MAE 315

Heat and Mass Transfer

Computer Project

June 19th, 2014

Chris Thurman

Chris Davis

Chris Shertzer

Lucas Mills

**Introduction:**

It is required to predict how long it would take for a long cylindrical log to ignite by solving the transient heat conduction problem. The finite different method with the Crank-Nicolson scheme will be used to solve this equation numerically and the results will be plotted against the analytical solution which uses the series solution.

**Governing Differential Equation:**

**Boundary Conditions:**

1. At t=0, T=25°C

2. At r=0,

3. At r=Ro,

**Results:**

**Crank-Nicolson Method**

% Given:

r\_0=0.05;

k=0.17;

alpha=1.28e-7;

T0=25;

Tinf=600;

h=13.6;

Tsur=420;

lamda1=1.9081;

A1=1.4698;

int=10000;

m2=(h\*2\*pi\*r\_0)/(k\*pi\*r\_0^2);

dr=r\_0/12;

dt=1;

lamda=(alpha\*dt)/(dr^2);

tetas=(420-600)/(25-600);

%Crank Nicolson Method

n=12;

T=ones(n+1,int);

r=ones(n+1,1);

for i=1:13

r(i)=dr\*(i-1);

end

r(1)=eps;

T(1:n+1,1)=25; %B.C.

r32=r(1)+(.5\*1\*dr);

r52=r(1)+(.5\*3\*dr);

r72=r(1)+(.5\*5\*dr);

r92=r(1)+(.5\*7\*dr);

r112=r(1)+(.5\*9\*dr);

r132=r(1)+(.5\*11\*dr);

r152=r(1)+(.5\*13\*dr);

r172=r(1)+(.5\*15\*dr);

r192=r(1)+(.5\*17\*dr);

r212=r(1)+(.5\*19\*dr);

r232=r(1)+(.5\*21\*dr);

r252=r(1)+(.5\*23\*dr);

%Creating the a vector

a=-((lamda/2).\*((r-(dr/2))./r)).\*diag(eye(n+1));

a(n+1)=-(lamda/2)\*((r(n+1)-dr/2)/r(n+1))-(lamda/2);

% Creating the b vector

b=(1+lamda)\*diag(eye(n+1));

b(1)=1+(lamda/2)\*((r32+r(1))/r(1));

b(n+1)=1+(lamda/2)\*((r(n+1)+r252)/r(n+1))+(lamda\*h\*dr)/k;

% Creating the c vector

c=-((lamda/2).\*((r+(dr/2))./r)).\*diag(eye(n+1));

c(1)=-(lamda/2)\*(1+r32/r(1));

for i=1:int

if T(13,i)>=420

ifin=i-1;

timefinal=ifin\*dt;

break

end

% Creating the d vector

d(1)=(lamda/2)\*T(2,i)+...

(1-(lamda/2)\*(r(1)+r32)/r(1))\*T(1,i)+...

((lamda/2)\*(r32/r(1)))\*T(2,i);

for j=2:n

d(j)=-a(j)\*T(j-1,i)+((1-lamda)\*T(j,i))-c(j)\*T(j+1,i);

end

d(n+1)=(lamda/2)\*((r252/r(n+1))+1)\*T(n,i)...

+(1-(lamda/2)\*((r252+r(n+1))/r(n+1))-(lamda\*h\*dr)/k)\*T(n+1,i)...

+(2\*lamda\*h\*dr\*Tinf)/k;

t=Tridiag(a,b,c,d);

T(:,i+1)=t;

end

% Plot

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

for n=0:4

t=ifin/2^n;

t=round(t);

x=time4/2^n;

Fo=(x\*alpha)/r\_0^2;

Tanal=Tinf+(T0-Tinf)\*(a1\*exp(-L1^2\*Fo).\*besselj(0,L1\*r/r\_0)...

