Heat Transfer - Homework 3

Rose Gebhardt and Harris Paspuleti

February 12, 2020

1 Add Capacitances

Problem Statement: Add capacitances to the circuit models for exercises 2.1 and 2.2 then plot temperature for each metal and heat transfer for each boundary as a function of time. The height for the wall in each exercise is 30 cm and the depth is 50 cm. Let the wall start at the higher temperature. Find approximately how long it takes for the wall to reach steady state. Find the steady state temperatures and determine if they agree with the expected values.

Results and Analysis:

To add capacitances to the models, the blocks — which were modelled as resistors — need to be modelled as two resistors with half the resistance for each. A capacitor with a battery source to show the initial temperature needs to be connected between the two resistors and ground. The results are the following two circuits:

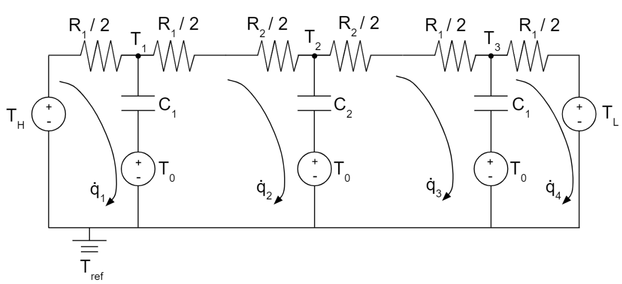


Figure 1: Circuit model of exercise 2.1

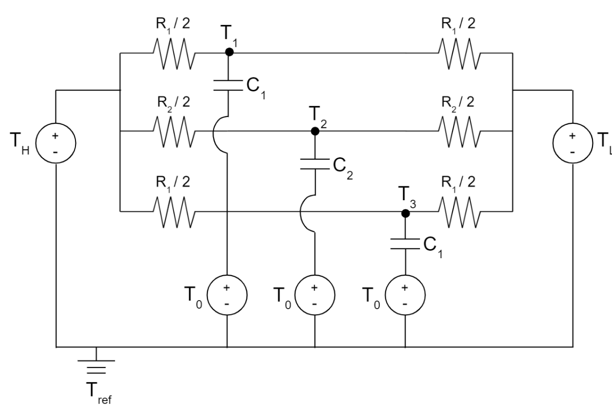


Figure 2: Circuit model of exercise 2.2

First, the values R_1, R_2, C_1, C_2 need to be found. $R_1 = \dfrac{\Delta x_1}{k_1 A}$, $R_2 = \dfrac{\Delta x_2}{k_2 A}$ where k_1 and k_2 are the thermal conductivities of each material, Δx_1 and Δx_2 are the distances through which heat is transferred, and A is the area through which heat is transferred. In order to compute C_1 and C_2, the densities (ρ) and specific heat capacitances (c) of copper and stainless steel need to be known. This is determined by looking up the material properties. From there, $C_1 = c_1\rho_1 V_1$ and $C_2 = c_2\rho_2 V_2$ where V is the volume of the material. The numeric values are calculated using MATLAB (see appendix).

1.1 Exercise 2.1 Circuit Analysis:

The temperatures in exercise 2.1 are best found using mesh analysis to find heat transfer rates:

$$\begin{pmatrix} \frac{R_1}{2}+\frac{1}{sC_1} & \frac{-1}{sC_1} & 0 & 0 \\ \frac{-1}{sC_1} & \frac{R_1+R_2}{2}+\frac{1}{sC_1}+\frac{1}{sC_2} & \frac{-1}{sC_2} & 0 \\ 0 & \frac{-1}{sC_2} & \frac{R_1+R_2}{2}+\frac{1}{sC_1}+\frac{1}{sC_2} & \frac{-1}{sC_1} \\ 0 & 0 & \frac{-1}{sC_1} & \frac{R_1}{2}+\frac{1}{sC_1} \end{pmatrix} \begin{pmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{Q}_3 \\ \dot{Q}_4 \end{pmatrix} = \begin{pmatrix} \frac{T_H-T_0}{s} \\ 0 \\ 0 \\ \frac{T_0-T_L}{s} \end{pmatrix}$$

Once MATLAB is used to find the heat transfer rates, the temperature at the nodes can be found using Ohm's Law:

$$T_1 = \frac{T_H}{s} - \left(\frac{R_1}{2}\right)\dot{Q}_1$$

$$T_2 = T_1 - \left(\frac{R_1+R_2}{2}\right)\dot{Q}_2$$

$$T_3 = T_2 - \left(\frac{R_1+R_2}{2}\right)\dot{Q}_3$$

1.2 Exercise 2.2 Circuit Analysis:

Since the circuit model for exercise 2.2 is non-planar, the temperatures should be determined using nodal analysis. Since the nodes in consideration do not directly interact with one another, the equations are easily separable:

$$\frac{\frac{T_H}{s}-T_1}{\frac{R_1}{2}} = \frac{T_1-\frac{T_0}{s}}{\frac{1}{sC_1}} + \frac{T_1-\frac{T_L}{s}}{\frac{R_1}{2}} \iff T_1\left(\frac{4}{R_1}+sC_1\right) = T_H\left(\frac{2}{sR_1}\right) + T_L\left(\frac{2}{sR_1}\right) + T_0(C_1)$$

$$\frac{\frac{T_H}{s}-T_2}{\frac{R_2}{2}} = \frac{T_2-\frac{T_0}{s}}{\frac{1}{sC_2}} + \frac{T_2-\frac{T_L}{s}}{\frac{R_2}{2}} \iff T_2\left(\frac{4}{R_2}+sC_2\right) = T_H\left(\frac{2}{sR_2}\right) + T_L\left(\frac{2}{sR_2}\right) + T_0(C_2)$$

$$\frac{\frac{T_H}{s}-T_3}{\frac{R_1}{2}} = \frac{T_3-\frac{T_0}{s}}{\frac{1}{sC_1}} + \frac{T_3-\frac{T_L}{s}}{\frac{R_1}{2}} \iff T_3\left(\frac{4}{R_1}+sC_1\right) = T_H\left(\frac{2}{sR_1}\right) + T_L\left(\frac{2}{sR_1}\right) + T_0(C_1)$$

Once the temperatures at each node are found, the heat transfer rates into and out of each node can be found with the governing equations

$$\dot{Q}_{1,in} = \frac{\frac{T_H}{s}-T_1}{\frac{R_1}{2}} \qquad \dot{Q}_{2,in} = \frac{\frac{T_H}{s}-T_2}{\frac{R_2}{2}} \qquad \dot{Q}_{3,in} = \frac{\frac{T_H}{s}-T_3}{\frac{R_1}{2}}$$

$$\dot{Q}_{1,out} = \frac{T_1-\frac{T_L}{s}}{\frac{R_1}{2}} \qquad \dot{Q}_{2,out} = \frac{T_2-\frac{T_L}{s}}{\frac{R_2}{2}} \qquad \dot{Q}_{3,out} = \frac{T_3-\frac{T_L}{s}}{\frac{R_1}{2}}$$

The temperatures of the nodes and the heat transfer rates are solved for in the Laplace domain in the equations above, but they can be converted into the time domain and plotted using the impulse function in MATLAB. The resulting plots are show below.

