# FINITE DIFFERENCE ANALYSIS OF 1-D UNSTEADY HEAT EQUATION

MEEN 689 – COMPUTATIONAL FLUID DYNAMICS

By

*B SATYA AKHIL REDDY*

*826006779*

Date: 10/17/2018

## INTRODUCTION

Finite-difference methods are numerical methods which have been developed for solving differential equations by approximating the derivatives with the difference equations. In this process, the continuous domain of the problem is converted into its discrete counterparts. Whenever a continuous space is discretized, a certain amount of discretization error is bound to creep into our analysis. Thus, for a model, our goal must be to reduce the error by performing discretization using several established schemes and choosing the best one amongst them.

In this report, we define our problem as one-dimensional unsteady heat flow in a wall of thickness 2 ft with infinite length in the other directions. It is composed of nickel with a diffusivity of 0.4 ft^2/hr. The wall has an initial temperature of 100°F and the surface temperatures of two sides are suddenly raised to 400°F. The differential equation is

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

The PDE is parabolic in nature, since the above process is time dependent phenomenon. We attempt to find the temperature profile of the wall using Forward Time Central Space (FTCS) and Crank Nicholson (CN) schemes, and then develop some performance characteristics to compare both the schemes.

The FTCS scheme is forward in time and central in space. It is an explicit scheme, where the values of the dependent variables at a later time are calculated directly from the values at current time. This scheme is conditionally stable, and the limitation is given by

$$\frac{\alpha \Delta t}{\Delta x^2} \leq 0.5$$

where, $\Delta t$ and $\Delta x$ are the step size in time and space respectively.

The CN scheme is central in space and time. It is an implicit scheme, where the values of the dependent variables are defined by a set of coupled equations. This scheme is unconditionally stable, meaning that the scheme is valid for any step size in space and time. But the step sizes are limited by the required accuracy of the solution.

In this analysis, we attempt to find the temperature profile of the wall due to the sudden elevation of temperatures at the boundaries. Then we assess the solution for different length and time scales to determine the efficiency of the schemes over the domain of the problem. Later, we try to compare both the schemes and determine the best scenario in which a scheme could be used.

In the next section, we start off with the methodology followed in the computation of the problem.

**METHOD OF SOLUTION**

The given PDE is

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

The boundary conditions are

$$T(x,0) = 100°\mathrm{F}$$
$$T(0,t) = 400°\mathrm{F}$$
$$T(L,t) = 400°\mathrm{F}$$

We transform the PDE into non-dimensional scale by using the following transformations

$$x = L * \bar{x}$$

$$\bar{T} = \frac{T - Ts}{Ti - Ts}$$

$$t = \left(\frac{L^2}{\alpha}\right) * \bar{t}$$

Thus, the PDE is transformed as

$$\frac{\partial \bar{T}}{\partial \bar{t}} = \frac{\partial^2 \bar{T}}{\partial \bar{x}^2}$$

The boundary conditions are also transformed as

$$\bar{T}(\bar{x},0) = 1$$
$$\bar{T}(0,\bar{t}) = 0$$
$$\bar{T}(1,\bar{t}) = 0$$

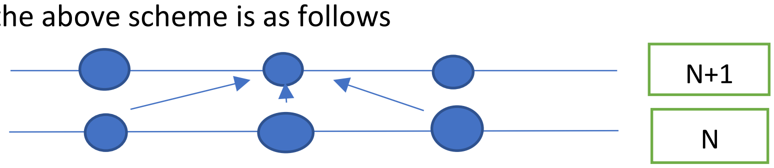
We use FTCS to discretize the non-dimensional unsteady heat flow PDE. FTCS is forward in time and central in space, thus having an order of O($\Delta x^2, \Delta t$).

$$\frac{\bar{T}_i^{n+1} - \bar{T}_i^n}{\Delta t} = \partial_{\tilde{x}x} T_i^n$$

Gathering the like terms on one side, we get

$$\overline{T_i^{n+1}} = \overline{T_i^n} + (\frac{\Delta t}{\Delta x^2})(\overline{T_{i-1}^n} - 2\overline{T_i^n} + \overline{T_{i+1}^n})$$

The stencil for the above scheme is as follows



The value of the dependent variable at the current time step is solely dependent on the values from the previous time step. Thus, the equations can be solved directly. We always know the value of the dependent variable on the boundaries (Dirichlet BC), therefore enabling us to solve only for the inner nodes.

