MAE 6263 Course Project 2

S M Abdullah Al Mamun
CWID : A20138451

April 17, 2018

1. Introduction:

This report is aimed to document the numerical results and analysis of a given 2D viscous Burgers equation for a plane channel flow. The domain is considered rectangular having twice unit length in horizontal direction then the vertical direction length. The basic finite difference scheme is used here to get the results in a proper iterative way. The main focus is given on the understanding the effect of the viscosity and grid discretization through the global Reynolds number and cell Peclet number.

1.1 Problem statement:

The 2D viscous Burgers equation for Cartesian coordinates is given as follows:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}.\nabla u = \nu \Delta \vec{u} \qquad (1.1)$$

Consider a rectangular domain (Ω) of unit vertical length and twice horizontal length. The initial conditions given by

$$At\ t = 0\ \{\vec{u}\,(x, y) = 0\ for\ 0 < x < 2, 0 < y < 1; \qquad (1.2)$$

and the boundary conditions given by

$$\begin{aligned} \vec{u}(x, y) &= 1\ for\ y = 1; \\ \vec{u}(x, y) &= 0\ for\ y = 0; \\ \vec{u}(x, y) &= \vec{u}(x-2, y) for\ x = 2; \\ \vec{u}(x, y) &= \vec{u}(x+2, y) for\ x = 0; \end{aligned} \qquad (1.3)$$

2. Code development

2.1.1 Explicit Scheme

For this problem we will discretize 2D Burgers equation using central difference scheme for the viscous term, both central difference and first order upwind for advective term.

Let us first assume central difference discretization for the advective term. The following explicit scheme can be devised for the x component of the fluid velocity u.

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\ \frac{u_{i+1,j}^{n}-u_{i-1,j}^{n}}{2\Delta x}+v_{i,j}^{n}\,\frac{u_{i,j+1}^{n}-u_{i,j-1}^{n}}{2\Delta y}=$$

$$\nu\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^{2}}+\ \frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^{2}}\right); \tag{2.1}$$

Similarly, equation can be devised for the y component of the fluid velocity v as follows

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\ \frac{v_{i+1,j}^{n}-v_{i-1,j}^{n}}{2\Delta x}+v_{i,j}^{n}\,\frac{v_{i,j+1}^{n}-v_{i,j-1}^{n}}{2\Delta y}=$$

$$\nu\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^{2}}+\ \frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^{2}}\right); \tag{2.2}$$

A similar set of expressions can also be derived for first order upwind scheme for the advective term. They may be represented as follows for both x component and y component of velocity.

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\ \frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\,\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}=$$

$$\nu\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^{2}}+\ \frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^{2}}\right); \tag{2.3}$$

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\ \frac{v_{i,j}^{n}-v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\,\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y}=$$

$$\nu\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^{2}}+\ \frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^{2}}\right); \tag{2.4}$$

The iterative Explicit method can be summarized in the following pseudocode -

Algorithm 1 : Explicit solver for 2D heat diffusion equation –
Input : $\gamma, Re, Pe, u_{i,j}^{n}, v_{i,j}^{n}$ Run : while ($u_{i,j}^{n+1}$-$u_{i,j}^{n}$) $> \epsilon$ and ($v_{i,j}^{n+1}$-$v_{i,j}^{n}$) $> \epsilon$ check for advective derivative discretization check for velocity direction if FOU for i=1:Δx:X_length for j =1: :Δy:Y_length Update $u_{i,j}^{n+1}$ from $u_{i,j}^{n}$ Update $v_{i,j}^{n+1}$ from $v_{i,j}^{n}$ end end end

2.1.2 Implicit Scheme

An implicit iterative scheme can be generated for the 2D Burgers equation using a backward in time discretization. The spatial derivatives are determined in a manner similar to the explicit method. We first describe the central difference scheme for the advective derivative.