1.3 Plots:

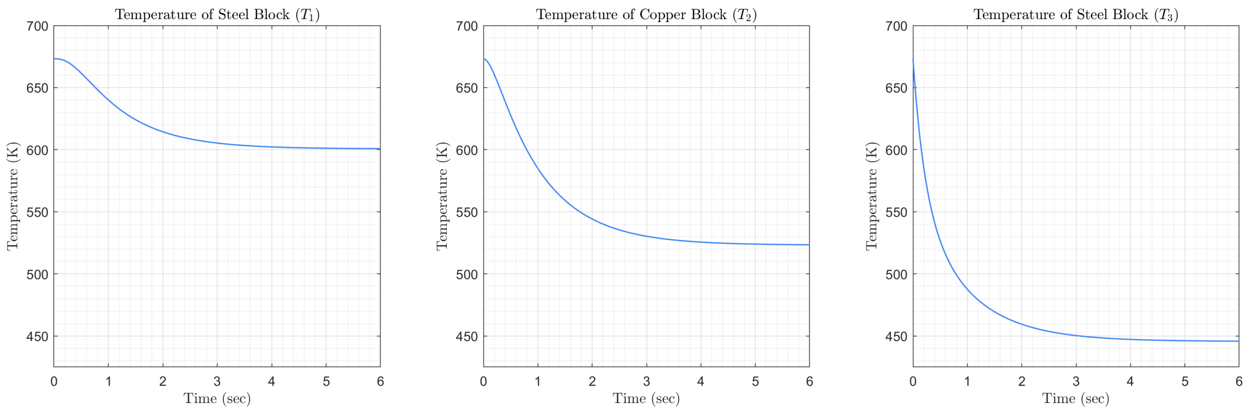


Figure 3: Temperature of Nodes in Exercise 2.1

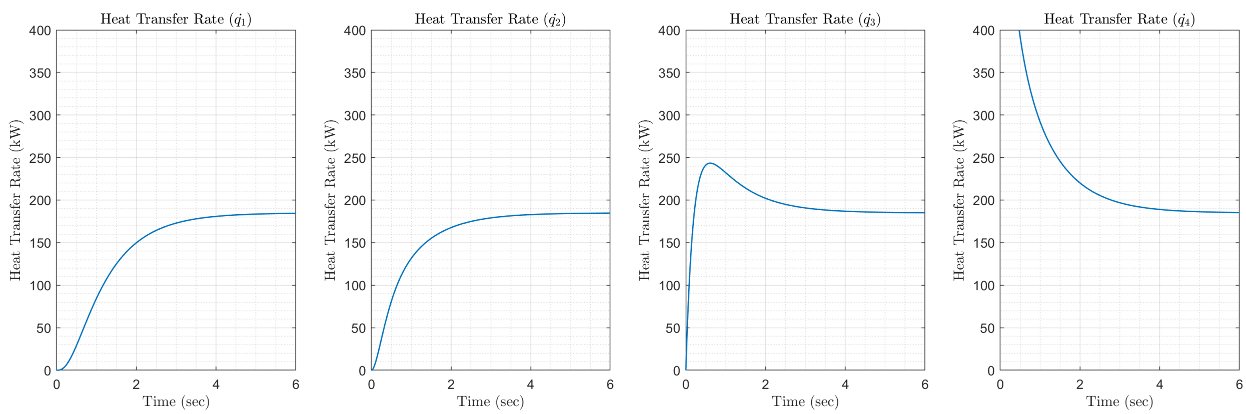


Figure 4: Heat Transfer Rates in Exercise 2.1

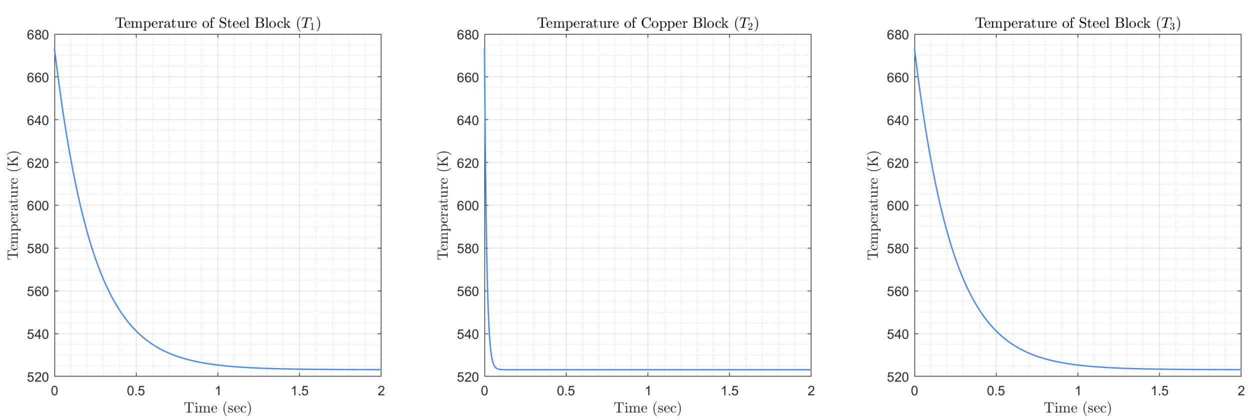


Figure 5: Temperature of Nodes in Exercise 2.2

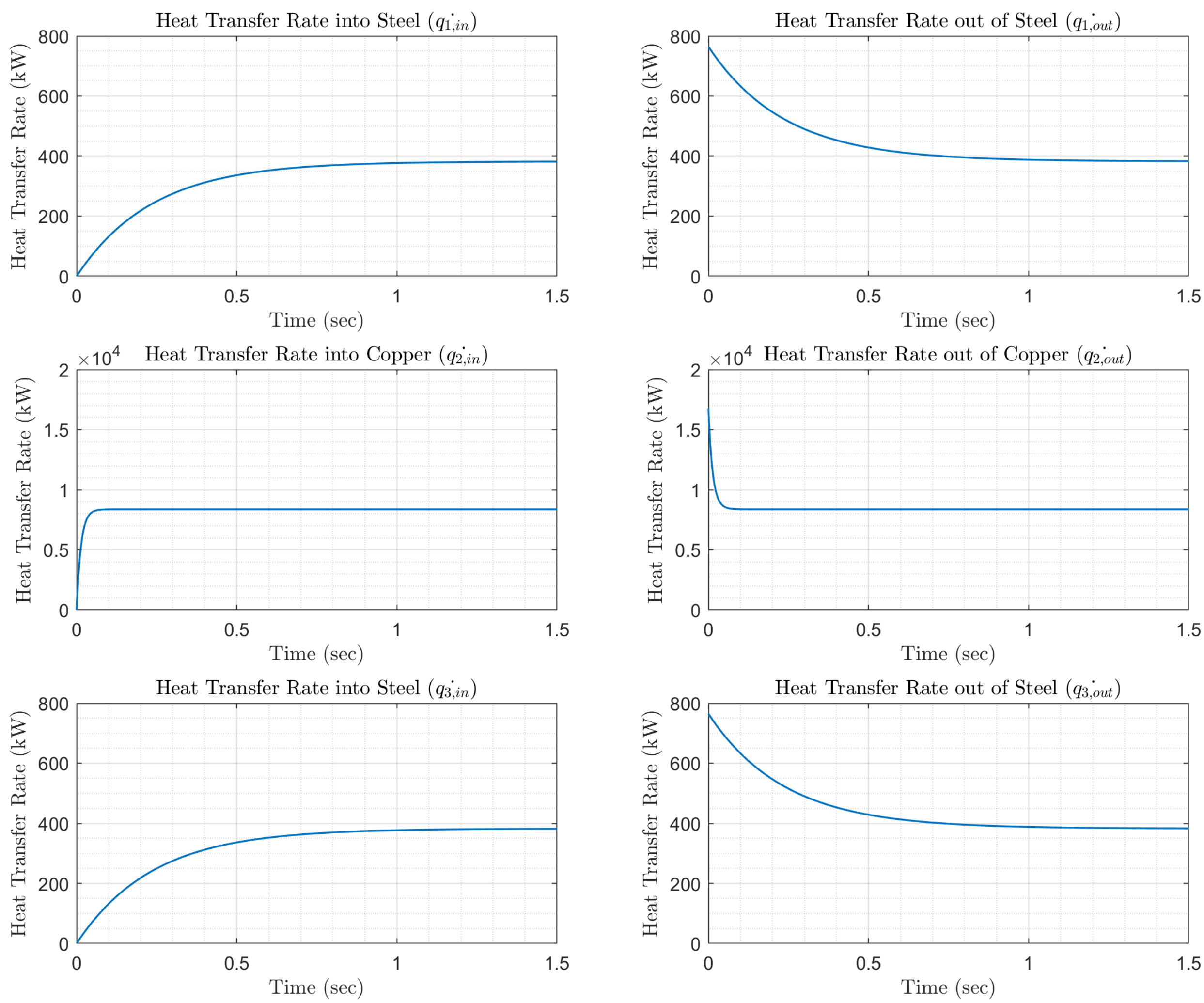


Figure 6: Heat Transfer Rates in Exercise 2.2

1.4 Steady-State Values

For both systems, it was assumed that the only type of heat transfer occurring is conduction, so the boundaries are not modelled with their own temperatures. The only values that need to be discussed are the temperatures of the metal blocks and the heat transfer rates at the boundaries.

For exercise 2.1, it takes about 5 seconds for the system to reach steady-state. The steady-state temperature of the left-most steel block is about 600 K (326.85 °C), the steady-state temperature of the copper block is approximately 525 K (251.85 °C), and the temperature of the right-most steel block is approximately 445 K (171.85 °C). Without explicitly calculating the limiting values, one could expect that the steady-state values would fall somewhere between T_L (373.15 K) and T_H (673.15 K). It also makes sense that the metals closer to T_H would have the higher steady-state temperatures.

For exercise 2.2, it takes the steel blocks about two seconds and the copper block about 0.2 seconds to reach their steady-state temperatures. Each block reaches a steady-state temperature of about 525 K (251.85 °C).

It makes sense that all the blocks would reach the same steady-state value because the are all the same width and the same distance from T_H and T_L. It also makes sense that the heat transfer rates would be different based on material properties because heat can flow through each block individually, unlike in exercise 2.1.