+a2\*exp(-L2^2\*Fo).\*besselj(0,L2\*r/r\_0)+a3\*exp(-L3^2\*Fo).\*...

besselj(0,L3\*r/r\_0)+a4\*exp(-L4^2\*Fo).\*besselj(0,L4\*r/r\_0)+...

a5\*exp(-L5^2\*Fo).\*besselj(0,L5\*r/r\_0)+a6\*exp(-L6^2\*Fo).\*...

besselj(0,L6\*r/r\_0)+a7\*exp(-L7^2\*Fo).\*besselj(0,L7\*r/r\_0)...

+a8\*exp(-L8^2\*Fo).\*besselj(0,L8\*r/r\_0)+a9\*exp(-L9^2\*Fo).\*...

besselj(0,L9\*r/r\_0)+a10\*exp(-L10^2\*Fo).\*besselj(0,L10\*r/r\_0));

hold on

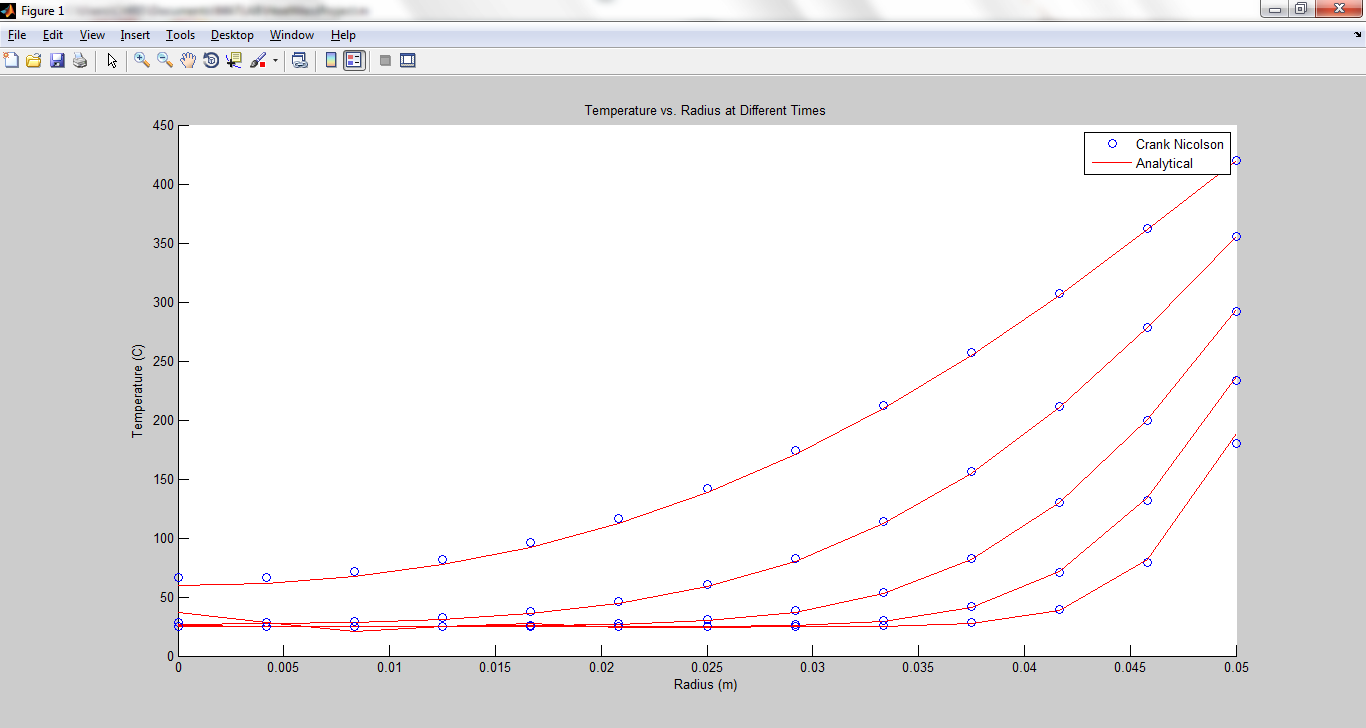
title('Temperature vs. Radius at Different Times');

xlabel('Radius (m)'); ylabel('Temperature (C)');

plot(r,T(:,t),'ob',r,Tanal,'-r'); legend('Crank Nicolson','Analytical')

end

hold off

**Numerical Results:**

**Time to Ignition =** 1966 seconds

**Analytical Solution:**

%Analytical Solution from Table 4-1

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% How to solve Problem 4.39 using multiple terms in the series (Table 4.1

