Computational Fluid Dynamics (AE 320)

Assignment 4

Submitted By

Krishna Wadhwani

160010031

Contents

1 Introduction **2**

2 Code Description **2**

3 Code Performance **2**

4 Formulation of Iteration Scheme **2**

5 Plots and results **4**

5.1 Diffusion Equation . . . 4

5.1.1 Surface Plots . . . 4

5.1.2 Conclusions . . . 6

5.2 Advection Equation . . . 6

5.2.1 Surface Plots . . . 6

5.2.2 Diagonal Plots . . . 8

5.2.3 Conclusions . . . 8

5.3 Burger's Equation . . . 9

5.3.1 $\nu = 0.0$ . . . 9

5.3.2 $\nu = 0.02$ . . . 10

5.3.3 $\nu = 0.04$ . . . 12

5.3.4 Diagonal Plots . . . 14

5.3.5 Conclusions . . . 17

1 Introduction

The aim of this assignment was to understand Finite Volume Methods and formulate a numerical scheme using Finite volume methods to solve diffusion, advection and burger's equation.

2 Code Description

The assignment contains the following three code files:

1. diffusion.py: Solves 2D diffusion equation with zero gradients boundary condition. The code generates an interactive surface plot and saves the surface and diagonal plots in a folder "diffusion_plots" for $t = 0, 10, 20, 30, 40s$
2. advection.py: Solves 2D advection equation with zero gradients boundary condition. The code generates an interactive surface plot and saves the surface and diagonal plots in a folder "advection_plots" for $t = 0, 5, 10, 15s$ using FTBS and FTCS2 schemes. (Lax-Friedrichs is unstable).
3. burger.py: Solves 2D inviscid burger's equation with zero gradients boundary condition. The code generates an interactive surface plot and saves the surface and diagonal plots in a folder "burger_plots" for $t = 0, 5, 15, 25s$ using FTBS, FTCS and Lax-Friedrichs scheme.

3 Code Performance

The initial code was written using 3 nested loops, which took a lot of time to run. The code was then vectorised using array slicing and simultaneous update of of numpy arrays.

The non-vectorized code takes 9 min 32 seconds to generate the final surface plot of advection equation for time = 10 seconds, using FTBS scheme whereas the vectoriezed code does this almost instantaneously.

Therefore all the three codes have been vectorized for optimized performance.

4 Formulation of Iteration Scheme

Integral form of conservation law is given as:

$$\frac{\partial}{\partial t} \iiint_{\underline{V}} \phi d\underline{V} + \oiint_{S} \mathcal{T} dS = 0$$

For the limits of $d\underline{V}$ and taking averaged quantities, we get:

$$\frac{\partial \bar{\phi}}{\partial t} + \nabla \mathcal{T} = 0$$

On discretizing:

$$\frac{\partial \bar{\phi}}{\partial t} = -\frac{\mathcal{T}_{x,i+1/2,j} - \mathcal{T}_{x,i-1/2,j}}{\Delta x} - \frac{\mathcal{T}_{y,i,j+1/2} - \mathcal{T}_{y,i,j-1/2}}{\Delta y}$$

$$\implies \bar{\phi}_{ij}^{n+1} = \bar{\phi}_{ij}^{n} - \frac{\Delta t}{\Delta x}(\mathcal{T}_{x,i+1/2,j} - \mathcal{T}_{x,i-1/2,j}) - \frac{\Delta t}{\Delta y}(\mathcal{T}_{y,i,j+1/2} - \mathcal{T}_{y,i,j-1/2})$$

In our problem, $\bar{\phi} = \bar{u}$ and $\mathcal{T}$ for different equations is as follows:

For Diffusion Equation, $\mathcal{T}_x = -\nu\frac{\partial u}{\partial x}$ and $\mathcal{T}_y = -\nu\frac{\partial u}{\partial y}$:

$$\mathcal{T}_{x,i+1/2,j} = -\nu\frac{(\bar{u}^n_{i+1,j} - \bar{u}^n_{i,j})}{\Delta x}$$
$$\mathcal{T}_{x,i-1/2,j} = -\nu\frac{(\bar{u}^n_{i,j} - \bar{u}^n_{i-1,j})}{\Delta x}$$

$$\mathcal{T}_{y,i,j+1/2} = -\nu\frac{(\bar{u}^n_{i,j+1} - \bar{u}^n_{i,j})}{\Delta y}$$
$$\mathcal{T}_{y,,j-1/2} = -\nu\frac{(\bar{u}^n_{i,j} - \bar{u}^n_{i,j-1})}{\Delta y}$$

For Advection Equation,

$$\mathcal{T}_{x,i+1/2,j} = \frac{(\bar{u}^n_{i+1,j} + \bar{u}^n_{i,j})}{2} - \frac{\lambda}{2}(\bar{u}^n_{i+1,j} - \bar{u}^n_{i,j})$$
$$\mathcal{T}_{x,i-1/2,j} = \frac{(\bar{u}^n_{i,j} + \bar{u}^n_{i-1,j})}{2} - \frac{\lambda}{2}(\bar{u}^n_{i,j} - \bar{u}^n_{i-1,j})$$

$$\mathcal{T}_{y,i,j+1/2} = \frac{(\bar{u}^n_{i,j+1} + \bar{u}^n_{i,j})}{2} - \frac{\lambda}{2}(\bar{u}^n_{i,j+1} - \bar{u}^n_{i,j})$$
$$\mathcal{T}_{y,i,j-1/2} = \frac{(\bar{u}^n_{i,j+1} + \bar{u}^n_{i,j})}{2} - \frac{\lambda}{2}(\bar{u}^n_{i,j} - \bar{u}^n_{i,j-1})$$

