MATLAB mini-tutorial

(please go through a more thorough tutorial online)

1. Matlab GUI: Become familiarized with the GUI.

2. Variables: don't have to explicitly declare type, as it can morph into any type (actually, pretty much everything in Matlab is a matrix).

```
EDU>> x=1
x =
     1

EDU>> x=[1, 2, 3]
x =
     1     2     3

EDU>> x = [1, 2; 3, 4]
x =
     1     2
     3     4
```

3. Indexing: Accessing specific parts of the matrix is easy

```
EDU>> x(1,2)
ans =
     2

EDU>> x(:,2)
ans =
     2
     4
```

4. Colon operator: As seen in the previous example, ":" means "get all indices in this dimension." We can be even more fancy, e.g.

- 1:10 -- get first ten indices
- 2:end -- get indices from position 2 all the way to the end
- 1:2:end -- get indices from start to end, in intervals of two
- end:-1:1 -- gets indices in reverse order

5. Some useful commands for getting info about your variables:

- "whos" lists all instantiated variables
- size(x) returns a vector containing the size of matrix x

6. Arithmetic:

```
EDU>> y = [-1,0;2,3]
y =
    -1     0
     2     3

EDU>> x + y
ans =
     0     2
     5     7

EDU>> x * y   (this is matrix multiplication)
ans =
     3     6
     5    12

EDU>> x .* y  (notice "." means element-wise operation)
ans =          ("." Is also needed when performing arithmetic
    -1     0    operation between a scalar and a matrix)
     6    12
```

7. Other useful operations:

- exp, log – exponential, and logarithmic function with base e
- x.^b raises x (element-wise) to a power of b
- x^b raises x (matrix-wise) to a power of b

8. Logical operators

- &, |, ~ -- element-wise AND, OR, and NOT
- Also: <, <=, >, >=, ==, ~= (last expression is "not equal")

9. "Find" command (returns indices that match the logical query):

```
EDU>> a = [10 4 6 -1 3]
a =
    10     4     6    -1     3

EDU>> find(a==4)
ans =
     2

EDU>> find(a < 5)
ans =
     2     4     5

EDU>> find(a < 5, 2)
ans =
     2     4
```

10. "Sort" command (sorts vector)

```
EDU>> [aSorted, aIndx] = sort(a)
aSorted =
    -1     3     4     6    10
aIndx =
     4     5     2     3     1
```

11. How to pre-allocate space:

- `x = zeros(M,N)` creates a zero M x N matrix
- `x = ones(M,N)` creates an MxN matrix filled with ones
- `x = rand(M,N)` creates matrix filled with rand uniform

12. For-loops: (while loops are also available in Matlab)

```
EDU>> for i=1:5
          c(i) = i^2;   (note semicolon suppresses output)
      end
EDU>> c
c =
     1     4     9    16    25
```

13. Plotting example, using "plot" (other commands: "hist", "bar", etc):

```
EDU>> x = 0:0.1:2;
EDU>> y = exp(x);
EDU>> figure
EDU>> plot(x,y,'r-*')
EDU>> hold on;
EDU>> z = x.^2;
EDU>> plot(x,z,'b-^')
EDU>> xlabel('some measurement')
EDU>> ylabel('some f(x)')
EDU>> legend('exp(x)','x squared')
```

14. For more information, use "help" command, e.g. `help plot`

15. You can also create "batch scripts" by putting all interactive commands in an .m file and running it in the console. Furthermore, you can create "functions" in a separate .m file -- use the following signature:

```
function [out1, out2] = myfun(in1, in2, in3)
% Here's my function (% allows you to comment)
```

16. Saving / loading variables:

- `save file.mat a b x y`
- `load file.mat`
- `load file.txt` (file.txt: numerical matrix form)