Electric Circuits &
The Heat Equation

ECE 461/661 Controls Systems

Jake Glower - Lecture #14

Please visit Bison Academy for corresponding
lecture notes, homework sets, and solutions

Circuits & Differential Equations

Each energy storage elements adds one differential equation

- Inductors: Energy = $\frac{1}{2}LI^2$
- Capacitors: Energy = $\frac{1}{2}CV^2$

Energy States are

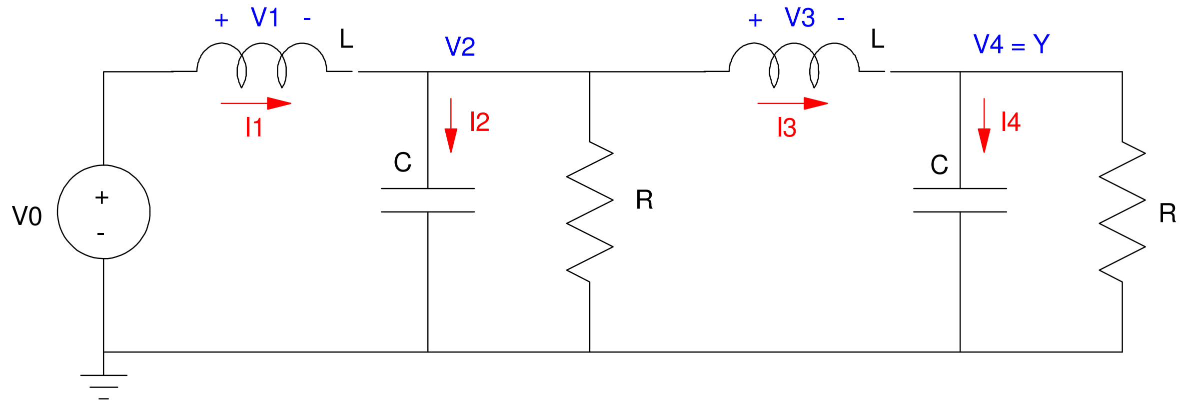
- I Inductor
- V Capacitor

The differential equations from from

$$V = L\frac{dI}{dt}$$

$$I = C\frac{dV}{dt}$$

Example #1: RLC Circuit



Dynamics

$$V_1 = LsI_1 = V_0 - V_2$$

$$I_2 = CsV_2 = I_1 - I_3 - \frac{V_2}{R}$$

$$V_3 = LsI_3 = V_2 - V_4$$

$$I_4 = CsV_4 = I_3 - \frac{V_4}{R}$$

State Space

$$s\begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{-1}{L}\right) & 0 & 0 \\ \left(\frac{1}{C}\right) & \left(\frac{-1}{RC}\right) & \left(\frac{-1}{C}\right) & 0 \\ 0 & \left(\frac{1}{L}\right) & 0 & \left(\frac{-1}{L}\right) \\ 0 & 0 & \left(\frac{1}{C}\right) & \left(\frac{-1}{RC}\right) \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{L}\right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Plugging in numbers in Matlab

$$s\begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & -10 & 0 & 0 \\ 100 & -1 & -100 & 0 \\ 0 & 10 & 0 & -10 \\ 0 & 0 & 100 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} X$$

```
A = [0,-10,0,0 ; 100,-1,-100,0 ; 0,10,0,-10, ; 0,0,100,-1];
B = [10;0;0;0];
C = [0,0,0,1];
D = 0;
G = ss(A,B,C,D);
zpk(G)

            1000000
--------------------------------
(s^2 + s + 382) (s^2 + s + 2618)
```

