Computational Heat Transfer: Final Project

Ran Li

December 20, 2016

1 Problem Statement

Problem 8.8:

Consider a heated cylindrical rod emerging at time t = 0, from a die at temperature T_0, as shown in Fig. 8.5b. Assuming axisymmetry, develop a numerical scheme to solve for the time-dependent temperature distribution in the material, if the rod loses energy by convection to ambient fluid at a constant value of heat transfer coefficient.

2 Mathematical Analysis

This is a modified 1-D heat conduction problem with convection terms at each point on the entity. Setup of the problem is shown down below. The basic idea is to take the changing length of the rod as constant over certain time interval Δt so that by solving for transient behavior of the beam we can approximate the real phenomena.

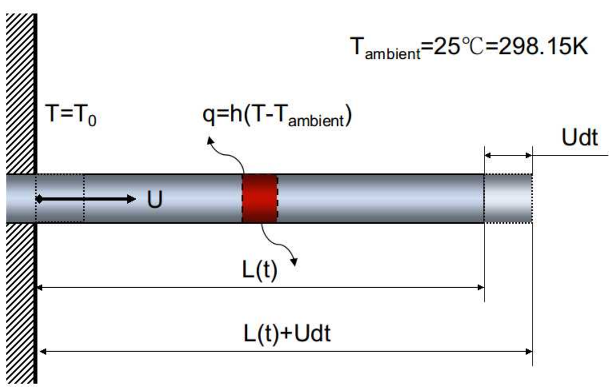

Figure 1: Problem Setup

Mathematically, governing equation for such problem is:

$$\rho C\left(\frac{\partial T}{\partial t}+U\frac{\partial T}{\partial x}\right)=k\frac{\partial^2 T}{\partial x^2}-\frac{hP}{A}\left(T-T_a\right) \tag{1}$$

Only convection term is taken into consideration as mentioned in the problem description. Boundary condition could be specified as:

$$x=0:T\left(x\right)=T_0 \tag{2}$$

$$x=L\left(t\right):-k\frac{\partial T}{\partial x}=h\left(T-T_{ambient}\right) \tag{3}$$

Since $L\left(t\right)$ is constantly changing with time:

$$L\left(t\right)=Ut \tag{4}$$

Thus we take $L\left(t\right)$ as constant over certain time interval Δt and by taking heat conduction within each interval as common transient problem of 1-D beam the problem could be solved. The iteration scheme is shown in figure down below.

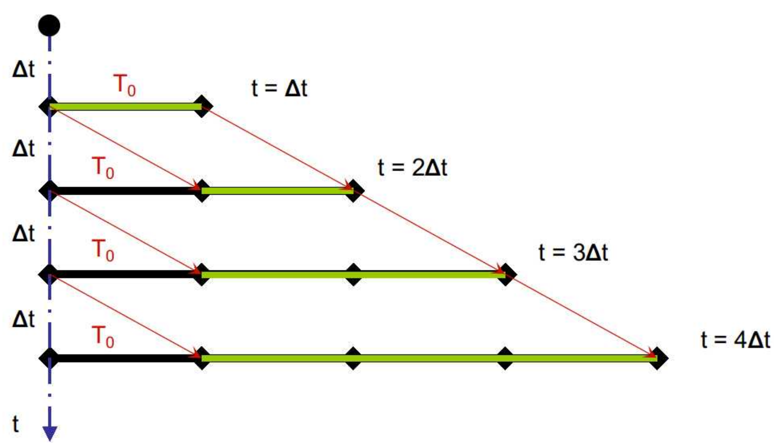

Figure 2: Iteration Scheme of the Problem

3 Method Applied and Grid Generation

For this problem I selected FTCS finite difference method. The reason of my choice is that this problem could be devided into superposition of multiple simple heat conduction problems which could be easily solved using explicit method. Also code for explicit method is much more easier to compose and modify.

This problem is a 1-D problem and thus it's discretization is done by dividing the longtitude region of the beam into multiple discrete nodes. Since it's quite simple and straight forward I incoupled the mesh generation within my solver.

