Computational study of temperature distribution with unsteady heat equation using FTCS and Crank-Nicolson finite difference methods

Name: Chinmay Rajesh Chavan

Course: MEEN 689 – Computational Fluid Dynamics

Date: 10/19/2022

Introduction

Case for study:

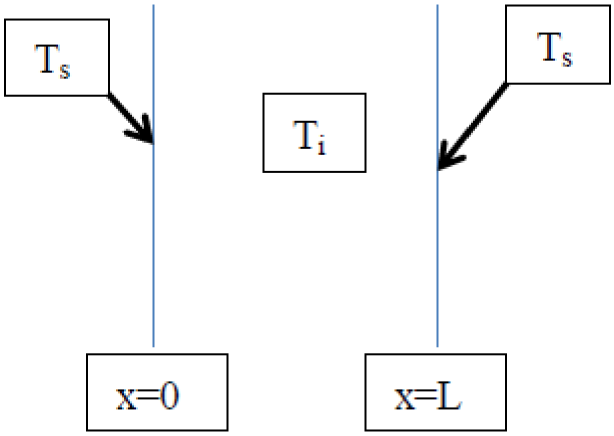

Fig.1. Problem depiction where blue lines are the wall boundaries

As shown in Fig.1. a wall of thickness (L) of **2ft** in the x-direction and infinite in the other directions has an initial uniform temperature (T_i) of **100°F**. The surface temperatures (T_s) at the two sides are suddenly increased and maintained at **400°F**. The wall is composed of nickel steel (40% Ni) with a diffusivity of α=**0.4** ft²/hr.

Governing unsteady heat equation is given by:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad \dots\dots\dots\dots\dots Eq.(1)$$

This is a **parabolic** PDE. …… (as here, A=α, B=0, C=0. Thus, B^2-4AC=0)
Analytical solution subject to initial and boundary conditions is given by:

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

To non-dimensionalize the equations, we take $t^* = \alpha t/L^2$, $x^* = x/L$, $T^* = (T-T_s)/(T_i-T_s)$

Upon substituting into Eq.(1) , we get non-dimensional governing equation as:

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} \quad \dots\dots\dots\dots\dots\dots Eq.(3)$$

Similarly, substituting into Eq.(2), we get non-dimensional solution as:

$$T^* = 2 \sum_{m=1}^{\infty} e^{-(m\pi)^2 t^*} \left[\frac{1-(-1)^m}{m\pi} \sin(m\pi x^*)\right] \quad \dots\dots\dots\dots\dots Eq.(4)$$

To get the temperature profile we conducted computational study using Forward time Central space (FTCS) and Crank-Nicolson finite difference schemes for several cases based on value of ***d*** which is defined as,

$$d = \frac{\Delta t^*}{\Delta x^{*2}} = \alpha \frac{\Delta t}{\Delta x^2} \quad \dots\dots\dots\dots\dots\dots\dots\dots Eq.(5)$$

For this study we have considered cases as shown in Table 1 and Table 2 for FTCS and Crank-Nicolson schemes respectively.

Case no.	Δt*	Δt (hr)	Δx*	Δx (ft)	d	Is it stable?
1	0.01	0.1	0.2	0.4	0.25	Yes
2	0.02	0.2	0.2	0.4	0.5	Yes
3	0.025	0.25	0.2	0.4	0.625	No
4	0.04	0.4	0.2	0.4	1	No
5	0.00001	0.0001	0.005	0.01	0.4	Yes
6	0.00001	0.0001	0.01	0.02	0.1	Yes
7	0.00001	0.0001	0.02	0.04	0.025	Yes
8	0.00001	0.0001	0.025	0.05	0.016	Yes
9	0.0002	0.002	0.05	0.1	0.08	Yes
10	0.0005	0.005	0.05	0.1	0.2	Yes
11	0.0008	0.008	0.05	0.1	0.32	Yes
12	0.001	0.01	0.05	0.1	0.4	Yes

Table 1. Cases considered for evaluating with FTCS explicit scheme

Case no.	Δt*	Δt (hr)	Δx*	Δx (ft)	d	Is it stable?
1	0.01	0.1	0.2	0.4	0.25	Yes
2	0.0001	0.001	0.02	0.04	0.25	Yes
3	0.0001	0.001	0.04	0.08	0.0625	Yes
4	0.0001	0.001	0.05	0.1	0.04	Yes
5	0.0001	0.001	0.1	0.2	0.01	Yes
6	0.00001	0.0001	0.02	0.04	0.025	Yes
7	0.0001	0.001	0.02	0.04	0.25	Yes
8	0.001	0.01	0.02	0.04	2.5	Yes
9	0.01	0.1	0.02	0.04	25	Yes

Table 2. Cases considered for evaluating with Crank-Nicolson implicit scheme

Method of Solution

Forward time-Central space scheme:

-This is an explicit scheme.

-It is found to be conditionally stable for $0 \leq d \leq 0.5$ using numerical analysis.

-Applying this scheme to the non-dimensional PDE from Eq.(3) gives us:

$$\frac{u_{(i)}^{(n+1)} - u_{(i)}^{(n)}}{\Delta t^*} = \frac{u_{(i+1)}^{(n)} - 2u_{(i)}^{(n)} + u_{(i-1)}^{(n)}}{\Delta x^{*2}} \quad \dots \dots \dots \dots Eq.(6)$$

-Upon solving Eq.(6) and substituting for d using Eq.(5) we get value of (n+1)th time step as:

$$u_{(i)}^{(n+1)} = u_{(i)}^{(n)} + d\left(u_{(i+1)}^{(n)} - 2u_{(i)}^{(n)} + u_{(i-1)}^{(n)}\right) \quad \dots \dots \dots \dots Eq.(7)$$

-From Eq.(7), we can see that this scheme depends on values of i-1, i, i+1 locations at the nth time step to give value of ith location at (n+1)th time step.

-The stencil for this scheme is as shown in Fig.2.

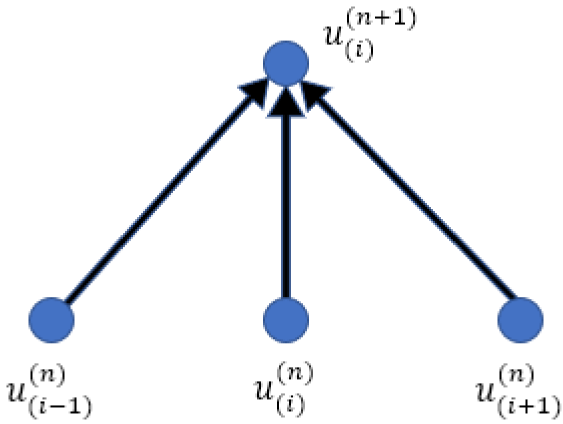

Fig.2. FTCS stencil

Crank-Nicolson scheme:

-This is an implicit scheme.

-It is found to be unconditionally stable for all $d > 0$ using numerical analysis.

-Applying this scheme to the non-dimensional PDE from Eq.(3) gives us:

$$\frac{u_{(i)}^{(n+1)} - u_{(i)}^{(n)}}{\Delta t^*} = \frac{u_{(i+1)}^{(n+1)} - 2u_{(i)}^{(n+1)} + u_{(i-1)}^{(n+1)}}{2\Delta x^{*2}} + \frac{u_{(i+1)}^{(n)} - 2u_{(i)}^{(n)} + u_{(i-1)}^{(n)}}{2\Delta x^{*2}} \quad \dots Eq.(8)$$

-Upon solving Eq.(8) and substituting for d using Eq.(5) we get value of (n+1)th time step as:

$$-\frac{d}{2}u_{(i-1)}^{(n+1)} + (1+d)u_{(i)}^{(n+1)} - \frac{d}{2}u_{(i+1)}^{(n+1)} = \frac{d}{2}u_{(i-1)}^{(n)} + (1-d)u_{(i)}^{(n)} + \frac{d}{2}u_{(i+1)}^{(n)} \quad \dots \dots Eq.(9)$$

-From Eq.(9), we can see that this scheme depends on values of i-1, i, i+1 locations at the n^{th} time step along with values of i-1, i+1 locations at the $(n+1)^{th}$ time step to give value of i^{th} location at $(n+1)^{th}$ time step.

-The stencil for this scheme is as shown in Fig.3.