1.5 Discussion

Final Results

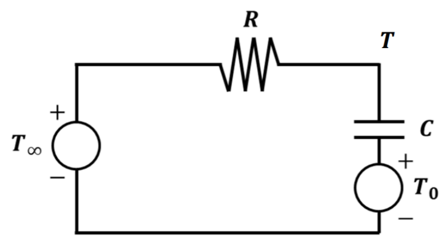
- Each block in exercise 2.1 reaches steady-state in about **5 seconds**.
- The leftmost steel block in 2.1 has a steady-state temperature of **600 K (326.85 °C)**.
- The copper block in 2.1 has a steady-state temperature of **525 K (251.85 °C)**.
- The rightmost steel block in 2.1 has a steady-state temperature of **445 K (171.85 °C)**.
- Each block in exercise 2.2 reaches a steady-state temperature of **525 K (251.85 °C)**.
- The steel blocks in 2.2 take about **2.0 seconds** to reach steady-state.
- The copper block in 2.2 takes about **0.2 seconds** to reach steady-state.

Concluding Statements

Exercise 2.1 shows heat transfer occurring through elements in series while exercise 2.2 shows heat transfer occurring through elements connected in parallel. At steady-state, elements connected in series will all experience the same heat transfer rate, but the temperature drop across the elements will not necessarily be the same. In parallel, however,the temperature drop across each element will be the same, but the heat transfer rate will depend on the material properties. This is completely analogous to properties of electric circuits in parallel and series.

2 Sinusoidal Ambient Temperature

Problem Statement: Input a time-varying forcing function for Homework 2. Make T_∞ a sine function with an amplitude of A = 25°C, an average value of 25°C, and a period of 2 minutes. Plot the temperature at node T and the heat transfer of the system as a function of time. Make R a vector and loop over the varying values with the new forcing function. How does varying R affect the behavior of the system?



