MAE 6263 Course Project 1

S M Abdullah Al Mamun
CWID : A20138451

April 1, 2018

1. Introduction:

This report is aimed to document the numerical results and analysis of a given heat diffusion problem within a 2D system. The domain is considered square and of unit length on each direction. 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 concepts of finite difference equation, consistency, stability and solution methods.

1.1 Problem statement:

The 2D heat diffusion equation for Cartesian coordinates is given as follows:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \qquad (1.1)$$

Consider a square domain (Ω) of unit side length with initial conditions given by

$$T(x,y)=0 \text{ for } 0<x<1,\ 0<y<1 \qquad (1.2)$$

and the time independent boundary conditions given by

$$T(x,y) = x+y \qquad (1.3)$$

2. Code development

2.1.1 Explicit Scheme

Considering the expansion point as $T_{i,j}^{n}$ the partial differential equation given by eq.(1.1) may be discretized by utilizing a forward time and central space discretization. Where $T_{i,j}^{n}$ represents the solution of the PDE at the node (i,j) in the discretized domain at time n. The discretized equation can be written as

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

Reorganizing the terms we can obtain an algebraic equation for the solution at a node for its new time step provided the solution is known at the previous time step and the equation becomes

$$T_{i,j}^{n+1} = \gamma\left(T_{i+1,j}^{n} + T_{i-1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n}\right) + (1-4\gamma)\, T_{i,j}^{n}$$

Where γ is considered as $\frac{\Delta t}{\Delta x^2}$ and for this problem $\Delta x = \Delta y$.
The pseudocode for the computation is given by

Algorithm 1 : Explicit solver for 2D heat diffusion equation –
Input : γ, Endtime, mesh-size, $T_{i,j}^{n}$ Run : for k=1: Δt:Endtime for i=1:Δx:X_length for j =1: :Δy:Y_length $T_{i,j}^{n+1} = \gamma\left(T_{i+1,j}^{n} + T_{i-1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n}\right) + (1-4\gamma)\, T_{i,j}^{n}$ end end end

2.1.2 <u>Implicit Scheme</u>

For the implicit scheme backward time and central space discretization is utilized considering $T_{i,j}^{n+1}$ as expansion point. On applying the discretization we get

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

Rearranging gives us

$$-\gamma T_{i,j-1}^{n+1} - \gamma T_{i-1,j}^{n+1} + (1+4\gamma) T_{i,j}^{n+1} - \gamma T_{i+1,j}^{n+1} - \gamma T_{i,j+1}^{n+1} = T_{i,j}^{n}$$

We can represent this equation in matrix form considering the mesh size of given problem. Then the equation becomes **Ax=B**.

Where $\mathbf{A} = \begin{bmatrix} 1+4\gamma & -\gamma & 0 & -\gamma & \cdots & 0 \\ -\gamma & 1+4\gamma & -\gamma & 0 & \dots & 0 \\ 0 & -\gamma & 1+4\gamma & \ddots & \cdots & \vdots \\ -\gamma & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & -\gamma \\ 0 & 0 & \dots & 0 & -\gamma & 1+4\gamma \end{bmatrix}$

x will be the vector of which includes the solution for upper time step i.e n+1 we intend to compute and **B** will be the vector of given initial and boundary values for time level n.

To get **x** from this equation we do need to invert **A** which is difficult. We will consider Gauss-Seidel method to break **A** into strict upper triangular matrix and Lower triangular matrix as $\boldsymbol{A} = \boldsymbol{L}_* + \boldsymbol{U}$.

Then the system of linear equation van be written as $\boldsymbol{L}_*\boldsymbol{x} = \boldsymbol{B} - \boldsymbol{U}\boldsymbol{x}$

Now using Gauss-Seidel method we can solve the left hand side of this expression for **x**, using previous value for **x** on the right hand side. Analytically this may be written as:

$\boldsymbol{x}^{\boldsymbol{k}+\boldsymbol{1}} = \boldsymbol{L}_*^{-\boldsymbol{1}}(\boldsymbol{B} - \boldsymbol{U}\boldsymbol{x}^{\boldsymbol{k}})$

However by taking advantage of the triangular form of $\boldsymbol{L}_*$, the elements of $\boldsymbol{x}^{\boldsymbol{k}+\boldsymbol{1}}$ can be computed sequentially using forward substitution:

$$x_i^{k+1} = \frac{1}{A_{ii}}\left(B_i - \sum_{j=1}^{i-1} A_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} A_{ij}x^k\right)$$

Where i=1,2,3,….n.

