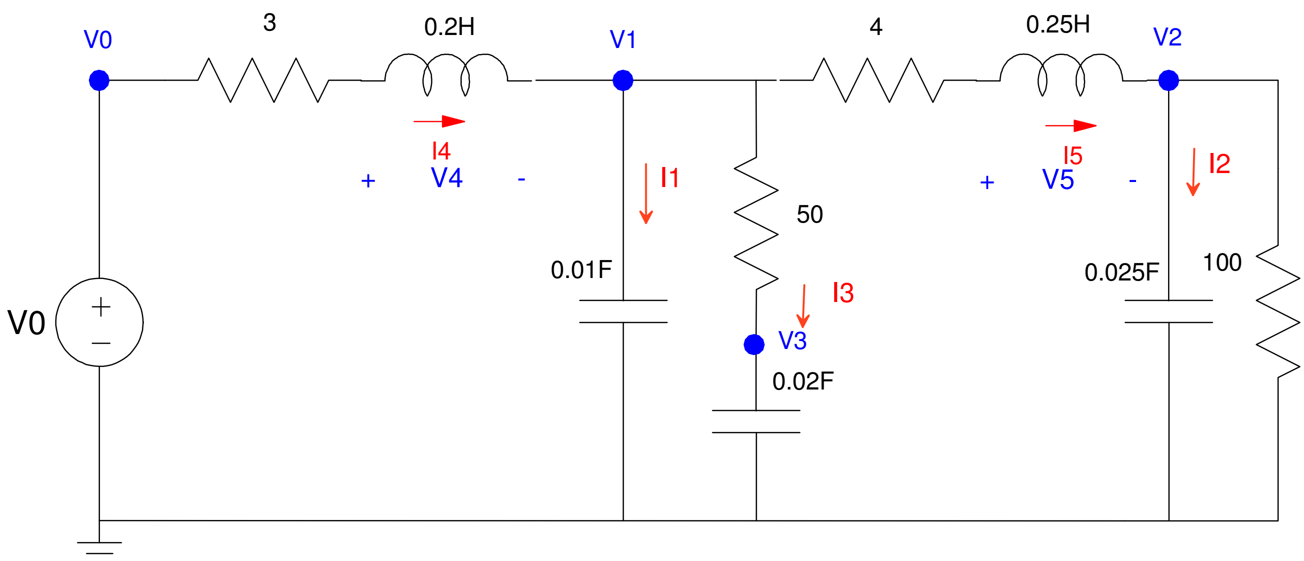
ECE 463/663 - Homework #2

State-Space, Eigenvalues, Eigenvectors. Due Monday, Jan 22nd

1) For the following RLC circuit

- Specify the dynamics for the system (write N coupled differential equations)
- Express these dynamics in state-space form
- Determine the transfer function from V0 to V2



Write the dynamics:

$$I_1 = 0.01sV_1 = I_4 - I_5 - \left(\frac{V_1 - V_3}{50}\right)$$

$$I_2 = 0.025sV_2 = I_5 - \left(\frac{V_2}{100}\right)$$

$$I_3 = 0.02sV_3 = \left(\frac{V_1 - V_3}{50}\right)$$

$$V_4 = 0.2sI_4 = V_0 - 3I_4 - V_1$$

$$V_5 = 0.25sI_5 = V_1 - 4I_5 - V_2$$

Solve for the derivative and group terms

$$sV_1 = 100I_4 - 100I_5 - 2V_1 + 2V_3$$

$$sV_2 = 40I_5 - 0.4V_2$$

$$sV_3 = V_1 - V_3$$

$$sI_4 = 5V_0 - 15I_4 - 5V_1$$

$$sI_5 = 4V_1 - 16I_5 - 4V_2$$

Express in state-space form

$$\begin{bmatrix} sV_1 \\ sV_2 \\ sV_3 \\ sI_4 \\ sI_5 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 2 & 100 & -100 \\ 0 & -0.4 & 0 & 0 & 40 \\ 1 & 0 & -1 & 0 & 0 \\ -5 & 0 & 0 & -15 & 0 \\ 4 & -4 & 0 & 0 & -16 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_4 \\ I_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 5 \\ 0 \end{bmatrix} V_0$$

$$Y = V_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_4 \\ I_5 \end{bmatrix}$$

Throw into Matlab

```
>> A = [-2,0,2,100,-100;0,-0.4,0,0,40;1,0,-1,0,0];
>> A = [A;-5,0,0,-15,0;4,-4,0,0,-16]

   -2.0000         0    2.0000  100.0000 -100.0000
         0   -0.4000         0         0   40.0000
    1.0000         0   -1.0000         0         0
   -5.0000         0         0  -15.0000         0
    4.0000   -4.0000         0         0  -16.0000

>> B = [0;0;0;5;0]

     0
     0
     0
     5
     0

>> C = [0,1,0,0,0];
>> D = 0;
>> G = ss(A,B,C,D);
>> zpk(G)

                     80000 (s+1)
-------------------------------------------------------
(s+0.9432) (s^2 + 16.04s + 90.27) (s^2 + 17.42s + 1005)

>>
```

