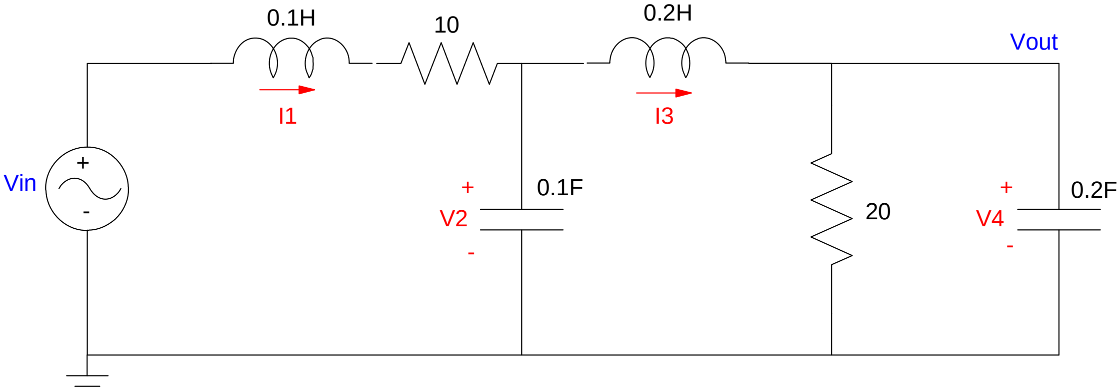
Eigenvalues and Eigenvectors

Eigenvalues and Eigenvectors are important for understanding systems in state-space form. Take for example the 4th-order RC filter:



4th-Order RC Filter from Lecture #3

The state-space model we found was:

$$\begin{bmatrix} sI_1 \\ sV_2 \\ sI_3 \\ sV_4 \end{bmatrix} = \begin{bmatrix} -100 & -10 & 0 & 0 \\ 10 & 0 & -10 & 0 \\ 0 & 5 & 0 & -5 \\ 0 & 0 & 5 & -0.25 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_{in}$$

$$V_{out} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + [0] V_{in}$$

along with the transfer function of

```
                    2500
--------------------------------------------
(s+98.99) (s+0.5025) (s^2 + 0.7525s + 75.38)
```

The states in the system are important: they tell you the energy in the system.

The 'A' matrix is all important: it tells you how the energy moves about the system. If you look at the eigenvalues of A:

```
>> A4 = [-100,-10,0,0;10,0,-10,0;0,5,0,-5;0,0,5,-0.25]

 -100.0000  -10.0000         0         0
   10.0000         0  -10.0000         0
         0    5.0000         0   -5.0000
         0         0    5.0000   -0.2500

>> eig(A4)

 -98.9950
  -0.3763 + 8.6740i
  -0.3763 - 8.6740i
  -0.5025
```

you'll note that **the eigenvalues of A are the poles of the system.** The eigenvalues tell you how the energy decays.

Eigenvalues tell you what behaves that way. If you have a system with an initial condition and no input:

$$sX = AX$$

$$X(0) = X_0$$

then the response of X(t) will be

$$X(t) = a_1\Lambda_1 e^{\lambda_1 t} + a_2\Lambda_2 e^{\lambda 2t} + a_3\Lambda_3 e^{\lambda 3t} + \ldots$$

where

a_i are constants determined by the initial condition,

Λ_i are the eigenvectors of A, and

λ_i are the eigenvalues of A.

At t=0, you get

$$X(0) = a_1\Lambda_1 + a_2\Lambda_2 + a_3\Lambda_3 + \ldots$$

or

$$X_0 = \Lambda A$$

$$A = \Lambda^{-1}X_0$$

In short,

- The eigenvalues tell you how the system behaves,
- The eigenvectors tell you what behaves that way,
- The initial condition tells you how much you excite each eigenmode.