Results and Analysis:

2.1 Calculations

To describe the sinusoidal forcing function in the time domain, one needs to specify the amplitude, frequency, phase shift, and vertical shift. The amplitude is given ($A = 25$K). Since $T_{period} = \dfrac{2\pi}{\omega}$ and it is known that $T_{period} = 120\,\text{sec}$, one finds that $\omega = \frac{2\pi}{120} = \frac{\pi}{60}$. The average temperature gives the vertical offset, but it need to be converted from Celsius to Kelvin by adding 273.15 K. Finally, it is assumed that this is a purely sinusoidal function and has no phase shift. So, in the time domain, the forcing function is

$$T_\infty(t) = 25\sin\left(\frac{\pi}{60}t\right) + 298.15$$

For most calculations, it is more convenient to have this in the frequency domain, so it converted using the Laplace transform:

$$\mathcal{L}\{T_\infty(t)\} = 25\mathcal{L}\{\sin\left(\frac{\pi}{60}t\right)\} + \mathcal{L}\{298.15\}$$

$$T_\infty(s) = 25\left(\frac{\frac{\pi}{60}}{s^2 + (\frac{\pi}{60})^2}\right) + \frac{298.15}{s}$$

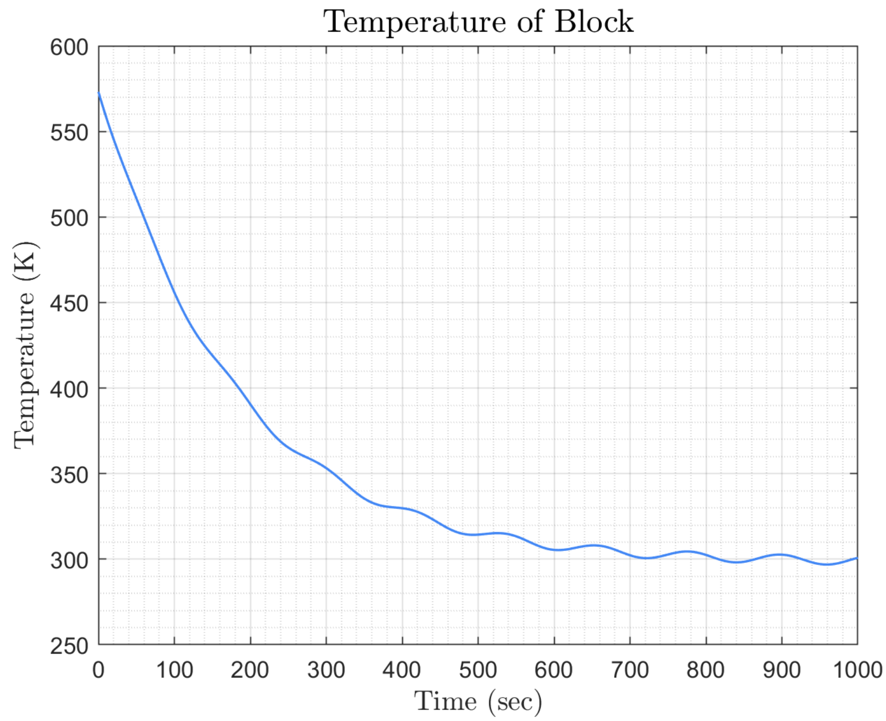
This can now be used to solve for the heat transfer rate of the system and the temperature of the block:

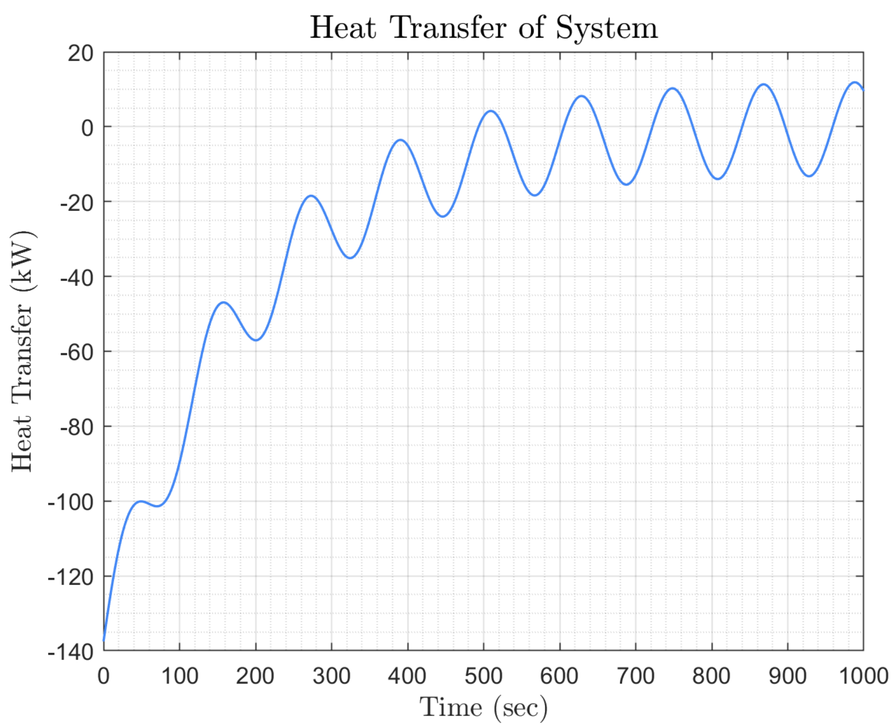
$$Q(s) = \frac{T_\infty(s) - \frac{T_0}{s}}{R + \frac{1}{sC}}$$