2) For the transfer function from V0 to V2

- Determine a 1st or 2nd-order approximation for this trasfer function
- Plot the step response of the actual 5th-order system and its approximation

Keep the dominant pole: (s + 09432)

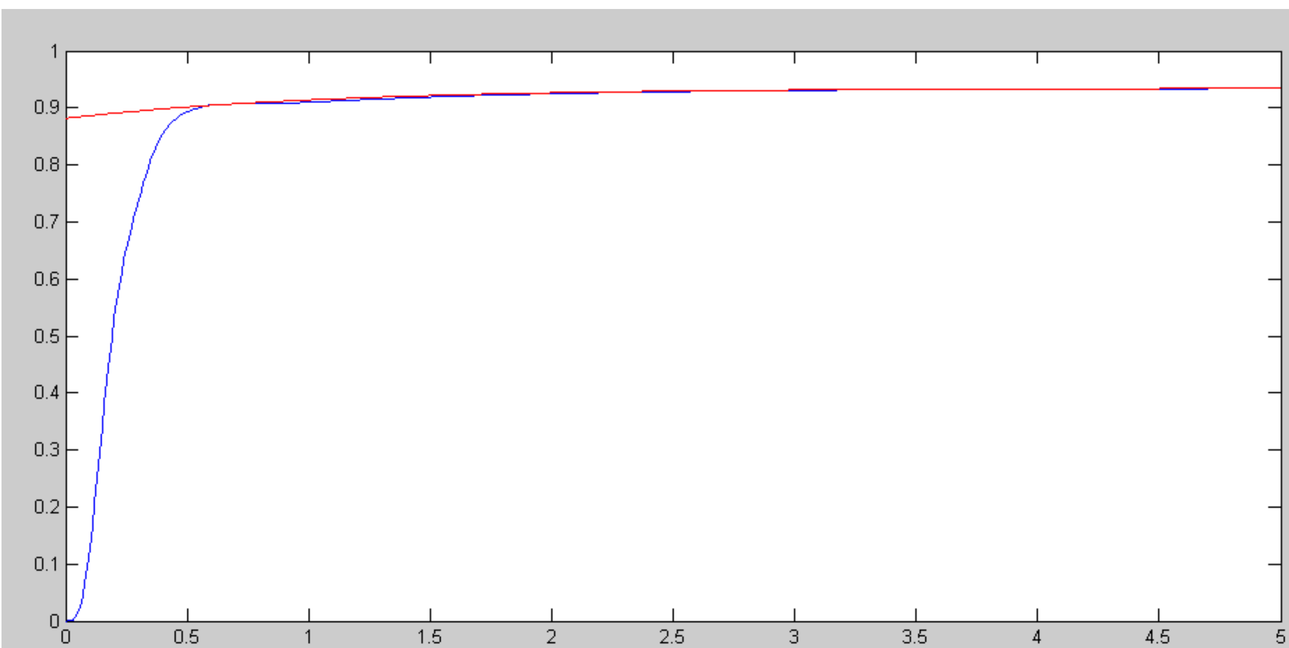
Keep the zero: (s+1)

Match the DC gain

```
>> G1 = zpk(-1,-0.9432,1)

   (s+1)
----------
(s+0.9432)

>> k = evalfr(G,0) / evalfr(G1,0)

    0.8815

>> G1 = zpk(-1,-0.9432,k)
```

```
0.8815 (s+1)
------------
 (s+0.9432)
```

```
>> t = [0:0.01:5]';
>> y5 = step(G,t);
>> y1 = step(G1,t);
>> plot(t,y5,'b',t,y1,'r')
```



5th-Order System (blue) & 1st-Order Approximation (red)

3) For this circuit

What initial condition will the energy in the system decay as slowly as possible?