% page 237)

%run the MATLAB script

Bi=4.0;

r\_0=0.05;

alpha=1.28e-7;

%define the Bessel function

fb=@(x)x\*besselj(1,x)/besselj(0,x)-Bi;

%determine first 10 roots as lambdas

L1=fzero(@(x)fb(x),[1 2]);

L2=fzero(@(x)fb(x),[3 5]);

L3=fzero(@(x)fb(x),[6 8]);

L4=fzero(@(x)fb(x),[9 11]);

L5=fzero(@(x)fb(x),[12 14]);

L6=fzero(@(x)fb(x),[15 17]);

L7=fzero(@(x)fb(x),[18 25]);

L8=fzero(@(x)fb(x),[26 27]);

L9=fzero(@(x)fb(x),[28 30]);

L10=fzero(@(x)fb(x),[31 33]);

%calculate the coefficients A's

a1=2/L1\*(besselj(1,L1)/(besselj(0,L1)^2+besselj(1,L1)^2));

a2=2/L2\*(besselj(1,L2)/(besselj(0,L2)^2+besselj(1,L2)^2));

a3=2/L3\*(besselj(1,L3)/(besselj(0,L3)^2+besselj(1,L3)^2));

a4=2/L4\*(besselj(1,L4)/(besselj(0,L4)^2+besselj(1,L4)^2));

a5=2/L5\*(besselj(1,L5)/(besselj(0,L5)^2+besselj(1,L5)^2));

a6=2/L6\*(besselj(1,L6)/(besselj(0,L6)^2+besselj(1,L6)^2));

a7=2/L7\*(besselj(1,L7)/(besselj(0,L7)^2+besselj(1,L7)^2));

a8=2/L8\*(besselj(1,L8)/(besselj(0,L8)^2+besselj(1,L8)^2));

a9=2/L9\*(besselj(1,L9)/(besselj(0,L9)^2+besselj(1,L9)^2));

a10=2/L10\*(besselj(1,L10)/(besselj(0,L10)^2+besselj(1,L10)^2));

% Series Solution

fth1=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)-tetas;

fth2=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)-tetas;

fth3=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)-tetas;

fth4=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)-tetas;

fth5=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)-tetas;

fth6=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)+a6\*exp(-L6^2\*x)\*besselj(0,16)-tetas;

fth7=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)+a6\*exp(-L6^2\*x)\*besselj(0,L6)+a7\*exp(-L7^2\*x)\*besselj(0,L7)-tetas;

fth8=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)+a6\*exp(-L6^2\*x)\*besselj(0,L6)+a7\*exp(-L7^2\*x)\*besselj(0,L7)+...

a8\*exp(-L8^2\*x)\*besselj(0,L8)-tetas;

fth9=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)+a6\*exp(-L6^2\*x)\*besselj(0,L6)+a7\*exp(-L7^2\*x)\*besselj(0,L7)+...

a8\*exp(-L8^2\*x)\*besselj(0,L8)+a9\*exp(-L9^2\*x)\*besselj(0,L9)-tetas;

fth10=@(x)a1\*exp(-L1^2\*x)\*besselj(0,L1)+a2\*exp(-L2^2\*x)\*besselj(0,L2)...

+a3\*exp(-L3^2\*x)\*besselj(0,L3)+a4\*exp(-L4^2\*x)\*besselj(0,L4)+a5\*exp(-L5^2\*x)...