On applying Von Neumann stability analysis, we find the stability criterion for the above non-dimensional discretization to be

$$\frac{\Delta t}{\Delta x^2} \leq 0.5$$

We must make sure that our discretization scales in length and time obey the above criterion, or else the scheme would lead to erroneous results.

We now use Crank Nicholson scheme to discretize the PDE. This scheme is central in time with respect to spatial discretization. Therefore, the order of this scheme is $O(\Delta t^2, \Delta x^2)$. Continuing with our analysis, the PDE is discretized as1

$$\frac{\bar{T}_i^{n+1} - \bar{T}_i^n}{\Delta t} = \partial_{\tilde{x}\tilde{x}} T_i^{n+\frac{1}{2}}$$

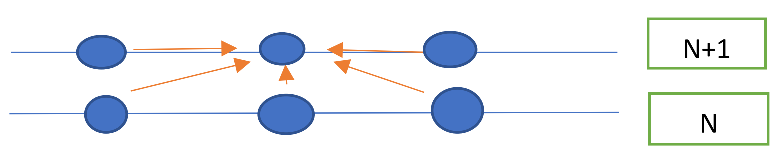
The differential on RHS can be written as

$$\frac{\bar{T}_i^{n+1} - \bar{T}_i^n}{\Delta t} = \frac{\partial_{\tilde{x}\tilde{x}} T_i^{n+1} + \partial_{\tilde{x}\tilde{x}} T_i^n}{2}$$

Upon expanding and gathering the like terms on one side,

$$-\frac{d}{2} * \bar{T}_{i-1}^{n+1} + (1+d) * \bar{T}_i^{n+1} - \frac{d}{2} * \bar{T}_{i+1}^{n+1} = (1-d) * \bar{T}_i^n + \frac{d}{2} * (\bar{T}_{i-1}^n + \bar{T}_{i+1}^n)$$

The stencil for this scheme is




As we can see, the above equation cannot be solved by itself. We need to develop a set of equations for a given time step and then use an appropriate matrix or iterative approach to solve for the dependent variables. We use TDMA approach in our case to solve this problem. Since we always know the value of the dependent variable on the boundaries , we solve only for the inner nodes.

We write the above equation for all the inner nodes in the form of a matrix. This matrix given below consists of Nx-2 elements, where Nx is the number of nodes. We must notice that, the equation for nodes 2 and Nx-1 are different since we included the node 1 and node Nx values at the current time level (boundary information) respectively.

$$
\begin{bmatrix}
1+d & -\frac{d}{2} & \cdots & & \\
-\frac{d}{2} & 1+d & -\frac{d}{2} & & \\
\vdots & -\frac{d}{2} & \ddots & -\frac{d}{2} & \vdots \\
 & & \cdots & 1+d & -\frac{d}{2} \\
 & & & -\frac{d}{2} & 1+d
\end{bmatrix}
\begin{bmatrix}
\bar{T}_2^{n+1} \\ \bar{T}_3^{n+1} \\ \vdots \\ \bar{T}_{Nx-2}^{n+1} \\ \bar{T}_{Nx-2}^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
(1-d)*\bar{T}_i^n + \frac{d}{2}*(\bar{T}_{i-1}^n + \bar{T}_{i+1}^n) + \frac{d}{2}*\bar{T}_1^{n+1} \\
(1-d)*\bar{T}_i^n + \frac{d}{2}*(\bar{T}_{i-1}^n + \bar{T}_{i+1}^n) \\
(1-d)*\bar{T}_i^n + \frac{d}{2}*(\bar{T}_{i-1}^n + \bar{T}_{i+1}^n) \\
(1-d)*\bar{T}_i^n + \frac{d}{2}*(\bar{T}_{i-1}^n + \bar{T}_{i+1}^n) \\
(1-d)*\bar{T}_i^n + \frac{d}{2}*(\bar{T}_{i-1}^n + \bar{T}_{i+1}^n) + \frac{d}{2}*\bar{T}_{Nx}^{n+1}
\end{bmatrix}
$$

On applying TDMA, we get the solution of all the nodes at the current time step. Upon performing Von Neumann stability analysis, we find that this scheme is unconditionally stable. Any combination of $\Delta t$ and $\Delta x$ give us the solution to the PDE, but one must know that dense grids and small time scales lead to more dependable solutions.

**DISCUSSION OF RESULTS**

The developed equations were coded MATLAB and the results obtained have been discussed below.