The convergence tolerance for this iterative substitution method is taken as $1.0e^{-4}$ in the sense of an L_1 norm which is given by

$$\sum_{i=1}^{N} \frac{\left|x_i^{k+1} - x_i^k\right|}{N} < 1.0e^{-4}$$

A pseudocode for the computation is given by

```
Algorithm 2 : Explicit solver for 2D heat diffusion equation –
Input : γ, Endtime, mesh-size, T^n_{i,j}
Create the matrices A,x,B
        for k = 1:Δt:Endtime
        norm = inf ; tolerance = 10^-4 ;
        while norm>tolerance
         x_old = x ;
              for i=1: n
              σ = 0;
              for j=1: i-1
              σ =  σ + A(i, j) * x(j);
              end
              for j=i+1:n
              σ =  σ + A(i, j) * x_old(j);
              end
          x(i) = (1/A(i, i))  * (B(i) − σ);
        end
        norm  = 1/N (sum( abs(x_old(i, j) − x(i, j))))
      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 $T_{i,j}^{n}$ = computed solution

$\overline{T_{i,j}^{n}}$ = exact solution of PDE

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

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

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} + \frac{\varepsilon_{i,j}^{n+1} - \varepsilon_{i,j}^{n}}{\Delta t} = \frac{T_{i+1,j}^{n} - 2T_{i,j}^{n} + T_{i-1,j}^{n}}{\Delta x^2} + \frac{T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}}{\Delta y^2} + \frac{\varepsilon_{i+1,j}^{n} - 2\varepsilon_{i,j}^{n} + \varepsilon_{i-1,j}^{n}}{\Delta x^2} + \frac{\varepsilon_{i,j+1}^{n} - 2\varepsilon_{i,j}^{n} + \varepsilon_{i,j-1}^{n}}{\Delta y^2} \qquad (2.1)$$

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

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

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

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

which gives the following two conditions for stability

$$\gamma \geq 0 \text{ and } \gamma \leq \frac{1}{4}$$

The explicit method thus can be said conditionally stable.

2.2.2 Implicit Method

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.

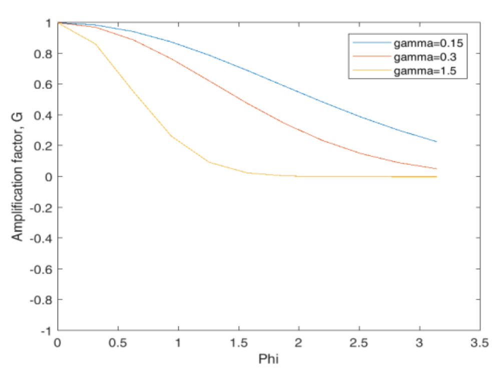