For example, the above system has four eigenvalues. It also has four eigenvectors.

```
>> [a,b] = eig(A)

a =  (eigenvector matrix)

  -0.9950             -0.0705 + 0.0061i  -0.0705 - 0.0061i  -0.0710
   0.1000              0.7075             0.7075              0.7067
  -0.0050             -0.0439 - 0.6076i  -0.0439 + 0.6076i  -0.0355
   0.0003             -0.3498 + 0.0304i  -0.3498 - 0.0304i   0.7031

b =  (eigenvalue matrix)

 -98.9950                   0                  0                  0
        0             -0.3763 + 8.6740i        0                  0
        0                   0            -0.3763 - 8.6740i        0
        0                   0                  0            -0.5025
```

Suppose the initial condition was

$$X_0 = \begin{bmatrix} I_1(0) \\ V_2(0) \\ I_3(0) \\ V_4(0) \end{bmatrix} = \begin{bmatrix} -0.9950 \\ 0.1000 \\ -0.0050 \\ 0.0003 \end{bmatrix}$$

Then the initial condition in terms of eigenmodes is

```
>> X0 = a(:,1)

   -0.9950
    0.1000
   -0.0050
    0.0003

>> inv(a)*X0

  1.0000
  0.0000
  0.0000
  0.0000
```

or you only excite the 1st mode (the one with an eigenvalue at -98.995)

$$X(t) = 1 \cdot \begin{pmatrix} -0.9950 \\ 0.1000 \\ -0.0050 \\ 0.0003 \end{pmatrix} e^{-98.995t} + 0 \cdot \Lambda_2 e^{\lambda_2 t} + 0 \cdot \Lambda_3 e^{\lambda_3 t} + 0 \cdot \Lambda_4 e^{\lambda_4 t}$$

If the initial condition matches the 4th eigenvalue, you only excite that mode:

```
>> X0 = a(:,4)

   -0.0710
    0.7067
   -0.0355
    0.7031

>> inv(a)*X0

   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -0.0000 + 0.0000i
   1.0000 - 0.0000i
```

and

$$X(t) = \begin{pmatrix} -0.0710 \\ 0.7067 \\ -0.0355 \\ 0.7031 \end{pmatrix} e^{-0.5025t}$$

If X0 excites one of the complex modes, X(t)'s behaviour is determined by the complex eigenvalues:

```
>> X0 = real(a(:,2))

   -0.0705
    0.7075
   -0.0439
   -0.3498

>> inv(a)*X0

   0.0000 - 0.0000i
   0.5000 + 0.0000i
   0.5000 - 0.0000i
  -0.0000 + 0.0000i
```

$$X(t) = 0.5\begin{pmatrix} -0.0705 + 0.0061i \\ 0.7075 \\ -0.0439 - 0.6076i \\ -0.3498 + 0.0304i \end{pmatrix} e^{(-0.3763+8.6740i)t} + 0.5\begin{pmatrix} -0.0705 - 0.0061i \\ 0.7075 \\ -0.0439 + 0.6076i \\ -0.3498 - 0.0304i \end{pmatrix} e^{(-0.3763-8.6740i)t}$$

Example 2: 10-Stage RC Filter

Consider next the 10-stage RC filter from lecture #3. The transfer function we found was:

```
                                              10000000000
------------------------------------------------------------------------------------------------------
(s+39.31) (s+36.72) (s+32.67) (s+27.51) (s+21.69) (s+15.75) (s+10.2) (s+5.539) (s+2.181) (s+0.4234)
```

Note that the poles are equal to the eigenvalues of the 10x10 A matrix:

```
A10 =

  -20.2000   10.0000         0         0         0         0         0         0         0         0
   10.0000  -20.2000   10.0000         0         0         0         0         0         0         0
         0   10.0000  -20.2000   10.0000         0         0         0         0         0         0
         0         0   10.0000  -20.2000   10.0000         0         0         0         0         0
         0         0         0   10.0000  -20.2000   10.0000         0         0         0         0
         0         0         0         0   10.0000  -20.2000   10.0000         0         0         0
         0         0         0         0         0   10.0000  -20.2000   10.0000         0         0
         0         0         0         0         0         0   10.0000  -20.2000   10.0000         0
         0         0         0         0         0         0         0   10.0000  -20.2000   10.0000
         0         0         0         0         0         0         0         0   10.0000  -10.2000
```