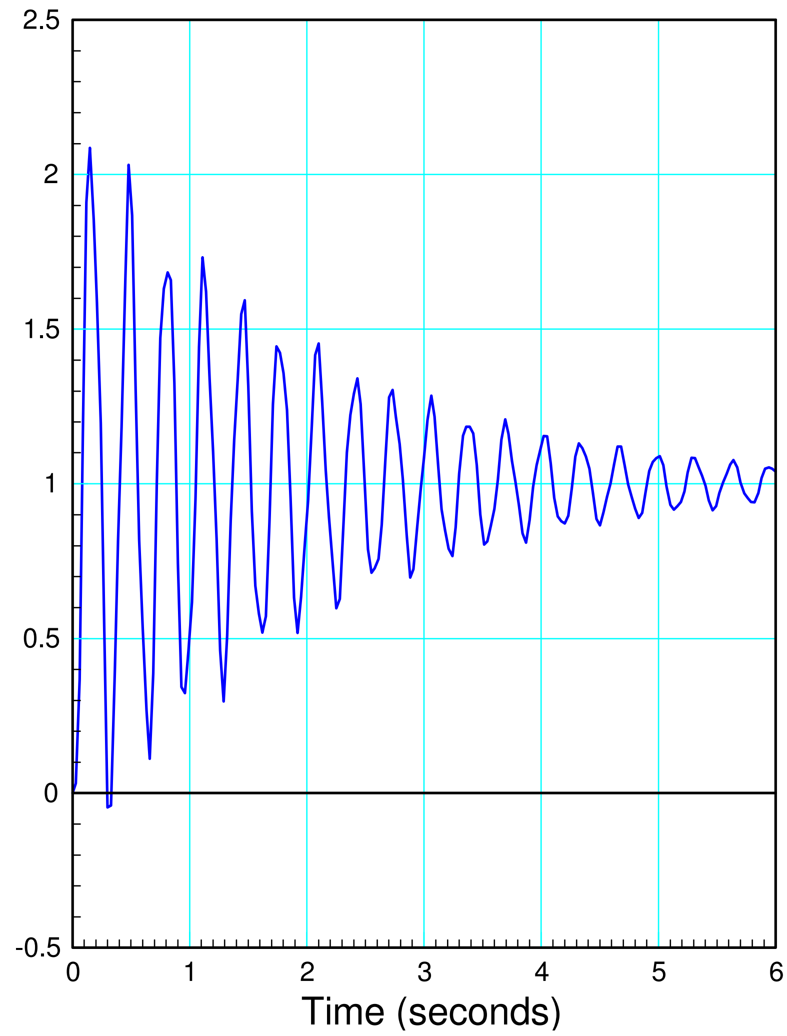
Poles = Eigenvalues

```
eig(G)

   -0.5000 +51.1643i
   -0.5000 -51.1643i
   -0.5000 +19.5376i
   -0.5000 -19.5376i
```

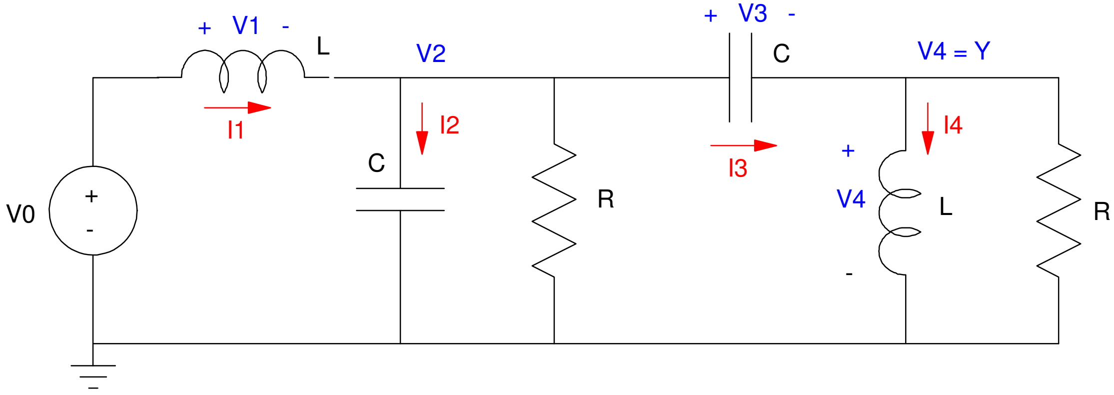
Step Response:

```
t = [0:0.03:6]';
y = step(G,t);
plot(t,y);
xlabel('Time (seconds)');
ylabel('y (Volts)');
```



Example 2:

Dynamics



$$V_1 = LsI_1 = V_0 - V_2$$

$$I_2 = CsV_2 = I_1 - \frac{V_2}{R} - I_4 - \frac{V_2 - V_3}{R}$$

$$I_3 = CsV_3 = I_4 + \frac{V_2 - V_3}{R}$$

$$V_4 = LsI_4 = V_2 - V_3$$

$$s\begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} \left(\frac{-1}{L}\right) & 0 & 0 & 0 \\ \left(\frac{1}{C}\right) & \left(\frac{-2}{RC}\right) & \left(\frac{1}{RC}\right) & \left(\frac{-1}{C}\right) \\ 0 & \left(\frac{1}{RC}\right) & \left(\frac{-1}{RC}\right) & \left(\frac{1}{C}\right) \\ 0 & \left(\frac{1}{L}\right) & \left(\frac{-1}{L}\right) & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{L}\right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Substituting L = 100mH, C = 10mF, R = 50

$$\begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} -10 & 0 & 0 & 0 \\ 100 & -4 & 2 & -100 \\ 0 & 2 & -2 & 100 \\ 0 & 10 & -10 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_{in}$$

$$Y = V_2 - V_3$$

Throwing it in MATLAB:

```
A = [-10,0,0,0;100,-4,2,-100;0,2,-2,100;0,10,-10,0];
B = [10;0;0;0];
C = [0,1,-1,0];
D = 0;

G = ss(A,B,C,D);
```