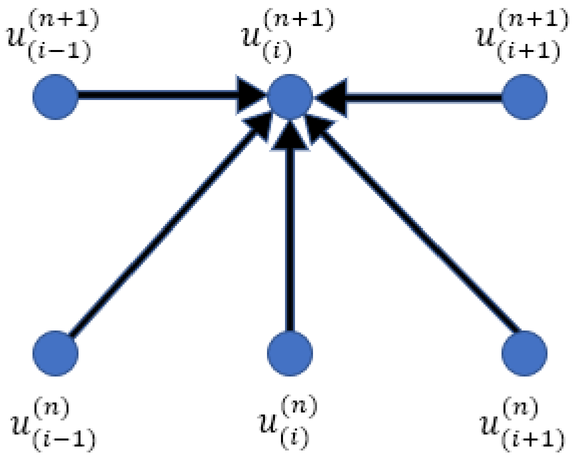

Fig.3. Crank-Nicolson stencil

Initial & boundary conditions:

The initial conditions for this problem in dimensional form are as follows:

At t = 0, T = T_i = 100°F for $0 \leq x \leq L$ …… (where L = 2 ft)

The boundary conditions for this problem in dimensional form are as follows:

At t = 0, T = T_s = 400°F for x = 0
& T = T_s = 400°F for x = L …… (where L = 2 ft)

The range changes for dimensional to non-dimensional parameters are shown in Table 3.

Parameter description	Dimensional range	Non-dimensional range
Time	$0 \leq t \leq 1hr$	$0 \leq t^* \leq 0.1$
Location	$0 \leq x \leq 2ft$	$0 \leq x^* \leq 1$
Temperature	$T_i = 100°F$ & $T_s = 400°F$	$T_i^* = 1$ & $T_s^* = 0$

Table 3. Parameter range in dimensional and non-dimensional form

The initial conditions for this problem in non-dimensional form are as follows:

At t^* = 0, T^* = 1 for $0 \leq x^* \leq 1$

The boundary conditions for this problem in non-dimensional form are as follows:

At t^* = 0, T^* = 0 for x^* = 0
& T^* = 0 for x^* = 1

Discussion of Results

For FTCS scheme:

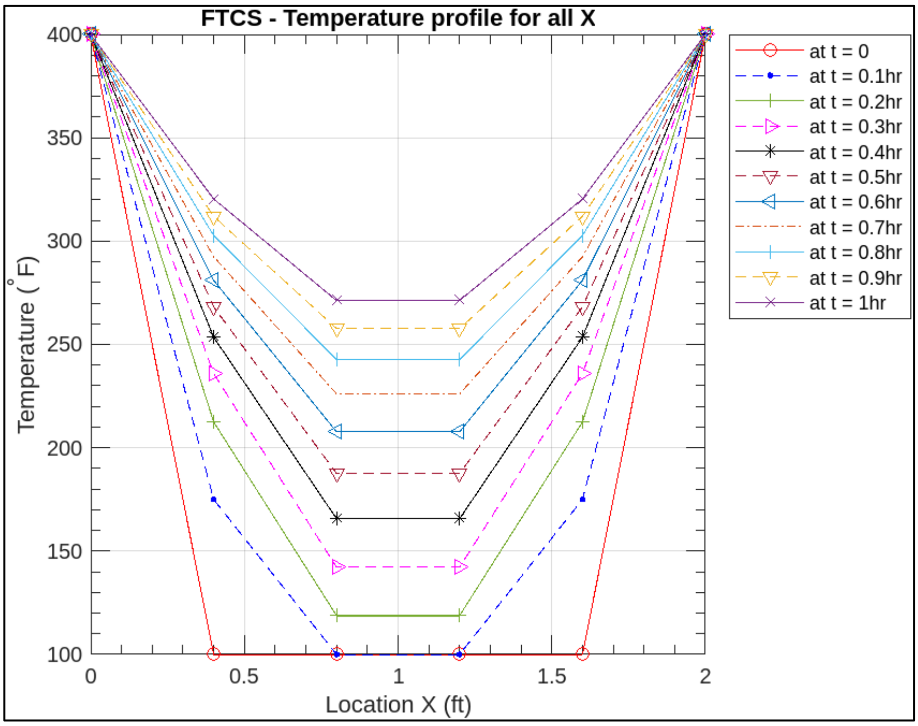

Fig.4. Temperature profile for all x location from 0 to 1hr using FTCS explicit scheme for Δx=0.4ft and Δt=0.1hr (i.e. d=0.25).

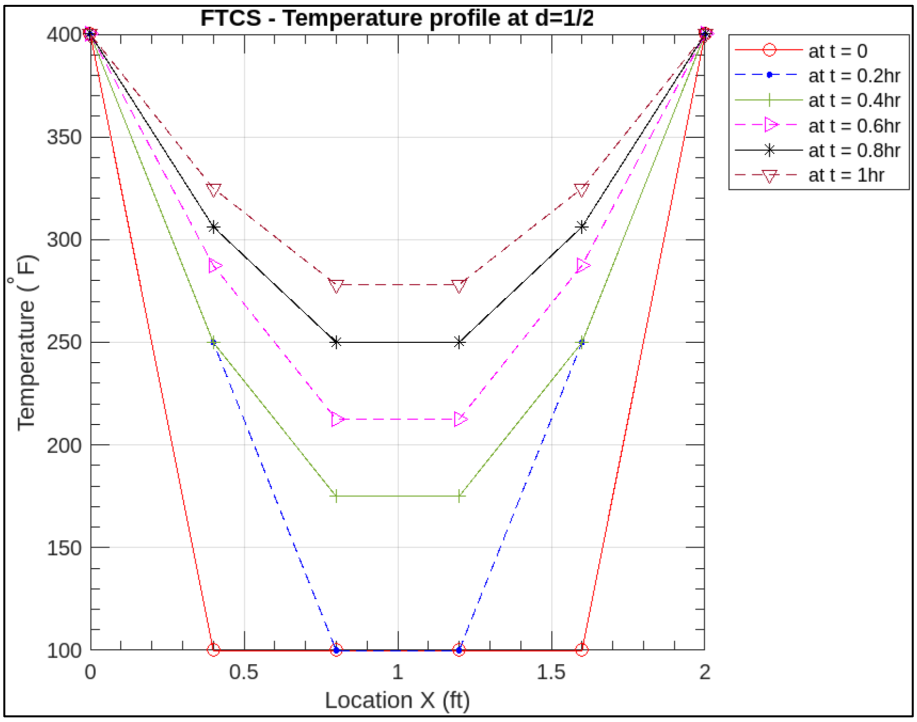

Fig.5. Temperature profile for all x location from 0 to 1hr using FTCS explicit scheme for Δx=0.4ft and Δt=0.2hr (i.e. d=0.5).

Figures 4 & 5 show the temperature profile as a function of location for cases 1 & 2 respectively. Here, we observe that we achieve stable plots for $d \leq 0.5$ using FTCS scheme.

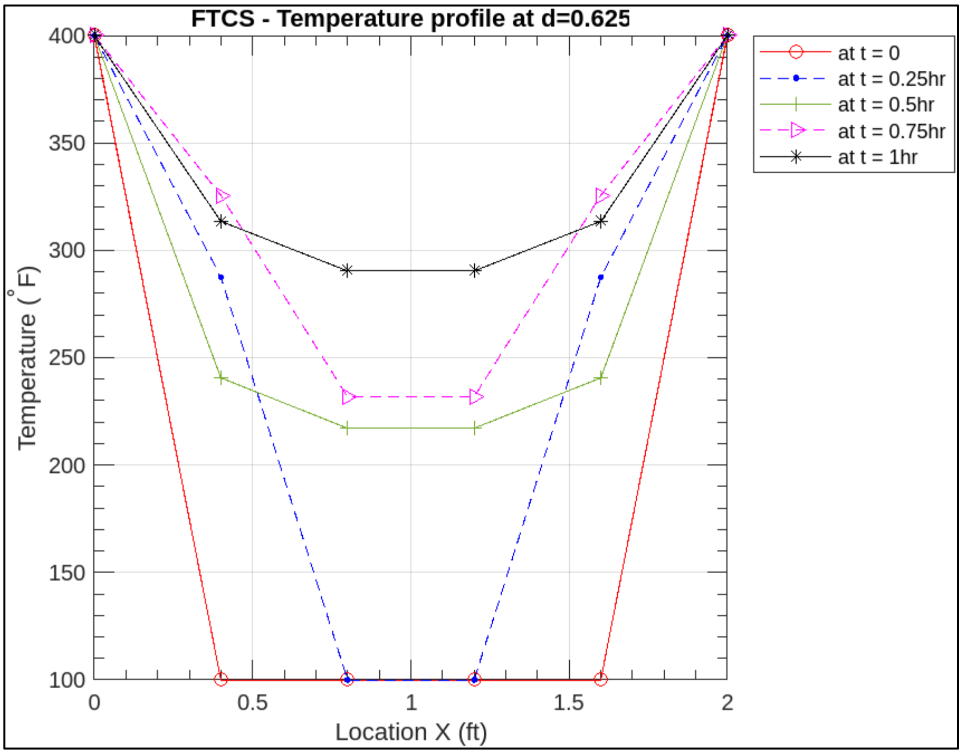

Fig.6. Temperature profile for all x location from 0 to 1hr using FTCS explicit scheme for Δx=0.4ft and Δt=0.25hr (i.e. d=0.625).