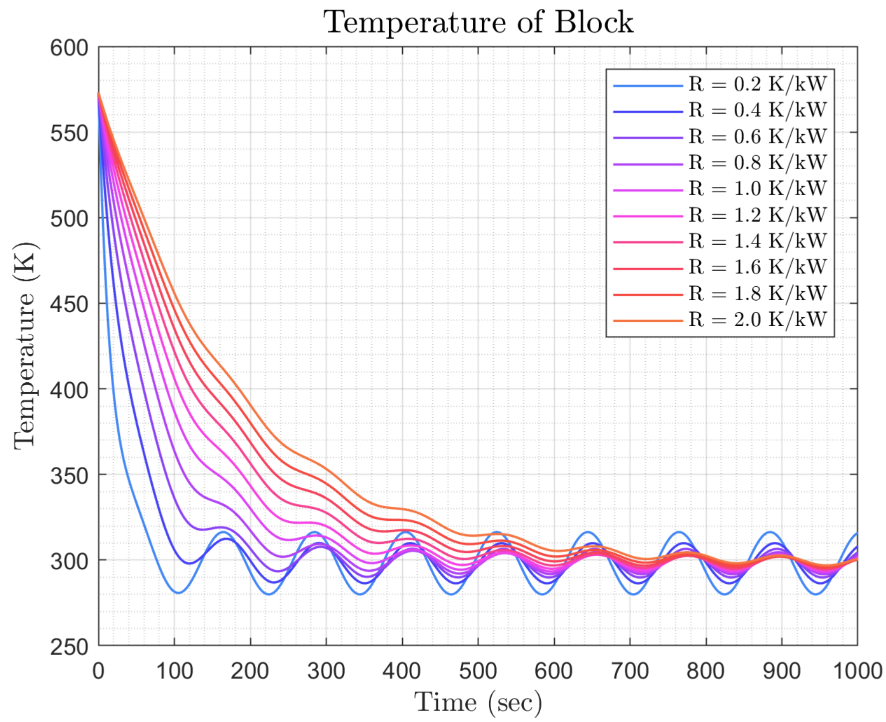
$$T(s) = T_\infty(s) - RQ(s)$$

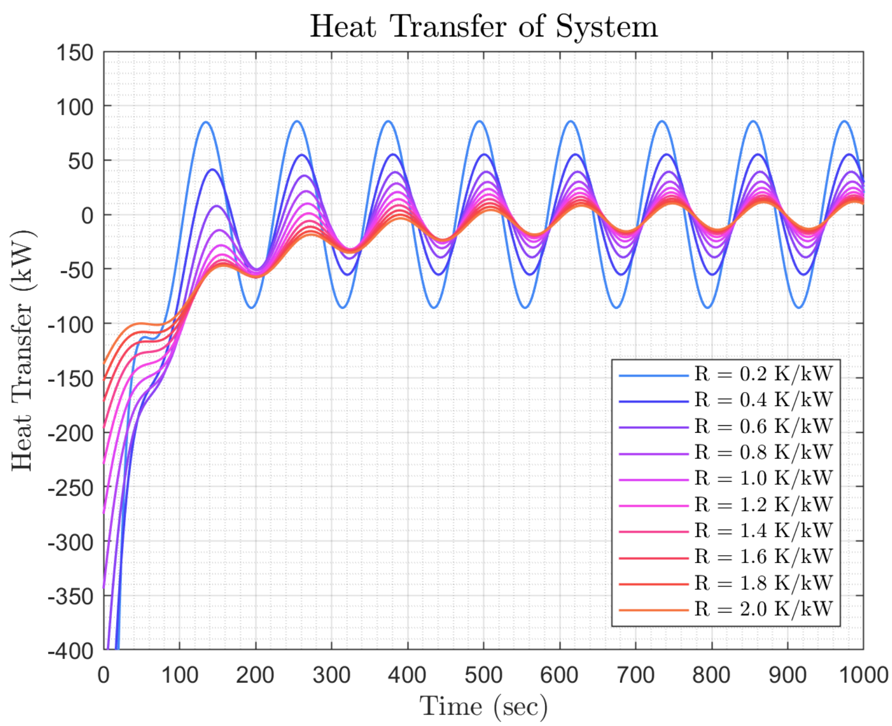
Once again, these equations are converted and plotted in the time domain using the impulse function.

2.2 Plots


Temperature of Block
Temperature (K)
Time (sec)
600
550
500
450
400
350
300
250
0
100
200
300
400
500
600
700
800
900
1000


Heat Transfer of System
Heat Transfer (kW)
Time (sec)
20
0
-20
-40
-60
-80
-100
-120
-140
0
100
200
300
400
500
600
700
800
900
1000


Temperature of Block
Temperature (K)
Time (sec)
R = 0.2 K/kW
R = 0.4 K/kW
R = 0.6 K/kW
R = 0.8 K/kW
R = 1.0 K/kW
R = 1.2 K/kW
R = 1.4 K/kW
R = 1.6 K/kW
R = 1.8 K/kW
R = 2.0 K/kW


Heat Transfer of System
Heat Transfer (kW)
Time (sec)
R = 0.2 K/kW
R = 0.4 K/kW
R = 0.6 K/kW
R = 0.8 K/kW
R = 1.0 K/kW
R = 1.2 K/kW
R = 1.4 K/kW
R = 1.6 K/kW
R = 1.8 K/kW
R = 2.0 K/kW

2.3 Effect of R on System

The plots above indicate that smaller values of R make the system more sensitive to changes in ambient temperature. Both the temperature and the heat transfer rate oscillate with larger amplitudes when R is small. Large values of R make the system react slower to changes in ambient temperature. The phase is more offset when R is large and the amplitude of oscillations are smaller.

2.4 Discussion

Final Results

- **The temperature of the block oscillates** as time reaches infinity.
- **The heat transfer rate of the block oscillates** as time reaches infinity.
- **Small R values make the temperature and heat transfer rate more sensitive** to changes in ambient temperature.

Concluding Statements

As time goes to infinity, the average temperature tends toward the ambient temperature and the average rate of heat transfer tends to zero. Since there is a sinusoidal forcing function rather than just an initial condition, both the temperature and heat transfer rates will continue to change sinusiodally, as seen in the plots above.

Initially there is a large difference in the temperature, which results in a greater rate of heat transfer and steeper change in the temperature. However over time, the system tends to oscillate and a consistent average.

Appendix: MATLAB Code

Contents

Rose Gebhardt and Harris Paspuleti – February 12, 2020 – Homework 3 Question 1

```
clear all; close all; clc;
```

Parameters Used in Both Problems

```
s = tf('s');
time_1 = linspace(0,6,1000);
time_2 = linspace(0,2,1000);

% Define material properties
rho_st = 8000; % kg/m^3
rho_cu = 8960; % kg/m^3
c_st = 502.416; % J/kgK
c_cu = 376.812; % J/kgK
k_st = 17; % W/mK
k_cu = 372; % W/mK

% Citations
    % https://www.engineersedge.com/materials/specific_heat_capacity_of_metals_13259.htm
    % https://hypertextbook.com/facts/2004/KarenSutherland.shtml
    % https://amesweb.info/Materials/Density_of_Copper.aspx

% Define dimensions
A = 0.3*0.5; % m^2
x_st = 0.002; % m
x_cu = 0.003; % m
x = 0.002; % m
```

```
V_st = A*x_st; % m^3
V_cu = A*x_cu; % m^3

% Define constant temperatures
T_H = 400 + 273.15; % K
T_L = 100 + 273.15; % K
T_0 = 400 + 273.15; % K
```

Example 2.1

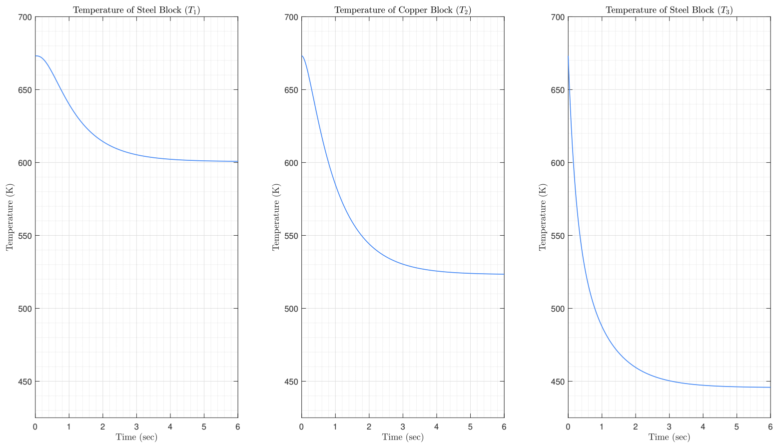
```
% Define variables
R_st1 = x_st/(k_st*A); % K/W
R_cu1 = x_cu/(k_cu*A); % K/W
C_st1 = rho_st*V_st*c_st; % J/K
C_cu1 = rho_cu*V_cu*c_cu; % J/K

% Define impedeances
Z_st1 = 1/(s*C_st1);
Z_cu1 = 1/(s*C_cu1);

% Define terms used in mesh analysis
z_1 = R_st1/2 + Z_st1;
z_2 = -1*Z_st1;
z_3 = R_st1/2 + R_cu1/2 + Z_st1 + Z_cu1;
z_4 = -1*Z_cu1;

% Do mesh analysis
Z = [z_1, z_2, 0, 0; z_2, z_3, z_4  0; 0, z_4, z_3, z_2; 0, 0, z_2, z_1];
f = [(T_H - T_0)/s; 0; 0; (T_0 - T_L)/s];

% Find heat transfer rate
q_dot_1 = Z\f;

% Calculate temperature at nodes
T_11 = (T_H/s) - (R_st1/2)*q_dot_1(1);
T_21 = (T_11) - (R_st1 + R_cu1)*q_dot_1(2)/2;
T_31 = (T_21) - (R_st1 + R_cu1)*q_dot_1(3)/2;

% Convert to time domain
t_11 = impulse(T_11,time_1); % K
t_21 = impulse(T_21,time_1); % K
t_31 = impulse(T_31,time_1); % K
q_11 = impulse(q_dot_1(1),time_1)/1000; % kW
q_21 = impulse(q_dot_1(2),time_1)/1000; % kW
```