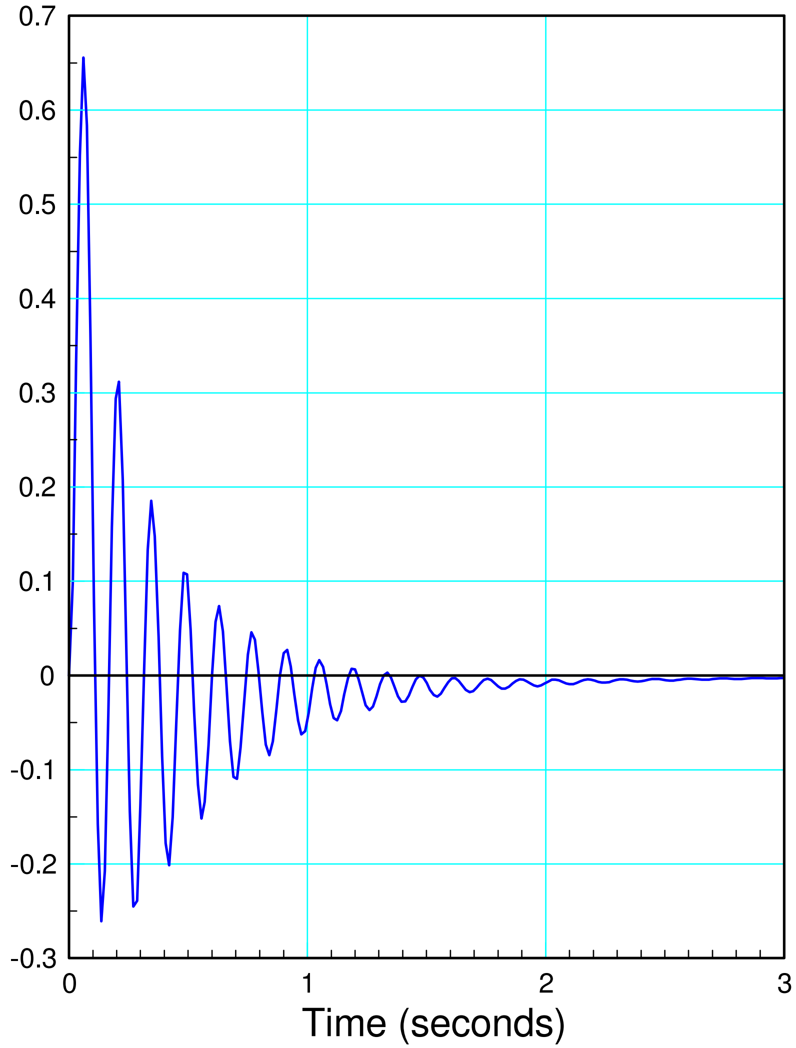
```
zpk(G)

                1000 s^2
--------------------------------------
(s+10) (s+1.001) (s^2 + 4.999s + 1999)

eig(G)

  -2.4997 +44.6402i
  -2.4997 -44.6402i
  -1.0005
 -10.0000

t = [0:0.015:3]';
y = step(G,t);
plot(t,y);
xlabel('Time (seconds)');
ylabel('y (Volts)');
```

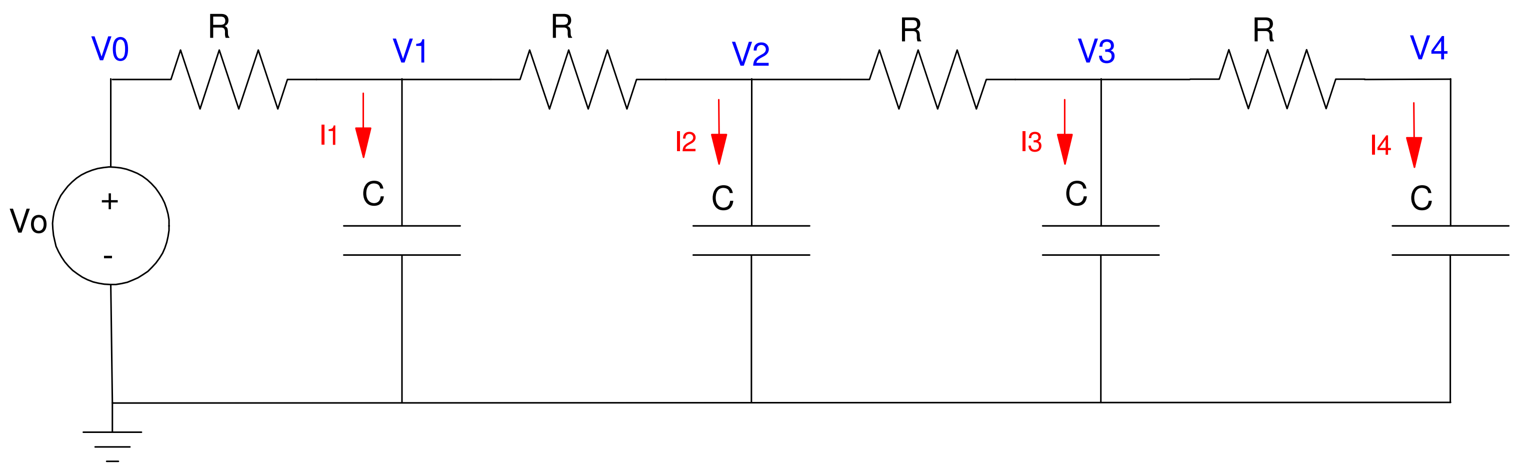


RC Circuits: Heat Equation

In one dimension, the heat equation is defined as coupled first-order differential equations where

