

Lecture 19

An introduction to Matlab

What is Matlab?

- MATLAB, short for "matrix laboratory", refers to both the numerical computing environment and to its core programming language.
- MATLAB allows one to easily manipulate matrices, plot functions and data, implement algorithms, create user interfaces, and interface with programs in other languages.
- It is used by more than one million people in industry and academia and runs on most modern operating systems, including Windows, Mac OS, Linux and Unix.
 - From <http://en.wikipedia.org/wiki/Matlab>

How does it work?

- It is an interactive language, interpreted on the fly.
 - (There is a compiler, but that isn't the normal use)
- It is, at its heart, a matrix manipulation tool.
 - It has a gob of built in functions to do everything from multiply matrices to reading Microsoft excel files.
 - It is *outstanding* at generating plots and graphs.
- It supports many of the features we've been working with
 - Loops, if statements, structures, etc.
 - But they are often significantly different in actual use.

ex0.m: Array generation

```
a=2;
b=[1 3 5];
x=(1:4);
y=[a b x]
x=(5:-2:-4)
bob=(x)'
linda=bob(2)
```

Basic array construction

- $[x_1, x_2, \dots, x_n]$
 - Concatenates the elements as columns
 - Any of the x 's can be arrays, but if so, they must *all* have the same number of rows.
- $(z:y)$
 - Make an array of values (with 1 row)
 - So $(1:4)$ is $[1\ 2\ 3\ 4]$
- $(z:a:y)$
 - As above, but count by a .
 - So $(1:2:6)$ is $[1\ 3\ 5]$

More arrays

- $(x)'$ is the transpose of x .
 - So rows become columns
 - (This is actually the complex transpose!)
- $[1;2;3;4]$
 - This is the transpose of $[1\ 2\ 3\ 4]$

ex1.m: Array manipulation

```
base=rand(1,4);
real=base*6;
d6=ceil(real)
roll=sum(d6)-min(d6)
```

building

- `rand()` is a built in “matrix builder” function. Others include:
 - `ones()` – all ones
 - `zeros()` – all zeros
 - `randn()` – random numbers with zero mean and unit variance
 - `eye()` – identity matrix

Matrix-scalar

- Matrix scalar operations in general involve the operation being applied to each element.
 - So if $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $b=3$
 - $A+b = \begin{bmatrix} 4 & 5 \\ 6 & 7 \end{bmatrix}$
- Some functions, like `ceil()`, are scalar operations that apply to each element.
- `sum()` and `min()` are functions that work on one dimension of an array
 - In general they work on *columns*. But if there is only one row, will operate on rows.

Scalar-scalar

- Just like C++.
 - `sum(d6) - min(d6)`
 - Both are scalars.

ex2.m: more array manipulation

```
base=rand(4,100000);
real=base*6;
d6=ceil(real);
roll=sum(d6)-min(d6);
bins=(3:18);
hist(roll,bins)
rr=(roll);
mean(rr)
median(rr)
```

- `base=rand(4,100000);`
 - Big array (and a good time not to forget the “;”)
 - Changed so 4 rows, 100,000 columns instead of 1 row, 4 columns.
 - Did this so `sum()` and `min()` would work as they “prefer” to operate on columns
- Notice the next 3 lines are unchanged!

me.m: Some real data manipulation

- Next, want to graph it.
 - Each graphing function is a bit different.
 - `hist(x,y)` graphs a histogram of data in `x` using bins in the array `y`.
- Finally, compute mean and median.
 - Note, using rows now as only 1 row!

```
biglist=xlsread('exam1scores.xls');
section=biglist(:,1);
scores=biglist(:,7);
one=(101:109);
two=(201:209);
both=[one two];
i=0;
for x=both
    i=i+1;
    idx=find(section==x);
    list=scores(idx);
    a(i)=mean(list);
end
```