DIAGNOSTIC 4 SOLUTION

MODELING AND SIMULATION

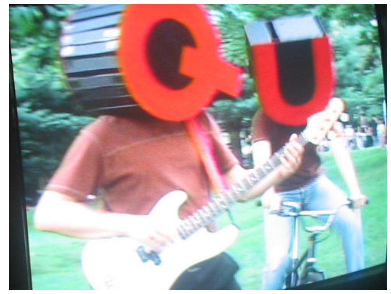

Figure 1: Image from the video "Q U," by They Might Be Giants, on their album, *Here Come the ABCs*.

copper_func

To convert `copper_script.m` to `copper_func.m`: (1) Add the function signature at the beginning. (2) Add `end` at the end. (3) Indent the body. (4) Make `T_block` an input variable (rather than a local variable). (5) Assign the result, `dTdt`, to the output variable, `res`. Here's what the result looks like:

```
function res = copper_func(t, T_block)
    % t is time in s
    % T_block is the temperature of the block in degC

    m = 10;          % mass of the block in kg
    T_env = 0;       % temperature of the environment in degC

    % find the size and area of the block
    density = 8.94 * 0.001 / 0.01^3;   %  g/cm3 * (kg/g) / (m/cm)^3
    volume = m / density;              %  m^3

    length = volume^(1/3);             %  m
    area = 6 * length^2;               %  m^2

    % convert american r-value to SI r-value
    r_value_english = 30;                  % dram fathoms per fortnight
    r_value = r_value_english * 0.1761;    % K m^2 / W

    % compute the rate of heat (energy) flow, dU/dt
    deltaT = T_env - T_block;              % K (also degC)
    dUdt = 1/r_value * area * deltaT;      % W (aka J/s)

    % compute the rate of temperature change, dT/dt
    c = 386;                     % specific heat of copper in J / (kg K)
    dTdt = dUdt / m / c;         % K / s
    res = dTdt;
end
```

In the original statement of the problem, we used Q to represent temperature (in order to avoid conflicts with t for time). But that was confusing[1] because the handout we gave you used Q for heat (energy). In these solutions, we will use T for temperature and U for heat (energy). So dT/dt is in units of K/s and dU/dt is in units of J/s, also known as W.

[1] We're sorry!

copper_euler

Here is a loop that uses Euler's method to estimate the temperature of the block after 24 hours.

```
t_start = 0  % initial time in seconds
t_end = 24 * 3600;      % final time in seconds
n = 4000;               % number of time steps
timestep = (t_end-t_start) / n;    % size of timestep in seconds

t(1) = t_start % initial time in s
T(1) = 30;              % initial temperature in degC

for i = 1:n
    dTdt = copper_func(0, T(i));      % temperature rate of change in degC/s
    T(i+1) = T(i) + dTdt * timestep;  % new temperature in decC
    t(i+1) = t(i) + timestep          % new time in s
end

t(end), T(end)
```

We used an integer loop variable to make it easy to store the results in a vector. `T(end)` is the last element of the vector `T`, which is the temperature of the block after 24 hours.

copper_ode45

Here is a script that uses `ode45` to estimate the temperature of the block after 24 hours.

```
t_start = 0;                  % initial time in seconds
t_end = 24 * 3600;            % final time in seconds
T_initial = 30;               % initial temperature in degC

[Time,Temp] = ode45(@copper_func, [t_start, t_end], T_initial);

Time(end), Temp(end)
```

The output variables from `ode45` are a vector of times, `Time`, and a vector of temperatures, `Temp`.