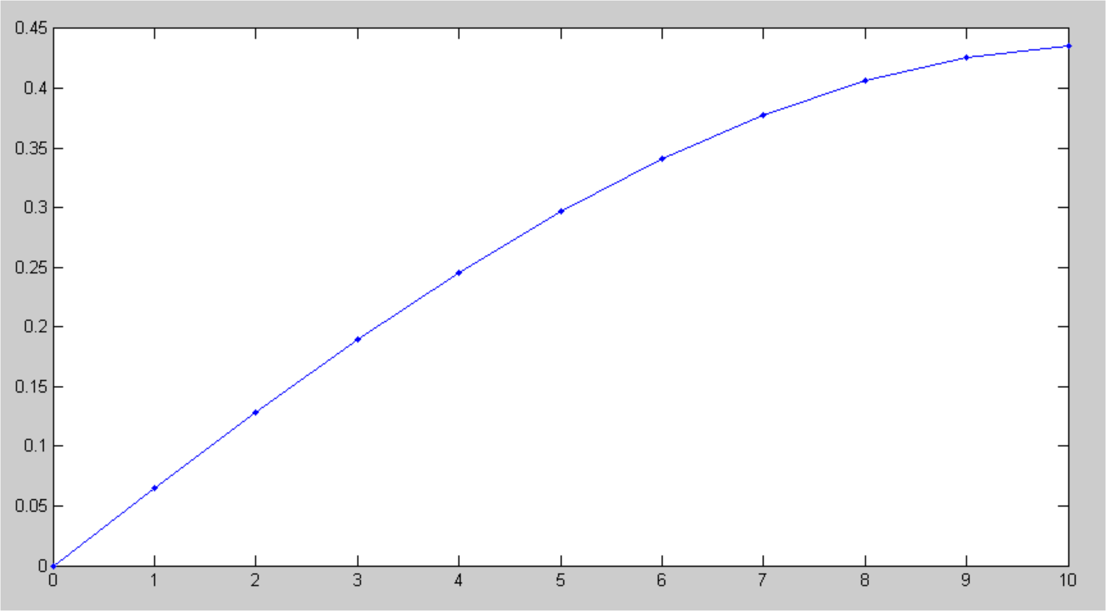
```
>> eig(A10)

  -39.3115
  -36.7248
  -32.6698
  -27.5068
  -21.6946
  -15.7496
  -10.2000
   -5.5390
   -2.1806
   -0.4234
```

Again, eigenvalues are the same as poles: they tell you how the system will behave. The eigenvectors tell you what behaves that way:

```
>> [a,b] = eig(A10)

a =   eigenvector

-0.1286   -0.2459    0.3412    0.4063    0.4352    0.4255    0.3780    0.2969   -0.1894    0.0650
 0.2459    0.4063   -0.4255   -0.2969   -0.0650    0.1894    0.3780    0.4352   -0.3412    0.1286
-0.3412   -0.4255    0.1894   -0.1894   -0.4255   -0.3412   -0.0000    0.3412   -0.4255    0.1894
 0.4063    0.2969    0.1894    0.4352    0.1286   -0.3412   -0.3780    0.0650   -0.4255    0.2459
-0.4352   -0.0650   -0.4255   -0.1286    0.4063    0.1894   -0.3780   -0.2459   -0.3412    0.2969
 0.4255   -0.1894    0.3412   -0.3412   -0.1894    0.4255    0.0000   -0.4255   -0.1894    0.3412
-0.3780    0.3780   -0.0000    0.3780   -0.3780    0.0000    0.3780   -0.3780   -0.0000    0.3780
 0.2969   -0.4352   -0.3412    0.0650    0.2459   -0.4255    0.3780   -0.1286    0.1894    0.4063
-0.1894    0.3412    0.4255   -0.4255    0.3412   -0.1894    0.0000    0.1894    0.3412    0.4255
 0.0650   -0.1286   -0.1894    0.2459   -0.2969    0.3412   -0.3780    0.4063    0.4255    0.4352

diag(b) - eigenvalues

-39.3115  -36.7248  -32.6698  -27.5068  -21.6946  -15.7496  -10.2000   -5.5390   -2.1806   -0.4234
```