\*besselj(0,L5)+a6\*exp(-L6^2\*x)\*besselj(0,L6)+a7\*exp(-L7^2\*x)\*besselj(0,L7)+...

a8\*exp(-L8^2\*x)\*besselj(0,L8)+a9\*exp(-L9^2\*x)\*besselj(0,L9)+a10\*exp(-L10^2\*x)\*...

besselj(0,L10)-tetas;

%solve for the Fourier numbers

t1=fzero(@(x)fth1(x),0.1);

t2=fzero(@(x)fth2(x),0.1);

t3=fzero(@(x)fth3(x),0.1);

t4=fzero(@(x)fth4(x),0.1);

t5=fzero(@(x)fth5(x),0.1);

t6=fzero(@(x)fth6(x),0.1);

t7=fzero(@(x)fth7(x),0.1);

t8=fzero(@(x)fth8(x),0.1);

t9=fzero(@(x)fth9(x),0.1);

t10=fzero(@(x)fth10(x),0.1);

format short g

%calculate the times in seconds

time1=t1\*r\_0^2/alpha;

time2=t2\*r\_0^2/alpha;

time3=t3\*r\_0^2/alpha;

time4=t4\*r\_0^2/alpha;

time5=t5\*r\_0^2/alpha;

time6=t6\*r\_0^2/alpha;

time7=t7\*r\_0^2/alpha;

time8=t8\*r\_0^2/alpha;

time9=t9\*r\_0^2/alpha;

time10=t10\*r\_0^2/alpha;

%Analytical Plot

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

r=linspace(0,0.05);

for n=0:4

t=time4/2^n;

Fo=(t\*alpha)/r\_0^2;

T=Tinf+(T0-Tinf)\*(a1\*exp(-L1^2\*Fo).\*besselj(0,L1\*r/r\_0)...

+a2\*exp(-L2^2\*Fo).\*besselj(0,L2\*r/r\_0)+a3\*exp(-L3^2\*Fo).\*...

besselj(0,L3\*r/r\_0)+a4\*exp(-L4^2\*Fo).\*besselj(0,L4\*r/r\_0)+...

a5\*exp(-L5^2\*Fo).\*besselj(0,L5\*r/r\_0)+a6\*exp(-L6^2\*Fo).\*...

besselj(0,L6\*r/r\_0)+a7\*exp(-L7^2\*Fo).\*besselj(0,L7\*r/r\_0)...

+a8\*exp(-L8^2\*Fo).\*besselj(0,L8\*r/r\_0)+a9\*exp(-L9^2\*Fo).\*...

besselj(0,L9\*r/r\_0)+a10\*exp(-L10^2\*Fo).\*besselj(0,L10\*r/r\_0));