To start with, the PDE for unsteady heat flow is parabolic. In a parabolic PDE, the disturbances due to any changes are only felt after a time 't'. In our case, this is justified since both the schemes showed that the solution at time 'n+1' is dependent on the solution at time 'n'. To solve a parabolic PDE, we need to have the knowledge of the initial and boundary conditions. In our problem, the initial and boundary conditions were defined clearly. Thus, all the necessary steps were taken to solve the unsteady heat equation.

Figure 1 shows the development of temperature profile over time using FTCS scheme.

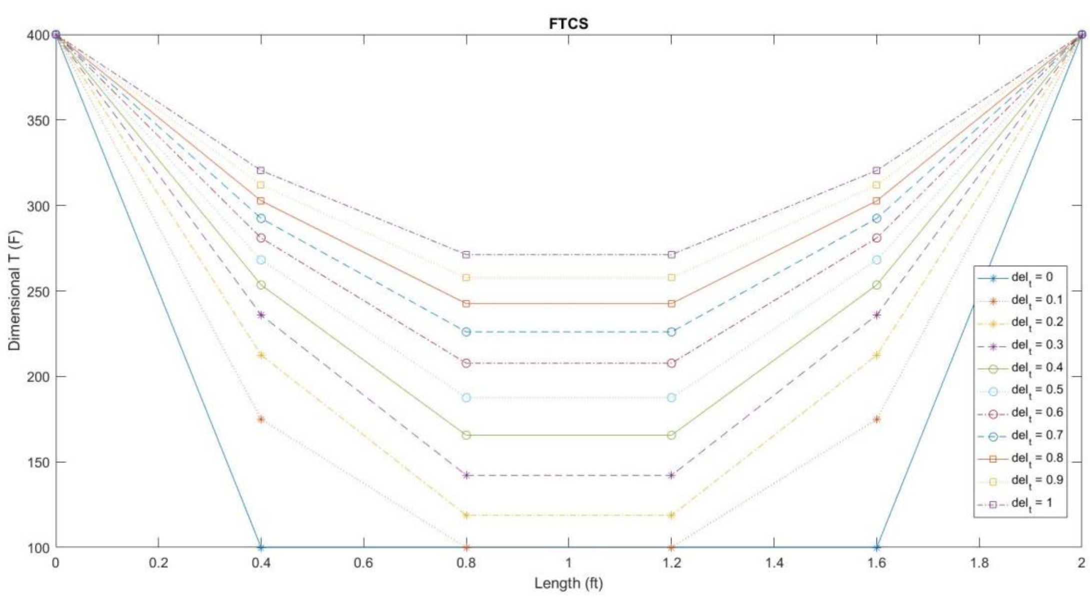



*Figure 1Temperature profile developed using FTCS scheme*

The wall was at an initial temperature of 100F. At time t-0, the temperature at the boundaries of the wall was elevated to 400F. This has been clearly depicted in Fig.1. As the time increases initially, the elevated temperature at the boundaries (disturbance) effects the temperature of the inner nodes since heat transfer takes place due to considerable difference in the temperatures. As a result, the temperature of the nodes nearer to the boundaries begins to increase steeply. The rate of increase of temperature of the inner nodes gradually decreases since the driving potential for heat transfer (difference in temperatures) decreases due to increase in temperature of the inner nodes. This can be confirmed from Fig.1 where we see that the curve loses its steepness over each time step. At time t=1 hour, the temperature profile of the wall looks like the topmost curve in Fig.1.

Figure 2 shows the development of temperature profile over time using Crank Nicholson Scheme.

