

## Useful MATLAB Tips

### (1) File etiquette – remember to fclose(f)

```
f=fopen('filename');
a = fread(...);    or a=fwrite(...);
fclose(f);
```

*How big is a?*

size(a) will give rows/columns or all dimensions if a has more than two  
 size(a,1) will give number of rows  
 length(a) will give you the length of a vector... but be careful using this on arrays!

*File size guessing*

```
a=fread(f, [1 inf], 'uint8');    although the type may not be uint8 necessarily
size(a)
```

*Then use functions like mod, factor, remainder, etc. In conjunction with find perhaps...*

*Type conversions, there are too many to list, but a couple are:*

```
b = double(a);    convert to 64 bit floating point
d = uint8(c);     convert to 8 bit bytes
```

*Other useful commands: help iofun*

Including: fscanf, fprintf	for formatted data
sscanf, sprintf	for string data
imread, imwrite	for image files
import	sometimes useful

### (2) Matlab help commands

help function or help category works best when you know function or category  
 lookfor something will search function descriptions for "something"  
 who displays your variables and their types/sizes  
 why just because

### (3) Making \*.tif files

maketiff and remember to use \*.tif extension.

*Tiffs can be viewed with most graphics software and some text editors on your PC.  
 Tiffs can be viewed with xv or kview, or others on Unix/Linux.  
 The imwrite function can also be used directly but maketiff on the webpage is simplest.*

(4) Plotting

`pix = pix'`;      *the apostrophe transposes, files are often read in "sideways"*  
`flipud` or `fliplr`      *will also flip arrays up/down and left/right*

`figure`;      *pops up a new figure window*  
`figure(3)`;      *command line toggle control of figure 3, same as clicking to bring to front*  
`hold on`;      *holds current plot on, allows overlap plotting*  
`hold off`;      *turns off the hold, next plot will overwrite current plot*

`plot(x,y)`  
`xlabel('x axis label here')` and `yaxis('y axis label here')` and `title('Title')` are all important

`subplot(2,1,1)`, `plot(x1, y1)`; `subplot(2,2,2)`, `plot(x2,y2)`; *makes two plots on the same figure*

*`subplot(r,c,n)` takes  $r$ =plots in a row down,  $c$ =plots in a column across, and  $n=1$  through  $rx$  where  $n$  counts across first and down second:*

*Figure 1: Six subplots,  $r=3, c=2, n=1:6$*

<code>subplot(3,2,1)</code>	<code>subplot(3,2,2)</code>
<code>subplot(3,2,3)</code>	<code>subplot(3,2,4)</code>
<code>subplot(3,2,5)</code>	<code>subplot(3,2,6)</code>

`imagesc(pix)`;      *scaled image*  
`axis equal` and `axis auto` and `axis tight` are all useful, `axis image` makes square pixels  
`axis([xbot xtop ybot ytop])` sets specifics  
`axis([0 10 0 5])` makes axes from 0 to 10 and 0 to 5

`clf`      *clears the current figure without closing it*  
`close all`      *closes all figures but Matlab stays open*  
`gtext('My nose is here')`      *will drop the text 'My nose is here' where you click on the figure*  
`plot(x,y, 'ro')`      *will plot red circles at each data point*  
`help plot`      *for more info on line types*

(5) Array indexing and "skip" or "stride" indexing*Assigning*

`x = 1:10`;      *will give  $x=1,2,3,4,5,6,7,8,9,10$*   
`x = 1:2:10`;      *will give  $x=1,3,5,7,9$*   
`x = 2:2:10`;      *will give  $x=2,4,6,8,10$*   
`x = 1:3:10`;      *will give  $x=1,4,7, 10$  (and so on)*

*In an existing named array*

`x = a(:,112)`;      *makes x a vector of all the rows in column 112 of a, the : means "all" here*  
`x = a(6:end, :)`;      *makes x a matrix the same as a but w/o the first five rows, end goes to end!*

*For zooming or cropping*  
`a = ones(1000, 1000);`  
`b = zeros(100, 100);`  
`a(1:100, 1:100) = b;`  
`a(1:10:100, 1:10:100) = b;`

*Commas separate columns, semicolons separate rows.*  
*Commas can also be spaces in an array (but not in a function call).*  
`a = [11, 12; 21, 22];` and `a = [11 12; 21 22]` are both