What initial condition will the energy in the system decay as fast as possible?

```
 0.9591              0.9591              0.3961 + 0.0100i   0.3961 - 0.0100i    0.0566
-0.1846 + 0.0069i  -0.1846 - 0.0069i    0.8655             0.8655              0.0597
-0.0075 - 0.0296i  -0.0075 + 0.0296i  -0.0363 - 0.0278i  -0.0363 + 0.0278i    0.9964
-0.0311 + 0.1509i  -0.0311 - 0.1509i  -0.1885 + 0.1304i  -0.1885 - 0.1304i   -0.0201
 0.0331 - 0.1421i   0.0331 + 0.1421i  -0.1648 + 0.1103i  -0.1648 - 0.1103i   -0.0008

-8.7099 -30.4885i  -8.7099 +30.4885i  -8.0185 - 5.0963i  -8.0185 + 5.0963i   -0.9432
```

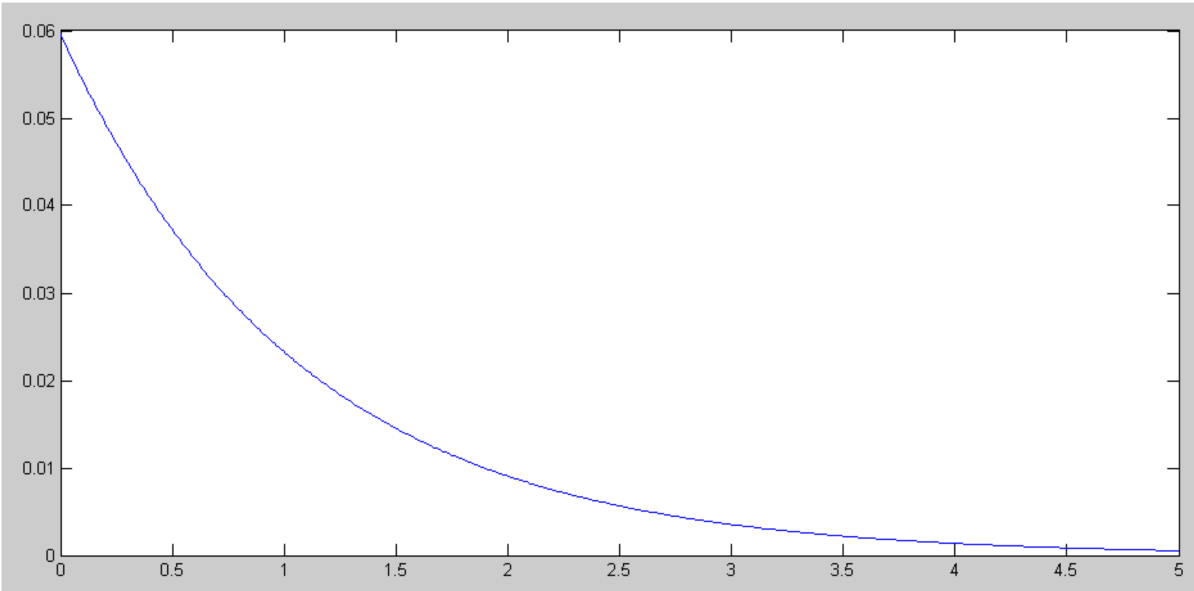
The slow mode is in red. If you make the initial condition proportional to the eigenvector

```
>> X0 = M(:,5)

V1     0.0566
V2     0.0597
V3     0.9964
I4    -0.0201
I5    -0.0008
```

the energy will decay as exp(-0.9432t)

In Matlab (not required - just illustrating this in the solutions)

```
>> X0 = M(:,5);
>> G = ss(A,X0,C,D);
>> y = impulse(G,t);
>> plot(t,y)
```



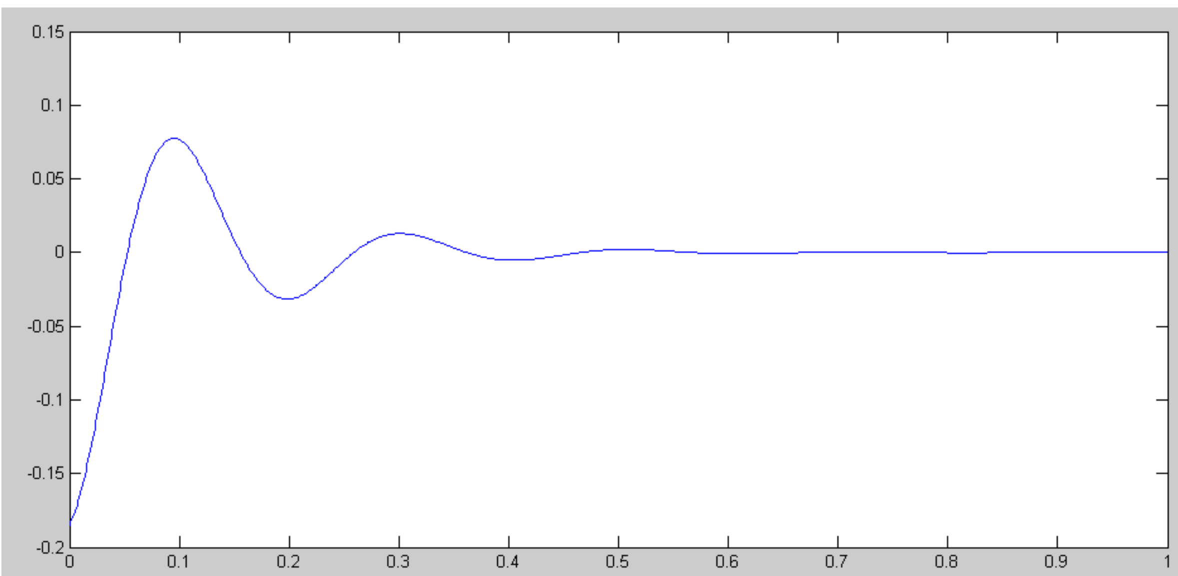
If you use another eigenvector, such as the first (real = cosine, imag = -sine)

```
>> X0 = real(M(:,1))

V1     0.9591
V2    -0.1846
V3    -0.0075
I4    -0.0311
I5     0.0331

>> X0 = imag(M(:,1))

V1          0
V2     0.0069
V3    -0.0296
I4     0.1509
I5    -0.1421
```

