Stability analysis in the unsteady problem

Aim:

To figure out and explain the following;

1. How does an unstable solution look like?

2. At what CFL number does the solution become unstable ?

Solution:

- For any transient problem (unsteady), there are two possible methods of solving such as explicit and implicit. The implicit schemes are solved by iterative techniques such as Jacobi, Gauss Seidel, SOR. In explicit schemes, the equations are solved at the very same node. Hence there is no use of iterative techniques.
- Stability of a solution in transient problem depends on two factors. They are time step and CFL number.

Unstable solution:

An unstable solution will provide an erroneous plot consisting of an empty contour or an evenly distributed contour inferring that the governing equations are not solved at every nodal point.

Time Step:

Time step is the incremental change in time for which the governing equations are being solved.

CFL number:

The Courant–Friedrichs–Lewy or CFL condition is a condition for the stability of unstable numerical methods that model convection or wave phenomena.

$$C = U \frac{\Delta t}{\Delta x} \leq 1$$

Where C is the courant number, U is the dependent variable, Δt is the time interval and Δx is the distance between any two consecutive nodes.

In case of 2D heat conduction problem,

$$C = \alpha\left(\frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2}\right) \leq 0.5$$

Δx and Δy is the distance between any two consecutive nodes along x and y axis respectively.

From the formulae, it is clear that the CFL number is influenced by the change in time step or distance between the nodes. Whenever time step is increased, CFL number is increased (directly proportional) and when distance between nodes is decreased or the mesh is more fine, CFL number is increased.

Let us consider the 2D heat conduction problem,

1. **Transient Explicit:**

Program:

```
clear all

close all

clc


%steady state eqn.

%length of the domain is 1 (Unit square)

%Tl=400,Tr=800,Tt=600,Tb=900


%Number of nodes along x&y

nx=10;

ny=10;


%Grid spacing

x=linspace(0,1,nx);

y=linspace(0,1,ny);

[X,Y]=meshgrid(x,y); %converts a vector into matrix

dx=x(2)-x(1);

dy=y(2)-y(1);
```

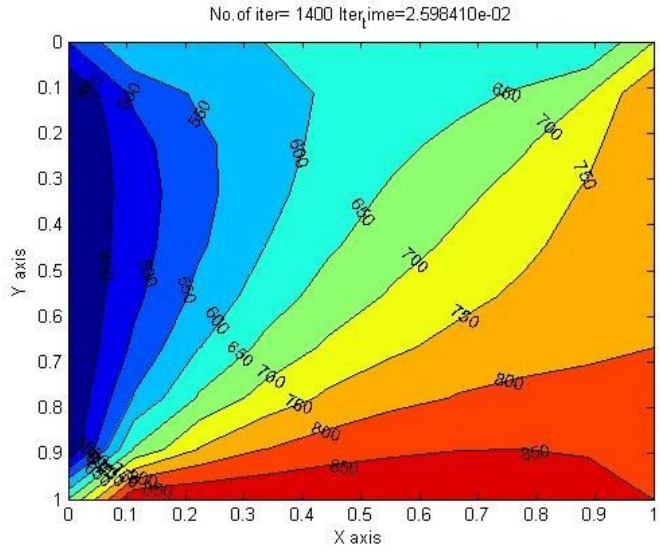
```
%BCs
T=ones(nx,ny);
T(:,1)=400;   %T at left
T(:,10)=800;  %T at right
T(1,:)=600;   %T at top
T(10,:)=900;  %T at bottom
T(1,1)= 500;  %[T(:,1)+T(1,:)]/2
T(1,10)= 700; %[T(1,:)+T(:,10)]/2
T(10,1)=650;  %[T(:,1)+T(10,:)]/2
T(10,10)=850; %[T(10,:)+T(:,10)]/2

Told=T;
dt=0.001;
alpha=1.4;
K1=(alpha*dt)/(dx^2);
K2=(alpha*dt)/(dy^2);

tol=1e-4;
error=9e9;
nt=1400;

tic
computation_start=0;
for k=1:nt
```

```
        while error>tol
        for i=2:nx-1
        for j=2:ny-1
           T1=Told(i,j);
           T2=K1*(Told(i+1,j)-2*Told(i,j)+Told(i-1,j));
           T3=K2*(Told(i,j+1)-2*Told(i,j)+Told(i,j-1));
           T(i,j)=T1+T2+T3;  %Tp=(1/K)*[((T+Tr)/(dx^2))+((Tt+Tb)/(dy^2))]
        end
        end
        error_vector=max(abs(Told-T));
        error=max(error_vector); %error single element
        Told=T;
        end
        computation_start=computation_start+1;
end
toc
time_taken=toc;

figure(1)
contourf(x,y,T);
[x,y]=contourf(x,y,T);
clabel(x,y);       %to label the points over the contour
xlabel('X axis');
ylabel('Y axis');
set(gca,'YDIR','reverse');
```

```
title_name=sprintf('No.of iter= %d Iter_time=%d',computation_start,time_taken);

title(title_name)
```