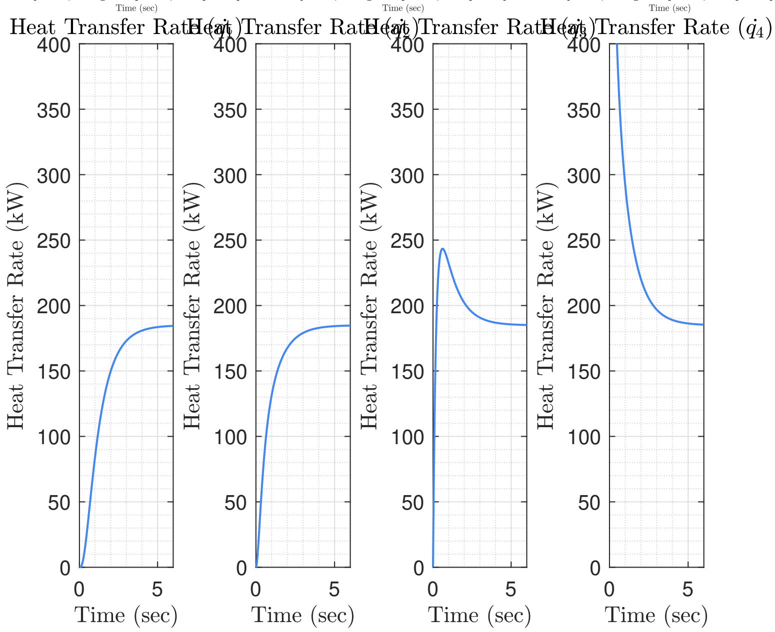
```
q_31 = impulse(q_dot_1(3),time_1)/1000; % kW
q_41 = impulse(q_dot_1(4),time_1)/1000; % kW
```

Example 2.1 Plots

```
set(gcf, 'Position', get(0, 'Screensize'));

% Temperature at Nodes
figure(1)
subplot(1,3,1)
plot(time_1,t_11,'LineWidth',1)
title('Temperature of Steel Block ($T_1$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([425,700]);

subplot(1,3,2)
plot(time_1,t_21,'LineWidth',1)
title('Temperature of Copper Block ($T_2$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([425,700]);

subplot(1,3,3)
plot(time_1,t_31,'LineWidth',1)
title('Temperature of Steel Block ($T_3$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([425,700]);

set(gcf, 'Position', get(0, 'Screensize'));

% Heat Transfer Rate at Boundaries
figure(2)
subplot(1,4,1)
```

```
plot(time_1,q_11,'LineWidth',1)
title('Heat Transfer Rate ($\dot{q_1}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([0,400]);

subplot(1,4,2)
plot(time_1,q_21,'LineWidth',1)
title('Heat Transfer Rate ($\dot{q_2}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([0,400]);

subplot(1,4,3)
plot(time_1,q_31,'LineWidth',1)
title('Heat Transfer Rate ($\dot{q_3}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([0,400]);

subplot(1,4,4)
plot(time_1,q_41,'LineWidth',1)
title('Heat Transfer Rate ($\dot{q_4}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,6]);
ylim([0,400]);
```





Example 2.2

```
% Define variables
R_st2 = x/(k_st*A); % K/W
R_cu2 = x/(k_cu*A); % K/W
C_st2 = rho_st*V_st*c_st; % J/K
C_cu2 = rho_cu*V_cu*c_cu; % J/K

% Define impedeances
Z_st2 = 1/(s*C_st2);
Z_cu2 = 1/(s*C_cu2);

% Define numerator and denominator used in nodal analysis
d_1 = (4/R_st2) + (1/Z_st2);
d_2 = (4/R_cu2) + (1/Z_cu2);
n_1 = ((2*T_H)/(s*R_st2)) + ((2*T_L)/(s*R_st2)) + T_0*C_st2;
n_2 = ((2*T_H)/(s*R_cu2)) + ((2*T_L)/(s*R_cu2)) + T_0*C_cu2;
```

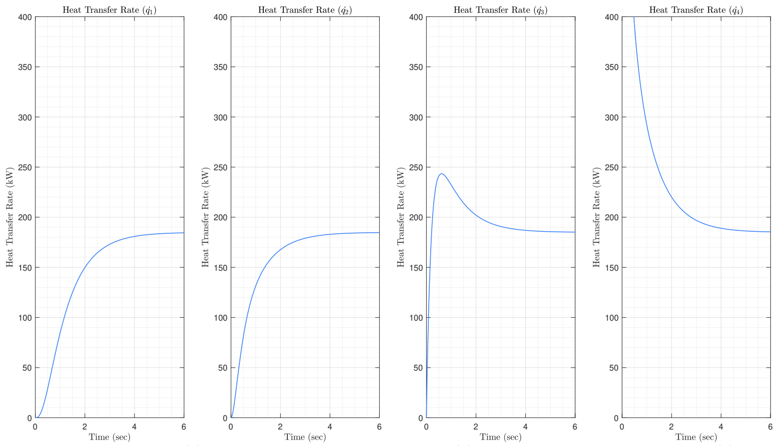
```
% Calculate temperature at nodes
T_12 = n_1/d_1;
T_22 = n_2/d_2;
T_32 = n_1/d_1;

% Calculate heat transfer rates
Q_12_in = ((T_H/s) - T_12)/(R_st2/2);
Q_22_in = ((T_H/s) - T_22)/(R_cu2/2);
Q_32_in = ((T_H/s) - T_32)/(R_st2/2);
Q_12_out = ((T_12 - T_L/s))/(R_st2/2);
Q_22_out = ((T_22 - T_L/s))/(R_cu2/2);
Q_32_out = ((T_32 - T_L/s))/(R_st2/2);

% Convert to time domain
t_12 = impulse(T_12,time_2); % K
t_22 = impulse(T_22,time_2); % K
t_32 = impulse(T_32,time_2); % K
q_12_in = impulse(Q_12_in,time_2)/1000; % kW
q_22_in = impulse(Q_22_in,time_2)/1000; % kW
q_32_in = impulse(Q_32_in,time_2)/1000; % kW
q_12_out = impulse(Q_12_out,time_2)/1000; % kW
q_22_out = impulse(Q_22_out,time_2)/1000; % kW
q_32_out = impulse(Q_32_out,time_2)/1000; % kW
```