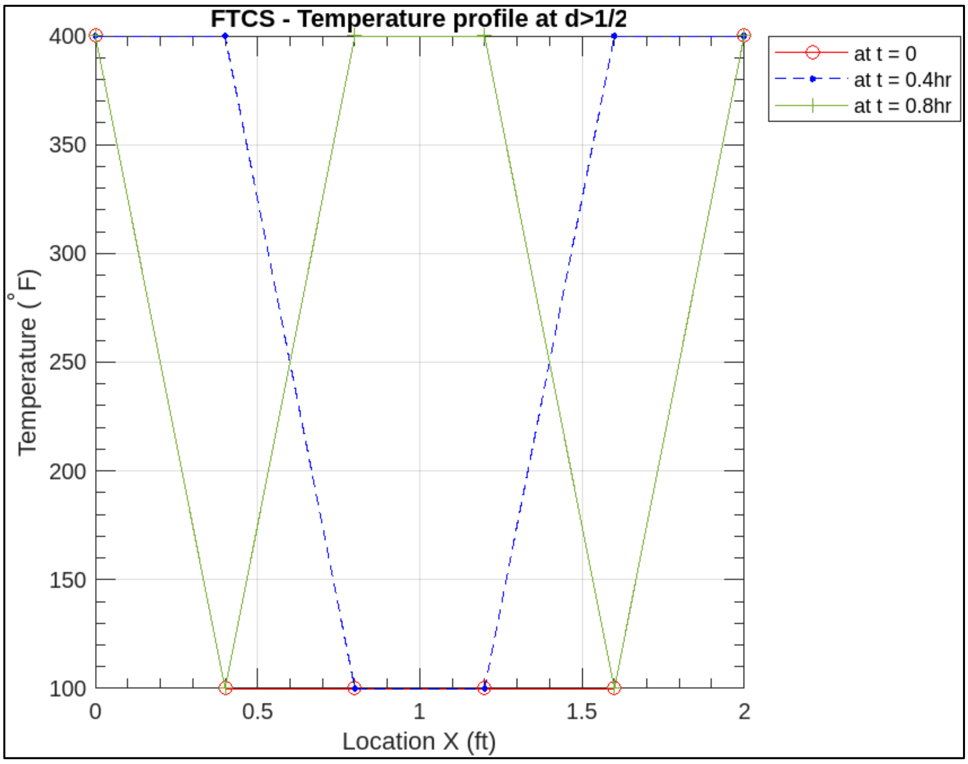

Fig.7. Temperature profile for all x location from 0 to 1hr using FTCS explicit scheme for Δx=0.4ft and Δt=0.4hr (i.e. d=1).

Figures 6 & 7 show the temperature profile as a function of location for cases 3 & 4 respectively. Here, we observe instability in the plots for d>0.5 using FTCS scheme. So, for further cases we consider d<0.5 to achieve stable results.

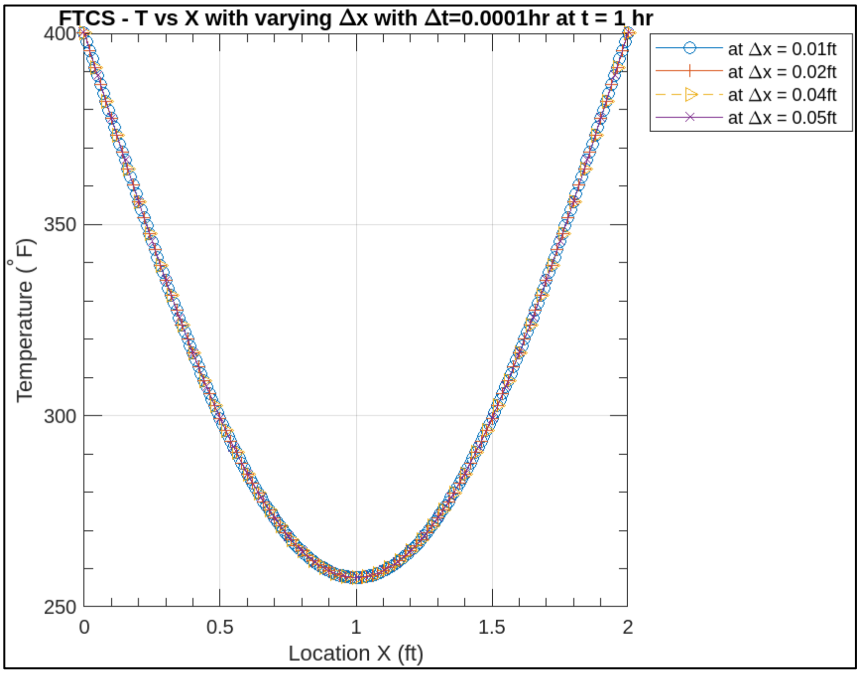

Fig.8. Temperature profile for all x location at t = 1hr using FTCS explicit scheme for varying Δx with fixed Δt=0.0001hr (d<0.5).

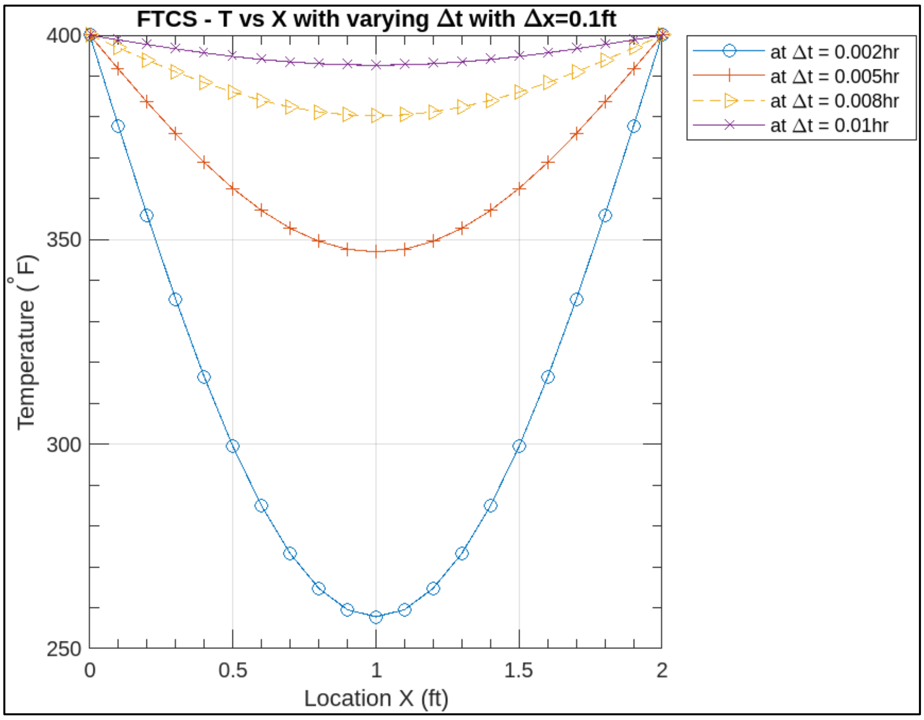

Fig.9. Temperature profile for all x location at t = 1hr using FTCS explicit scheme for varying Δt with fixed Δx=0.1ft (d<0.5).

Figure 8 shows the effect of spatial steps on the temperature profile for cases 5-8. We observe that all plots overlap showing that variation of spatial steps does not effect the solution accuracy and only results in a smoother curve when higher resolution is selected. Whereas the effect of temporal steps on solution accuracy is eminent as seen in Fig. 9 which displays results for cases 9-12. This shows that using higher resolution in time steps will give more accurate results, here we can see a difference of about 130°F at the peaks.