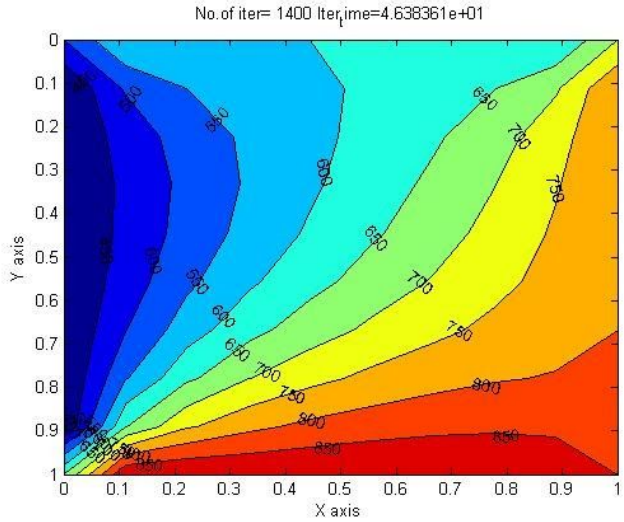
Output:

a)When dt=0.001, dx=dy=10, CFL =0.28



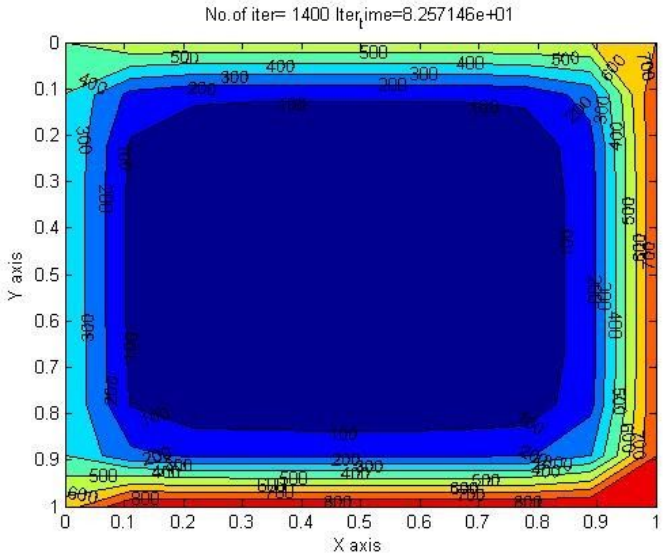
In the above case, the result is stable with a reasonable computation time.

b) When dt = 0.0000001 ,dx=dy=10, CFL= 0.000028



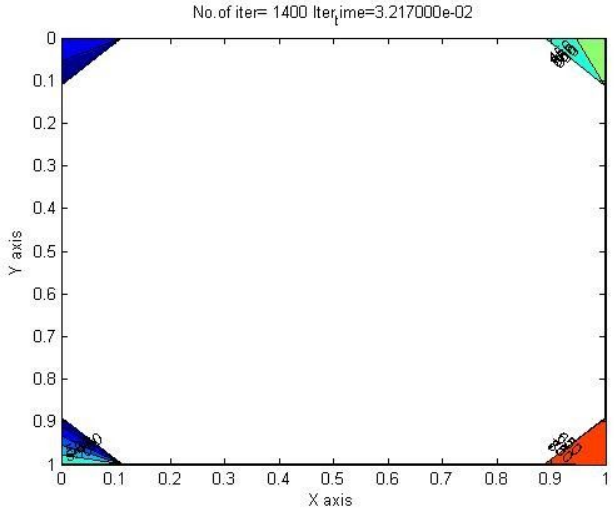
As the CFL number is decreased by marginally, the computation time taken for the same iterations has been increased.

c) When dt = 0.000000001 ,dx=dy=10, CFL= 0.00000028



In the above case, even though the CFL is well below 0.5, due to lower value of dt, truncation error arises and disturbs the stability of the solution.

d) When dt=0.005, dx=dy=10, CFL =1.4



When the CFL number is increased above 0.5, the result is completely unstable as solutions of all nodes are not captured.