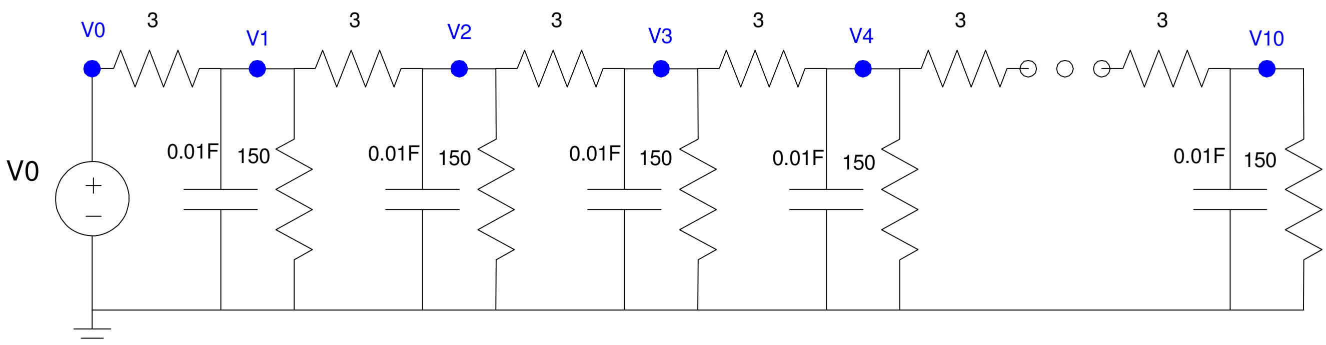
the trasient will decay as exp(-0.87099t) cos(30.4885t + phi)

Checking this in Matlab (not required - just illustrating this for the solutions)

```
>> G = ss(A,X0,C,D);
>> t = [0:0.001:1]';
>> y = impulse(G,t);
>> plot(t,y)
```



Problem 4-7: 10-Stage RC Filter.



Problems 4-7

4) For the 10-stage RC circuit

- Specify the dynamics for the system (write N coupled differential equations)
  - note: Nodes 1..9 have the same form. Just write the node equation for node 1 and node 10.
- Express these dynamics in state-space form
- Determine the transfer function from V0 to V10

Take node #1

$$I_1 = 0.01sV_1 = \left(\frac{V_0 - V_1}{3}\right) + \left(\frac{V_2 - V_1}{3}\right) - \left(\frac{V_1}{150}\right)$$

$$sV_1 = 33.33V_0 - 67.33V_1 + 33.33V_2$$

The same patter repeats for nodes 2..9

$$sV_2 = 33.33V_1 - 67.33V_2 + 33.33V_3$$

$$sV_3 = 33.33V_2 - 67.33V_3 + 33.33V_4$$

$$\vdots$$

$$sV_9 = 33.33V_8 - 67.33V_9 + 33.33V_{10}$$

Node #10 is a little difference since there is only a single 3-Ohm resistor connected to it

$$I_{10} = 0.01sV_{10} = \left(\frac{V_9 - V_{10}}{3}\right) - \left(\frac{V_{10}}{150}\right)$$

