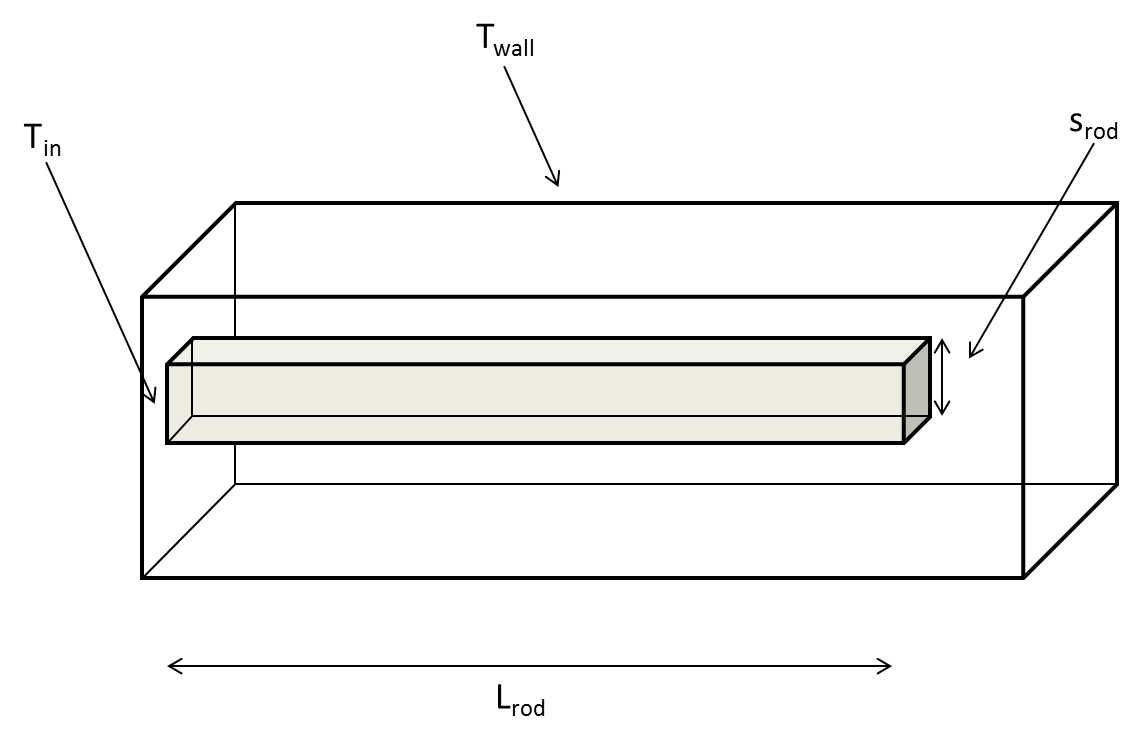
**Heated Rod in an Enclosure**

Jon Rea

4-19-15

Problem Statement:

A square rod with side length s and length L is extended in an enclosure. One end of the rod is heated to a known temperature Tin. The walls of the enclosure are kept at a known temperature Twall. The rod has thermal conductivity k and emissivity ε; conduction through the rod and radiation from the rod to the walls are significant while convective heat transfer from the rod is negligible. Find the temperature profile in the rod and the temperature at the end of the rod.



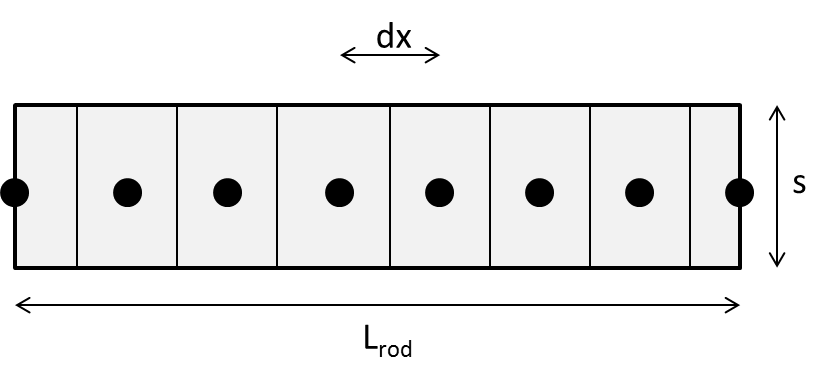
**Figure 1:**  Problem statement.