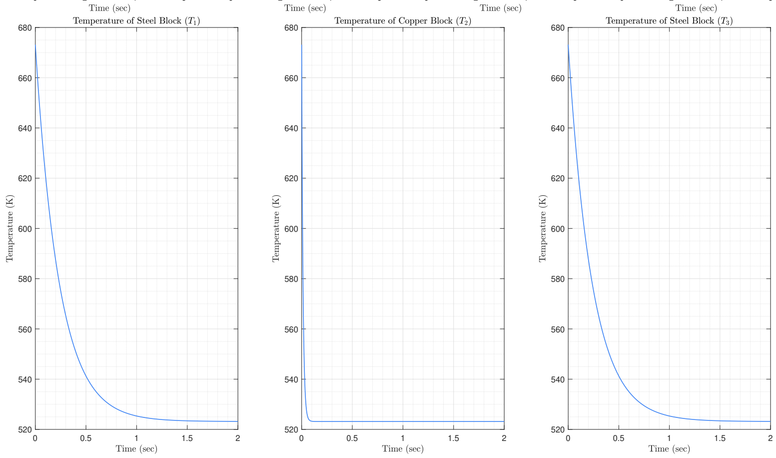
Example 2.2 Plots

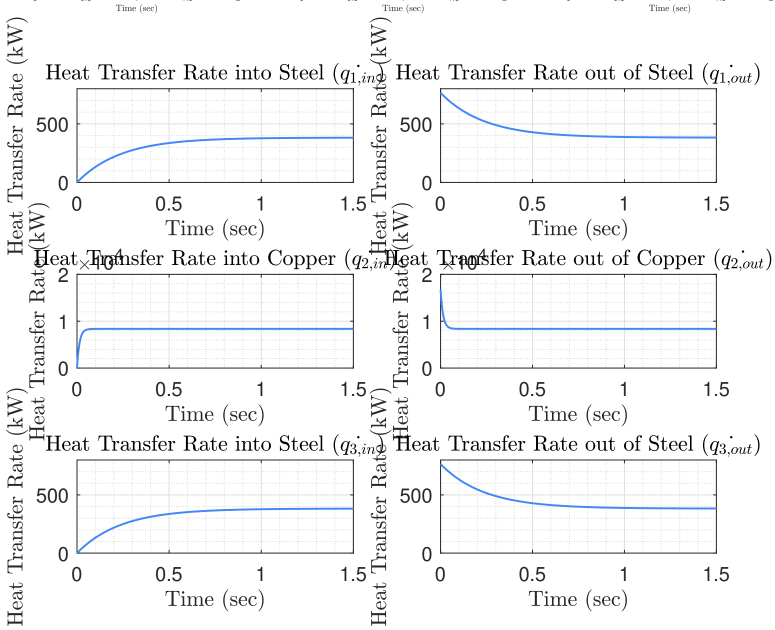
```
set(gcf, 'Position', get(0, 'Screensize'));

% Temperature at Nodes
figure(3)
subplot(1,3,1)
plot(time_2,t_12,'LineWidth',1)
title('Temperature of Steel Block ($T_1$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,2]);
ylim([520,680]);

subplot(1,3,2)
plot(time_2,t_22,'LineWidth',1)
title('Temperature of Copper Block ($T_2$)','interpreter','latex')
```

```
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,2]);
ylim([520,680]);

subplot(1,3,3)
plot(time_2,t_32,'LineWidth',1)
title('Temperature of Steel Block ($T_3$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Temperature (K)','interpreter','latex')
grid on
grid minor
xlim([0,2]);
ylim([520,680]);

set(gcf, 'Position', get(0, 'Screensize'));

% Heat Transfer Rates at Boundaries
figure(4)
subplot(3,2,1)
plot(time_2,q_12_in,'LineWidth',1)
title('Heat Transfer Rate into Steel ($\dot{q_{1,in}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,800]);

subplot(3,2,2)
plot(time_2,q_12_out,'LineWidth',1)
title('Heat Transfer Rate out of Steel ($\dot{q_{1,out}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,800]);

subplot(3,2,3)
plot(time_2,q_22_in,'LineWidth',1)
```

```
title('Heat Transfer Rate into Copper ($\dot{q_{2,in}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,20000]);

subplot(3,2,4)
plot(time_2,q_22_out,'LineWidth',1)
title('Heat Transfer Rate out of Copper ($\dot{q_{2,out}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,20000]);

subplot(3,2,5)
plot(time_2,q_32_in,'LineWidth',1)
title('Heat Transfer Rate into Steel ($\dot{q_{3,in}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,800]);

subplot(3,2,6)
plot(time_2,q_32_out,'LineWidth',1)
title('Heat Transfer Rate out of Steel ($\dot{q_{3,out}}$)','interpreter','latex')
xlabel('Time (sec)','interpreter','latex')
ylabel('Heat Transfer Rate (kW)','interpreter','latex')
grid on
grid minor
xlim([0,1.5]);
ylim([0,800]);
```


Heat Transfer Rate ($\dot{q}_1$)
Heat Transfer Rate ($\dot{q}_2$)
Heat Transfer Rate ($\dot{q}_3$)
Heat Transfer Rate ($\dot{q}_4$)
Heat Transfer Rate (kW)
Time (sec)


Temperature of Steel Block (T_1)
Temperature of Copper Block (T_2)
Temperature of Steel Block (T_3)
Temperature (K)
Time (sec)


Heat Transfer Rate into Steel ($\dot{q_{1,in}}$)
Heat Transfer Rate out of Steel ($\dot{q_{1,out}}$)
Heat Transfer Rate into Copper ($\dot{q_{2,in}}$)
Heat Transfer Rate out of Copper ($\dot{q_{2,out}}$)
Heat Transfer Rate into Steel ($\dot{q_{3,in}}$)
Heat Transfer Rate out of Steel ($\dot{q_{3,out}}$)
Heat Transfer Rate (kW)
Time (sec)