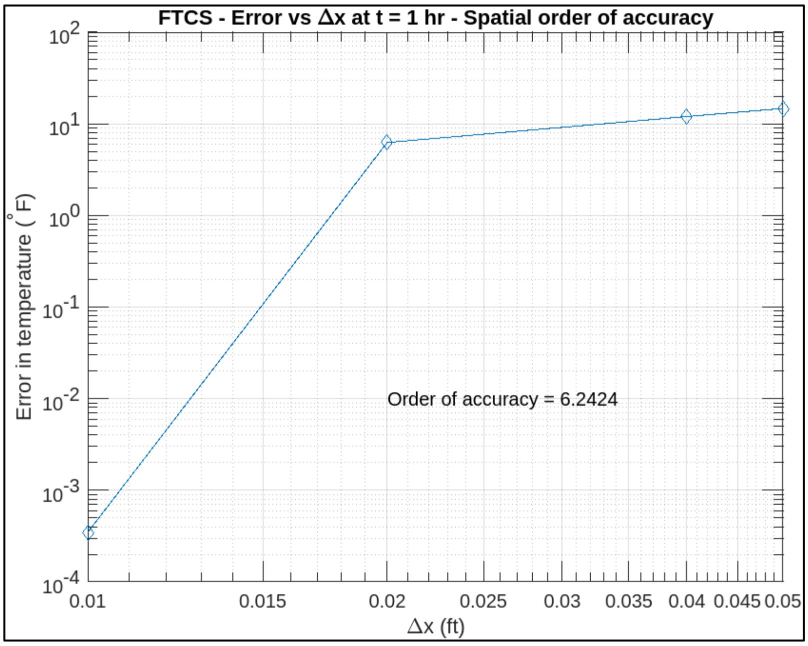

Fig.10. Error plot w.r.t. Δx in log-log scale at t = 1hr using FTCS explicit scheme to obtain spatial order of accuracy.

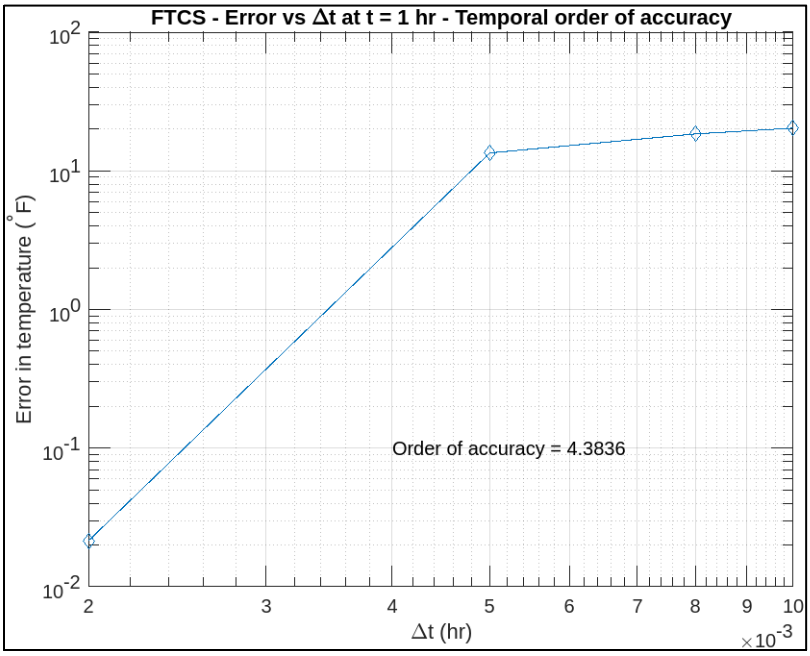

Fig.11. Error plot w.r.t. Δt in log-log scale at t = 1hr using FTCS explicit scheme to obtain temporal order of accuracy.

We get order of accuracy from slope of error plot in log-log scale. Figure 10 shows error as a function of Δx and from the plot we are getting spatial order of accuracy to be 6 using FTCS scheme. Figure 11 shows error as a function of Δt and from the plot we are getting temporal order of accuracy to be 4 using FTCS scheme. Theoretically, we should have second order accurate with spatial steps and first order accurate with time steps. This gap in computed and theoretical order of accuracy seems to exist because of narrow stability range of FTCS. To achieve stable plots, one must carefully select Δt & Δx values to get good results. Here, as it is only feasible for $d<0.5$, it is difficult to achieve a case where error due to Δt or Δx doesn't overpower the other and what we are seeing is the combination of both.

For Crank-Nicolson scheme:

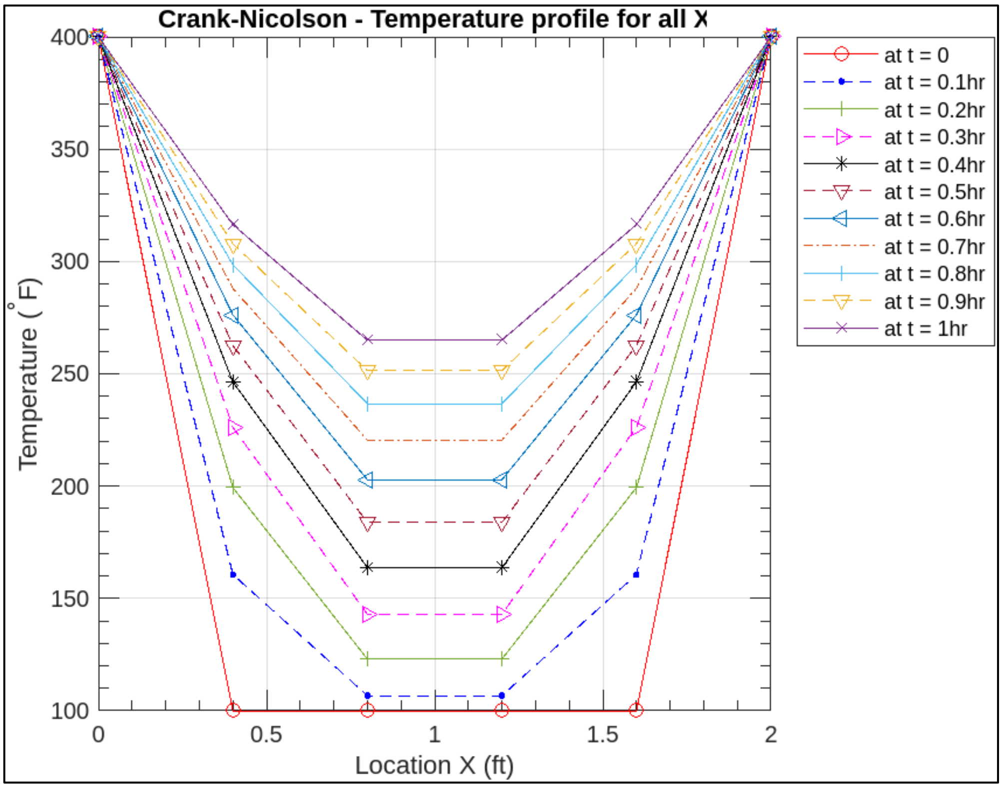

Fig.12. Temperature profile for all x location from 0 to 1hr using Crank-Nicolson implicit scheme for Δx=0.4ft and Δt=0.1hr (i.e. d=0.25).

Figure 12 shows the temperature profile as a function of location for case 1. Here, we observe that we achieve stable plots for d=0.25 using Crank-Nicolson scheme.

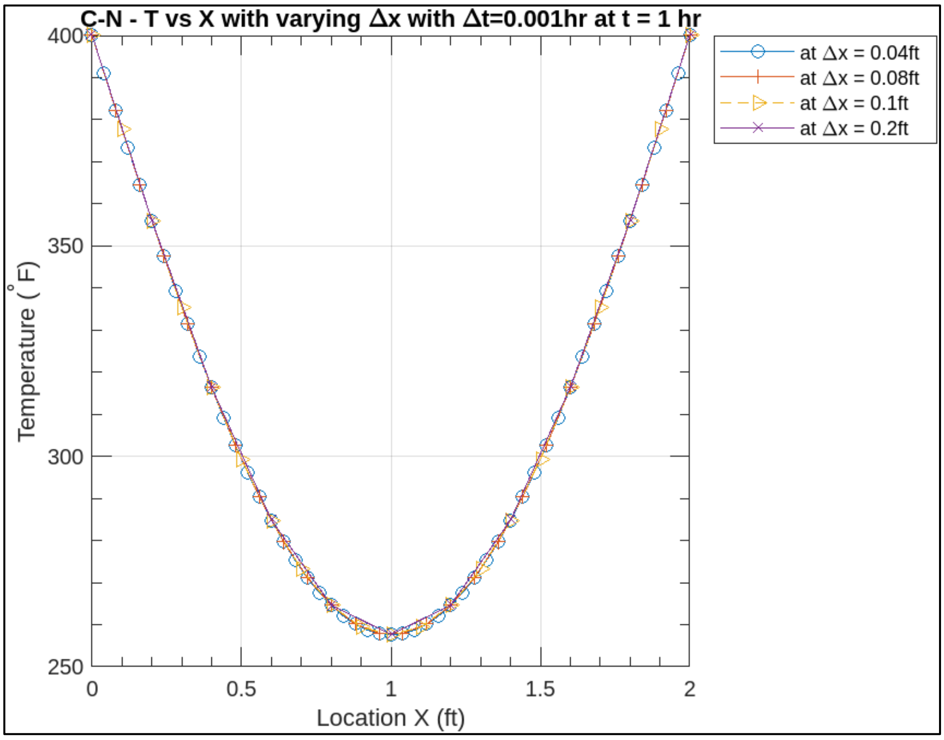

Fig.13. Temperature profile for all x location at t = 1hr using Crank-Nicolson implicit scheme for varying Δx with fixed Δt=0.001hr (0.01≤d≤0.25).