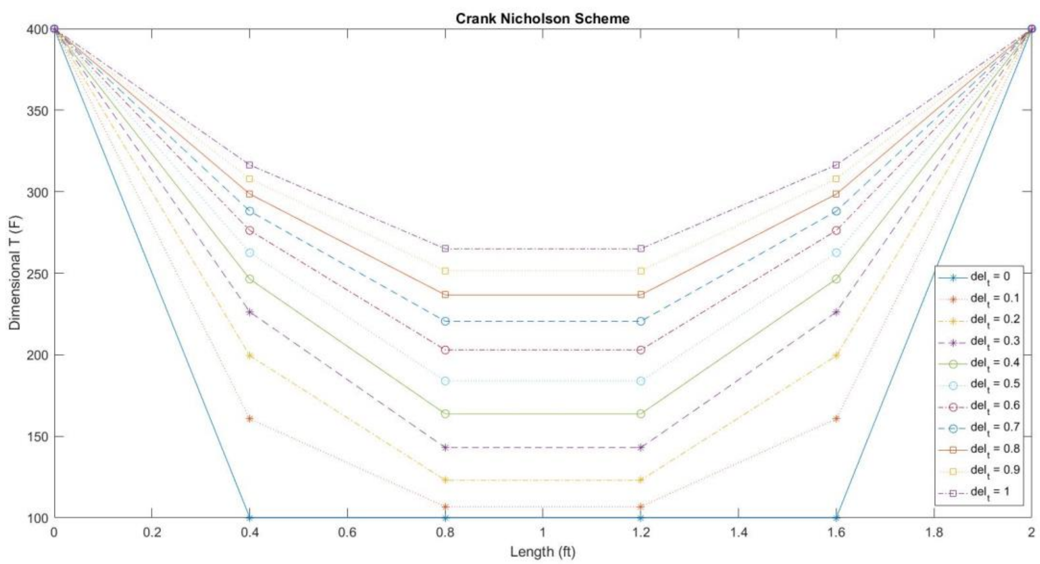



*Figure 2Development of temperature profile using Crank Nicholson Scheme*

The discussion made above for FTCS scheme can also be applied to Crank Nicholson scheme. Since the physics of the problem was correctly justified by both the schemes, we must now look at both the schemes and assess their numerical stability.

Figure 3 below shows the temperature profile at time t=1 hour for the length scales 0.4ft and 0.26ft using FTCS scheme. The time scale was kept constant at 0.1 hour. Going by the stability criterion developed for FTCS scheme, we obtain $\Delta x \geq 0.2828$. From the Fig.3, for a length scale of 0.4 which is greater than 02828, the scheme is able to develop the temperature profile as per the physics of the problem. But when we decrease the length scale to 0.26 which is lesser than 0.2828, the solution begins to oscillate around the actual value of the problem. This oscillation increases as we decrease the length scale further. Thus, for a given time scale, there is a limited size up to which a grid can be refined. For small size grids, extremely small time steps must be considered to obtain the solution according to the physics of the problem.