For the x component of velocity we have

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t} + u_{i,j}^{n+1}\ \frac{u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}}{2\Delta x} + v_{i,j}^{n+1}\ \frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2\Delta y} =$$

$$\nu\left(\frac{u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}}{\Delta y^2}\right); \tag{2.5}$$

Similarly for the y component of velocity we have

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t} + u_{i,j}^{n+1}\ \frac{v_{i+1,j}^{n+1}-v_{i-1,j}^{n+1}}{2\Delta x} + v_{i,j}^{n+1}\ \frac{v_{i,j+1}^{n+1}-v_{i,j-1}^{n+1}}{2\Delta y} =$$

$$\nu\left(\frac{v_{i+1,j}^{n+1}-2v_{i,j}^{n+1}+v_{i-1,j}^{n+1}}{\Delta x^2} + \frac{v_{i,j+1}^{n+1}-2v_{i,j}^{n+1}+v_{i,j-1}^{n+1}}{\Delta y^2}\right); \tag{2.6}$$

An implicit scheme can be devised in exactly the same manner but with first order upwinding used for the advective term and is shown below.

For x component of velocity

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t} + u_{i,j}^{n+1}\ \frac{u_{i,j}^{n+1}-u_{i-1,j}^{n+1}}{\Delta x} + v_{i,j}^{n}\,\frac{u_{i,j}^{n+1}-u_{i,j-1}^{n+1}}{\Delta y} =$$

$$\nu\left(\frac{u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}}{\Delta y^2}\right); \qquad (2.7)$$

And for the y component of velocity

$$\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t} + u_{i,j}^{n+1}\ \frac{v_{i,j}^{n+1}-v_{i-1,j}^{n+1}}{\Delta x} + v_{i,j}^{n}\,\frac{v_{i,j}^{n+1}-v_{i,j-1}^{n+1}}{\Delta y} =$$

$$\nu\left(\frac{v_{i+1,j}^{n+1}-2v_{i,j}^{n+1}+v_{i-1,j}^{n+1}}{\Delta x^2} + \frac{v_{i,j+1}^{n+1}-2v_{i,j}^{n+1}+v_{i,j-1}^{n+1}}{\Delta y^2}\right); \qquad (2.8)$$

For both the case of explicit and implicit we should check the velocity of each component whether it is negative or positive to determine the upwinding direction as well as the discretization equation.

A pseudocode for the computation is given by

```
Algorithm 2 : Explicit solver for 2D Burgers equation –
Input : γ, Re, Pe, u^n_{i,j}, v^n_{i,j}
Run :   while (u^{n+1}_{i,j}-u^n_{i,j}) > ϵ and (v^{n+1}_{i,j}-v^n_{i,j}) > ϵ
        u^{n+1}_{i,j}= u^n_{i,j} ;    v^{n+1}_{i,j}= v^n_{i,j} ;
        Do Gauss-Seidel
        while (u^{k+1}_{i,j}-u^k_{i,j}) > norm and (v^{k+1}_{i,j}-v^k_{i,j}) > norm
                for i=1:Δx:X_length
                        for j =1: :Δy:Y_length
                                Update u^{k+1}_{i,j} from u^k_{i,j}
                                Update v^{k+1}_{i,j} from v^k_{i,j}
                        end
                end
        end
        end
```

2.2 Von Neumann Stability Analysis

2.2.1 Explicit method

Here the 2D PDE for equation is given by

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

Let $T^n_{i,j}$ = computed solution

$\overline{T^n_{i,j}}$ = exact solution of PDE

$\varepsilon^n_{i,j}$ = error at time level n in mesh point (i,j)

