Lab 5

Limiting Behaviour of Age-structured Populations

Eigenvalues and eigenvectors

Handout – print version of Lecture on *Marine Modelling* February 04, 2018

Katja Fennel, Oceanography, Dalhousie University

5.1

1 Recap

Recall that the *dominant eigenvalue* (i.e. the eigenvalue with largest absolute magnitude) of Leslie matrix M tells us about the *limiting behavior* of an age-structured population. Specifically, the total population will:

- increase for $\lambda_1 > 1$
- be stable for $\lambda_1 = 1$
- decrease to extinction for $\lambda_1 < 1$

The eigenvalues of M are the *roots of the characteristic polynomial*:

$$p(\lambda) = \det(M - \lambda I)$$

If M has dimensions $n \times n$ there are n *eigenvalues* $\lambda_1, \lambda_2, \cdots, \lambda_n$ (not necessarily distinct) and n *eigenvectors* $v_1, v_2, \cdots v_n$.

The eigenvector corresponding to the dominant eigenvalue (v_1) gives us the *relative age distribution* for large values of time. In other words, the age distribution of the population will approach some scalar multiple of that eigenvector with time.

5.2

2 Examples

Suppose we have three age classes, females in 2nd and 3rd age class produce 4 and 3 female offspring, and suppose that 50% and 25% of females live to second and third age class, respectively. The Leslie matrix is:

$$M = \begin{pmatrix} 0 & 4 & 3 \\ 0.5 & 0 & 0 \\ 0 & 0.25 & 0 \end{pmatrix}$$

Suppose the initial age distribution is:

$$\underline{x}_0 = \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$$

Enter both

5.3

```
>> M =[0 4 3; 0.5 0 0; 0 0.25 0];
>> x0 = [10;10;10];
```

We will follow the population over 10 years and want to hold all population vectors in one array X. Define this array, initialize with zeros, and store initial age distribution in first column.

```
>> X = zeros(3,11);
>> X(:,1) = x0;
```

Now calculate age-distribution over the next 10 years:

```
>> for k=2:11, X(:,k) = M*X(:,k-1); end
```

View the results:

```
>> X
```

5.4

```
>> X


 X =

   1.0e+03 *

 Columns 1 through 6

   0.0100   0.0700   0.0275   0.1437   0.0813   0.2978
   0.0100   0.0050   0.0350   0.0138   0.0719   0.0406
   0.0100   0.0025   0.0013   0.0088   0.0034   0.0180

 Columns 7 through 11

   0.2164   0.6261   0.5445   1.3333   1.3238
   0.1489   0.1082   0.3130   0.2722   0.6667
   0.0102   0.0372   0.0271   0.0783   0.0681
```

5.5

You noted the scale factor 10^3. Let's try a friendlier format:

```
>> format short g
>> X

 X =

 Columns 1 through 6

   10      70     27.5   143.75     81.25    297.81
   10       5       35    13.75    71.875    40.625
   10     2.5     1.25     8.75    3.4375    17.969

   Columns 7 through 11

   216.41    626.09    544.49   1333.3    1323.8
   148.91     108.2    313.05   272.25    666.67
   10.156    37.227    27.051   78.262    68.062
```

5.6

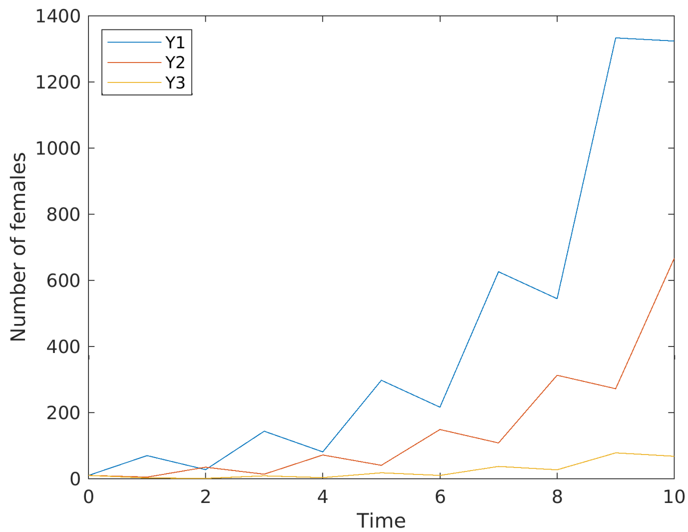
Plot:

```
>> t=0:10;
>> plot(t,X')
>> xlabel('Time')
>> ylabel('Number of females')
>> legend('Y1','Y2','Y3','Location', 'Best')
```