$$\frac{dx_i}{dt} = f(x_{i-1}, x_i, x_{i+1})$$

This also describes RC circuits where the capacitors are grounded such as



4-Stage RC Filter (heat equation)

The analogy with heat flow is

- Voltage Temperature
- Current Heat Flow
- Resistance Thermal Resistance
- Electrical Energy: Thermal Energy

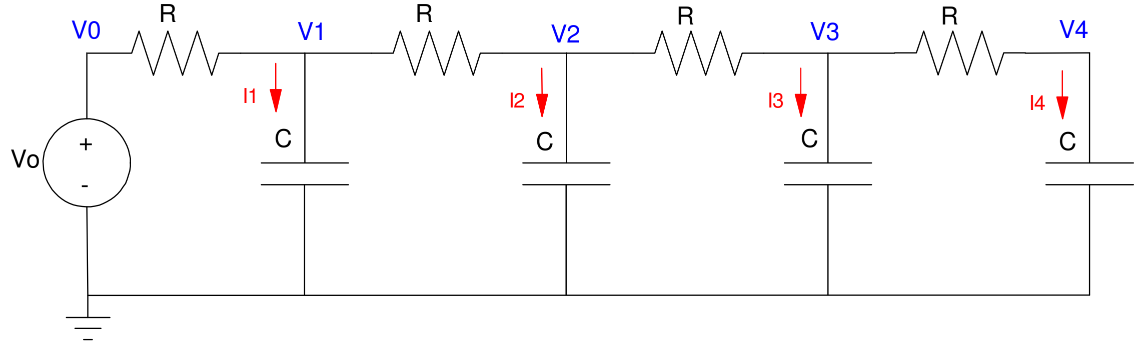
Heat	Thermal Inertia (J / degree)	Temperature (degrees C)	Heat Flow (Watts)	Thermal Resistance (degree C / Watt)
Electrical	Capacitance (J / volt)	Voltage (Volts)	Current (Amps)	Resistance (R) (Ohm = Volts / Amp)

$$I_1 = CsV_1 = \left(\frac{V_0 - V_1}{R}\right) + \left(\frac{V_2 - V_1}{R}\right)$$

$$I_2 = CsV_2 = \left(\frac{V_1 - V_2}{R}\right) + \left(\frac{V_3 - V_2}{R}\right)$$

$$I_3 = CsV_3 = \left(\frac{V_2 - V_3}{R}\right) + \left(\frac{V_4 - V_3}{R}\right)$$

$$I_4 = CsV_4 = \left(\frac{V_3 - V_4}{R}\right)$$



$$\begin{bmatrix} sV_1 \\ sV_2 \\ sV_3 \\ sV_4 \end{bmatrix} = \begin{bmatrix} \left(\frac{-2}{RC}\right) & \left(\frac{1}{RC}\right) & 0 & 0 \\ \left(\frac{1}{RC}\right) & \left(\frac{-2}{RC}\right) & \left(\frac{1}{RC}\right) & 0 \\ 0 & \left(\frac{1}{RC}\right) & \left(\frac{-2}{RC}\right) & \left(\frac{1}{RC}\right) \\ 0 & 0 & \left(\frac{1}{RC}\right) & \left(\frac{-1}{RC}\right) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{RC}\right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Dynamics when C = 0.01F, R = 10 Ohms