If we substitute these into the FDE using the explicit scheme we get

$$\frac{T^{n+1}_{i,j} - T^n_{i,j}}{\Delta t} + \frac{\varepsilon^{n+1}_{i,j} - \varepsilon^n_{i,j}}{\Delta t} = \frac{T^n_{i+1,j} - 2T^n_{i,j} + T^n_{i-1,j}}{\Delta x^2} + \frac{T^n_{i,j+1} - 2T^n_{i,j} + T^n_{i,j-1}}{\Delta y^2} + \frac{\varepsilon^n_{i+1,j} - 2\varepsilon^n_{i,j} + \varepsilon^n_{i-1,j}}{\Delta x^2} + \frac{\varepsilon^n_{i,j+1} - 2\varepsilon^n_{i,j} + \varepsilon^n_{i,j-1}}{\Delta y^2} \qquad (2.1)$$

Which then simplifies to the following

$$\frac{\varepsilon_{i,j}^{n+1}-\varepsilon_{i,j}^{n}}{\Delta t}=\frac{\varepsilon_{i+1,j}^{n}-2\varepsilon_{i,j}^{n}+\varepsilon_{i-1,j}^{n}}{\Delta x^2}+\frac{\varepsilon_{i,j+1}^{n}-2\varepsilon_{i,j}^{n}+\varepsilon_{i,j-1}^{n}}{\Delta y^2} \quad (2.2)$$

So we get that the error satisfies the original differential equation.

Represnting $\varepsilon_{i,j}^{n}$ as a Fourier series gives

$$\varepsilon_{i,j}^{n} = \sum_{k_x=-N}^{N}\sum_{k_y=-N}^{N} E_{k_x k_y}^{n}\, e^{i\frac{\pi k_x x}{L}}\; e^{i\frac{\pi k_y y}{L}}$$

Where $i=\sqrt{-1}$

Here $i\frac{\pi k_x x}{L}=i\frac{\pi k_x (j\Delta x)}{L}=ij\varphi_{k_x}$ & $i\frac{\pi k_y y}{L}=i\frac{\pi k_y (j\Delta y)}{L}=ij\varphi_{k_y}$

So we get

$$\varepsilon_{i,j}^{n} = \sum_{k_x=-N}^{N}\sum_{k_y=-N}^{N} E_{k_x k_y}^{n}\, e^{ij\varphi_{k_x}}\; e^{ij\varphi_{k_y}} \qquad (2.3)$$

Since we are dealing with linear schemes the error equation must be satisfied by each harmonic separately. Substituting eq. (2.2.3) into eq. (2.2.2) and taking one harmonic with the consideration $\Delta x=\Delta y$ we can write

$$e^{ij\varphi_k}\frac{(E_k^{n+1}-E_k^{n})}{\Delta t}=\frac{2}{(\Delta x)^2}E_k^{n}\left(e^{i(j+1)\varphi_k}-2e^{ij\varphi_k}+\ e^{i(j-1)\varphi_k}\right)$$

Further simplification gives

$$\frac{(E_k^{n+1}-E_k^{n})}{E_k^{n}}=2\gamma(e^{i\varphi}-2+e^{-i\varphi}) \qquad \text{where } \gamma=\frac{\Delta t}{\Delta x^2}$$

Which further becomes $\frac{E_k^{n+1}}{E_k^{n}}=1-8\gamma\sin^2\frac{\varphi}{2}=\mathrm{G}$

where G is defined as the amplification factor which must be less than 1 for stability for all φ i.e $\left|1-8\gamma\sin^2\frac{\varphi}{2}\right|\le 1$

which gives the following two conditions for stability

$$\gamma\ge 0 \ \text{ and } \ \gamma\le\frac{1}{4}$$

The explicit method thus can be said conditionally stable.

2.2.2 <u>**Implicit Method**</u>

Using similar procedure as above we can derive amplification factor for implicit scheme as follows

$$G = \frac{E_k^{n+1}}{E_k^n} = \frac{1}{1+8\gamma\sin^2\frac{\varphi}{2}}$$

for stability for all φ we get $\left|\frac{1}{1+8\gamma\sin^2\frac{\varphi}{2}}\right| \leq 1$

which gives us two conditions for stability

$$\gamma \geq 0$$

$$\gamma \geq -\frac{1}{4}$$

Which implies that the implicit method for the 2D heat equation is stable since γ is always greater or equal to zero.