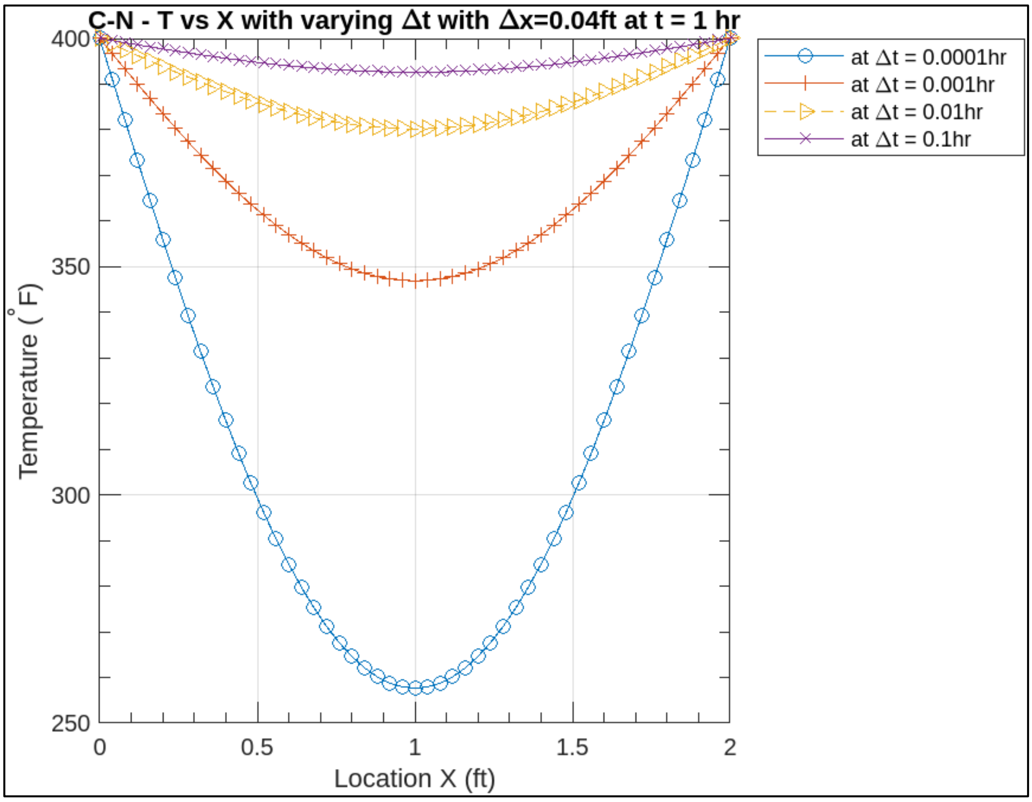

Fig.14. Temperature profile for all x location at t = 1hr using Crank-Nicolson implicit scheme for varying Δt with fixed Δx=0.04ft ($0.025 \leq d \leq 25$).

Figure 13 shows the effect of spatial steps on the temperature profile for cases 2-5. We observe that all plots overlap showing that variation of spatial steps does not effect the solution accuracy and only results in a smoother curve when higher resolution is selected. Whereas the effect of temporal steps on solution accuracy is eminent as seen in Fig. 14 which displays results for cases 6-9. This shows that using higher resolution in time steps will give more accurate results, here we can see a difference of about 140°F at the peaks. This result is similar to what we observed in FTCS scheme. From Fig. 14, we also observe that Crank-Nicolson scheme is stable for $0.025 \leq d \leq 25$ which aligns with the theoretically derived stability criteria of unconditional stability in Crank-Nicolson scheme.

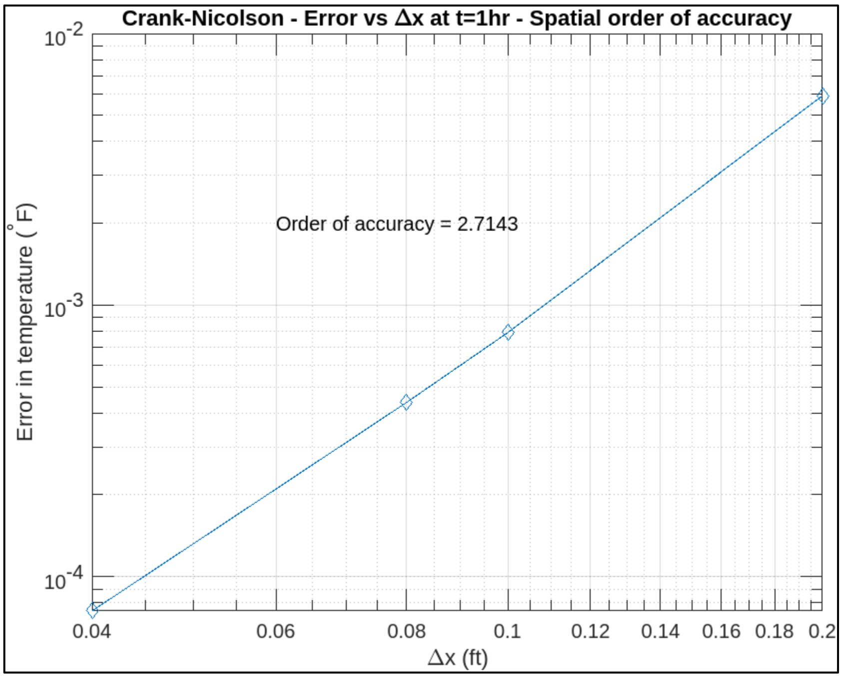

Fig.15. Error plot w.r.t. Δx in log-log scale at t = 1hr using Crank-Nicolson implicit scheme to obtain spatial order of accuracy.

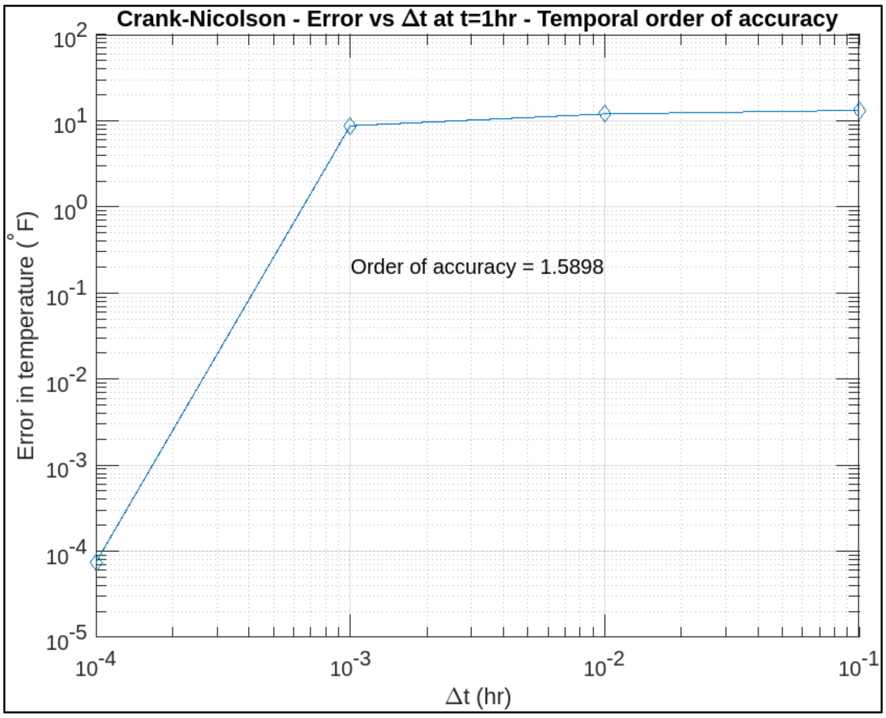

Fig.16. Error plot w.r.t. Δt in log-log scale at t = 1hr using Crank-Nicolson implicit scheme to obtain temporal order of accuracy.

