Objectives:

Be able to:

- Write the differential equations for a system modeled by the heat equation
- Write the electrical RC circuit equivalent for such a system
- Write the state-space equations for a heat system
- Find the transfer function using MATLAB
- Given the transfer function, predict how the system will respond to a step input

Definition: The Heat Equation:

If you want to model the temperature behaviour of a system, you could break the system into a large number of finite elements. The temperature at each element is proportional to the energy in that element. The energy at this node can flow to neighboring elements and is described by a first-order differential equation.

At each node:

\[
\frac{dx_j}{dt} = -a_{ij}x_i + \sum_{i \neq j} a_{ij}x_j
\]

where \( x_i \) is the temperature at node \( i \) and \( a_{ij} \) are real positive constants:

- \( a_{ij} > 0 \)
- \( a_{ii} > \sum_{i \neq j} a_{ij} \)

The last requirement is conservation of energy: the energy gained by neighboring nodes cannot be greater than the energy lost at node \( i \). If could be less (if you have a lossy, i.e. poorly insulated) system, however.

RC Circuit Analogy:

The electrical circuit which also satisfied the heat equation is a passive RC network with capacitors to ground and resistors connecting the capacitors. For example, suppose you wanted to model the temperature along a long rod.

If you treat this as five finite elements, each element has

- Thermal inertia: the energy stored is proportional to the temperature at that node, and
- Thermal conductivity: Between elements, energy can flow and is proportional to the temperature difference between adjacent nodes times the thermal conductivity.

The electrical analogy is an RC network where each voltage node has

- A capacitor to ground: the energy stored is proportional to the voltage at the node, and
- Resistance: between each capacitor, resistors are placed. The current flow is proportional to the voltage between adjacent nodes times the electrical conductivity of that resistor.
The analogy is

<table>
<thead>
<tr>
<th>Heat</th>
<th>Thermal Inertia (J / degree)</th>
<th>Temperature (degrees C)</th>
<th>Heat Flow (Watts)</th>
<th>Thermal Resistance (degree C / Watt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>Capacitance (J / volt)</td>
<td>Voltage (Volts)</td>
<td>Current (Amps)</td>
<td>Resistance (R) (Ohm = Volts / Amp)</td>
</tr>
</tbody>
</table>

1-Dimensional Example:

For example, find the mathematical model for the temperature along a long thin rod. For convenience, assume you split this rod into five finite elements. For each element,

- Assume each element has a thermal inertia of C Joules / degree. (the volume of each element times the thermal capacitance of the material)
- Between elements, assume that 1 Watts flows if the temperature difference is R degrees. (Thermal conductivity times area divided by distance).

\[
x_1, x_2, x_3, x_4, x_5
\]

The electrical analogy is an RC circuit is as follows:

\[
x_1, x_2, x_3, x_4, x_5
\]

The differential equations for this system are then:

\[
C_1 \frac{dx_1}{dt} = -\left(\frac{1}{R_{01}} + \frac{1}{R_{12}}\right)x_1 + \left(\frac{1}{R_{12}}\right)x_2 + \left(\frac{1}{R_{01}}\right)u
\]

\[
C_2 \frac{dx_2}{dt} = -\left(\frac{1}{R_{12}} + \frac{1}{R_{23}}\right)x_2 + \left(\frac{1}{R_{12}}\right)x_1 + \left(\frac{1}{R_{23}}\right)x_3
\]

\[
C_3 \frac{dx_3}{dt} = -\left(\frac{1}{R_{23}} + \frac{1}{R_{34}}\right)x_3 + \left(\frac{1}{R_{23}}\right)x_2 + \left(\frac{1}{R_{34}}\right)x_4
\]
\[ C^4 \frac{dx_4}{dt} = -\left( \frac{1}{R_{34}} + \frac{1}{R_{45}} \right)x_4 + \left( \frac{1}{R_{34}} \right)x_3 + \left( \frac{1}{R_{45}} \right)x_5 \]

\[ C^5 \frac{dx_5}{dt} = -\left( \frac{1}{R_{45}} \right)x_5 + \left( \frac{1}{R_{45}} \right)x_4 \]