2.2.3 Amplification factor for exact solution

Amplification factor for the exact solution is given by

$$G_k^e = e^{-\gamma\varphi_k^2}$$

3. Numerical Results and Analysis

Amplification Factor (G)

Using the amplification factor expressions derived in the previous section for implicit, explicit and exact solution three different plots are made for varying phase angel φ_k and stability factor γ. These are shown in figs. 3.1-3.3.

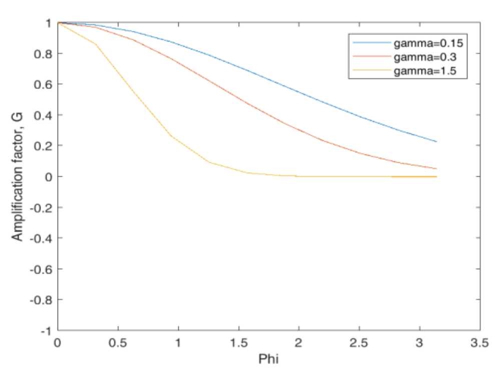

Figure 3.1 : Amplification factor for exact solution of the PDE at different γ

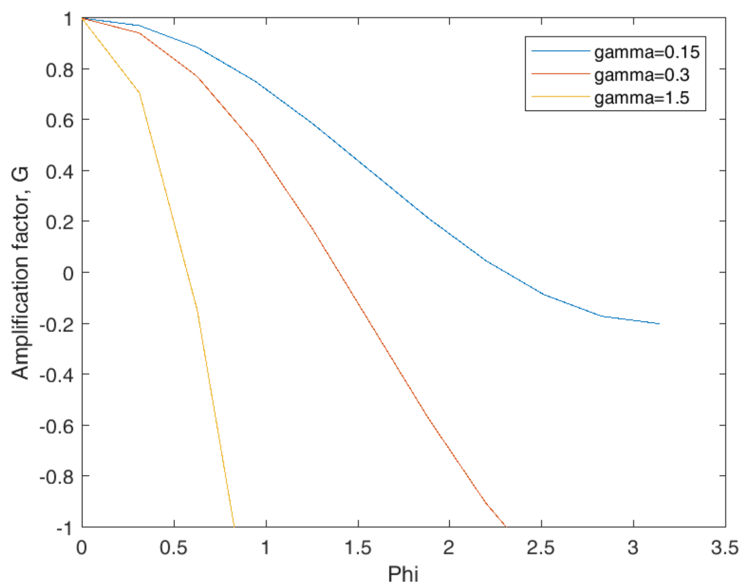

Figure 3.2: Amplification factor of the explicit method for different γ

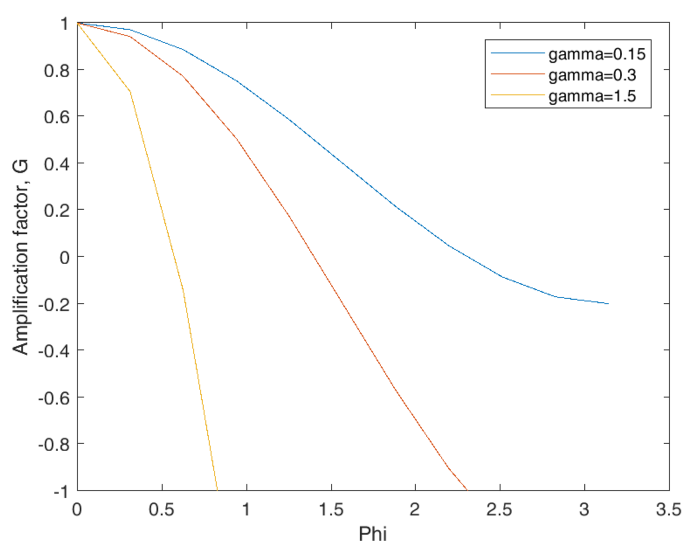

Figure 3.3: Amplification factor of the implicit method for different γ

Iso-contour

Iso-contour plot of implicit and explicit one, also the error plot from the exact (considering the implicit 100x100 mesh at γ=1.5 at time level 0.1s as exact solution) given below.