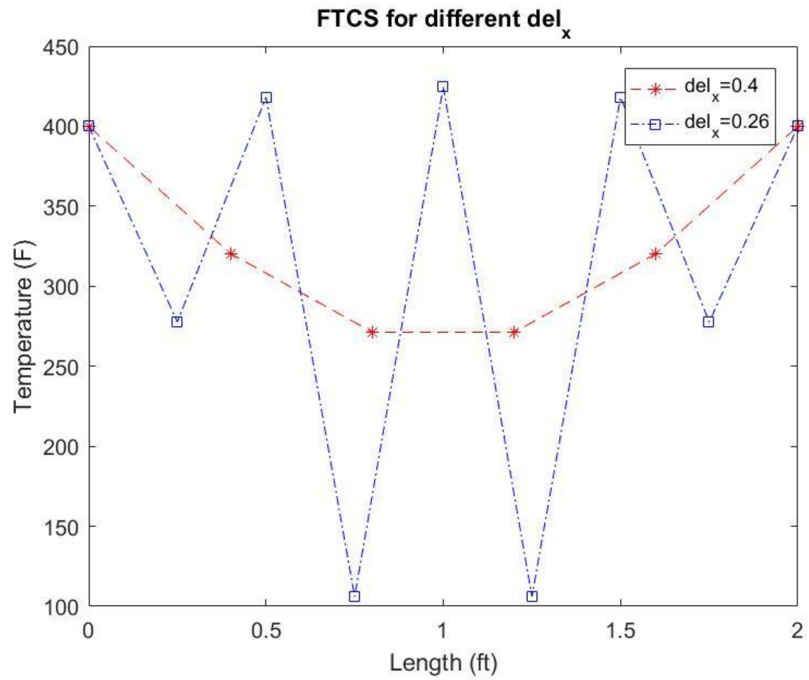



*Figure 3 FTCS stability - Temperature profile at different length scales*

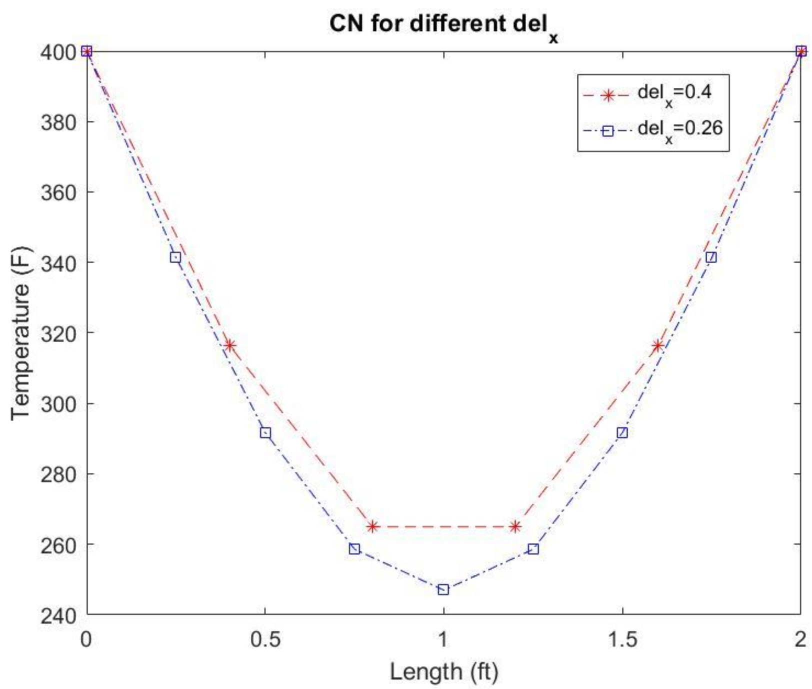



*Figure 4 Crank Nicholson Stability - Temperature profile for different length scales*

Figure 4 shows the temperature profile at time t=1 hour for the length scales 0.26ft and 0.4ft using Crank Nicholson scheme. The time scale was kept constant at 0.1 hour. We already know that Crank Nicholson scheme is unconditionally stable. Thus, at both the length scales, this scheme is able to develop the temperature profile according to the physics of the problem.

Though the scheme is unconditionally stable, we must define the scheme based on its accuracy which in turn is defined by how strong the implicit formulation is. For a strongly formulated implicit scheme, there wouldn't be much difference in temperature for different scales , which contrasts with this formulation.

Now, we look at both the schemes together and make some comparisons based on their performance.

Once the numerical solutions are calculated, the error associated is calculated by comparing the numerical solution to the analytical solution. The analytical solution is given as

$$T = T_s + 2(T_i - T_s)\sum_{m=1}^{\infty} e^{-(\frac{m\pi}{L})^2 \alpha t}\left[\frac{1-(-1)^m}{m\pi}\sin(\frac{m\pi x}{L})\right]$$

The error is calculated as below and then the discussions are carried out.

$$error = \frac{1}{IM}\sqrt{\sum_{i=1}^{IM}(T_{(i)}^{computed} - T_{(i)}^{analytical})^2}$$

Here, IM is the number of nodes in the discretization.

Figure 5 shows the loglog plot between error and length scale at a constant time scale for both the schemes.