MATLAB code composed for this problem is as shown:

Listing 1: MATLAB function

```
function [T,Fo,aa,bb] = test_final(dt,dx,U,DetT,duration)
%   FTCS iteration method for FINAL PROJECT of CHT
%   dt ---- time step for iteration
%   dx ---- grid size of the mesh
%   DetT ---- time inteval
%   U ---- translation speed of the cylinder
%   duration ---- total time of our calculation

%   Fo ---- preset Fourier number
%   T ---- output matrix of result
%   Ta ---- ambient tempreture of atmosphere
%   T0 ---- boundary tempreture
%   aa ---- hD/k
%   bb ---- UD/a

a = 5*10^(-3);  %     Diffusion coeff
k = 25;  %     Conductivity coeff
h = 50; %     Convection film coeff
r = 0.001; % Radius of the rod
rhoC = k/a;

P = 2*pi*r;
A = pi*r^2;

aa=(h*r*2)/k;
bb=(U*r*2)/a;

T0 = 783.15; % initial temeprature : 500 degree Celsius assumed
Ta = 298.15; % ambient temeprature : 25 degree Celsius assumed

Fo= a*dt/(dx^2);

T=zeros(duration/dt,duration*U/dx);
T(:,1) = T0;% T(3,1)=T0;T(3,2)=T0;T(3,3)=T0;

L=zeros(duration/DetT,1);
N=zeros(duration/DetT,1);

lim = duration/DetT;

X0 = DetT*U/dx;
S = DetT/dt;

% T(1,1:X0)=T0;
```

```
N(1) = X0;

K=1;

while K<=lim

    L(K) = K*DetT*U; % Length of extruded rod as defined
    N(K) = K*X0; % Total number of nodes at this time step

    M1 =(K-1)*S+1;
    M2 =K*S;

    T(M1,1:X0)=T0;

    if K~=1
        T(M1,(X0+1):N(K))=T(M1-1,1:N(K-1));
    end



    for i=(M1+1):M2
    % calculate at each time step

        for j=2:N(K)
        % FTCS finite difference method for each time step

            if j~= N(K)
                T(i,j)= (Fo - U*dt/(2*dx))*T(i-1,j+1)+(1-2*Fo-h*P*dt/(rhoC*A))*T
                % - U*dt/(2*dx)
                % + U*dt/(2*dx)

            elseif j==N(K)
                T(i,j) = (k*T(i,j-1)-h*dx*Ta)/(k-h*dx);%h*dx*(T(i-1,j-1)-Ta)/k +
            end

        end

    end

    K=K+1;

end

T = (T-Ta)/(T0-Ta);

end
```

4 Numerical Scheme Setup

Discretizing the physical domain using finite difference approach:(within certain Δt)

$$\rho C\left[\frac{T_i^{(n+1)}-T_i^{(n)}}{\Delta t}+U\frac{T_{i+1}^{(n)}-T_{i-1}^{(n)}}{2\Delta x}\right]=k\frac{T_{i+1}^{(n)}-2T_i^{(n)}+T_{i-1}^{(n)}}{\Delta x^2}-\frac{hP}{A}\left(T_i^{(n)}-T_a\right) \tag{5}$$

Rearranging terms and thus the forward iteration could be derived:

$$T_i^{(n+1)}=\left(F_o-\frac{U\Delta t}{2\Delta x}\right)T_{i+1}^{(n)}+\left(1-2F_o-\frac{hP\Delta t}{\rho CA}\right)T_i^{(n)}+\left(F_o+\frac{U\Delta t}{2\Delta x}\right)T_{i-1}^{(n)}+\frac{hP\Delta t}{\rho CA}T_a \tag{6}$$

This is our formular to solve for tempreture distribution within certain Δt. As for when it comes to a time step betweenΔt and $\Delta t+1$(into next iteration), region from $U\Delta t$ to Ut(which is the end point of the rod) will inherite tempreture distribution from last step while region from 1 to $U\Delta t$ will be automatically preset to T_0 to initialize new iteration step.