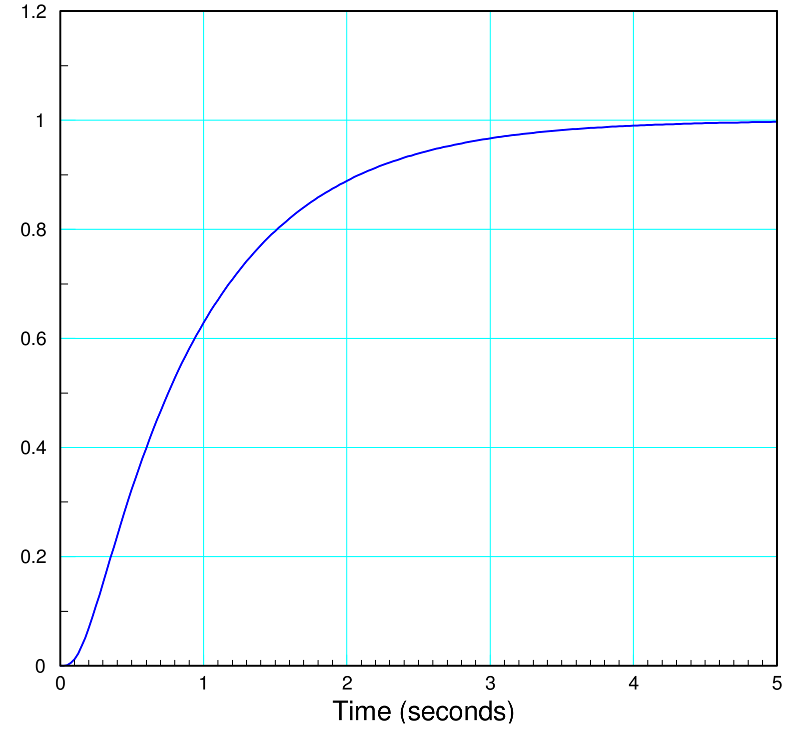
- DC gain = 1.000
- Dominant pole: s = -1.206

```
A = [-20,10,0,0 ; 10,-20,10,0 ; 0,10,-20,10 ; 0,0,10,-10];
B = [10;0;0;0];
C = [0,0,0,1];
D = 0;
G = ss(A,B,C,D);

DC = evalfr(G,0)
DC = 1

zpk(G)

               10000
------------------------------------
(s+35.32) (s+23.47) (s+10) (s+1.206)
```



Animation and Heat Equations (fun stuff):

Matlab actually does animation really well

- Expand 4-stage RC filter to 10 stages
- Simulates the temperature along a metal bar

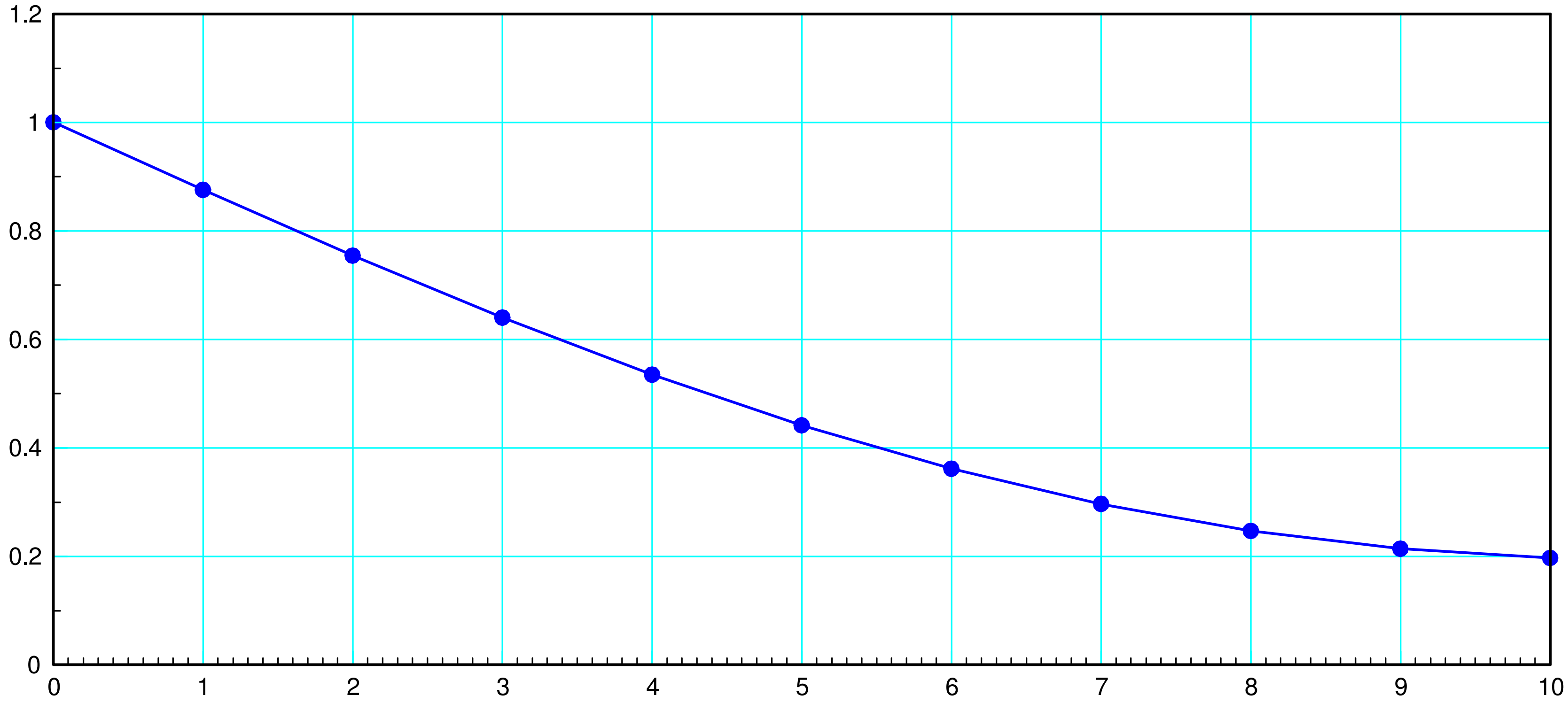
```
while(t < 100)

   dV(1)  = 10*V0   - 20*V(1) + 10*V(2);
   dV(2)  = 10*V(1) - 20*V(2) + 10*V(3);
   dV(3)  = 10*V(2) - 20*V(3) + 10*V(4);
   :
   dV(8)  = 10*V(7) - 20*V(8) + 10*V(9);
   dV(9)  = 10*V(8) - 20*V(9) + 10*V(10);
   dV(10) = 10*V(9) - 10*V(10);

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

   plot([0:10], [V0;V], '.-');
   ylim([0,1.2]);
   pause(0.01);
   end
```