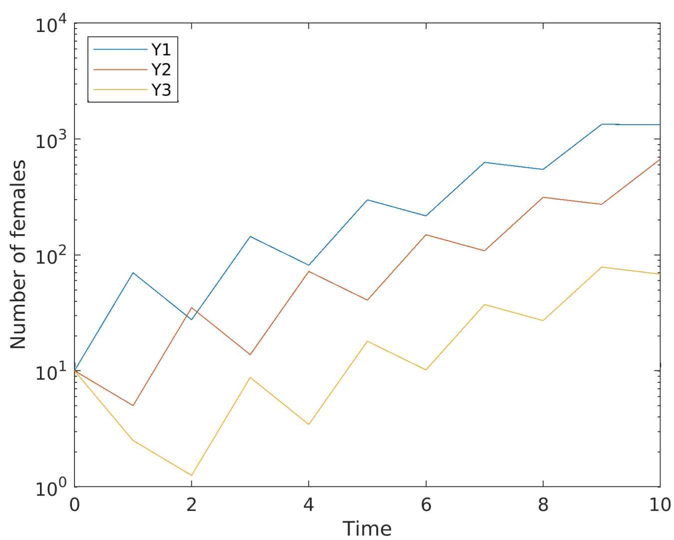
or plot as semi-log plot:

```
>> figure
>> semilogy(t,X')
>> xlabel('Time')
>> ylabel('Number of females')
>> legend('Y1','Y2','Y3','Location', 'Best')
```

5.7



5.8



5.9

Now let's calculate the eigenvalues and eigenvectors of M:

```
>> [V,D] = eig(M);
```

Try `help eig` and read the first two paragraphs. The matrix `V` contains the eigenvectors. The matrix `D` contains the corresponding eigenvalues on it's main diagonal. Let's look at both:

```
>> V
   V =
      0.94737     -0.93201      0.22588
      0.31579        0.356     -0.59137
     0.052632    -0.067989      0.77412

>> D
```

```
D =
       1.5            0            0
         0       -1.309            0
         0            0     -0.19098
```

Let's check that $M = VDV^{-1}$ 5.10

```
>> V*D*inv(V)

ans =

  -1.1839e-16            4            3
          0.5  -8.6606e-16  -8.3267e-17
   -6.245e-17         0.25  -5.5511e-17
```

Is this equal to our Leslie matrix M? 5.11

Our dominant eigenvalue is 1.5 (`D(1,1)`) and the corresponding eigenvector is the first column of V (i.e. `V(:,1)`)

```
>> v1 = V(:,1)

   v1 =

      0.94737
      0.31579
     0.052632
```

Note that Matlab returns normalized eigenvectors by default:

```
>> v1'*v1

   ans =

       1
```

5.12

To get the relative age distribution in the future, we can simply divide each element of `v1` by the sum of all elements in `v1`:

```
>> v1=v1/sum(v1)
   v1 =
         0.72
         0.24
         0.04
```

I.e. 72% of the total population will be in age class 1, 24% in age class 2, and 4% in age class 3. Let's test this by calculating age distribution after 100 years:

```
>> x100=M^100*x0
   x100 =
   1.1555e+19
   3.8516e+18
   6.4194e+17

>> x = x100/sum(x100)
   x =
         0.72
         0.24
         0.04
```

5.13