5 Results and Compare

In my computation, physical parameters could be found in section 3 where I inserted MATLAB code, while my inputs for function are chosen as shown in the table below:

Input Name	Meaning	Value
dt	Time Step for Iteration	$1\times10^{-3}s$
dx	Grid Size	$1\times10^{-2}m$
U	Extrusion Speed	$1m/s$
DetT	Time Inteval	$1\times10^{-1}s$
duration	Total Time	$50s$

Table 1: Input Parameters

The result from my code could be hardly called satisfactory, yet I believe it does explained about behavior of rod when L is big enough. This figure shows tempreture distribution when time is quite short.

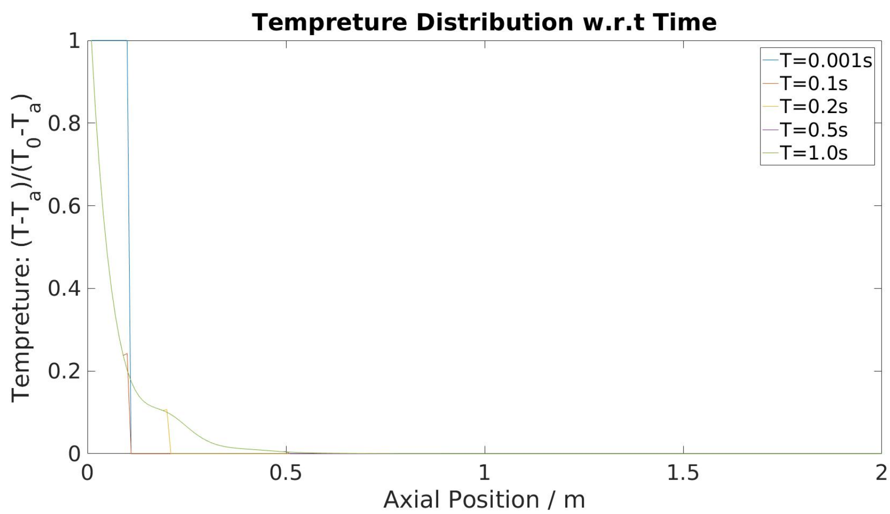

Figure 3: Tempreture Distribution w.r.t. Time

In comparison with what I found in the textbook(down below), it is clear that my calculation is not the same with the standard solution. There is no siginificant difference between each step in my calculation, which is totally unlike the standard solution. I double checked my differential scheme multiple times and still didn't find a clue about it.

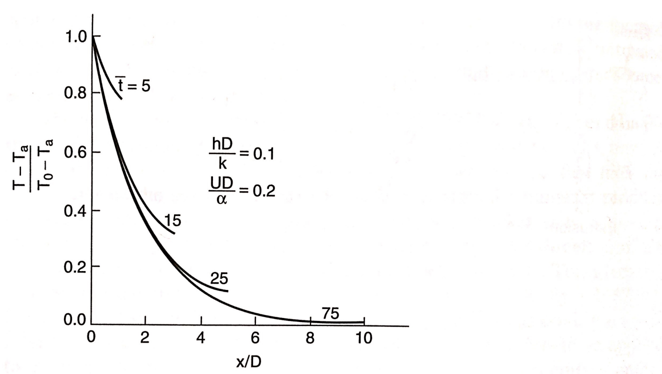

Figure 8.6 Temperature distributions in a continuously moving cylindrical rod of diameter D, at various time intervals after the start of the process, assuming one-dimensional conduction [From Jaluria and Singh (1983). Reprinted with permission of Elsevier/North-Holland.]