Figure 15 shows error as a function of Δx and from the plot we are getting spatial order of accuracy to be 2.7 using Crank-Nicolson scheme. Figure 16 shows error as a function of Δt and from the plot we are getting temporal order of accuracy to be 1.5 using Crank-Nicolson scheme. This closely aligns with the theoretical values of second order accurate with spatial steps and first order accurate with time steps. Here, as we have freedom to choose Δt & Δx without worrying about stability, we can achieve proper results. We also observe that as the resolution is increased the error reduces and converges towards zero.

Summary & Conclusions

- From this study we got good alignment with theoretical stability criteria for both FTCS explicit and Crank-Nicolson implicit schemes.
- We were able to show that FTCS is stable for $0 \leq d \leq 0.5$ and Crank-Nicolson is unconditionally stable for all values of $d>0$.
- We observed that both schemes converge towards zero error when Δt & Δx are closer to zero.
- The order of accuracy obtained from FTCS scheme was not aligning with theoretical values for both temporal and spatial order of accuracy due to limitations on selecting value for d.
- We achieved good correlation between order of accuracy obtained using Crank-Nicolson and the theoretical values of first order temporal accuracy and second order spatial accuracy.
- To conclude, it can be said that FTCS explicit scheme is trickier to be applied because of small range of feasible d values as compared to no restriction on d values while using Crank-Nicolson implicit scheme.

Appendix

Calculations:

-Numerical stability analysis for FTCS scheme:

$$\frac{u_{(i)}^{(n+1)} - u_{(i)}^{(n)}}{\Delta t} - \alpha \frac{u_{(i+1)}^{(n)} - 2u_{(i)}^{(n)} + u_{(i-1)}^{(n)}}{\Delta x^2} = 0$$

error eqn: $\frac{e_{(i)}^{(n+1)} - e_{(i)}^{(n)}}{\Delta t} - \alpha \frac{e_{(i+1)}^{(n)} - 2e_{(i)}^{(n)} + e_{(i-1)}^{(n)}}{\Delta x^2} = 0$

we know, $e_{(i)}^{(n)} = E^{(n)} e^{J\theta i}$ & multiply by Δt

$\therefore E^{(n+1)} e^{J\theta i} - E^{(n)} e^{J\theta i} - \alpha \frac{\Delta t}{\Delta x^2}\left(E^{(n)} e^{J\theta (i+1)} - 2E^{(n)} e^{J\theta i} + E^{(n)} e^{J\theta (i-1)}\right) = 0$

put $d = \alpha \frac{\Delta t}{\Delta x^2}$ & divide by $e^{J\theta i}$

$\therefore E^{(n+1)} - E^{(n)} - d E^{(n)}\left(e^{J\theta} - 2 + e^{-J\theta}\right) = 0$

$\therefore E^{(n+1)} = E^{(n)} + d E^{(n)}(\cos\theta + J\sin\theta - 2 + \cos\theta - J\sin\theta) = 0$

$\therefore E^{(n+1)} = E^{(n)}(1 + d(2\cos\theta - 2))$

$\therefore \frac{E^{(n+1)}}{E^{(n)}} = 1 + 2d(\cos\theta - 1)$

For stability, $|G| \leq 1$

$\therefore -1 \leq 1 + 2d(\cos\theta - 1) \leq 1$

$\therefore -2 \leq 2d(\cos\theta - 1) \leq 0$

$\therefore -1 \leq d(\cos\theta - 1) \leq 0$

for $-\pi \leq \theta \leq \pi$

Worst case, $\cos\theta = -1 \Rightarrow \cos\theta - 1 = -2$

$\therefore -1 \leq d(-2) \leq 0$

$\therefore \frac{1}{2} \geq d \geq 0$

$\therefore \boxed{0 \leq d \leq \frac{1}{2}} \Rightarrow$ FTCS conditionally stable

-Numerical stability analysis for Crank-Nicolson scheme:

$$\frac{u_{(i)}^{(n+1)} - u_{(i)}^{(n)}}{\Delta t} - \alpha\left[\frac{u_{(i-1)}^{(n+1)} - 2u_{(i)}^{(n+1)} + u_{(i+1)}^{(n+1)}}{2\Delta x^2} + \frac{u_{(i-1)}^{(n)} - 2u_{(i)}^{(n)} + u_{(i+1)}^{(n)}}{2\Delta x^2}\right] = 0$$

error eqn.:

$$\frac{e_{(i)}^{(n+1)} - e_{(i)}^{(n)}}{\Delta t} - \alpha\left[\frac{e_{(i-1)}^{(n+1)} - 2e_{(i)}^{(n+1)} + e_{(i+1)}^{(n+1)}}{2\Delta x^2} + \frac{e_{(i-1)}^{(n)} - 2e_{(i)}^{(n)} + e_{(i+1)}^{(n)}}{2\Delta x^2}\right] = 0$$

we know, $e_{(i)}^{(n)} = E^{(n)} e^{J\theta i}$ & multiply by Δt

$$\therefore E^{(n+1)}e^{J\theta i} - E^{(n)}e^{J\theta i} - \frac{1}{2}\frac{\alpha\Delta t}{\Delta x^2}\left(E^{(n+1)}e^{J\theta(i-1)} - 2E^{(n+1)}e^{J\theta i} + E^{(n+1)}e^{J\theta(i+1)} + E^{(n)}e^{J\theta(i-1)} - 2E^{(n)}e^{J\theta i} + E^{(n)}e^{J\theta(i+1)}\right) = 0$$

put $d = \frac{\alpha\Delta t}{\Delta x^2}$ & divide by $e^{J\theta i}$

$$\therefore E^{(n+1)} - E^{(n)} - \frac{d}{2}\left[E^{(n+1)}(e^{-J\theta} - 2 + e^{J\theta}) + E^{(n)}(e^{-J\theta} - 2 + e^{J\theta})\right] = 0$$

we know, $e^{-J\theta} - 2 + e^{J\theta} = \cos\theta - J\sin\theta - 2 + \cos\theta + J\sin\theta = 2(\cos\theta - 1)$

$$\therefore E^{(n+1)} - E^{(n)} - \frac{d}{2}\left(E^{(n+1)} + E^{(n)}\right)\left(2(\cos\theta - 1)\right) = 0$$

$$\therefore E^{(n+1)}\left(1 - d(\cos\theta - 1)\right) = E^{(n)}\left(1 + d(\cos\theta - 1)\right)$$

$$\therefore \frac{E^{(n+1)}}{E^{(n)}} = \frac{1 + d(\cos\theta - 1)}{1 - d(\cos\theta - 1)}$$

for $\cos\theta = -1 \Rightarrow \cos\theta - 1 = -2$

$$\therefore \frac{E^{(n+1)}}{E^{(n)}} = \frac{1 - 2d}{1 + 2d} < 1$$ as $1 - 2d < 1 + 2d$ for $d > 0$

. Thus, $|G| < 1$

Alternatively,

for stability, $|G| \leq 1$

$$\therefore -1 \leq \frac{1 + d(\cos\theta - 1)}{1 - d(\cos\theta - 1)} \leq 1$$

$$-1+d(\cos\theta-1) \le 1+d(\cos\theta-1) \quad \& \quad 1+d(\cos\theta-1) \le 1-d(\cos\theta-1)$$

$$\therefore \; -1 \le 1 \qquad \& \qquad d(\cos\theta-1) \le -d(\cos\theta-1)$$

Always true for any $d>0$

$$\therefore \; \cos\theta - 1 \le -\cos\theta + 1$$

$$\therefore \; \cos\theta \le 1$$

Always true for any $d>0$

Thus, $|G| \le 1$ for any $d>0$

Thus, Crank-Nicolson scheme is <u>unconditionally stable</u>.