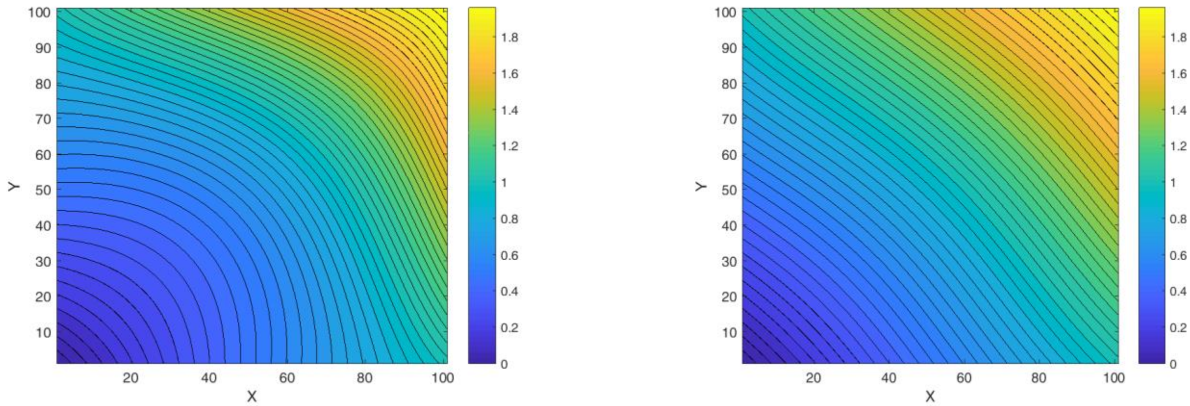

Figure 3.4 : Iso-contour plot of Implicit(left) and Explicit(right) solution from 100x100 mesh at γ=0.15 and at t=0.1s

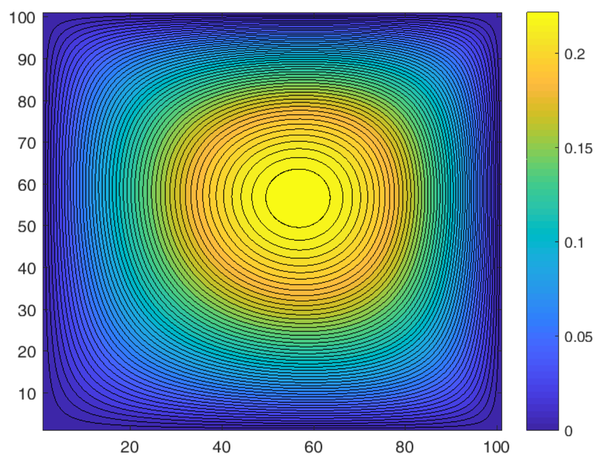

Figure : 3.5: Iso-contour plot of error between Implicit and Explicit solution from 100x100 mesh at γ=0.15 and at t=0.1s (Considering the Implicit one as exact solution)

<u>Comment</u>

From the above iso-contour figure it can be derived that it will take more than 0.1s to reach steady state for implicit one and the explicit one is close to steady state but require a little more time too. The error iso-contour plot shows that the error is lager at the center position of the system which represent that the heat reaches steady state at the center later than the other positions.