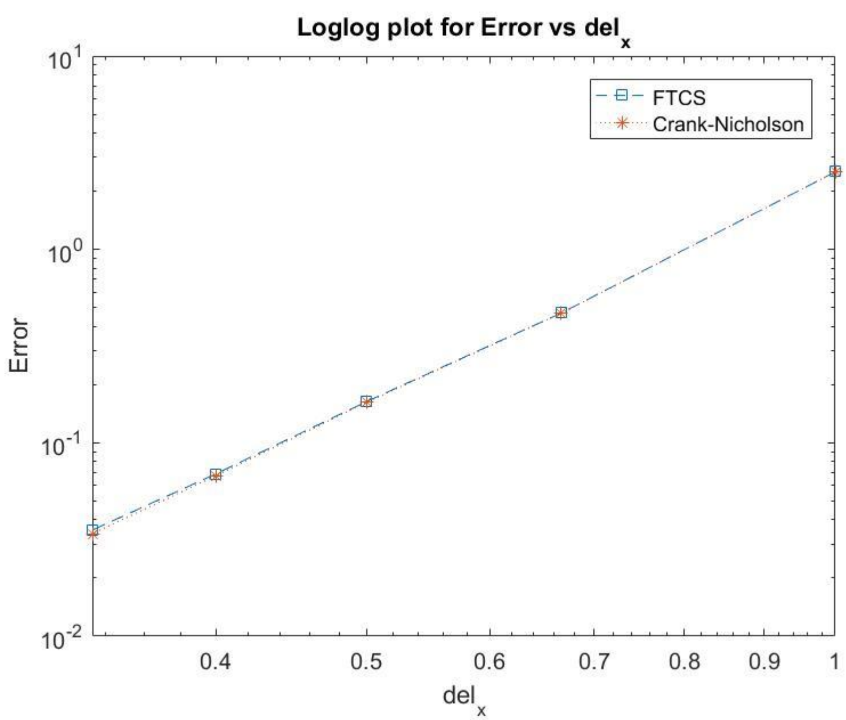



*Figure 5 Error plot of length scale for FTCS and Crank Nicholson schemes*

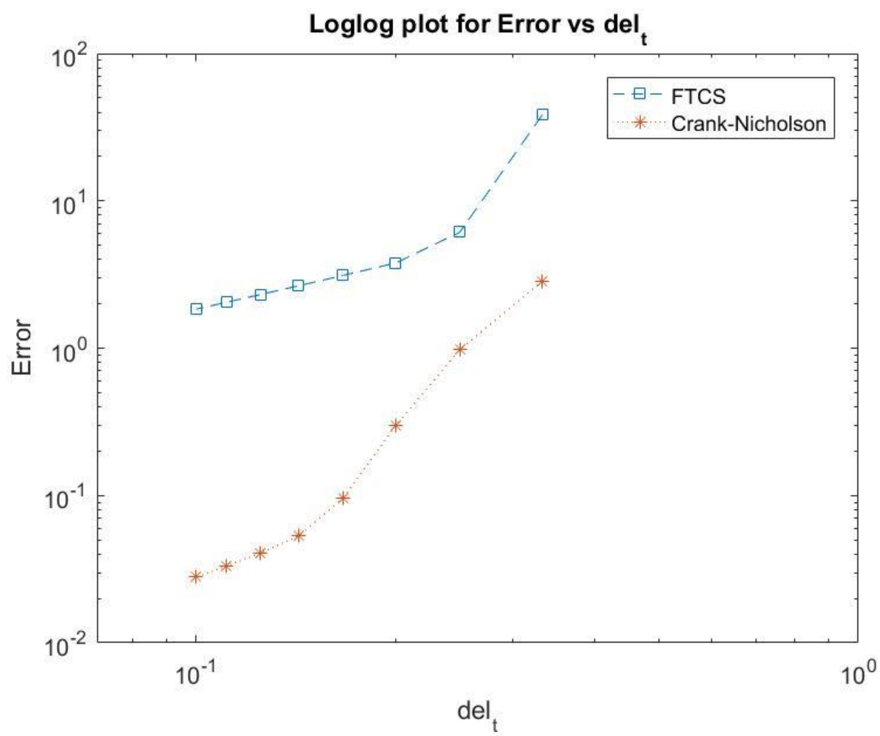
From Fig.5, we see that the plots for both the schemes coincide. The considered values for the length scale were all above the critical length scale value for FTCS. For a small time scale, there is no difference in the performance of both the schemes.

From the literature, we find that the slope of loglog plot corresponds to the order of accuracy of the discretization scheme. Since both the schemes have the same spatial order of accuracy, the nature of the plot is rightly justified. But the calculation of the slope of the loglog plot resulted in the following values at a time scale of 0.0001

$$Slope\ of\ FTCS = 3.8647$$
$$Slope\ of\ CN = 3.9211$$

This erroneous result may be attributed to the inclusion of time scale error over the range of length scales. This error can be minimized by assuming much lower time scales so that the time scale error is negligible.

Figure 6 shows the loglog plot between error and time scale at a constant length scale for both the schemes.




We can see a clear difference in the performance of both the schemes. The error is reduced by almost 99% at a time scale of 0.1 hr. Thus, this tells us that for a given length scale, the performance of Crank Nicholson is superior to FTCS.

The temporal order of accuracy of Crank Nicholson is $O(\Delta t^2)$ whereas for FTCS, it is $O(\Delta t)$. Since the slope of loglog plot corresponds to the order of accuracy of the discretization scheme, the order of accuracy is calculated from the data. The nature of the plot tells us that the temporal order of accuracy of the Crank Nicholson scheme is higher than FTCS scheme, which is a correct justification of the theory. For the FTCS scheme, the length scale was calculated every time scale to minimize the error.

$$Slope\ of\ FTCS = 1.0644$$
$$Slope\ of\ CN = 3.7356$$

The calculated value for FTCS is in order with the theory. But the calculated value for Crank Nicholson doesn't agree with the theoretical value. The error maybe attributed to contribution of error from length scale over the range of time scale.

**SUMMARY**

In our problem, we find that both the schemes follow the physics of the problem in their stability domain. From our analysis, for a given time scale, both the schemes provide the same reliability in their stable domain, thus leaving the choice to the CFD programmer. For a given length scale, if the problem is time accuracy dependent, Crank Nicholson provided the best results relatively.