Assumptions:

1. The temperature at the base of the heated rod is fixed.
2. The temperature of the enclosure is fixed.
3. No convection.
4. The rod is a gray body such that absorptivity α is equal to emissivity ε.
5. The view factor from each rod node to the enclosure is 1.
6. The rod is in steady-state with conditions not changing in time.

**Numerical Solution**

The rod is discretized into nodes according to figure 2.



**Figure 2:**  Discretization of the rod into nodes.

An energy balance is then applied to each node in order to obtain equations for the temperature of each node in terms of the temperatures of other nodes. Drawings of each energy balance are given in figure 3.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. Node 1 | 1. Nodes i = 2:N-1 | (c) Node N |

**Figure 3:**  Energy balance applied to each node.

For node 1, because the temperature at the entrance of the rod is known, we simply use

|  |  |
| --- | --- |
|  | (1) |

For node 2, the energy balance equation becomes

|  |  |
| --- | --- |
| *= – +* | (2) |

For the final node N, the energy balance equation reduces to

|  |  |
| --- | --- |
| *= – +* | (3) |

Equations 1, 2 and 3 now represent a system of equations that can be solved using the matrix inversion method. In matrix form, we have

|  |  |
| --- | --- |
|  | (4) |

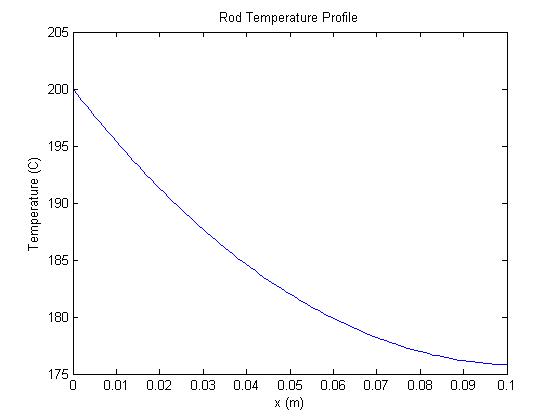
where [*T*] is a column vector of temperatures that we are solving for. Each row in [*A*] represents a node, and the values in [*A*] correspond to the coefficients of each temperature from the system of equations developed above. Terms that do not include a temperature of a node or include a radiation term of T4 are put into the column vector [*b*]. To write the code, we populate the matrix [*A*] and the vector [*b*], then solve for [*T*].

This system of equations can only be solved if we can put in values for [*b*], which requires that we have known values of Ti. However, these values are unknown because our goal of the problem is to find all Ti. We then need to make an initial guess of the temperature profile so that we can solve the system of equations for the temperature profile. This new calculated temperature profile is not correct, but gives an updated guess that is closer to the correct. This new temperature profile is used to update the initial guess at the temperature profile, and the matrix inversion method is again used to solve for a more accurate temperature profile. This process is repeated until the guess of the temperature profile is equal to the calculated temperature profile, in which case our guess is correct and we know the true temperature profile.

Table 1 gives the input parameters of the system and figure 4 shows the simulation results.