$$sV_{10} = 33.33V_9 - 34V_{10}$$

In state-space form

$$\begin{bmatrix} sV_1 \\ sV_2 \\ sV_3 \\ sV_4 \\ sV_5 \\ sV_6 \\ sV_7 \\ sV_8 \\ sV_9 \\ sV_{10} \end{bmatrix} = \begin{bmatrix} -67.33 & 33.33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 33.33 & -67.33 & 33.33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 33.33 & -67.33 & 33.33 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 33.33 & -67.33 & 33.33 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 33.33 & -67.33 & 33.33 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 33.33 & -67.33 & 33.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 33.33 & -67.33 & 33.33 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 33.33 & -67.33 & 33.33 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 33.33 & -67.33 & 33.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 33.33 & -34 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \\ V_{10} \end{bmatrix} + \begin{bmatrix} 33.33 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Throw into Matlab to find the tgransfer function

```
>> A = zeros(10,10);
>> for i=1:9
A(i,i) = -67.33;
A(i+1,i) = 33.33;
A(i,i+1) = 33.33;
end
>> A(10,10) = -34;
>> A

A =

  -67.3300   33.3300         0         0         0         0         0         0         0         0
   33.3300  -67.3300   33.3300         0         0         0         0         0         0         0
         0   33.3300  -67.3300   33.3300         0         0         0         0         0         0
         0         0   33.3300  -67.3300   33.3300         0         0         0         0         0
         0         0         0   33.3300  -67.3300   33.3300         0         0         0         0
         0         0         0         0   33.3300  -67.3300   33.3300         0         0         0
         0         0         0         0         0   33.3300  -67.3300   33.3300         0         0
         0         0         0         0         0         0   33.3300  -67.3300   33.3300         0
         0         0         0         0         0         0         0   33.3300  -67.3300   33.3300
         0         0         0         0         0         0         0         0   33.3300  -34.0000

>> B = [33.33;0;0;0;0;0;0;0;0;0];
>> C = [0,0,0,0,0,0,0,0,0,1];
>> D = 0;
>> G = ss(A,B,C,D);
>> zpk(G)


                                 1691816033937951
-----------------------------------------------------------------------------------------
(s+131) (s+122.4) (s+108.9) (s+91.68) (s+72.31) (s+52.5) (s+34) (s+18.46) (s+7.271) (s+1.415)

>>
```

5) For the transfer function for problem #4

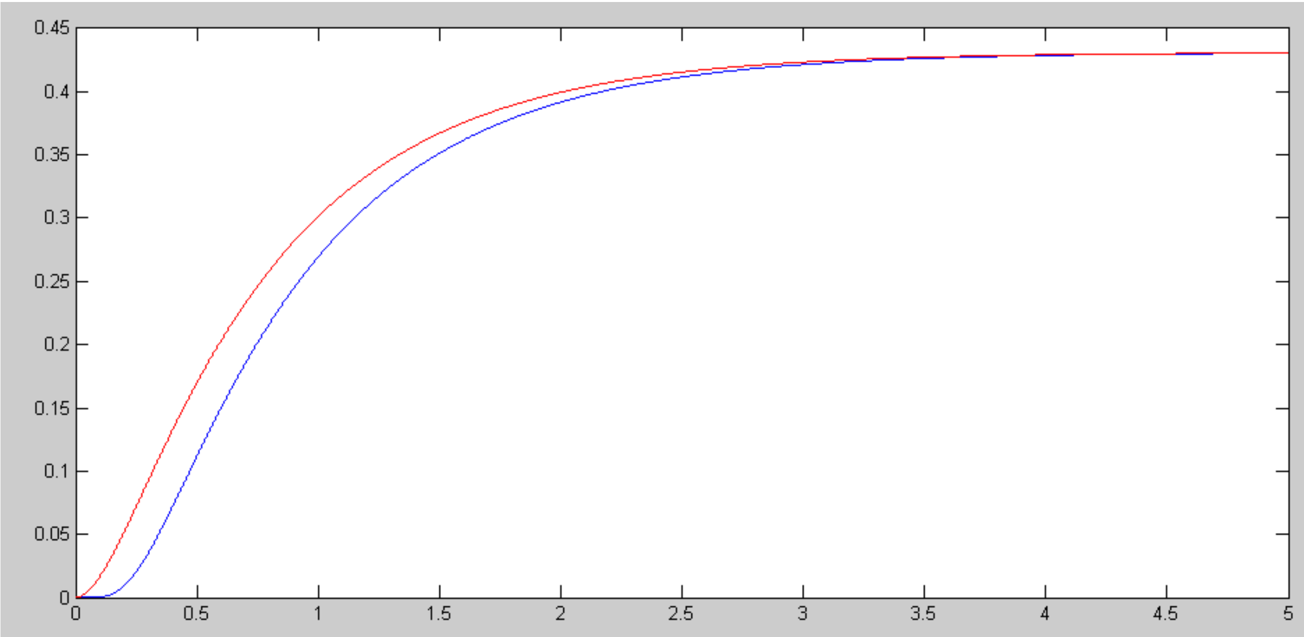
- Determine a 2nd-order approximation for this trasfer function
- Plot the step response of the actual 10th-order system and its 2nd-order approximation

```
                                        1691816033937951
  ------------------------------------------------------------------------------------------
  (s+131) (s+122.4) (s+108.9) (s+91.68) (s+72.31) (s+52.5) (s+34) (s+18.46) (s+7.271) (s+1.415)
```