Impact of time-step size:

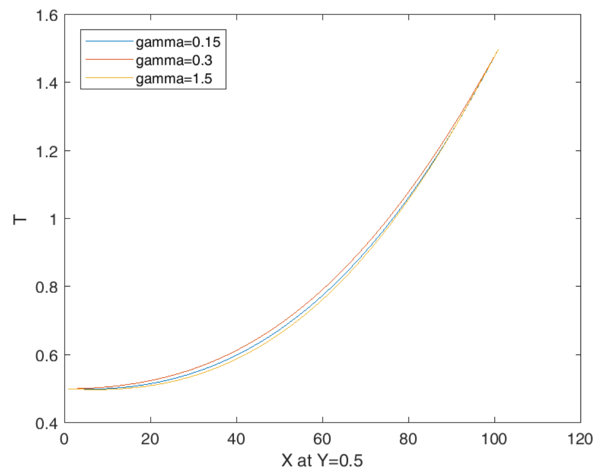

Figure 3.6 : T-x at y=0.5 for implicit method with resolution 100x100 at t=0.1s for different values of γ

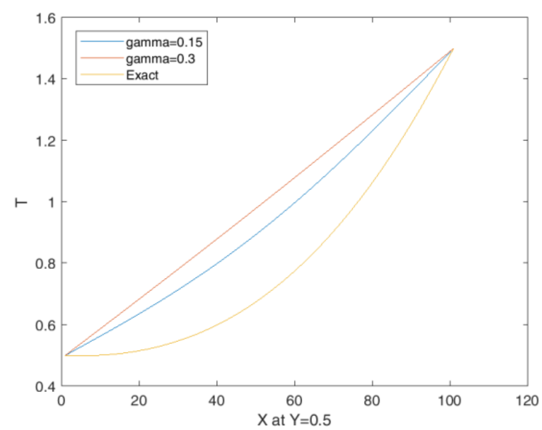

Figure 3.7 : T-x at y=0.5 for explicit method with resolution 100x100 at t=0.1s for different values of γ

Comment

From the above figures we can derive that for implicit method the solution is stable for any values of γ greater than zero. For the explicit one the solution is not stable when the value of γ is greater than $\frac{1}{4}$.

Impact of spatial resolution:

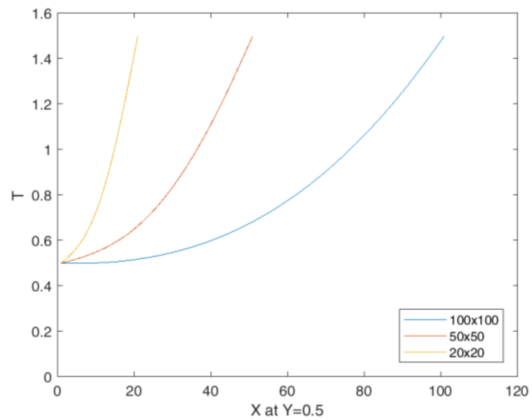

Figure 3.8: T-x at y=0.5 for implicit method and for γ=0.15 at t=0.1s for various mesh size

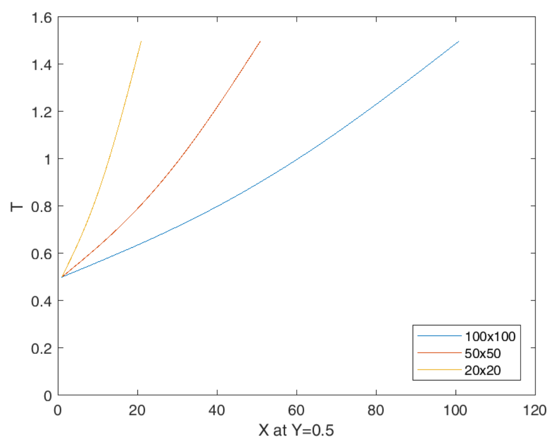

Figure 3.9: T-x at y=0.5 for explicit method and for γ=0.15 at t=0.1s for various mesh size

Comment

From the above figures of spatial resolution it is observed that larger mesh gives smooth solution.