Figure 4: Standard Solution from Textbook

When it comes to the long term behavior, it seems that the solution is quite alike the standard result. According to analytical analysis, when $L(t)$ is big enough, the governing equation will collapse into:

$$\rho C\left(U\frac{\partial T}{\partial x}\right) = k\frac{\partial^2 T}{\partial x^2} - \frac{hP}{A}\left(T - T_a\right) \tag{7}$$

This solution will be quite alike to standard 1-D beam heat conduction problem. What I got from my code also shows this property. The result I got for long time scale is shown down below:

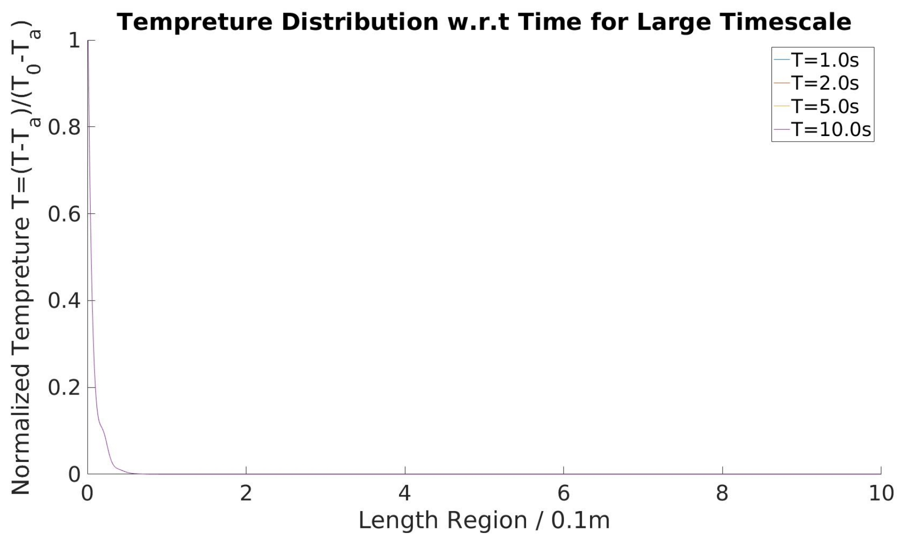

Figure 5: Result for long timescale

Also what I'm interested in is how tempreture at a specific point change during this process. In the figure down below I shown tempreture variation with time for 3 indivitual nodes, which initially ($t = \Delta t = 0.001s$) at $x = 0.01m$,$x = 0.05m$ and $x = 0.1m$ respectively. It turns out that their variation are very similar to each other.

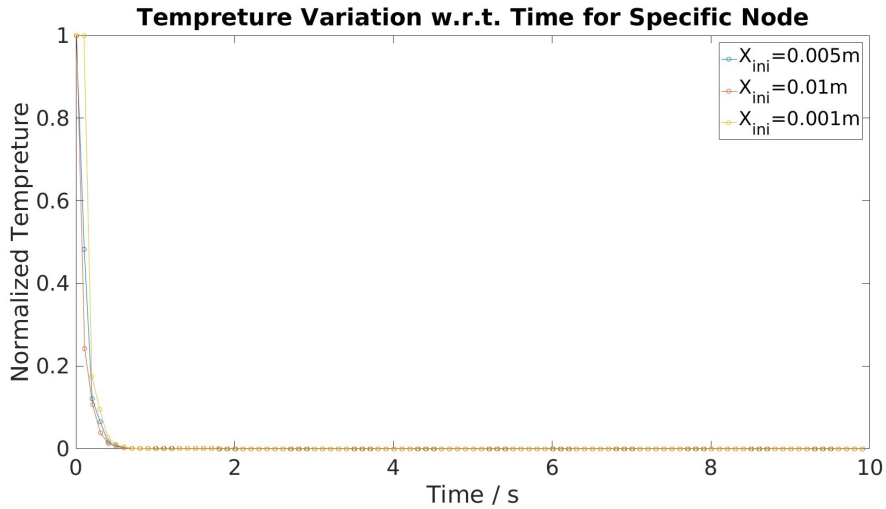

Figure 6: Tempreture Variation at Specific Node

6 Conclusions

In this program I practiced setting up speciallized FDM scheme from governing equation for certain problem. But my results doesn't match with standard result. I'm still working on fixing this bug. However I believe this model I set up could show some property of the phenomena when $L(t)$ is large enough since it is prediceted analytically that the governing equation will collapse into steady-state like format. Though I have no clue why it is not working well with short timescale in my case, I've already felt that it may because of stability of my algorithm couldn't sustain those situations or there's some fatal error in my FDM scheme setting up. I'll dig more into this problem after the final exam.