In State Space form:

\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4 \\
    x_5
\end{bmatrix}
= \begin{bmatrix}
    \left( \frac{-1/R_{01} + 1/R_{12}}{C_1} \right) & \left( \frac{1/R_{12}}{C_1} \right) & 0 & 0 & 0 \\
    \left( \frac{1/R_{12}}{C_2} \right) & \left( \frac{-1/R_{12} + 1/R_{23}}{C_2} \right) & 0 & 0 & 0 \\
    0 & \left( \frac{-1/R_{23} + 1/R_{34}}{C_3} \right) & \left( \frac{1/R_{34}}{C_3} \right) & 0 & 0 \\
    0 & 0 & \left( \frac{-1/R_{34} + 1/R_{45}}{C_4} \right) & \left( \frac{1/R_{45}}{C_4} \right) & 0 \\
    0 & 0 & 0 & \left( \frac{-1/R_{45} + 1/R_{55}}{C_5} \right) & \left( \frac{1/R_{55}}{C_5} \right)
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4 \\
    x_5
\end{bmatrix}
+ \begin{bmatrix}
    0 \\
    0 \\
    0 \\
    0 \\
    0
\end{bmatrix}u
\]

If \( C = 0.01 \text{F} \) and \( R = 10 \text{ Ohms} \),

\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4 \\
    x_5
\end{bmatrix}
= \begin{bmatrix}
    -20 & 10 & 0 & 0 & 0 \\
    10 & -20 & 10 & 0 & 0 \\
    0 & 10 & -20 & 10 & 0 \\
    0 & 0 & 10 & -20 & 10 \\
    0 & 0 & 0 & 10 & -10
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3 \\
    x_4 \\
    x_5
\end{bmatrix}
+ \begin{bmatrix}
    0 \\
    0 \\
    0 \\
    0 \\
    0
\end{bmatrix}u
\]

Note that this is a lossless system: there is no thermal conductivity between the rod and the environment. This shows up in the above matrix with each row adding to zero. If there were losses, the diagonal elements would be more negative and the rows would sum to a negative number (lossy).

**Problem:** Estimate how this system will behave:

**Solution:** You need to know the systems dominant pole(s) and DC gain. In MATLAB:

```matlab
A = [-20,10,0,0,0;10,-20,10,0,0;0,10,-20,10,0;0,0,10,-20,10;0,0,0,10,-10];
B = [10;0;0;0;0];
C = [0,0,0,0,1];
D = 0;
G = ss(A,B,C,D);
DC = evalfr(G, 0)
DC = 1.0000
zpks(G)

\[ G(s) = \frac{100000}{(s + 36.8)(s + 28.3)(s + 17.1)(s + 6.9)(s + 0.81)} \]
```
• The system has a DC gain of one. If the input is increased to 100°C, the output will go to 100°C as well.
• The system has a dominant pole at -0.8101. It will take approximately 4.93 seconds (4/0.8101) to reach steady state.
• The dominant pole is real. There should be no overshoot or oscillations in the step response.

Problem: Determine the actual step response:

Solution: Using MATLAB:

```matlab
t = [0:0.001:10]';
y = step(G,t);
plot(t,y)
xlabel('Time (seconds)');
ylabel('Y (Celsius)');
```

![Graph showing the step response](image)
2-Dimensional Heat Equations:

The above also works for plates (2 dimensions), and solids (3-dimensions). For example, find the heat flow for a plate, modeled as five finite elements:

\[
\begin{bmatrix}
  x_1 & x_2 & x_3 \\
  x_4 & x_5 & x_6
\end{bmatrix}
\]

Assuming all resistances are 10 Ohms and \( C = 0.01 \text{F} \) again,

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6
\end{bmatrix} = \begin{bmatrix}
  -30 & 10 & 0 & 10 & 0 & 0 \\
  10 & -30 & 10 & 0 & 10 & 0 \\
  0 & 10 & -20 & 0 & 0 & 10 \\
  10 & 0 & 0 & -30 & 10 & 0 \\
  0 & 10 & 0 & 10 & -30 & 10 \\
  0 & 0 & 10 & 0 & 10 & -20
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6
\end{bmatrix} + \begin{bmatrix}
  10 \\
  0 \\
  0 \\
  10 \\
  0 \\
  0
\end{bmatrix}
\]