MATLAB codes:

1. Code for obtaining temperature profile for all x using FTCS explicit scheme at each Δt = 0.1hr interval from 0 to 1 hr.

```
clc;
figure;
newcolors =
{'#FF0000','#0000FF','#77AC30','#FF00FF','#000000','#A2142F','#0072BD','#D95319','
#4DBEEE','#EDB120','#7E2F8E'};
colororder(newcolors)
%% Define grid & Initialize T (at t=0)
t=0;
tD(1,1)=0;
Ti=100;
Ts=400;
L=2;
A=0.4;
%dx=0.25; %case1
dx=0.2; %case2
NX=1+1/dx;
for j=2:(NX-1)
    X(j)=((j-1)*dx);
    T0(j)=1;
end
X(1,NX)=1;
T0(1,1)=0;
T0(1,NX)=0;
T(1,:)=Ts+(Ti-Ts).*T0; %Dimensional temperature IC
%% Time marching (dt=0.1hr)
dt=0.01;
Nt=0.1/dt;
for ts=1:Nt
    t=t+dt;
    tD(1,ts+1)=t/0.1;
    for k=2:(NX-1)
        T1(1,k)=T0(1,k)+(dt/dx^2).*(T0(1,k+1)-2.*T0(1,k)+T0(1,k-1));
```

```
    end
    T1(1,1)=0;
    T1(1,NX)=0;
    for j=1:NX
        T0(1,j)=T1(1,j);
        T(ts+1,j)=Ts+(Ti-Ts)*T1(1,j); %Dimensional temperature matrix
    end
end
%% Plot temperature profile for all x at 0.1hr interval
mkrdata =["-o" "--." "-+" "-->" "-*" "--v" "-<" "-." "-|" "--v" "-x"];
for ts1=1:(Nt+1)
    plot(2*X,T(ts1,:),mkrdata(1,ts1));
    hold on;
    lgddata(1,ts1)="at t = " + tD(1,ts1)+ "hr";
end
lgddata(1,1)= "at t = 0";
%% Plot detailing for T vs X
title('FTCS - Temperature profile for all X');
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
legend(lgddata,"Location","northeastoutside");
grid on;
hold off;
```

2. Code for Temperature profile using FTCS with varying Δx & to obtain error vs Δx plot.

```
clc;
figure;
%% Define grid & Initialize T (at t=0)
t=0;
tD(1,1)=0;
Ti=100;
Ts=400;
L=2;
A=0.4;
n=0;
mkrdata =["-o" "-+" "-->" "-x"];
for dx=[0.005 0.01 0.02 0.025]
    n=n+1;
    NX=1+1/dx;
    X(1:NX)=0;
    for j=2:(NX-1)
        X(j)=((j-1)*dx);
        T0(j)=1;
    end
    X(1,NX)=1;
    T0(1,1)=0;
    T0(1,NX)=0;
    %% Time marching (dt=0.1hr)
    dt=0.00001;
    Nt=0.1/dt;
    for ts=1:Nt
        t=t+dt;
        tD(1,ts+1)=t/0.1;
        T1(1,:)=0;
        for k=2:(NX-1)
            T1(1,k)=T0(1,k)+(dt/dx^2).*(T0(1,k+1)-2.*T0(1,k)+T0(1,k-1));
```

```
        end
        T1(1,1)=0;
        T1(1,NX)=0;
        for j=1:NX
            T0(1,j)=T1(1,j);
        end
    end
    T(n,:)=Ts+(Ti-Ts).*T1; %Dimensional temperature value
    Temp=T(n,1:NX);
    plot(2*X(1:NX),T(n,1:NX),mkrdata(1,n));  %Plot with varying dx
    hold on;
    %% Analytical solution at t=1hr
    for j=1:NX
        T2=0;
        t1=A*tD(1,Nt+1)/(L^2);
        x1=X(1,j);
        for m=1:1000000
            T2=T2+2.*(exp(-t1.*(m.*pi)^2).*(1-(-1)^m).*sin(x1.*pi.*m)./(m.*pi));
        end
        TA(1,j)=T2.*(Ti-Ts)+Ts;   %Analytical temperature matrix
    end
    e=0;
    for j=1:NX
        e=e+(Temp(1,j)-TA(1,j))^2;
    end
    Er(n)=sqrt(e)/NX;
end
title("FTCS - T vs X with varying \Deltax with \Deltat=" + dt/0.1 + "hr at t = 1
hr");
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
dx=[0.005 0.01 0.02 0.025];
legend("at \Deltax = " + 2*dx +"ft","Location","northeastoutside");
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
hold off;
%plot(2*X,TA);  %To check analytical plot
%% Error plot
figure;
lg=loglog(2*dx,Er,"Marker","diamond");
p = polyfit(log(2*dx),log(Er),1);
acry=p(1,1);  %Order of accuracy
txt = "Order of accuracy = " + acry;
text(0.02,0.01,txt);
title('FTCS - Error vs \Deltax at t = 1 hr - Spatial order of accuracy');
xlabel('\Deltax (ft)');
ylabel('Error in temperature (^{\circ}F)');

set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
```

3. Code for Temperature profile using FTCS with varying Δt & to obtain error vs Δt plot.

```
clc;
figure;
%% Define grid & Initialize T (at t=0)
t=0;
Ti=100;
Ts=400;
```

```
L=2;
A=0.4;
dx=0.05;
n=0;
mkrdata =["-o" "-+" "-->" "-x"];
NX=1+1/dx;
X(1:NX)=0;
    for j=2:(NX-1)
        X(j)=((j-1)*dx);
        T0(j)=1;
    end
    X(1,NX)=1;
    T0(1,1)=0;
    T0(1,NX)=0;
    %% Time marching
for dt=[0.0002 0.0005 0.0008 0.001]
    n=n+1;
    Nt=0.1/dt;
    for ts=1:Nt
        t=t+dt;
        T1(1,:)=0;
        for k=2:(NX-1)
            T1(1,k)=T0(1,k)+(dt/dx^2).*(T0(1,k+1)-2.*T0(1,k)+T0(1,k-1));
        end
        T1(1,1)=0;
        T1(1,NX)=0;
        for j=1:NX
            T0(1,j)=T1(1,j);
        end
    end
    T=Ts+(Ti-Ts)*T1; %Dimensional temperature value
    Temp=T(1,1:NX);
    plot(2*X(1:NX),T(1,1:NX),mkrdata(1,n));  %Plot with varying dt
    hold on;
    %% Analytical solution at t=1hr
    for j=1:NX
        T2=0;
        t1=A/(L^2);
        x1=X(1,j);
        for m=1:1000000
            T2=T2+2.*(exp(-t1.*(m.*pi)^2).*(1-(-1)^m).*sin(x1.*pi.*m)./(m.*pi));
        end
        TA(1,j)=T2.*(Ti-Ts)+Ts;   %Analytical temperature matrix
    end
    e=0;
    for j=1:NX
        e=e+(Temp(1,j)-TA(1,j))^2;
    end
    Er(n)=sqrt(e)/NX;
end
title("FTCS - T vs X with varying \Deltat with \Deltax=" + dx*2 +"ft");
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
dt=[0.0002 0.0005 0.0008 0.001];
legend("at \Deltat = " + dt/0.1 +"hr","Location","northeastoutside");
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
hold off;
%% Error plot
```

```
figure;
lg=loglog(dt/0.1,Er,"Marker","diamond");
p = polyfit(log(dt/0.1),log(Er),1);
acry=p(1,1);  %Order of accuracy
txt = "Order of accuracy = " + acry;
text(0.004,0.1,txt);
title('FTCS - Error vs \Deltat at t = 1 hr - Temporal order of accuracy');
xlabel('\Deltat (hr)');
ylabel('Error in temperature (^{\circ}F)');
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
```

4. Code for obtaining temperature profile for all x using Crank-Nicolson implicit scheme at each Δt = 0.1hr interval from 0 to 1 hr.

```
clc;
figure;
newcolors =
{'#FF0000','#0000FF','#77AC30','#FF00FF','#000000','#A2142F','#0072BD','#D95319','
#4DBEEE','#EDB120','#7E2F8E'};
colororder(newcolors)
%% Define grid & Initialize T (at t=0)
t=0;
tD(1,1)=0;
Ti=100;
Ts=400;
L=2;
%dx=0.25; %case1
dx=0.2; %case2
NX=1+1/dx;
for j=2:(NX-1)
    X(j)=((j-1)*dx);
    T0(j)=1;
end
T0(1,1)=0;
T0(1,NX)=0;
T(1,:)=Ts+(Ti-Ts).*T0; %Dimensional temperature IC
%% Time marching (dt=0.1hr)
dt=0.01;
d=dt/dx^2;
Nt=0.1/dt;
for ts=1:Nt
    t=t+dt;
    tD(1,ts+1)=t/0.1;
    %TRI Tridiagonal solver
        for j=2:(NX-1)
            A(j)=-d/2;
            B(j)=1+d;
            C(j)=-d/2;
            D(j)=(d/2)*T0(1,j-1)+(1-d)*T0(1,j)+(d/2)*T0(1,j+1);
        end
        B(1,1)=1+d;
        B(1,NX)=1+d;
        for i=2:(NX-1)
            R=A(1,i)/B(1,i-1);
            B(1,i)=B(1,i)-R*C(1,i-1);
            D(1,i)=D(1,i)-R*D(1,i-1);
        end
        D(1,NX-1)=D(1,NX-1)/B(1,NX-1);
```

```
        for i=(NX-2):-1:2
            D(1,i)=(D(1,i)-C(1,i)*D(1,i+1))/B(1,i);
        end
    for k=2:(NX-1)
        T1(1,k)=D(1,k);
    end
    T1(1,1)=0;
    T1(1,NX)=0;
    for j=1:NX
        T0(1,j)=T1(1,j);
        T(ts+1,j)=Ts+(Ti-Ts)*T1(1,j); %Dimensional temperature matrix
    end
end
%% Plot temperature profile for all x at 0.1hr interval
x=linspace(0,L,NX);
mkrdata =["-o" "--." "-+" "-->" "-*" "--v" "-<" "-." "-|" "--v" "-x"];
for ts1=1:(Nt+1)
    plot(x,T(ts1,:),mkrdata(1,ts1));
    hold on;
    lgddata(1,ts1)="at t = " + tD(1,ts1)+ "hr";
end
lgddata(1,1)= "at t = 0";
%% Plot detailing
title('Crank-Nicolson - Temperature profile for all X');
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
legend(lgddata,"Location","northeastoutside");
grid on;
hold off;
```