Any given initial condition will excite the ten different modes (eigenvectors). If you want to excite just the slowest mode, the initial conidtion should look like its eigenvector.

```
>> Node = [0:10]';
>> plot(Node, [0; a(:,10)], '.-');
```



Mode Shape for the Eigenvalue At -0.5025

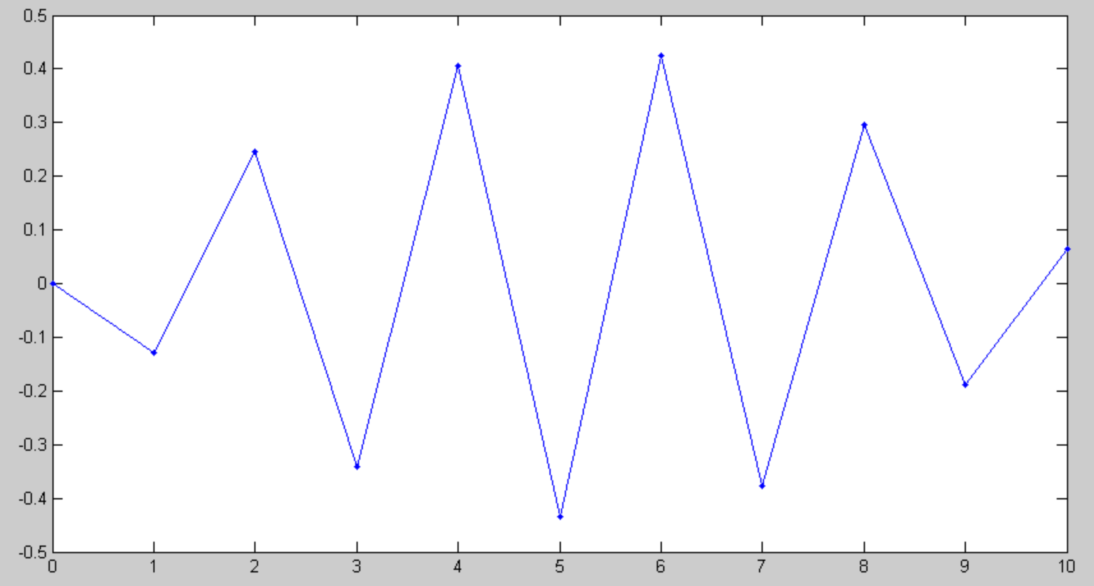
If this is your initial conidtion, the shape will remain the same (the eigenvector remains unchanged) but the amplitude decays as

$$X(t) = \Lambda_{10} e^{-0.4234t}$$

Since this is the dominant mode, if you simulate this system the fast modes should die out quickly leaving the dominant mode (this one).

If you excite the fast mode (the first column), then the shape of the voltages will remain the same (defined by the eigenvector) and decay as

$$X(t) = \Lambda_1 e^{-39.31t}$$



Initial Condition to Excite the Fastest Mode (eigenvalue = -39.31)

If you give the system some other initial condition, such as

- V(i) = 1   i <= 5
- V(i) = 0   i > 5

then all 10 modes will be excited as

```
>> X0 = [1;1;1;1;1;0;0;0;0;0]

     1
     1
     1
     1
     1
     0
     0
     0
     0
     0

>> inv(a)*X0

   -0.2530         initial condition for the fast mode ( pole = -39.31 )
   -0.0333
   -0.1310
    0.2266
    0.4796
    0.1218
   -0.0000
    0.8925
   -1.7228
    0.9258         initial conidtion for the slow mode ( pole = -0.4234 )
```