2. **Transient Implicit (Jacobi):**

Program:

```
clear all

close all

clc


%steady state eqn

%length of the domain is 1 (Unit square)

%Tl=400,Tr=800,Tt=600,Tb=900


%Number of nodes along x&y

nx=10;

ny=10;


%Grid spacing

x=linspace(0,1,nx);

y=linspace(0,1,ny);

[X,Y]=meshgrid(x,y); %converts a vector into matrix

dx=x(2)-x(1);

dy=y(2)-y(1);


%BCs

T=ones(nx,ny);
```

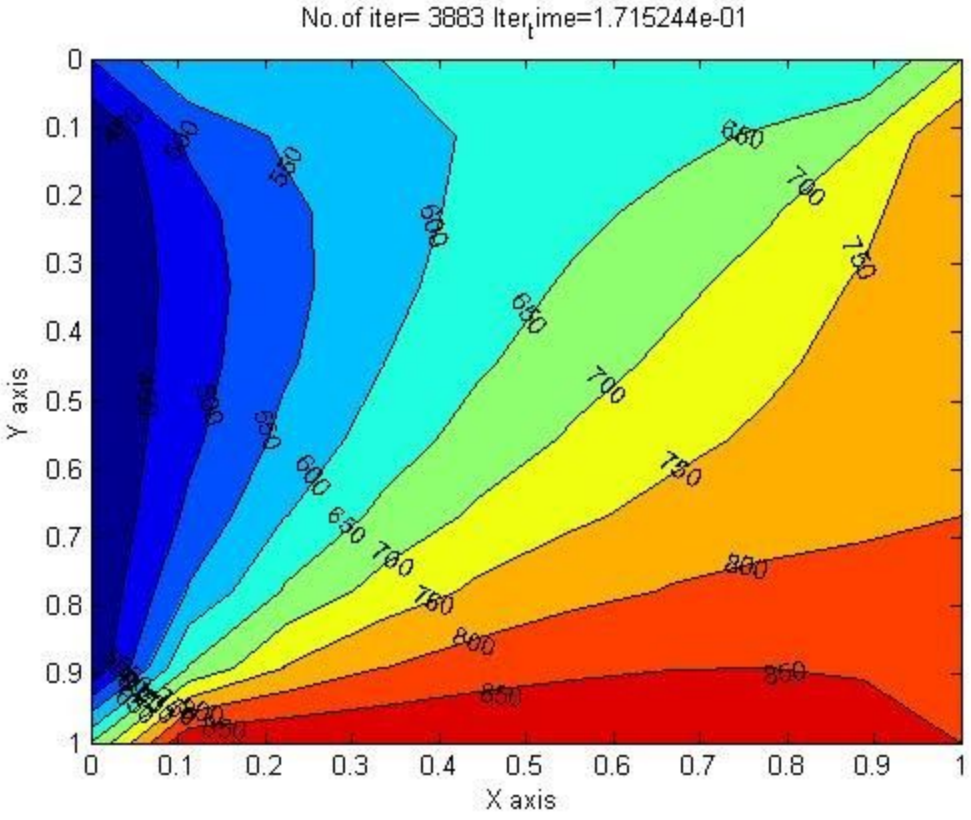
```
T(:,1)=400;   %T at left
T(:,10)=800;  %T at right
T(1,:)=600;   %T at top
T(10,:)=900;  %T at bottom
T(1,1)= 500;  %[T(:,1)+T(1,:)]/2
T(1,10)= 700; %[T(1,:)+T(:,10)]/2
T(10,1)=650;  %[T(:,1)+T(10,:)]/2
T(10,10)=850; %[T(10,:)+T(:,10)]/2

Told=T;
T_prev=Told;
dt=0.0001;
alpha=1.4;
K1=(alpha*dt)/(dx^2);
K2=(alpha*dt)/(dy^2);
nt=1400;

tic
jacobi_iter=1;
if (jacobi_iter==1)
for t=1:nt
   tol=1e-4;
   error=9e9;
       while error>tol
       for i=2:nx-1
```

```
        for j=2:ny-1
           term1=1/(1+(2*K1)+(2*K2));
           term2=K1*term1;
           term3=K2*term1;
           H=(Told(i-1,j)+Told(i+1,j));
           V=(Told(i,j-1)+Told(i,j+1));
           T(i,j)=(T_prev(i,j)*term1)+(H*term2)+(V*term3);
        end
        end
        errori=max(abs(Told-T));
        error=max(errori); %error single element
        Told=T;
        jacobi_iter=jacobi_iter+1;
        end
       T_prev=T;
end
end
toc
time_taken=toc;

figure(1)

[x,y]=contourf(x,y,T);
clabel(x,y);       %to label the points over the contour
xlabel('X axis');
```