**CONCLUSION**

From the above discussion, we can conclude that each scheme has its own positives and negatives. For FTCS, if the grid is dense, the time step associated with it must be small enough to satisfy the stability criterion. This causes the process to be computationally expensive. In such case, CN can be applied with relative ease, though it is complex to implement. Similarly, when the process in time accuracy dependent, FTCS is more superior to CN since it is computationally inexpensive and easy to implement.

**APPENDIX**

**FTCS**

```
%% Parameters

L=2;
Ti=100;
Ts=400;
alpha=0.4;
del_t=0.1;
t_start=0;
t_end=1;

%% Modifying the BCs' and other variables according to non-
dimensional setting

% x=L*x_nd
% T_nd=(T-Ts)/(Ti-Ts)
% t=(L^2/alpha)*t_nd

tnd_start=(alpha/L^2)*t_start;
tnd_end=(alpha/L^2)*t_end;
del_tnd=(alpha/L^2)*del_t;

% Number of time steps
Nt=round((tnd_end-tnd_start)/del_tnd);

% IC
Ti_nd=(Ti-Ts)/(Ti-Ts);

% BC
Ts_nd=(Ts-Ts)/(Ti-Ts);

%% Setting up of del_xnd for FTCS

% T_nd(j) = T_nd_old(j) + (del_tnd/del_xnd^2)*(T_nd_old(j-1)-
2*T_nd_old(j)+T_nd_old(j+1))
% Let us assume that the stability of FTCS is 0.5, then
% (del_tnd/del_xnd^2) must be less than or equal to 0.5

del_xnd_min=(2*del_tnd)^0.5;

% Since del_xnd_min=0.1414, let us assume that
del_xnd=0.13;

% Number of grid points
```

```
Nx=round(1/del_xnd)+1;

%% Initial Conditions setup

T=zeros(Nt+1,Nx);
T_nd=zeros(1,Nx);
T_nd(1)=Ts_nd;
T_nd(Nx)=Ts_nd;
for j=2:(Nx-1)
    T_nd(j)=Ti_nd;
end
T(1,:)=T_nd*(Ti-Ts)+Ts;
time=0;

%% Time Marching of the solution

T_nd_old=zeros(1,Nx);
for i=1:Nt
    time=time+del_tnd;

    % Keeping track of previous time step values
    T_nd_old=T_nd;

    % Evaluating T at the current time step
    for j=2:(Nx-1)
        T_nd(j)=T_nd_old(j)+(del_tnd/del_xnd^2)*(T_nd_old(j-1)-
2*T_nd_old(j)+T_nd_old(j+1));
    end

    % Setting BCs'
    T_nd(1)=Ts_nd;
    T_nd(Nx)=Ts_nd;

    T(i+1,:)=T_nd*(Ti-Ts)+Ts;
end

%% Plot of temperature profile

line={'*-','*:','*-.','*--','o-','o:','o-.','o--','s-','s:','s-
.'};
for t=1:Nt+1
    plot(linspace(0,L,Nx),T(t,:),line{t})
    hold on
    legendInfo{t} = ['del_t = ' num2str((t-1)*0.1)];
end
xlabel('Length (ft)');
ylabel('Dimensional T (F)');
```

```
title('FTCS');
legend(legendInfo)

%% Analytical Solution at t=1hr

syms m;
x=linspace(0,L,Nx);
sum_fun=2*(Ti-Ts)*exp(-1*((m*pi)/L)^2*alpha*1)*(((1-(-
1)^m)/(m*pi))*sin((m*pi*x)/L));
T_an=vpa(Ts+symsum(sum_fun,m,1,inf),4);

%% Error

Err=(1/Nx)*sqrt(sum((T(Nt+1,:)-T_an).^2));
```