**Table 1:** Input system parameters

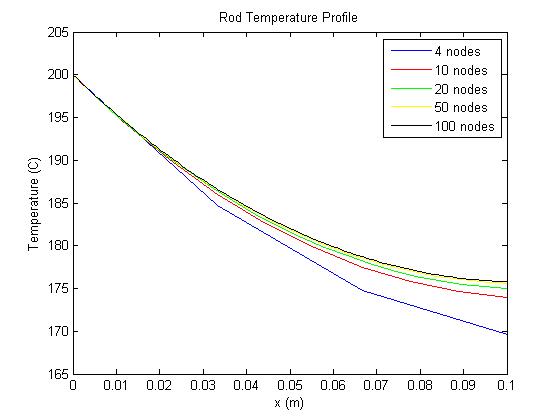
|  |  |  |
| --- | --- | --- |
| Parameter | Value | Units |
| Tin | 200 | C |
| Twall | 30 | C |
| s | .01 | m |
| L | .1 | m |
| k | 167 | W/mK |
| ε | 1 |  |



**Figure 4:**  Numerical solution for rod temperature profile

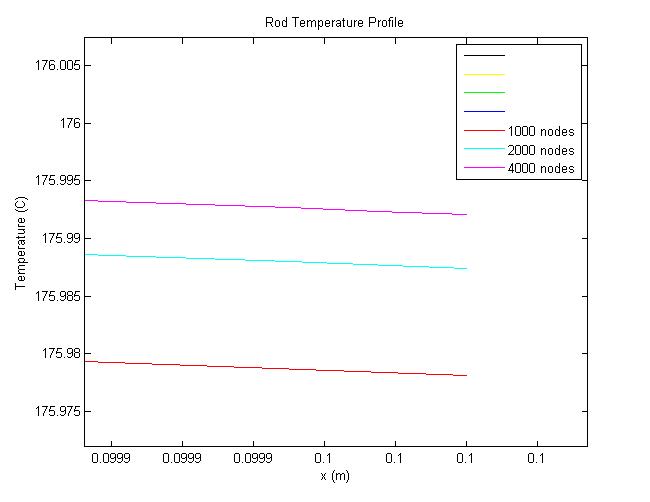
**Code Verification**

Figure 5 shows the temperature profile for different numbers of nodes. Once more than 50 nodes are used, the numerical solution has converged and does not change significantly with increased number of nodes. Because very little computation time is required, the solution in figure 4 is given for 500 nodes.



**Figure 5:**  Numerical solution convergence for > 50 nodes.

Figure 6 shows the temperature profile at the rod tip for even higher numbers of nodes.



**Figure 6:**  Numerical solution accuracy for large numbers of nodes

**MATLAB code**

% Known Values

Tin = 400 ;%C

Twall = 30 ;%C

s = 0.01 ;%m

L = 0.1 ;%m

k = 167 ;%W/mK

sigma = 5.67\*10^-8 ;%W/m^2K^4

epsilon = 0.5 ;% (Emissivity of rod)

% Discretize the rod into nodes

N = 1000;% (Number of nodes)

x = zeros(N,1) ;% (Location of each node)

dx = L/(N-1) ;% (Distance between each node)

for i = 1:N

x(i) = dx\*(i-1) ;

end

% Initialize Matrix Inversion Method

A = zeros(N) ;

b = zeros(N,1) ;

T = zeros(N,1) ;

% Guess a temperature profile

Tguess = zeros(N,1) ;

for i = 1:N

Tguess(i) = Tin ;

end

% Run a while loop until the guess matches the calculated temperature

while sum(abs(Tguess-T)) > 0.01

% Node 1

A(1,1) = 1 ;

b(1) = Tin ;

% Nodes i = 2:N-1

for i = 2:N-1

A(i,i-1) = k\*s^2/dx ;

A(i,i) = -2\*k\*s^2/dx ;

A(i,i+1) = k\*s^2/dx ;

b(i) = sigma\*epsilon\*4\*s\*dx\*((Tguess(i)+273)^4 - (Twall+273)^4) ;

end

% Node N

A(N,N-1) = k\*s^2/dx ;

A(N,N) = -k\*s^2/dx ;

b(N) = sigma\*epsilon\*(4\*s\*dx+s^2)\*((Tguess(i)+273)^4 - (Twall+273)^4) ;

% Calculate Temperatures

T = A\b ;

Tguess = Tguess + (T-Tguess)\*0.1 ;

end

% Plot Results

plot(x,T,'k')

xlabel('x (m)')

ylabel('Temperature (C)')

title('Rod Temperature Profile')