Math 160 Spring 2010, Lowman, Week 15 Monday
This is a listing of an Octave session (Matlab was used in the classroom) used in Monday's lecture.

Note: Octave is a Matlab clone and is a free downlowd for Windows, Macs and Linux.
Matlab is available in the campus PC-Labs. Both Octave and Matlab were used in lectures for linear algebra. It is assumed that students can repeat the following on Octave or Matlab.

Note: In Octave, comment lines begin with # symbol and in Matlab comment lines begin with the % symbol. If you use Matlab replace the #'s with %'s

```
octave-3.0.5:19> # math160s10 Lowman, w15L1
octave-3.0.5:19> #Markov process problem
octave-3.0.5:19> #Rusty Rent-O-Car has offices in NYC and LA. Customers can make local rentals or one-way rentals to
octave-3.0.5:19> #the other location. Each month: 1/2 of the cars that start the month in NYC end in LA ans 1/3 of the cars
octave-3.0.5:19> #that start the month in LA end up in NYC. At the start of the operation Tusty has 1000 cars in each city.
octave-3.0.5:19> #
octave-3.0.5:19> #Find the state matrix (distributin matrix) and monthly transition matrix.
octave-3.0.5:19> #Find Xs the stable state and As the stable matrix.
octave-3.0.5:19> #Find the state matrix, and the number of cars in each city for different months.
octave-3.0.5:19> #Find the transition matrix that takes a state from month 0 to month n
octave-3.0.5:19> #
octave-3.0.5:19> n = 2000
n =  2000
octave-3.0.5:20> #x0 the initial state contains the fraction of cars in each city
octave-3.0.5:20> x0 = [1/2;1/2]
x0 =

   0.50000
   0.50000

octave-3.0.5:21> #n0 = x0*n gives the number of cars in each city at month 0.
octave-3.0.5:21> n0 = x0 * n
n0 =

   1000
   1000

octave-3.0.5:22> # a the transition matrix contains the fracion of cars that start in one city and end in another city
octave-3.0.5:22> # for any month. Here the columns and rows are labeled N=R1 and C1 and L=R2 and C2
octave-3.0.5:22> a = [1/2 1/3; 1/2 2/3]
a =

   0.50000   0.33333
   0.50000   0.66667

octave-3.0.5:23> # note: the columns of a and xi must add to one and be non-negative.
octave-3.0.5:23> # x1 = a * x0 is the state matrix at the end of month 1
octave-3.0.5:23> x1 = a * x0
x1 =
```

```
   0.41667
   0.58333

octave-3.0.5:24> # x1 gives the distribution (fractions) of cars in each city at the end of
month 1
octave-3.0.5:24> # the number of cars in each city at the end of month n=1 is n1 = x1*n
octave-3.0.5:24> n1 = x1*n
n1 =

    833.33
   1166.67

octave-3.0.5:25> # at the end of month n=2
octave-3.0.5:25> x2 = a * x1
x2 =

   0.40278
   0.59722

octave-3.0.5:26> n2 = x2 * n
n2 =

    805.56
   1194.44

octave-3.0.5:27> # a2 = a^2 is the transition matrix from state n=0 to state n=1
octave-3.0.5:27> # x2 = a2 * x0 is an alternate way to find x2
octave-3.0.5:27> a2 = a^2
a2 =

   0.41667   0.38889
   0.58333   0.61111

octave-3.0.5:28> x2 = a2 * x0
x2 =

   0.40278
   0.59722

octave-3.0.5:29> n2 = x2 * n
n2 =

    805.56
   1194.44

octave-3.0.5:30> # x2 gives the fraction of cars in each city at n=2
octave-3.0.5:30> # n2 gives the number of cars in each city at n=2 (round to nearest car).
octave-3.0.5:30> # x3 = a * x3 or use x3 = a^3 * x0,
octave-3.0.5:30> x3 = a * x2
x3 =

   0.40046
   0.59954

octave-3.0.5:31> x3 = a^3 * x0
x3 =
```

```
   0.40046
   0.59954

octave-3.0.5:32> # a3 = a^3 is the transition matrix from state x0 to x3
octave-3.0.5:32> a3 = a^3
a3 =

   0.40278   0.39815
   0.59722   0.60185

octave-3.0.5:33> # eventially the system will reach a stable state xs where the
distribution of cars in each city will
octave-3.0.5:33> # no longer change. The number of cars in each city will stay the same
from month to month.