Contents

Rose Gebhardt and Harris Paspuleti – February 12, 2020 – Homework 3 Question 2

```
clear all; close all; clc;
```

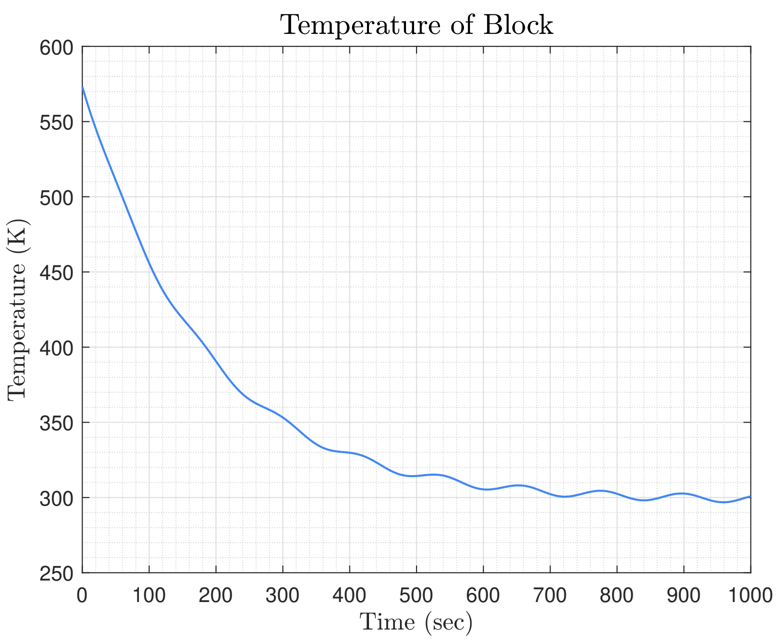
Define parameters used

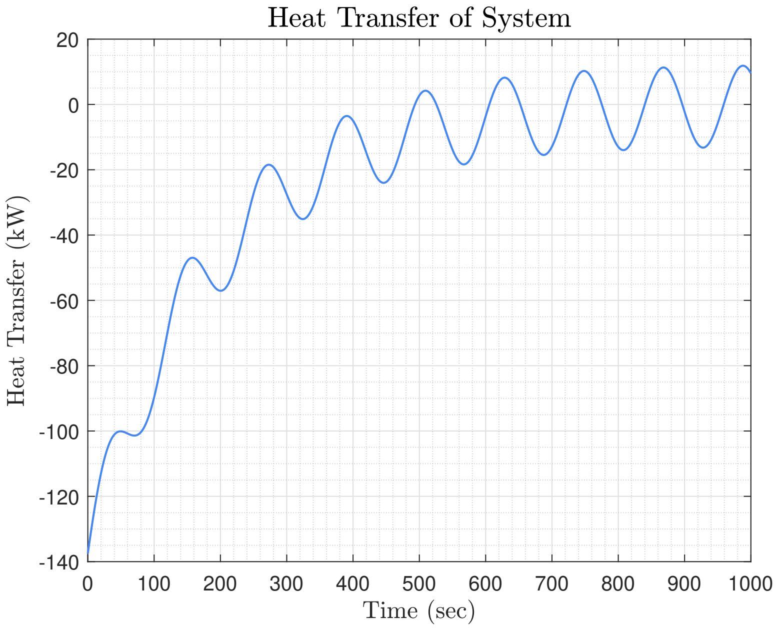
```
% Define system
s = tf('s');
m = 200; % kg
c = 0.45; % kJ/kgK
R = 2; % K/kW
C = m*c; % kJ/K
T_room = 25 + 273.15; % K
T_0 = 300 + 273.15; % K
omega = pi/60; % rad/s

% Vectors of varying time, resistance, and capacitance
time = linspace(0,1000,1000);

% Color
chromatic = [66, 135, 245; 69, 66, 245; 138, 66, 245; 176, 66, 245; 221, 66, 245;...
    245, 66, 227; 245, 66, 144; 245, 66, 93; 245, 75, 66; 245, 117, 66]/255;
set(groot,'defaultAxesColorOrder',chromatic)
```

1. Plot System in Frequency Domain

```
% Define ambient temperature in frequency domain
T_inf = (25*omega)/(s^2 + omega^2) + (T_room/s);

% Define heat transfer rate in frequency domain
Q_dot = (T_inf - (T_0/s))/(R + (1/(s*C)));

% Define temperature of block in frequency domain
T = T_inf - R*Q_dot;

% Switch to time domain
t_inf = impulse(T_inf,time); % Used to compare phase
q_dot = impulse(Q_dot,time);
t = impulse(T,time);
```

```
% Plot temperature
figure(1)
plot(time,t,'LineWidth',1)
title('Temperature of Block','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (K)','interpreter','latex','FontSize',12)
grid on
grid minor
xlim([0,1000]);
ylim([250,600]);

% Plot heat transfer
figure(2)
plot(time,q_dot,'LineWidth',1)
title('Heat Transfer of System','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Heat Transfer (kW)','interpreter','latex','FontSize',12)
grid on
grid minor
xlim([0,1000]);
ylim([-140,20]);
```





2. Vary resistance values

```
% Get temperature and heat transfer behavior for several resistances
t_R = zeros(length(time),10);
q_R = zeros(length(time),10);
for index = 1:10
    Q_it = (T_inf - (T_0/s))/((index*0.2) + (1/(s*C)));
    T_it = T_inf - index*0.2*Q_it;
    q_R(:,index) = impulse(Q_it,time);
    t_R(:,index) = impulse(T_it,time);
end

% Plot temperature for several resistances
figure(3)
plot(time,t_R,'LineWidth',1)
title('Temperature of Block','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (K)','interpreter','latex','FontSize',12)
legendTR = legend('R = 0.2 K/kW','R = 0.4 K/kW','R = 0.6 K/kW','R = 0.8 K/kW','R = 1.0 K/kW'...
    ,'R = 1.2 K/kW','R = 1.4 K/kW','R = 1.6 K/kW','R = 1.8 K/kW','R = 2.0 K/kW');
set(legendTR,'Interpreter','latex');
grid on
grid minor
xlim([0,1000]);
ylim([250,600]);

% Plot heat transfer for several resistances
figure(4)
```

```
plot(time,q_R,'LineWidth',1)
title('Heat Transfer of System','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Heat Transfer (kW)','interpreter','latex','FontSize',12)
legendQR = legend('R = 0.2 K/kW','R = 0.4 K/kW','R = 0.6 K/kW','R = 0.8 K/kW','R = 1.0 K/kW'...
    ,'R = 1.2 K/kW','R = 1.4 K/kW','R = 1.6 K/kW','R = 1.8 K/kW','R = 2.0 K/kW','Location','southeast');
set(legendQR,'Interpreter','latex');
grid on
grid minor
xlim([0,1000]);
ylim([-400,150]);
```