses color map.

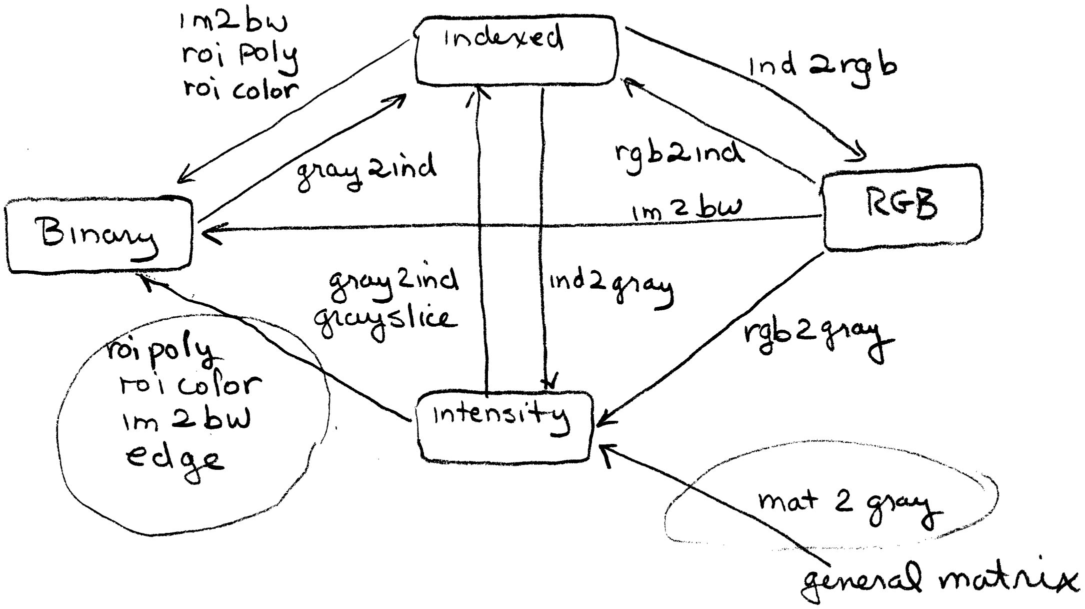
COLOR = [R G B]
n × 3 matrix
for an image containing n colors.

intensity images — what we will use
double precision 0 (black) to 1 (white)

binary images — 0 (black), 1 (white).

RGB images — scanners, etc.
uses three separate matrices

image deck - similar to MRI image slice.



Reading & writing images

GIF. (Graphics Interchange Format)

```
[x, map] = gifread ('img.gif');
```

indexed image X

with associated colormap map.

```
gifwrite ( X, map, '<filename>');
```

TIFF (tagged image file format).

```
[r, g, b] = tiffread ('rgb, tif')
```

returns R,G,B for rgb image.
returns image & color map for indexed file

```
type = tiffread ('<filename>')
```

1 = binary
8 = indexed image
24 = RGB

```
tiffwrite (X, map, '<filename>');
```

Can also do

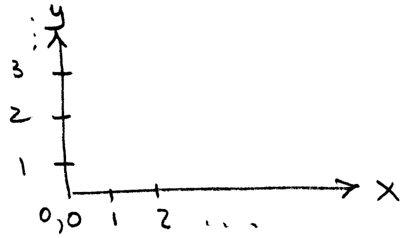
HDF	
BMP	MS windows.
PCX	ZSoft Paint
XWD	(X-windows).

all work with indexed image matrices & colormaps.

Coordinates:

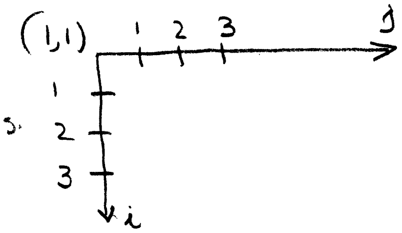
<u>cartesian</u>

(matlab graphics routines).



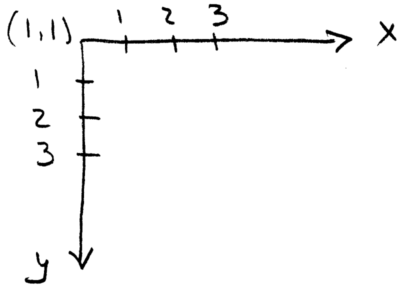
<u>matrix</u>

matrix subscripts.



<u>pixel</u>

coordinate system used by image processing for almost everything



imshow - display image.

imshow (X, map)	indexed images
imshow (I, 64)	display intensity image I with 64 gray levels.
imshow (BW, 2)	binary images.
imshow (~BW, 2)	display inverted image
imshow (R,G,B)	

simple program

```
load kids
subplot (1,2,1), imshow (X, map), title ('Before Rotation')
subplot (1,2,2), imshow (imrotate (X, 35, 'crop'), map)
title ('After Rotation')                optional
```

subplot (m, n, 1) — makes first subarea active

m, n: divide graphics window into m x n subareas.

B = imrotate (A, angle) — rotates by angle in ccw direction

B = imrotate (A, angle, 'method')

B = imrotate (A, angle, 'method', 'crop')

which we will talk about

method

nearest	nearest neighbor interpolation
bilinear	bilinear interpolation
bicubic	bicubic interpolation.

'crop' rotates but only returns central valid section which is same size as A.

imrotate (A, angle, ...)   displays rotated image in current figure

```
load tire
Y = imrotate (X, 135, 'crop')
imshow (Y, map).
```

to read in an image.

load forest ← typically stored as X.

I = ind2gray(X, map) %convert to intensity

imhist (I, n) ← plot histogram.

↑ # of bins

B = imresize (I, [mrows, ncols], 'method')

nearest
bilinear
bicubic

B = imrotate (I, angle)

imshow (I, n)

↑ default is 256.

B = mfilter2 (h, A, filtmask).

output.

↑ 2D filter

0's and 1's to mark where.