11	12
21	22

## (6) Functions and Operators

*For rotating, angles must be in radians to use sin and cos unless you use sind or cosd*  
`rotmat = [cos(ang_rad) sin(ang_rad); -sin(ang_rad) cos(ang_rad)];`

*Apply rotmat to the locations of each pixel. For pix(i,j) its new location is going to be m,n. Try your code on an example matrix first. How much room will you need? How to make m,n be integers? Where on the new array to start placing the new m,n? Do you need an offset?*

`fix`        *round towards zero*  
`floor`     *round towards minus infinity (aka round down)*  
`ceil`      *round towards plus infinity (aka round up)*  
`round`    *round towards nearest integer (the halfway condition)*

*Operators help ops*

`*` *matrix multiply, i.e. c = a\*b*  
`a = [ 1 2; 3 4]` and `b = [1 2; 3 4]` gives `c = [7 10; 15 22]`

`.*` *element wise multiply, i.e. c = a.\*b “dot times”*  
`a = [ 1 2; 3 4]` and `b = [1 2; 3 4]` gives `c = [1 4; 9 16]`

*Same with “divide” / and “dot-divide” ./ and “power” ^ and “dot power” .^*  
`a = [ 1 2 3]` .^2 = `[ 1 4 9]` and if you forgot the dot, it would be an error!

*Relational operators i.e., if(a ~= b) or find(a == 0)*

`==`        *is equal*  
`~=`        *is not equal*  
`>`        *greater than, < less than, <= less than or equal to, >= greater than or equal to*

*Logical operators, i.e., 1&1 = 1 and 1&0 = 0*  
`&` and, `|` or (pipe), `~` not, `xor`, exclusive or

*Bit operators*

bitand, bitcmp, bitor, bitxor, bitset, etc.

*Other elementary Matrix ops help elmat*zeros *all zeros*ones *all ones*repmat *useful replicating matrices, but tricky to use*linspace *makes evenly spaced line or you specify spacing, similar to logspace*meshgrid *connects arrays of x and y values*inf *infinity, can use isinf() to test for inf*NaN *not a number, can us isnan() to test for NaN*abs *takes absolute value*pi *is 3.14159...*cosd and sind take arguments in degrees but still be careful(7) Loopity loop and other language constructs*It helps immensely to put a % descriptive comment with each end statement. A couple of examples are included below. Good coding habit... keep track of those ends!*

```

if(expression)
elseif(expression)
else(expression)
end % testing for expression

```

```

for i=1:10
    y = sin(i);
    x = cos(i);
    plot(x,y);
end % loop to put away rotated row indices

```

```

while(expression)
end

```

*Also useful in loops:*break *exit from loop with no action*pause *stop until user input*input *(user input)*disp *display text in command window*sprintf *print a string*clear all *eeek! Careful! Clears all variables in workspace*clc *just clears up the screen, leaves variables and figures alone*% *comment lead character (there is no end comment character)*

(8) Unix Commands – some good in Matlab, some require ! in front of them to execute from Matlab

kinit -t    login to your Leland/AFS account  
 cd         change directory, cd .. goes up a level  
 pwd        present working directory  
 ls -al     list all files and sizes/access  
 more filename    displays content of a text file to screen, spacebar scrolls on pause  
 mkdir dirname    make directory dirname  
 man command    shows help page for command (not necessarily readable... but still)  
 rm filename     remove a file, as in, "delete"  
 rm -r directoryname    removes a directory, from top to bottom  
 mv filename newlocation/name    moves a file to a new place, can be used to rename  
 cp filename newlocation/name    copies a file to a new location, note . means same name  
 cp -r filename newlocation/name copies a directory  
 ssh         creates a secure connection to an AFS machine

(9) Bug checking

Are you in the right directory?  
 Is the file name typed correctly?  
 Did you remember to fclose(f)?  
 Are you saving the file with the right extension?  
 Are things you're assigning to each other the same size?  
 Have you overwritten an existing Matlab function with a variable name? (Type who)  
 Have you overwritten one of your own variables?  
 Which figure is the active one you're plotting in?  
 Hold on / hold off working or not?  
 Check your indexes!  
 Are your angles/functions in degrees or radians?  
 Did you colormap(gray)?  
 Did you axis equal?  
 Do you have your rows and columns right or are they switched?