octave-3.0.5:33> # When this happens an = a^n will also stop changing giving the stable
matrix as.
octave-3.0.5:33> # when the system reaches a stable state xs = a * xs => transition matrix
has no effect,
octave-3.0.5:33> # and as = a * as.
octave-3.0.5:33> # In addition the columns of as are the same as xs.
octave-3.0.5:33> # This gives an easy way to find xs and as.
octave-3.0.5:33> # (1) raise a to a high power
octave-3.0.5:33> # (2) check by increasing the power by 1 to see if there is no change. Try
higher powers until no change.
octave-3.0.5:33> # (3) as = a^(high power) and xs is a column of as.
octave-3.0.5:33> # The number of cars in each city when in a stable state is ns = xs * n
octave-3.0.5:33> #
octave-3.0.5:33> # An alternate method is solve for xs by solving:
octave-3.0.5:33> # a * xs = xs where the sum of xs elements must add to one. This was also
octave-3.0.5:33> # demonstrated in class.
octave-3.0.5:33> #
octave-3.0.5:33> # Try as = a^100 (use a computer or calculator)
octave-3.0.5:33> # as = a^100
octave-3.0.5:33> as = a^100
as =

   0.40000   0.40000
   0.60000   0.60000

octave-3.0.5:34> # now check if the power is high enough
octave-3.0.5:34> a^101
ans =

   0.40000   0.40000
   0.60000   0.60000

octave-3.0.5:35> # no change => stable matrix
octave-3.0.5:35> as
as =

   0.40000   0.40000
   0.60000   0.60000

octave-3.0.5:36> xs = as(:,1)  # all rows of col 1
xs =

   0.40000
   0.60000
```

```
octave-3.0.5:37> nx = xs * n
nx =

    800.00
   1200.00

octave-3.0.5:38> # the system stablizes with 800 cars in NYC and 1200 in LA for every month
after stabilization.
octave-3.0.5:38> # The only question left is how long does it take for the system to
stablize.
octave-3.0.5:38> #
octave-3.0.5:38> i=0, ai = a^i, xi=ai*x0, ni = xi * n
i = 0
ai =

   1   0
   0   1

xi =

   0.50000
   0.50000

ni =

   1000
   1000

octave-3.0.5:39> i=1, ai = a^i, xi=ai*x0, ni = xi * n
i =  1
ai =

   0.50000   0.33333
   0.50000   0.66667

xi =

   0.41667
   0.58333

ni =

    833.33
   1166.67

octave-3.0.5:40> i=2, ai = a^i, xi=ai*x0, ni = xi * n
i =  2
ai =

   0.41667   0.38889
   0.58333   0.61111

xi =

   0.40278
   0.59722
```

```
ni =

    805.56
   1194.44

octave-3.0.5:41> i=3, ai = a^i, xi=ai*x0, ni = xi * n
i =  3
ai =

   0.40278   0.39815
   0.59722   0.60185

xi =

   0.40046
   0.59954

ni =

    800.93
   1199.07

octave-3.0.5:42> i=4, ai = a^i, xi=ai*x0, ni = xi * n
i =  4
ai =

   0.40046   0.39969
   0.59954   0.60031

xi =

   0.40008
   0.59992

ni =

    800.15
   1199.85

octave-3.0.5:43> # observe the system is almost stable after only 4 months
octave-3.0.5:43> i=5, ai = a^i, xi=ai*x0, ni = xi * n
i =  5
ai =

   0.40008   0.39995
   0.59992   0.60005

xi =

   0.40001
   0.59999

ni =

    800.03
   1199.97

octave-3.0.5:44> i=6, ai = a^i, xi=ai*x0, ni = xi * n
```

```
i =  6
ai =

   0.40001   0.39999
   0.59999   0.60001

xi =

   0.40000
   0.60000

ni =

    800.00
   1200.00

octave-3.0.5:45> i=7, ai = a^i, xi=ai*x0, ni = xi * n
i =  7
ai =

   0.40000   0.40000
   0.60000   0.60000

xi =

   0.40000
   0.60000

ni =

    800.00
   1200.00

octave-3.0.5:46> diary
```