This results in the following plot at t = 2.0 seconds



Step response at t = 2.00 seconds (votlage at each node)

Eigenvectors and Eigenvalues:

- Eigenvalues: How the system will behave
- Eigenvectors: What behaves that way

A-matrix in state-space form:

-20	10	0	0	0	0	0	0	0	0
10	-20	10	0	0	0	0	0	0	0
0	10	-20	10	0	0	0	0	0	0
0	0	10	-20	10	0	0	0	0	0
0	0	0	10	-20	10	0	0	0	0
0	0	0	0	10	-20	10	0	0	0
0	0	0	0	0	10	-20	10	0	0
0	0	0	0	0	0	10	-20	10	0
0	0	0	0	0	0	0	10	-20	10
0	0	0	0	0	0	0	0	10	-10

```
G = ss(A,B,C,D);

zpk(G)

                                        10000000000
------------------------------------------------------------------------------------------
(s+39.11) (s+36.52) (s+32.47) (s+27.31) (s+21.49) (s+15.55) (s+10) (s+5.339) (s+1.981) (s+0.2234)

eig(A)

  -39.1115  -36.5248  -32.4698  -27.3068  -21.4946  -15.5496  -10.0000   -5.3390   -1.9806   -0.2234
```

The eigenvalues are the poles

- *how* the system behaves

Eigenvectors:

- What behaves that way
- Fast mode (red)
- Slow mode (blue)

```
[m,n] = eig(A)

m =  Eigenvectors

   -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


v = Eigenvalues

  -39.1115  -36.5248  -32.4698  -27.3068  -21.4946  -15.5496  -10.0000   -5.3390   -1.9806   -0.2234
```

Eigenvectors in Matlab

The step response uses the MATLAB command *step()*

```
G = ss(A, B, C, D);
y = step(G, t);
```

The natural response to initial conditions can be found from the impulse response

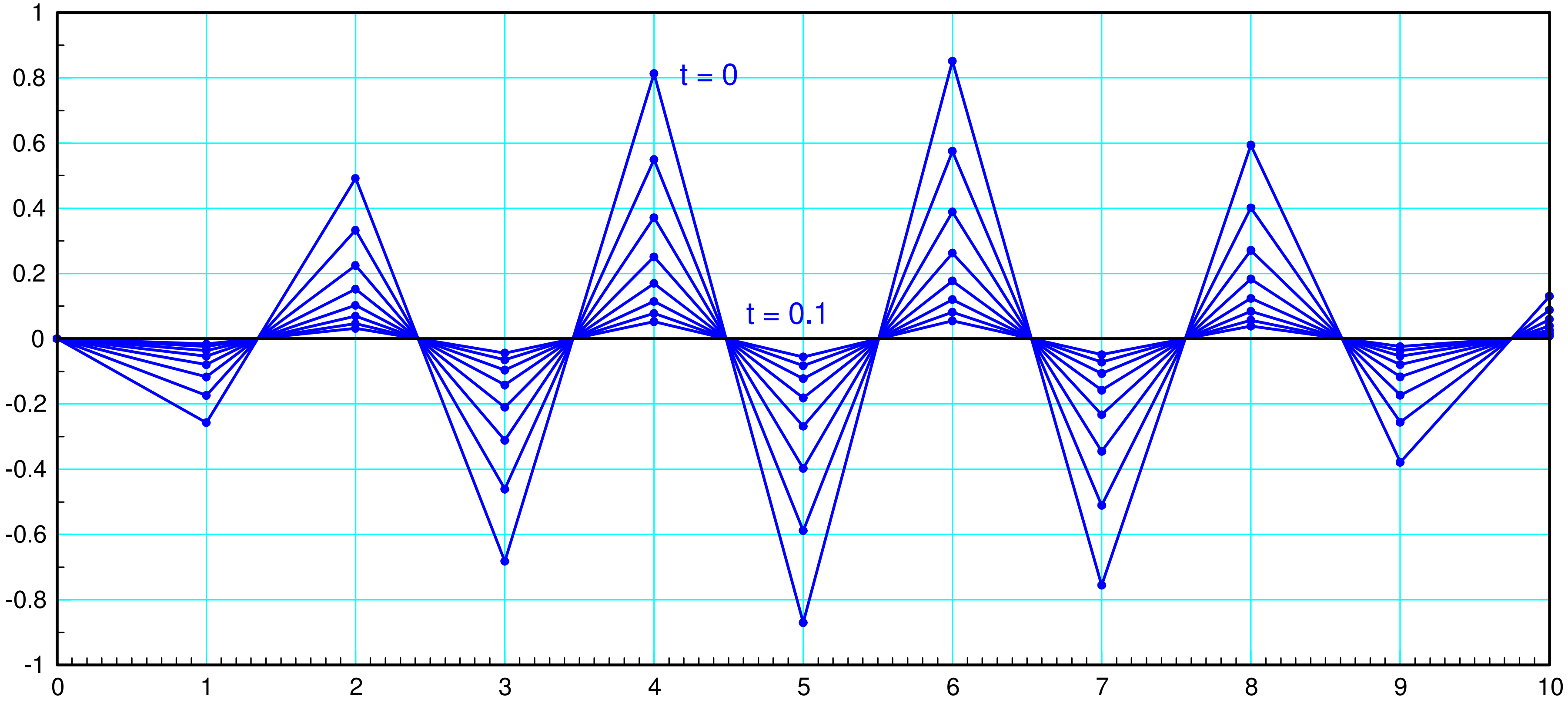
```
X0 = [1;2;3;4];
G = ss(A, X0, C, D);
y = impulse(G, t);
```