Problem: Estimate what the step response from \( U \) to \( x_3 \) will look like

Solution: In MATLAB:

\[
A1 = [-30,10,0,10,0;10,-30,10,0,10;0,10,-20,0,0;0,0,10,0,0;0,0,0,10,0;0,0,0,0,10]
\]
\[
A2 = [10,0,0,-30,10,0;0,10,0,10,-30,10;0,0,10,0,10,-20]
\]
\[
A = [A1;A2]
\]
\[
B = [10;0;0;10;0;0];
\]
\[
C = [0,0,1,0,0,0];
\]
\[
D = 0;
\]
\[
G = ss(A,B,C,D);
\]
\[
DC = evalfr(G, 0)
\]
\[
DC = 1.0000
\]
\[
zpk(G)
\]
\[ G(s) = \frac{1000 (s+21.98)(s+35.54)(s+52.46)}{(s+1.98)(s+15.54)(s+21.98)(s+32.46)(s+35.54)(s+52.46)} \]

- The system has a DC gain of one. If the input is increased to 100\textdegree C, the output will go to 100\textdegree C as well.
- The system has a dominant pole at -1.98. It will take approximately 2.02 seconds \((4/1.98)\) to reach steady state.
- The dominant pole is real. There should be no overshoot or oscillations in the step response.

The step response is

\begin{verbatim}
-->t = [0:0.001:10]';
-->y = step(G,t);
-->plot(t,y)
-->xlabel('Time (seconds)');
-->ylabel('Y (Celsius)');
\end{verbatim}
Animation and Heat Equations (fun stuff):

With Matlab, you can show the metal bar heating up and cooling down. For example, simulate a 10-node RC filter with $1/RC = 10$ and the base temperature of 100°C ($V_0 = 100$)

```matlab
% 10-stage RC Filter
V = zeros(10,1);
dV = zeros(10,1);
V0 = 100;
dt = 0.01;
t = 0;

while(t < 100)
    dV(1) = 10*V0 - 20.1*V(1) + 10*V(2);
    dV(2) = 10*V(1) - 20.1*V(2) + 10*V(3);
    dV(3) = 10*V(2) - 20.1*V(3) + 10*V(4);
    dV(4) = 10*V(3) - 20.1*V(4) + 10*V(5);
    dV(5) = 10*V(4) - 20.1*V(5) + 10*V(6);
    dV(6) = 10*V(5) - 20.1*V(6) + 10*V(7);
    dV(7) = 10*V(6) - 20.1*V(7) + 10*V(8);
    dV(8) = 10*V(7) - 20.1*V(8) + 10*V(9);
    dV(9) = 10*V(8) - 20.1*V(9) + 10*V(10);
    dV(10) = 10*V(9) - 10.1*V(10);

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

    hold off
    plot([0,10],[0,100],'w.');
    hold on
    plot([0:10], [V0;V], '.-');
    pause(0.01);
end
```

This results in the following plot at $t = 2.0$ seconds
A more efficient way to simulate this is to use matrices (i.e. state-space form)

\[ sX = AX + BU \]

\[ Y = CX \]