hold on

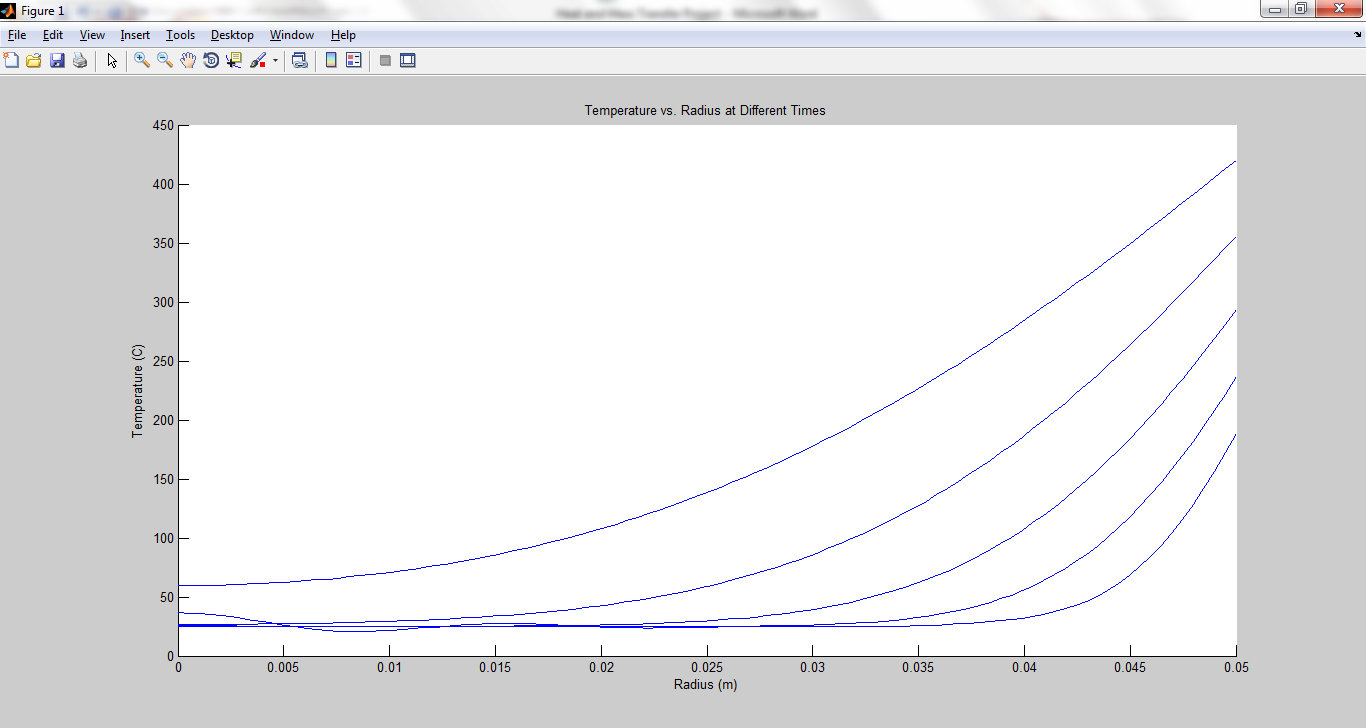
title('Temperature vs. Radius at Different Times');

xlabel('Radius (m)'); ylabel('Temperature (C)');

plot(r,T);

end

hold off

**Analytical Results:**

**Time to Ignition for Series Solution (1-10 terms):**

time1 = 1412.2

time2 =1902.6

time3 = 1907.9

time4 = 1907.9

time5 = 1907.9

time6 = 1907.9

time7 = 1907.9

time8 = 1907.9

time9 = 1907.9

time10 = 1907.9

**Error in Ignition Time:**

**Discussion:**

This project compares the solution obtained by using numerical methods to that obtained by using the analytical series solution. In this problem, a log is placed in a fire at a temperature of 600 degrees Celsius and it is required to find the time that it takes for the surface of the log to ignite which occurs at 420 degrees Celsius. By using the boundary conditions given, the temperature distribution over the radius of the log can be solved for over the time interval which will also solved for. In this problem, the temperature varies with time as well as with space, meaning that the system involves transient heat transfer. Because the Biot Number is greater than 0.1, a lumped system analysis is invalid thus requiring approximate analytical solutions involving series. The number of series can be determined by the Fourier Number, which, in this case, is less than 0.2, meaning that the problem requires more than a one-term approximation. When four terms were used in the series solution, there were fluctuations in the graph rather than a smooth curve. Although it was not particularly necessary to use 10 series terms, it was beneficial and wise to do so because the more terms used, the smoother and more accurate the graph became.

**Results:**

From the graphs above, it can be inferred that the Crank-Nicolson numerical method closely approximated the analytical series solution. Because of the difficulty in solving partial differential equations for transient heat transfer, analytical solutions have been derived and tabulated for use but are still only approximations of the actual solution. For this reason, it is very beneficial to use numerical methods, which simplify these partial differential equations into algebraic equations which can easily be solved, however, these algebraic equations contain many terms which can be very tedious to solve by hand. Because of this, using computer programs such as MATLAB can be very beneficial in solving and plotting the solutions to these numerical methods in a matter of seconds after the correct code has been written for a given problem. The error in ignition time was only 3.045 percent, which is very small, thus validating the use and accuracy of numerical methods in solving transient heat transfer problems.