If you make the initial conditions an eigenvector, you'll see the response

- for the fast mode (fast eigenvector), or
- for the slow mode (slow eigenector)

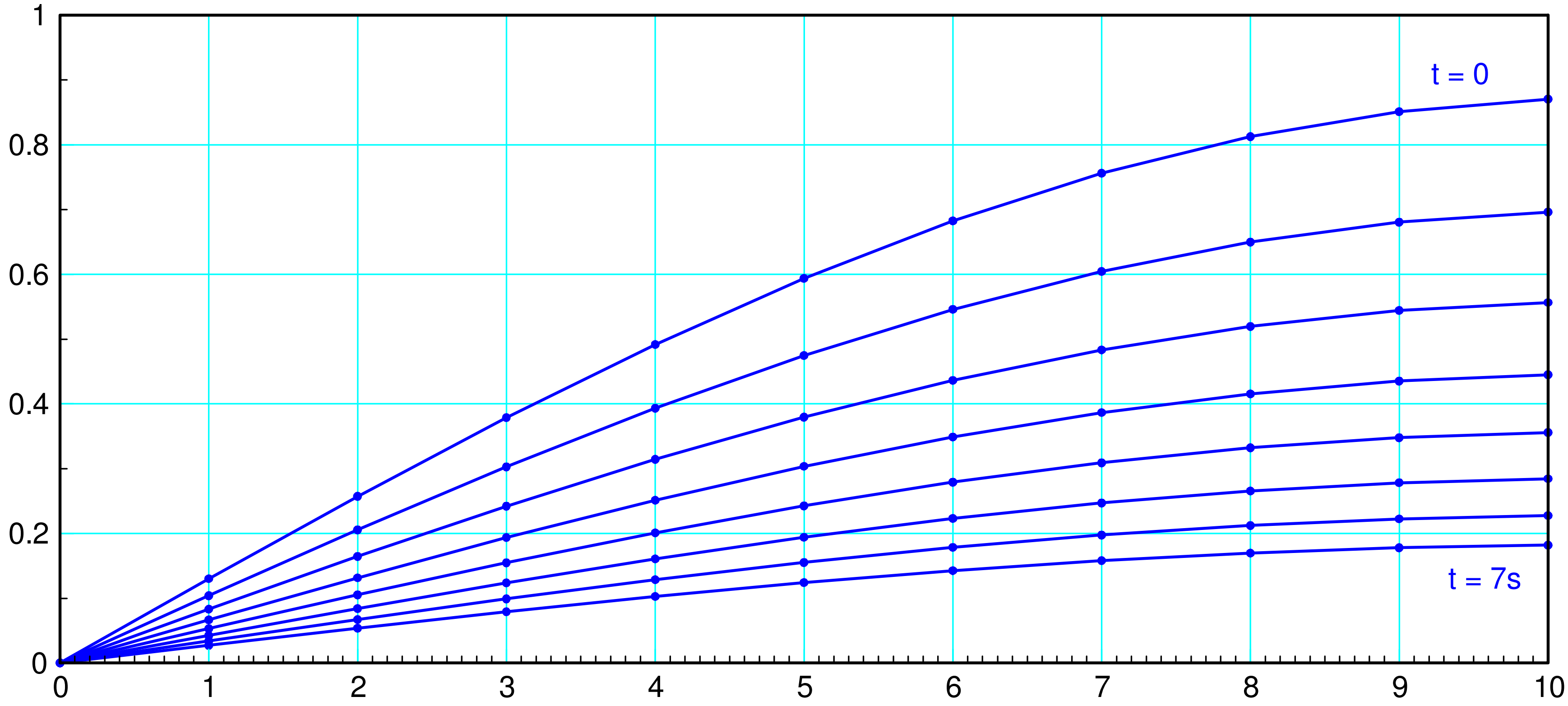
Fast Eigenvector (heat.m simulation)