**Crank Nicholson**

```
%% Homework-4 -- Crank Nicholson -- CFD

%% Parameters

L=2;
Ti=100;
Ts=400;
alpha=0.4;
del_t=0.1;
t_start=0;
t_end=1;

%% Modifying the BCs' and other variables according to non-
dimensional setting

% x=L*x_nd
% T_nd=(T-Ts)/(Ti-Ts)
% t=(L^2/alpha)*t_nd

tnd_start=(alpha/L^2)*t_start;
tnd_end=(alpha/L^2)*t_end;
del_tnd=(alpha/L^2)*del_t;

% Number of time steps
Nt=round((tnd_end-tnd_start)/del_tnd);

% IC
Ti_nd=(Ti-Ts)/(Ti-Ts);
```

```
% BC
Ts_nd=(Ts-Ts)/(Ti-Ts);

%% Setting up of del_xnd for Crank Nicholson

% Since Crank Nicholson scheme is unconditionally stable, we can
take any
% value of del_xnd. For uniformity, let us take del_xnd as 0.2
del_xnd=0.13;

% Number of grid points
Nx=round(1/del_xnd)+1;

%% Initial Conditions setup

T=zeros(Nt+1,Nx);
T_nd=zeros(1,Nx);
T_nd(1)=Ts_nd;
T_nd(Nx)=Ts_nd;
for j=2:(Nx-1)
    T_nd(j)=Ti_nd;
end
T(1,:)=T_nd*(Ti-Ts)+Ts;
time=0;

%% Time Marching of the solution

d=del_tnd/(del_xnd)^2;
a=zeros(Nx,1);
b=zeros(Nx,1);
c=zeros(Nx,1);
e=zeros(Nx,1);
for t=1:Nt
    T_nd_old=T_nd;

    for j=2:(Nx-1)
        a(j)=-d/2;
        b(j)=(1+d);
        c(j)=-d/2;
        e(j)=(1-d)*T_nd_old(j)+(d/2)*(T_nd_old(j-
1)+T_nd_old(j+1));
    end

    e(2)=e(2)+(d/2)*T_nd(1);
    e(Nx-1)=e(Nx-1)+(d/2)*T_nd(Nx);
```

```
    e=TDMA(2,Nx-1,a,b,c,e);

    for j=2:(Nx-1)
        T_nd(j)=e(j);
    end

    % Updating BCs
    T_nd(1)=Ts_nd;
    T_nd(Nx)=Ts_nd;

    T(t+1,:)=T_nd*(Ti-Ts)+Ts;
end

%% Plot of temperature profile

line={'*-','*:','*-.','*--','o-','o:','o-.','o--','s-','s:','s-
.'};
for t=1:Nt+1
    plot(linspace(0,L,Nx),T(t,:),line{t})
    hold on
    legendInfo{t} = ['del_t = ' num2str((t-1)*0.1)];
end
xlabel('Length (ft)');
ylabel('Dimensional T (F)');
title('Crank Nicholson Scheme');
legend(legendInfo)

%% Analytical Solution at t=1hr

syms m;
x=linspace(0,L,Nx);
sum_fun=2*(Ti-Ts)*exp(-1*((m*pi)/L)^2*alpha*1)*(((1-(-
1)^m)/(m*pi))*sin((m*pi*x)/L));
T_an=vpa(Ts+symsum(sum_fun,m,1,inf),4);

%% Error

Err=(1/Nx)*sqrt(sum((T(Nt+1,:)-T_an).^2));
```

**Error Plot**

```
%% Error Plots

Errx_FTCS=[2.517 0.4684 0.1634 0.06898 0.03538];
Errx_CN=[2.516 0.4666 0.1618 0.06745 0.0339];
dx=[1 0.666666667 0.5 0.4 0.333333333];
```

```
loglog(dx,Errx_FTCS,'s--')
hold on
loglog(dx,Errx_CN,'*:')
title('Loglog plot for Error vs del_x');
xlabel('del_x');
ylabel('Error');
legend('FTCS','Crank-Nicholson');

mx_FTCS=(log(Errx_FTCS(4))-log(Errx_FTCS(3)))/(log(dx(4))-
log(dx(3)));
mx_CN=(log(Errx_CN(4))-log(Errx_CN(3)))/(log(dx(4))-log(dx(3)));

Errt_CN=[2.859 0.9761 0.2974 0.09625 0.05315 0.04071 0.03321
0.02789];
Errt_FTCS=[38.5 6.121 3.787 3.119 2.654 2.312 2.05 1.842];
dt=[0.333333333 0.25 0.2 0.166666667 0.142857143 0.125
0.111111111 0.1];

figure
loglog(dt,Errt_FTCS,'s--')
hold on
loglog(dt,Errt_CN,'*:')
xlim([10^-(1.15) 1]);
title('Loglog plot for Error vs del_t');
xlabel('del_t');
ylabel('Error');
legend('FTCS','Crank-Nicholson');

mt_FTCS=(log(Errt_FTCS(4))-log(Errt_FTCS(3)))/(log(dt(4))-
log(dt(3)));
mt_CN=(log(Errt_CN(2))-log(Errt_CN(1)))/(log(dt(2))-log(dt(1)));
```