Figure 3.1 : Amplification factor for exact solution of the PDE at different γ

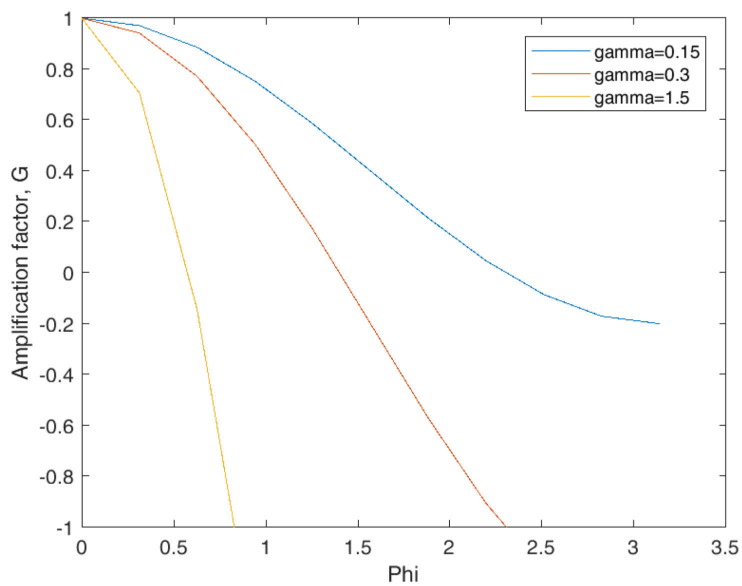

Figure 3.2: Amplification factor of the explicit method for different γ

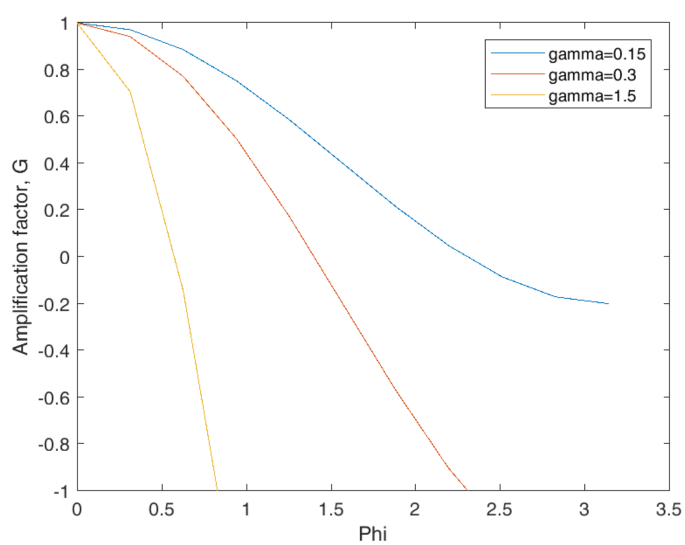

Figure 3.3: Amplification factor of the implicit method for different γ

<u>**Iso-contour**</u>

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.

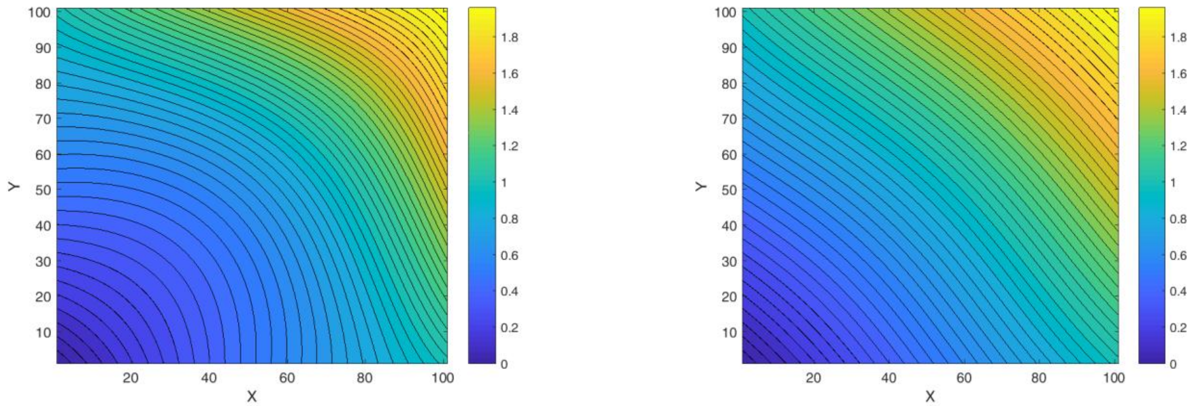

Figure 3.4 : Iso-contour plot of Implicit(left) and Explicit(right) solution from 100x100 mesh at γ=0.15 and at t=0.1s

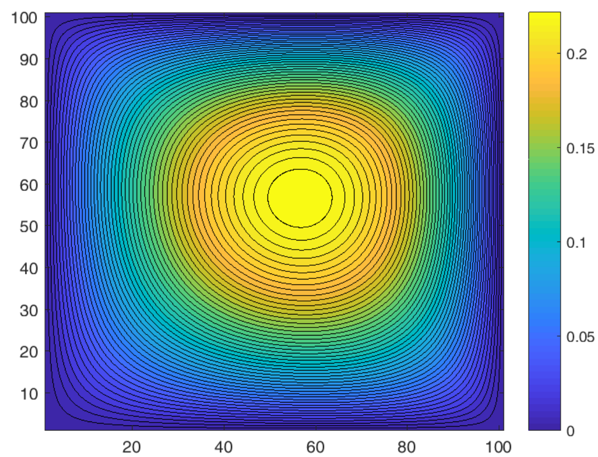

Figure : 3.5: Iso-contour plot of error between Implicit and Explicit solution from 100x100 mesh at γ=0.15 and at t=0.1s (Considering the Implicit one as exact solution)

<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:

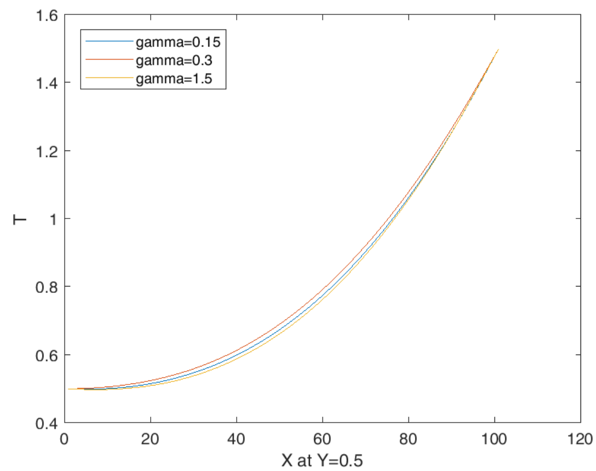

Figure 3.6 : T-x at y=0.5 for implicit method with resolution 100x100 at t=0.1s for different values of γ

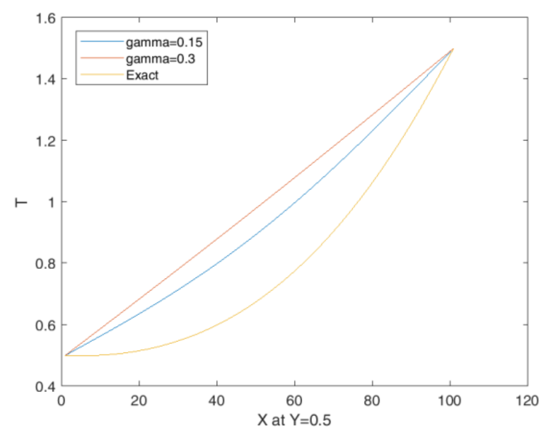