5. Code for Temperature profile using Crank-Nicolson with varying Δx & to obtain error vs Δx plot.

```
clc;
figure;
%% Define grid & Initialize T (at t=0)
t=0;
tD(1,1)=0;
Ti=100;
Ts=400;
L=2;
a=0.4;
n=0;
mkrdata =["-o" "-+" "-->" "-x"];
for dx=[0.02 0.04 0.05 0.1]
    n=n+1;
    NX=1+1/dx;
    X(1:NX)=0;
    for j=2:(NX-1)
        X(j)=((j-1)*dx);
        T0(j)=1;
    end
    X(1,NX)=1;
    T0(1,1)=0;
    T0(1,NX)=0;
    %% Time marching (dt=0.1hr)
    dt=0.0001;
    d=dt/dx^2;
```

```
    Nt=0.1/dt;
    for ts=1:Nt
        t=t+dt;
        tD(1,ts+1)=t/0.1;
        %TRI Tridiagonal solver
            for j=2:(NX-1)
                A(j)=-d/2;
                B(j)=1+d;
                C(j)=-d/2;
                D(j)=(d/2)*T0(1,j-1)+(1-d)*T0(1,j)+(d/2)*T0(1,j+1);
            end
            B(1,1)=1+d;
            B(1,NX)=1+d;
            for i=2:(NX-1)
                R=A(1,i)/B(1,i-1);
                B(1,i)=B(1,i)-R*C(1,i-1);
                D(1,i)=D(1,i)-R*D(1,i-1);
            end
            D(1,NX-1)=D(1,NX-1)/B(1,NX-1);
            for i=(NX-2):-1:2
                D(1,i)=(D(1,i)-C(1,i)*D(1,i+1))/B(1,i);
            end
        for k=2:(NX-1)
            T1(1,k)=D(1,k);
        end
        T1(1,1)=0;
        T1(1,NX)=0;
        for j=1:NX
            T0(1,j)=T1(1,j);
        end
    end
    T(n,:)=Ts+(Ti-Ts).*T1; %Dimensional temperature value
    Temp=T(n,1:NX);
    plot(2*X(1:NX),T(n,1:NX),mkrdata(1,n));  %Plot with varying dx
    hold on;
    %% Analytical solution at t=1hr
    for j=1:NX
        T2=0;
        t2=a/(L^2);
        x1=X(1,j);
        for m=1:1000000
            T2=T2+2.*(exp(-t2.*(m.*pi)^2).*(1-(-1)^m).*sin(x1.*pi.*m)./(m.*pi));
        end
        TA(1,j)=T2.*(Ti-Ts)+Ts;   %Analytical temperature matrix
    end
    e=0;
    for j=1:NX
        e=e+(Temp(1,j)-TA(1,j))^2;
    end
    Er(n)=sqrt(e)/NX;
end
title("C-N - T vs X with varying \Deltax with \Deltat=" + dt/0.1 +"hr at t = 1
hr");
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
dx=[0.02 0.04 0.05 0.1];
legend("at \Deltax = " + 2*dx +"ft","Location","northeastoutside");
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
```

```
hold off;
%plot(2*X,TA);  %To check analytical plot
%% Error plot
figure;
lg=loglog(2*dx,Er,"Marker","diamond");
p = polyfit(log(2*dx),log(Er),1);
acry=p(1,1);  %Order of accuracy
txt = "Order of accuracy = " + acry;
text(0.06,0.002,txt);
title('Crank-Nicolson - Error vs \Deltax at t=1hr - Spatial order of accuracy');
xlabel('\Deltax (ft)');
ylabel('Error in temperature (^{\circ}F)');
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
```

6. Code for Temperature profile using Crank-Nicolson with varying Δt & to obtain error vs Δt plot.

```
clc;
figure;
%% Define grid & Initialize T (at t=0)
t=0;
tD(1,1)=0;
Ti=100;
Ts=400;
L=2;
a=0.4;
dx=0.02;
n=0;
mkrdata =["-o" "-+" "-->" "-x"];
NX=1+1/dx;
X(1:NX)=0;
    for j=2:(NX-1)
        X(j)=((j-1)*dx);
        T0(j)=1;
    end
X(1,NX)=1;
T0(1,1)=0;
T0(1,NX)=0;
%% Time marching
for dt=[0.00001 0.0001 0.001 0.01]
    n=n+1;
    d=dt/dx^2;
    Nt=0.1/dt;
    for ts=1:Nt
        t=t+dt;
        tD(1,ts+1)=t/0.1;
        %TRI Tridiagonal solver
            for j=2:(NX-1)
                A(j)=-d/2;
                B(j)=1+d;
                C(j)=-d/2;
                D(j)=(d/2)*T0(1,j-1)+(1-d)*T0(1,j)+(d/2)*T0(1,j+1);
            end
            B(1,1)=1+d;
            B(1,NX)=1+d;
            for i=2:(NX-1)
                R=A(1,i)/B(1,i-1);
                B(1,i)=B(1,i)-R*C(1,i-1);
```

```
                D(1,i)=D(1,i)-R*D(1,i-1);
            end
            D(1,NX-1)=D(1,NX-1)/B(1,NX-1);
            for i=(NX-2):-1:2
                D(1,i)=(D(1,i)-C(1,i)*D(1,i+1))/B(1,i);
            end
        for k=2:(NX-1)
            T1(1,k)=D(1,k);
        end
        T1(1,1)=0;
        T1(1,NX)=0;
        for j=1:NX
            T0(1,j)=T1(1,j);
        end
    end
    T(n,:)=Ts+(Ti-Ts).*T1; %Dimensional temperature value
    Temp=T(n,1:NX);
    plot(2*X(1:NX),T(n,1:NX),mkrdata(1,n));  %Plot with varying dx
    hold on;
    %% Analytical solution at t=1hr
    for j=1:NX
        T2=0;
        t2=a/(L^2);
        x1=X(1,j);
        for m=1:1000000
            T2=T2+2.*(exp(-t2.*(m.*pi)^2).*(1-(-1)^m).*sin(x1.*pi.*m)./(m.*pi));
        end
        TA(1,j)=T2.*(Ti-Ts)+Ts;   %Analytical temperature matrix
    end
    e=0;
    for j=1:NX
        e=e+(Temp(1,j)-TA(1,j))^2;
    end
    Er(n)=sqrt(e)/NX;
end
title("C-N - T vs X with varying \Deltat with \Deltax=" + dx*2 +"ft at t = 1 hr");
xlabel('Location X (ft)');
ylabel('Temperature (^{\circ}F)');
dt=[0.00001 0.0001 0.001 0.01];
legend("at \Deltat = " + dt/0.1 +"hr","Location","northeastoutside");
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
hold off;
%% Error plot
figure;
lg=loglog(dt/0.1,Er,"Marker","diamond");
p = polyfit(log(dt/0.1),log(Er),1);
acry=p(1,1);  %Order of accuracy
txt = "Order of accuracy = " + acry;
text(0.001,0.2,txt);
title('Crank-Nicolson - Error vs \Deltat at t=1hr - Temporal order of accuracy');
xlabel('\Deltat (hr)');
ylabel('Error in temperature (^{\circ}F)');
set(gca,'XMinorTick','on','YMinorTick','on','TickLength',[0.03, 0.005]);
grid on;
```