Keep the slowest two poles

Match the DC gain

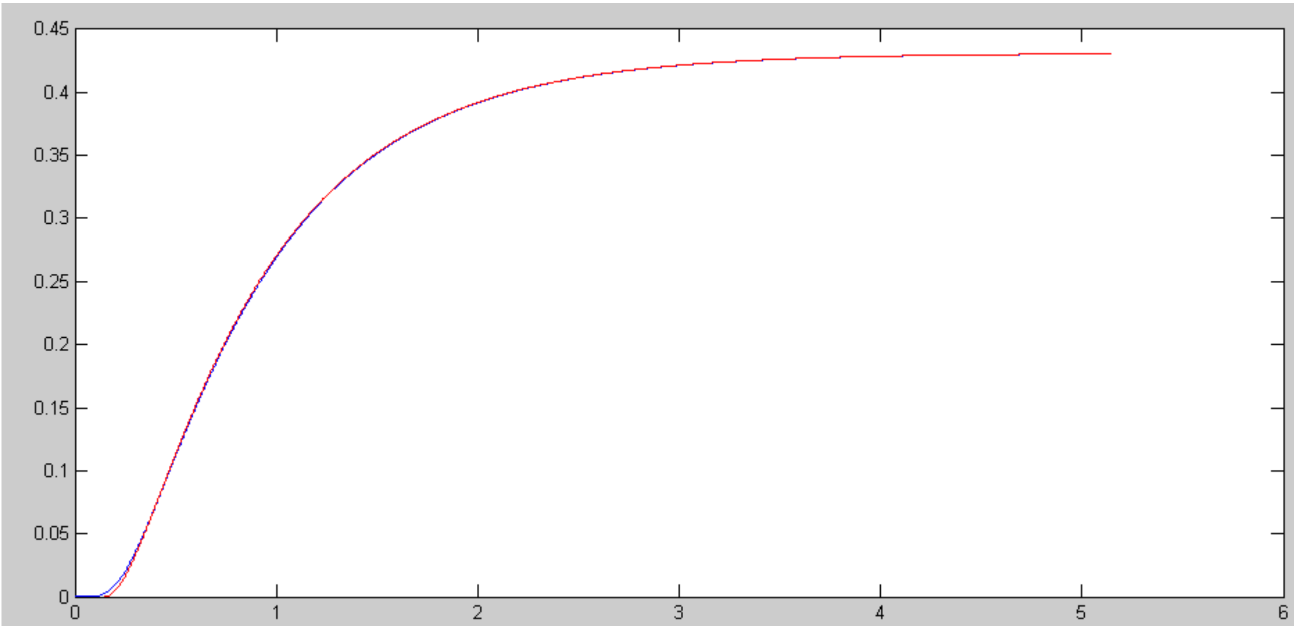
```
>> G2 = zpk([],[-1.415, -7.271],1);
>> k = evalfr(G,0) / evalfr(G2,0)

k =    4.4345

>> G2 = zpk([],[-1.415, -7.271],k);
>> G2

      4.4345
-------------------
(s+1.415) (s+7.271)

>> t = [0:0.001:5]';
>> y10 = step(G,t);
>> y2 = step(G2,t);
>> plot(t,y10,'b',t,y2,'r');
>>
```



Step resonse of the 10th-order system (blue) and 2nd-order approximation (red)

Note: A slightly better model adds a delay of 0.15 seconds (found with trial and error)

```
>> plot(t,y10,'b',t+0.15,y2,'r');
```



Step response of a 10th-order system (blue) and 2nd-order model with a 150ms delay (red)

6) For the circuit for problem #4

- What initial condition will decay as slowly as possible?
- What initial condition will decay as fast as possible?

