Ÿ Lab 2: Lists and matrices in Mathematica

Matrix calculations, can be done very easily and conveniently with *Mathematica*. In Mathematics, vectors and matrices are special cases of data lists. In this lab you will learn how to define lists and matrices, and how to manipulate them, e.g. how to apply matrices to vectors, and how to calculate eigenvalues and eigenvectors of matrices. Such calculations are very important for the analysis of dynamical systems.

Vectors in Mathematica are written as a list:

{element 1, element 2, ... , element n}

For exmaple, the list

```
l = 81, 2, 3, 4, 5, 6, 7, 8, 9, 10<
```

is a list of the first 10 integers, and it is also a vector with the ith component equal to i. You don't have to distinguish between row and column vectors in Mathematica, since Mathematica uses the context in order to determine which kind of vector you are using.

Matrices in mathematica are written as a list of lists, more precisely as a list of (row) vectors:

{{row1},{row 2}, ... , {row n}}

For example, a 2x2 matrix is

```
88a, b<, 8c, d<<
```

To be able to do calculations it is useful to give vectors and matrices a name,
e.g. x for a vector:

```
x = 8e, f<
```

and m for a matrix:

```
m = 88a, b<, 8c, d<<
```

You can display vectors and matrices in their typical matrix form by typing

```
MatrixForm@xD
MatrixForm@mD
```

The matrix and vector multiplication that we've been talking about in class (see also homework assignment 2) is technically known as the "dot product" and is represented by a "." For example, you get the vector that is obtained by applying the matrix m to the vector x simply by typing m.x Try it:

```
m.x
```

[Notice that *Mathematica* knew to interpret {e,f} as a a column vector in this calculation.]

Similarly, if you have two matrices you can multiply them together to get a new matrix. For example, try what you get by typing m.m

```
m.m
```

If that's too unwieldy, try

```
MatrixForm@m.mD
```

Ÿ Functions of Matrices in Mathematica

Two very important commands for matrix calculations are

```
Eigenvalues@mD
Eigenvectors@mD
```

These commands allow you to calculate eigenvalues and the corresponding eigenvectors for any matrix m. Try them with the following examples (see also homework assignment 2!):

```
m1 = 88a, b<, 8c, d<<
```

```
m2 = 881, 2<, 80.6, 0<<
```

```
m3 = 881, -3, 3<, 83, -5, 3<, 86, -6, 4<<
```

You can find both the eigenvalues and the eigenvectors of a matrix using Eigensystem[ ]. By the way, If you type

```
Information@"Eigensystem", LongForm ® FalseD
```

you will get information on the command in quotation marks. Apply this command to the matrices m1, m2 and m3.

Now check that you got the right answers: a matrix times its eigenvector equals the eigenvector times its associated eigenvalue (see class notes). Don't forget to use the dot product. For example

```
m3.8-1, 0, 1<
```

```
82, 0, -2<
```

is -2 times the eigenvector {-1,0,1}.

```
m3.81, 1, 0<
```

is also -2 times the eigenvector {1,1,0}. And

```
m3.81, 1, 2<
```

is 4 times the eigenvector {1,1,2}.

Another very useful matrix operation is MatrixPower[matrix,t],
this takes the matrix to the t power.

For example

```
MatrixPower@m2, 3D
```

gives youu the third power of m2. Check that this gives the same answer as:

```
m2.m2.m2
```

If the matrix considered happens to be a Leslie matrix describing the dynamics of a population, then these commands tell you how much change will occur over three generations. For example, consider the matrix m2: it describes the dynamics of a hypothetical population with two age classes (why?). If you start out with a population given by the vector {n1,n2}, where n1 is the size of age class 1 and n2 is the size of age class 2, then

```
MatrixPower@m2, sD.8n1, n2<
```

gives you the population vector after s generations. Choose specific values for n1 and n2 (e.g. n1=2 and n2=3.2} and, using the command above, check out what happens to the population vector as s becomes large.