Impact of numerical method:

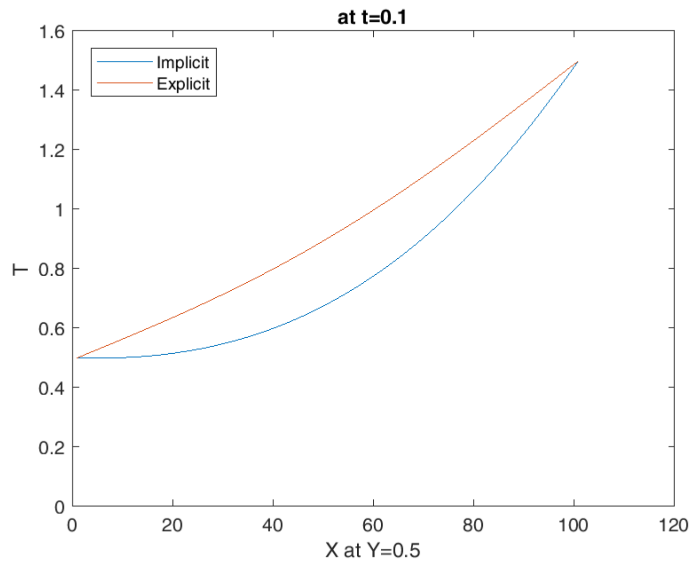

Figure 3.10: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.1s.

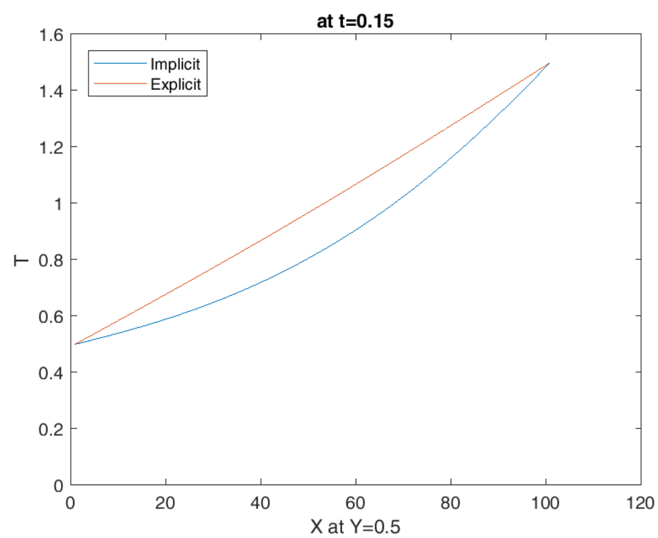

Figure 3.11: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.15s.

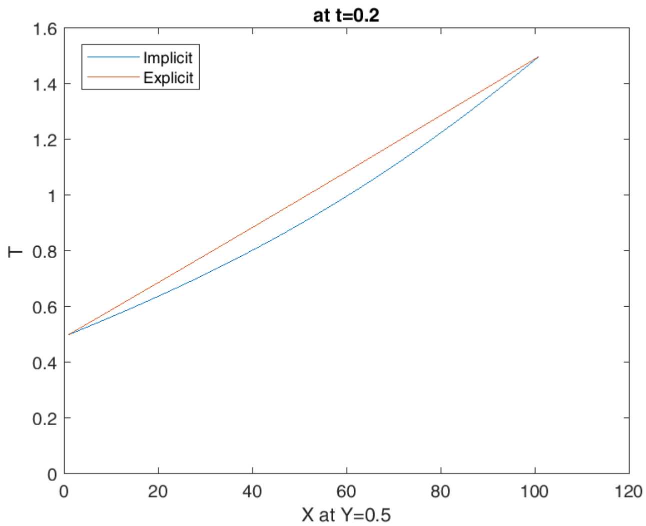

Figure 3.12: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.2s.

Comment

From the above figures we can conclude that with the increasing time both implicit and explicit results come to steady state and they both take more than 0.15s to reach steady state.

4. Comment on efficacy

All simulations were carried out using MATLAB R2017a. It was found that implicit method takes longer time for the computation. If considering the stability and accuracy then the implicit scheme is better scheme than the explicit one.