This is an eigenvector problem

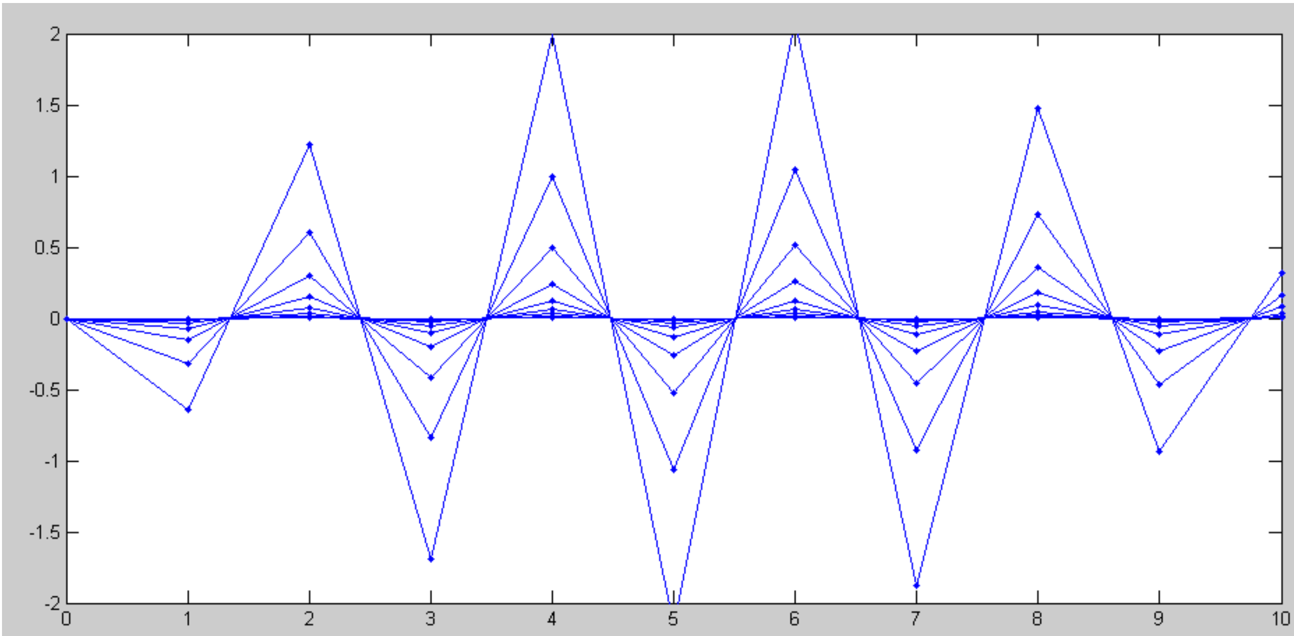
```
>> [M,V] = eig(A)

M =
     fast                                                                                 slow
   -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


>> eig(A)'

ans =

 -131.0285 -122.4071 -108.8918  -91.6836  -72.3115  -52.4968  -34.0000  -18.4648   -7.2714   -1.4145

>>
```

If you make the initial conditions proportional to the fast eigenvector, the states will decay as exp(-131t)

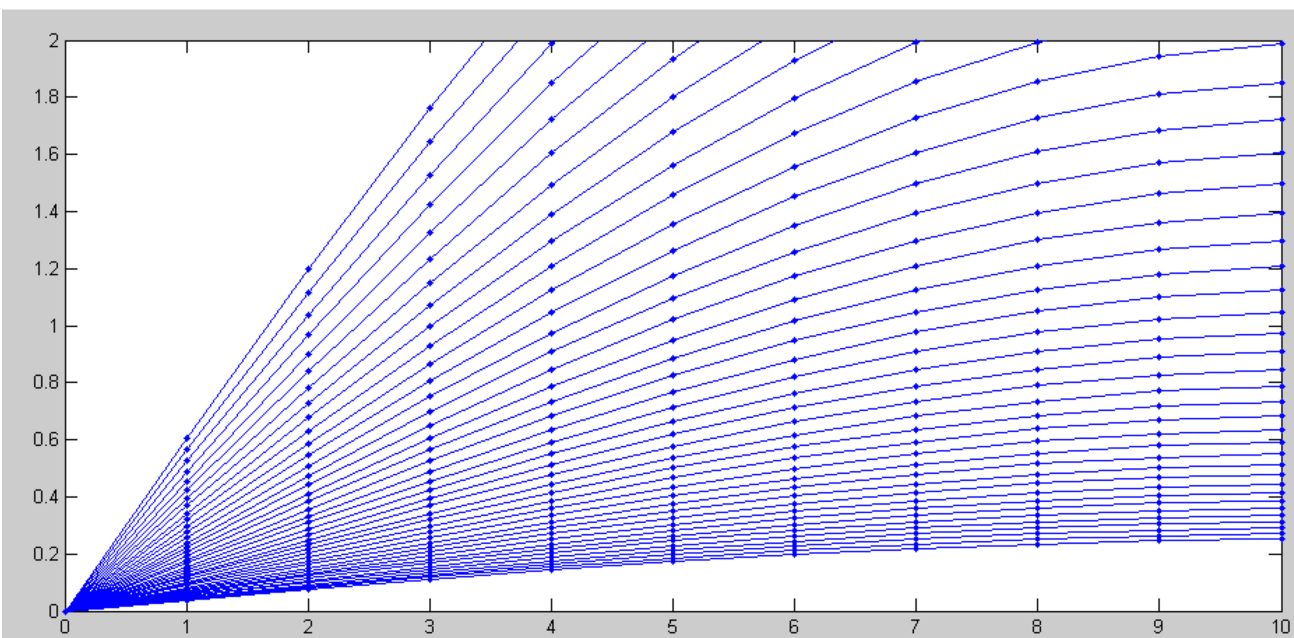
If you make the initial conidtions proportional to the slow eigenvector, the states will decay as exp(-1.41t)

7) Modify the program *heat.m* to match the dynamics you calculated for this problem.