Figure 3.7 : T-x at y=0.5 for explicit method with resolution 100x100 at t=0.1s for different values of γ

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:

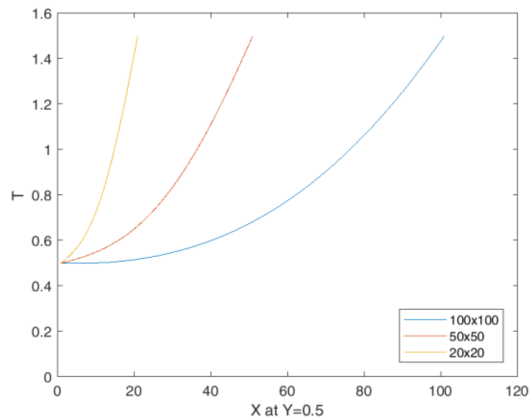

Figure 3.8: T-x at y=0.5 for implicit method and for γ=0.15 at t=0.1s for various mesh size

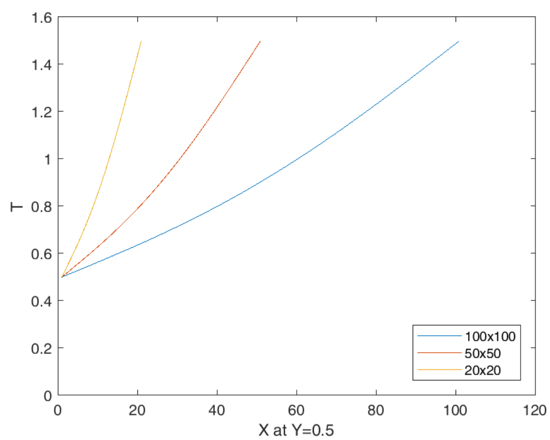

Figure 3.9: T-x at y=0.5 for explicit method and for γ=0.15 at t=0.1s for various mesh size

Comment

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

Impact of numerical method:

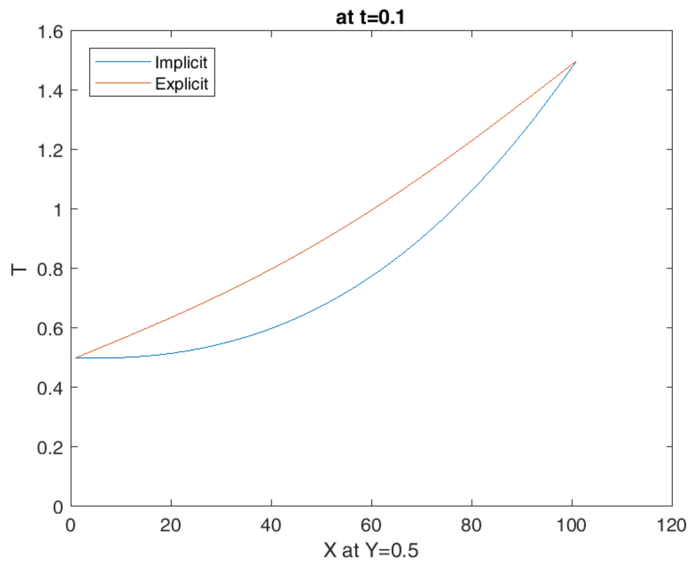

Figure 3.10: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.1s.

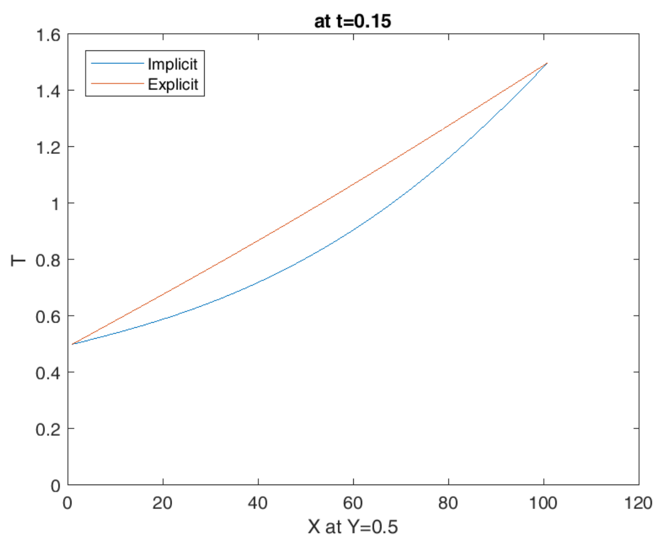

Figure 3.11: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.15s.

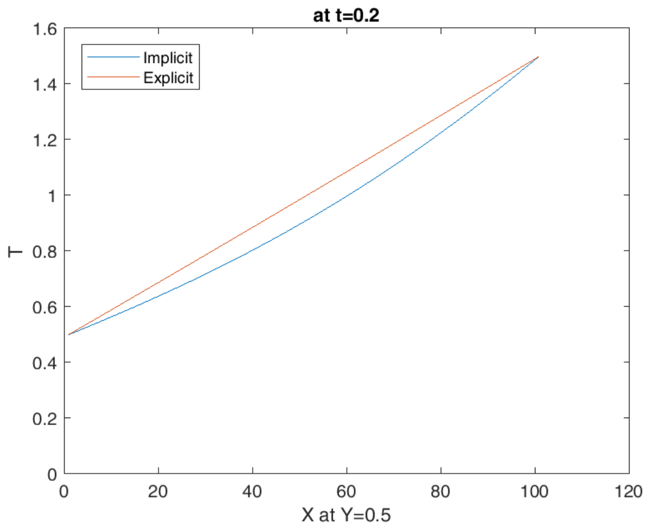

Figure 3.12: T-x at y=0.5 for implicit and explicit methods with γ=0.15 and 100x100 resolution at t=0.2s.

<u>Comment</u>

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. <u>Comment on efficacy</u>

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.