In MATLAB:

```matlab
% 10-stage RC Filter
V = zeros(10,1);
dV = zeros(10,1);
V0 = 100;
dt = 0.01;
t = 0;

a = 10;   % 1 / RC = 10
A = zeros(10,10);
for i = 1:9
    A(i,i) = -20.1;
    A(i+1,i) =  10;
    A(i,i+1) =  10;
end
A(10,10) = -10;

B = zeros(10,1);
B(1) = 10;

C = zeros(1,10);
C(10) = 1;

X = zeros(10,1);

U = 100;

while(t < 10)
    dX = A*X + B*U;
    X = X + dX*dt;
    t = t + dt;

    hold off
    plot([0,10],[0,100],'w.);
    hold on
    plot([0:10], [U;X], '.-');
    pause(0.01);
end
```
The resulting (A, B, C) matrices are:

```
>> A

-20.1  10.0   0   0   0   0   0   0   0   0
 10.0  -20.1  10.0   0   0   0   0   0   0   0
  0   10.0   10.0   0   0   0   0   0   0   0
  0   0   10.0  -20.1  10.0   0   0   0   0   0
  0   0   0   10.0  -20.1  10.0   0   0   0   0
  0   0   0   0   10.0  -20.1  10.0   0   0   0
  0   0   0   0   0   10.0  -20.1  10.0   0   0
  0   0   0   0   0   0   10.0  -20.1  10.0   0
  0   0   0   0   0   0   0   10.0  -10.0   0
  0   0   0   0   0   0   0   0   10.0  -10.0

>> B

B =

10
 0
 0
 0
 0
 0
 0
 0
 0
 0

>> C

C =

0   0   0   0   0   0   0   0   0   1

>>
```
Eigenvectors and Eigenvalues:

The eigenvectors and eigenvalues of A are:

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

```
> m = Eigenvectors
> n = eigenvalues
```

```
m = Eigenvectors
dominant pole
-0.1287 -0.2459  0.3413  0.4064  0.4353  0.4256  0.3780  0.2967 -0.1890  0.0637
 0.2459  0.4064 -0.4255 -0.2967 -0.0647  0.1899  0.3785  0.4355 -0.3409  0.1261
-0.3413 -0.4255  0.1891 -0.1898 -0.4257 -0.3408  0.0011  0.3425 -0.4259  0.1859
 0.4064  0.2967  0.1897  0.4352  0.1279 -0.3420 -0.3774  0.0672 -0.4273  0.2419
-0.4353 -0.0647 -0.4257 -0.1280  0.4067  0.1863 -0.3790 -0.2439 -0.3449  0.2930
 0.4255 -0.1897  0.3409 -0.3418 -0.1884  0.4260 -0.0022 -0.4292 -0.1947  0.3391
-0.3779  0.3782  0.0007  0.3775 -0.3787  0.0018  0.3769 -0.3801 -0.0064  0.3763
 0.2967 -0.4352 -0.3418  0.0662  0.2446 -0.4252  0.3796 -0.1328  0.1832  0.4068
-0.1891  0.3409  0.4254 -0.4258  0.3424 -0.1915  0.0032  0.1852  0.3369  0.4290
 0.0647 -0.1280 -0.1885  0.2447 -0.2955  0.3397 -0.3763  0.4047  0.4244  0.4425

n = eigenvalues
dominant pole
-39.2110   0      0      0      0      0      0      0      0      0
 0   0      0      0      0      0      0      0      0      0
 0   0   -32.5662  0      0      0      0      0      0      0
 0   0   27.4008  0      0      0      0      0      0      0
 0   0      0      0      0   -21.5858  0      0      0      0
 0   0      0      0      0      0      0      0      0      0
 0   0      0      0      0      0      0      0      0      0
 0   0      0      0      0      0      0      0      0      0
 0   0      0      0      0      0      0      0      0      0
 0   0      0      0      0      0      0      0      0      0
```

The eigenvalues tell you how the system will behave (i.e. the same as the system's poles) The dominant pole is the slowest eigenvalue (0.3041)

The eigenvectors tell you how each mode behaves (i.e. the mode shapes). The dominant pole has an eigenvector that goes from 0.06 to 0.44 in a smoothe fashion. If you start with a random initial condition, shortly the dominant mode will prevail.

For example, let the initial condition be random temperatures

```matlab
V = rand(10,1) * 100;
```

This initial condition excites all of the modes (eigenvectors). Shortly, the fast modes dies out and leave the dominant mode (and its corresponding eigenvector).

In simulation, this looks like the following:
Temperature for a random initial condition at $t = 0$ (blue) and $t = 1$ (green)