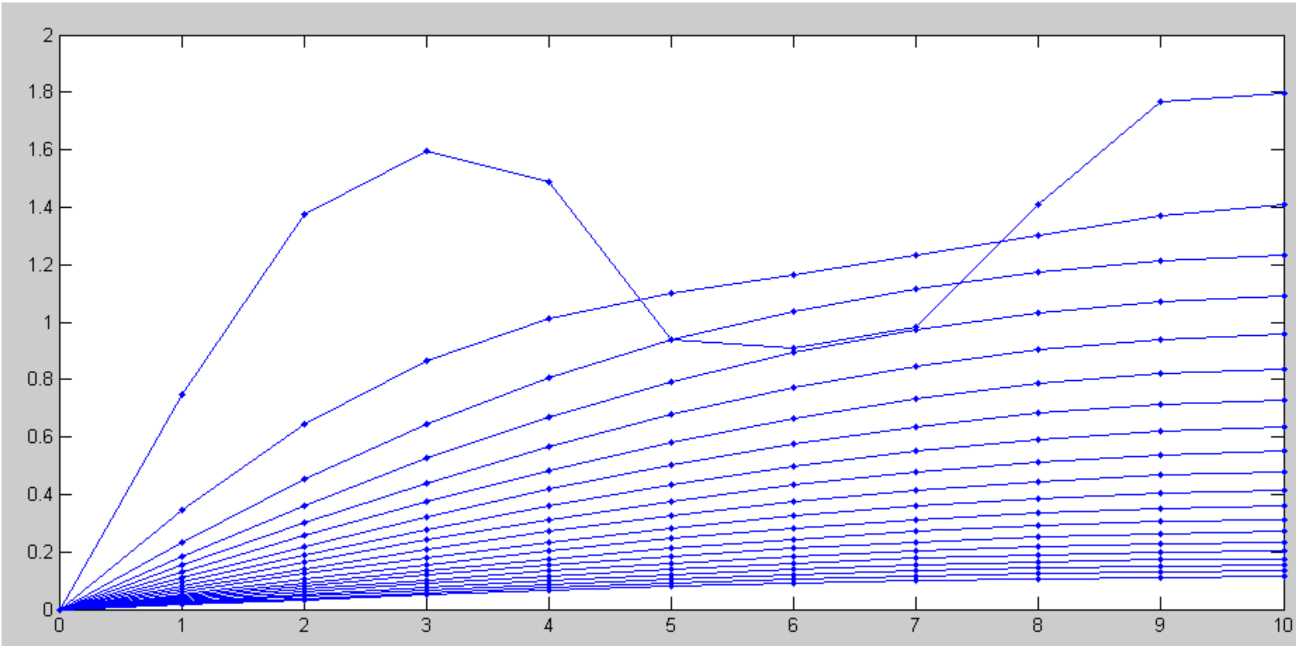
Fast mode



Slow Mode



Random Initial Condition



Give the program listing

```
% 10-stage RC Filter

V = 2*rand(10,1);
dV = 0*V;

dt = 0.01;
t = 0;
V0 = 0;

REF = 1;
DATA = [];

n = -1;
hold off

while(t < 2)

   dV(1) = 33.33*V0   - 67.33*V(1) + 33.33*V(2);
   dV(2) = 33.33*V(1) - 67.33*V(2) + 33.33*V(3);
   dV(3) = 33.33*V(2) - 67.33*V(3) + 33.33*V(4);
   dV(4) = 33.33*V(3) - 67.33*V(4) + 33.33*V(5);
   dV(5) = 33.33*V(4) - 67.33*V(5) + 33.33*V(6);
   dV(6) = 33.33*V(5) - 67.33*V(6) + 33.33*V(7);
   dV(7) = 33.33*V(6) - 67.33*V(7) + 33.33*V(8);
   dV(8) = 33.33*V(7) - 67.33*V(8) + 33.33*V(9);
   dV(9) = 33.33*V(8) - 67.33*V(9) + 33.33*V(10);
   dV(10) = 33.33*V(9) - 34*V(10);

   V = V + dV * dt;
   t = t + dt;

   n = mod(n+1,10);
   if(n == 0)
       plot([0:10], [V0;V], 'b.-');
       ylim([0,2]);
       xlim([0,10]);
       pause(0.01);
       hold on
   end
   DATA = [DATA ; V'];
end
```

Give the response for Vin = 0 and the initial conditions being

- The slowest eigenvector
- The fastest eigenvector
- A random set of voltages