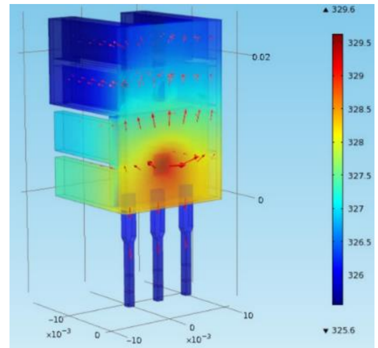
Temperature Control Lab A: SISO (Single Input, Single Output) Model



The temperature control lab is a hands-on lab where an objective is to develop a transient model between the heater power and temperature. An energy balance describes the transient temperature response of a single heater with one temperature sensor. The heater and temperature sensor are assumed to be at the same temperature. You can assume that conduction is negligible and that the only heat transferred is through radiation to the surroundings or convection to the surrounding air. The heater is initially off and the heater and sensor are initially at ambient temperature. Use the following values:

Quantity	**Value**	**Value (SI Units)**
Initial temperature (T_0)	23˚C	296.15 K
Ambient temperature (T_∞)	23˚C	296.15 K
Heater Output (Q)	0 to 3 W	0 to 3 W
Heat Capacity (C_p)	500 J/kg-K	500 J/kg-K
Surface Area (A)	12 cm^2	1.2×10^{-3} m^2
Mass (m)	4 gm	0.004 kg
Overall Heat Transfer Coefficient (U)	10 W/m^2-K	10 W/m^2-K
Emissivity (ε)	0.9	0.9
Stefan Boltzmann Constant (σ)	5.67×10^{-8} W/m^2-K^4	5.67×10^{-8} W/m^2-K^4

Starting with the following energy balance equation with enthalpy (h), convective heat transfer, radiative heat transfer, and the heater energy input (Q):

$$mC_p \frac{dT}{dt} = UA\,(T_\infty - T) + \epsilon\,\sigma\,A\,(T_\infty^4 - T^4)\; +\; Q$$

Simulate a temperature change when changing the heater output from 0 to 1 Watt for 10 minutes. Show the temperature response as it reaches a new steady state value.

See http://apmonitor.com/pdc/index.php/Main/ArduinoModeling for source code solutions (if needed).

Below are code excerpts to help in getting started with the energy balance simulation.

Starting ODEINT Python Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# define energy balance model
def heat(x,t,Q):
    # Parameters
    Ta = 23 + 273.15    # K
    U = 10.0            # W/m^2-K
    m = 4.0/1000.0      # kg
    Cp = 0.5 * 1000.0   # J/kg-K
    A = 12.0 / 100.0**2 # Area in m^2
    alpha = 0.01        # W / % heater
    eps = 0.9           # Emissivity
    sigma = 5.67e-8     # Stefan-Boltzman
```

Starting ode23 MATLAB Code

```
% save as heat.m
% define energy balance model
function dTdt = heat(time,x,Q)
    % Parameters
    Ta = 23 + 273.15;    % K
    U = 10.0;            % W/m^2-K
    m = 4.0/1000.0;      % kg
    Cp = 0.5 * 1000.0;   % J/kg-K
    A = 12.0 / 100.0^2;  % Area in m^2
    alpha = 0.01;        % W / % heater
    eps = 0.9;           % Emissivity
    sigma = 5.67e-8;     % Stefan-Boltzman
```

Starting GEKKO Python Code

```
#Load packages
import numpy as np
import matplotlib.pyplot as plt
from gekko import GEKKO

#initialize GEKKO model
m = GEKKO()

#model discretized time
n = 60*10+1  # Number of second time points (10min)
m.time = np.linspace(0,n-1,n) # Time vector

# Parameters
Q = m.Param(value=100.0) # Percent Heater (0-100%)

T0 = m.Param(value=23.0+273.15)     # Initial temperature
Ta = m.Param(value=23.0+273.15)     # K
U =  m.Param(value=10.0)            # W/m^2-K
mass = m.Param(value=4.0/1000.0)    # kg
Cp = m.Param(value=0.5*1000.0)      # J/kg-K
A = m.Param(value=12.0/100.0**2)    # Area in m^2
alpha = m.Param(value=0.01)         # W / % heater
eps = m.Param(value=0.9)            # Emissivity
sigma = m.Const(5.67e-8)            # Stefan-Boltzman
```