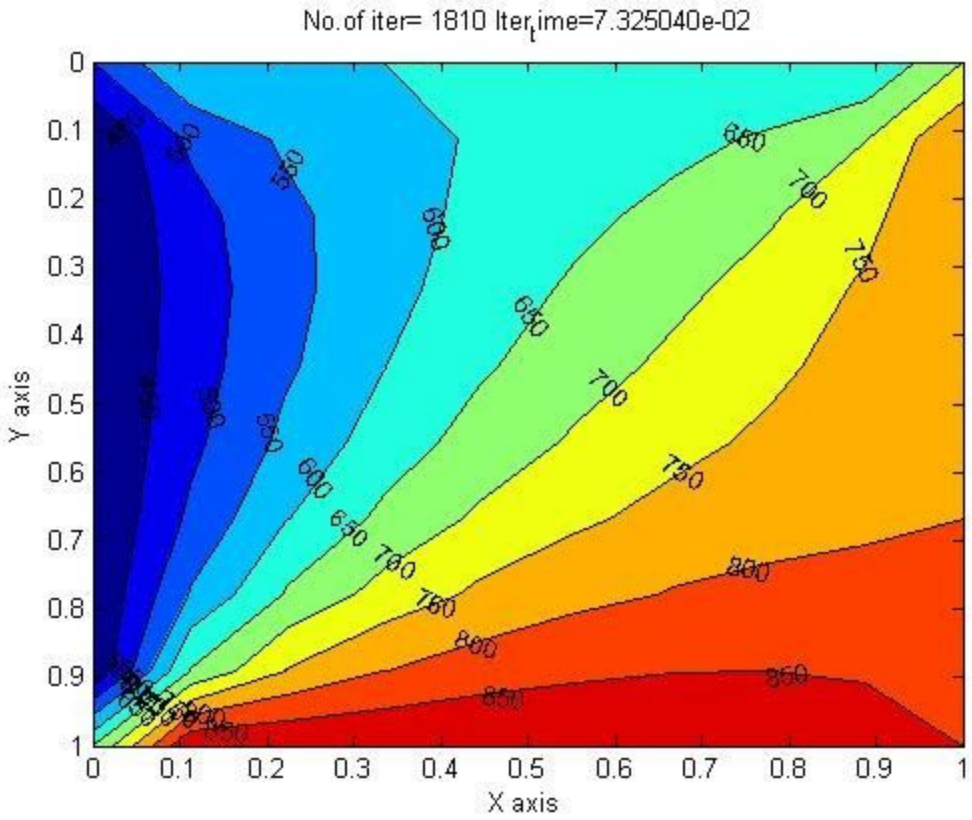
```
ylabel('Y axis');
set(gca,'YDIR','reverse');
title_name=sprintf('No.of iter= %d Iter_time=%d',jacobi_iter,time_taken);
title(title_name)
```

Output:

a) **When dt = 0.001**



b)When dt = 5



The system remains stable even after increasing the time step by a huge margin.

Inference:

1. In transient explicit simulation, large values of time step are not tolerated by the numerical scheme since it would raise the CFL value beyond 0.5 resulting in an unstable solution. Thus they are conditionally stable.
2. In transient implicit simulations using jacobi, Gauss Seidel or SOR, CFL value does not influence the stability of the solution since the solutions are solved iteratively using the time data from neighbourhood points. Thus they are called as unconditionally stable.