- Shape remains the same (the eigenvector)
- The amplitude dops as exp(-39.11t) (the eigenvalue)



Slow Eigenvector

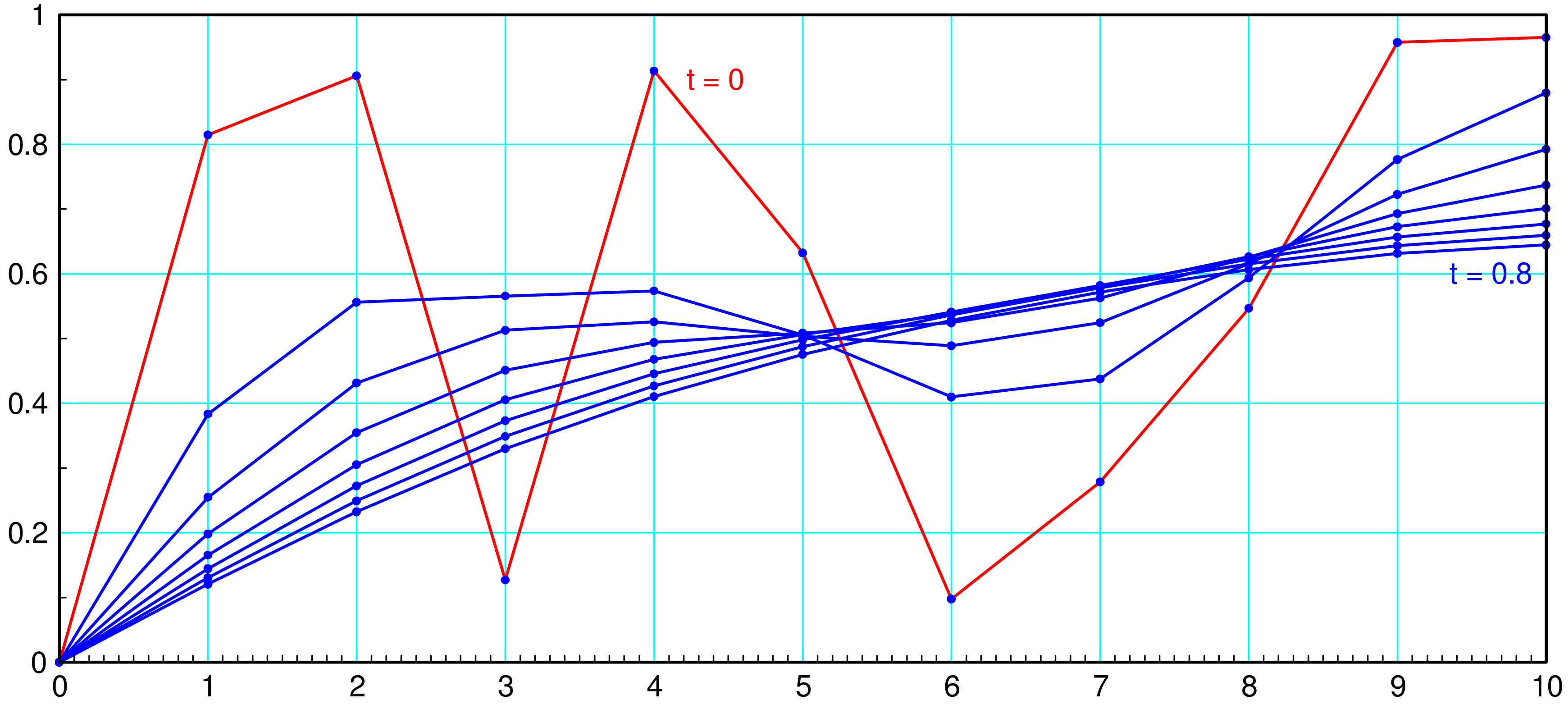
- Shape remains the same (the eigenvector)
- The amplitude dops as exp(-0.2234t) (the eigenvalue)



Natural Response with the initial condition being the slow eigenvector

Random Initial Condition

- All 10 eigenvectors excited
- Fast ones die out quickly, leaving the slow (dominant) mode



Natural Response with Random Initial Conditions