For Burger equation:

$$\mathcal{T}_{x,i+1/2,j} = \frac{(\bar{u}^2_{i+1,j} + \bar{u}^2_{i,j})^n}{4} - \frac{\lambda}{2}(\bar{u}^n_{i+1,j} - \bar{u}^n_{i,j}) - \nu\frac{(\bar{u}^2_{i+1,j} - \bar{u}^2_{i,j})}{\Delta x}$$

$$\mathcal{T}_{x,i-1/2,j} = \frac{(\bar{u}^2_{i,j} + \bar{u}^2_{i-1,j})^n}{4} - \frac{\lambda}{2}(\bar{u}^n_{i,j} - \bar{u}^n_{i-1,j}) - \nu\frac{(\bar{u}^n_{i,j} - \bar{u}^n_{i-1,j})}{\Delta x}$$

$$\mathcal{T}_{y,i,j+1/2} = \frac{(\bar{u}^2_{i,j+1} + \bar{u}^2_{i,j})^n}{4} - \frac{\lambda}{2}(\bar{u}^n_{i,j+1} - \bar{u}^n_{i,j}) - \nu\frac{(\bar{u}^n_{i,j+1} - \bar{u}^n_{i,j})}{\Delta y}$$

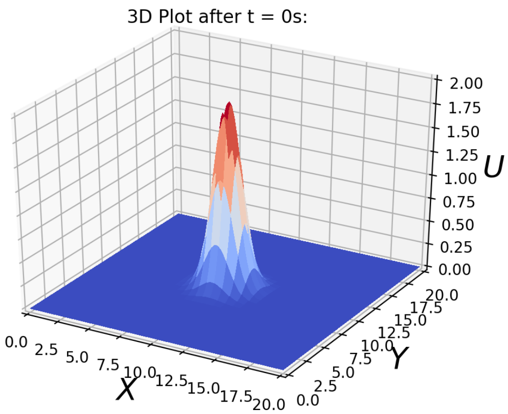
$$\mathcal{T}_{y,i,j-1/2} = \frac{(\bar{u}^2_{i,j+1} + \bar{u}^2_{i,j})^n}{4} - \frac{\lambda}{2}(\bar{u}^n_{i,j} - \bar{u}^n_{i,j-1}) - \nu\frac{(\bar{u}^n_{i,j} - \bar{u}^n_{i,j-1})}{\Delta y}$$

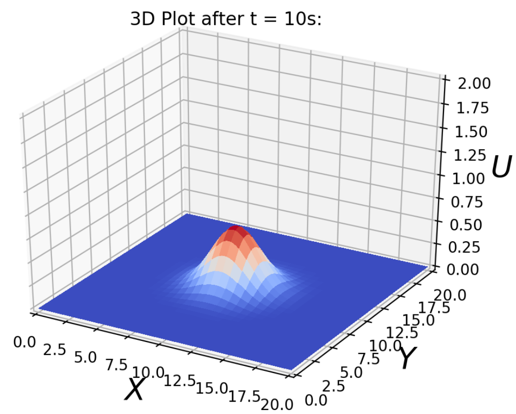
For FTBS: $\lambda = 1$
For FTCS2: $\lambda = a\sigma$
For Lax-Friedrich: $\lambda = dx/dt$

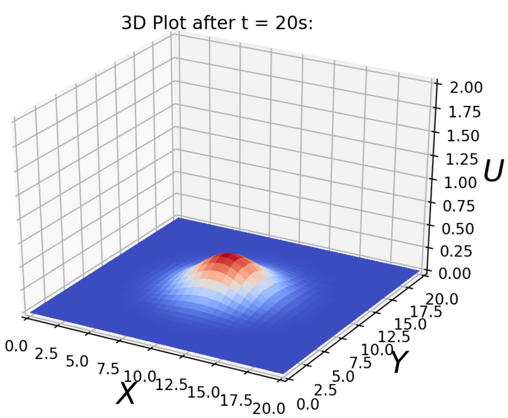
5 Plots and results

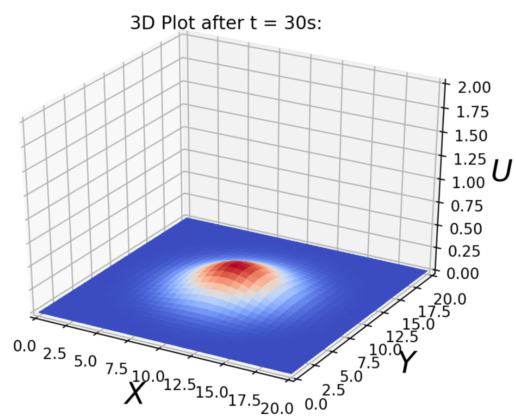
5.1 Diffusion Equation

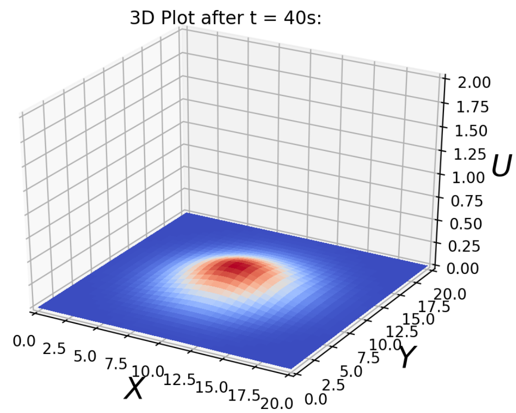
5.1.1 Surface Plots

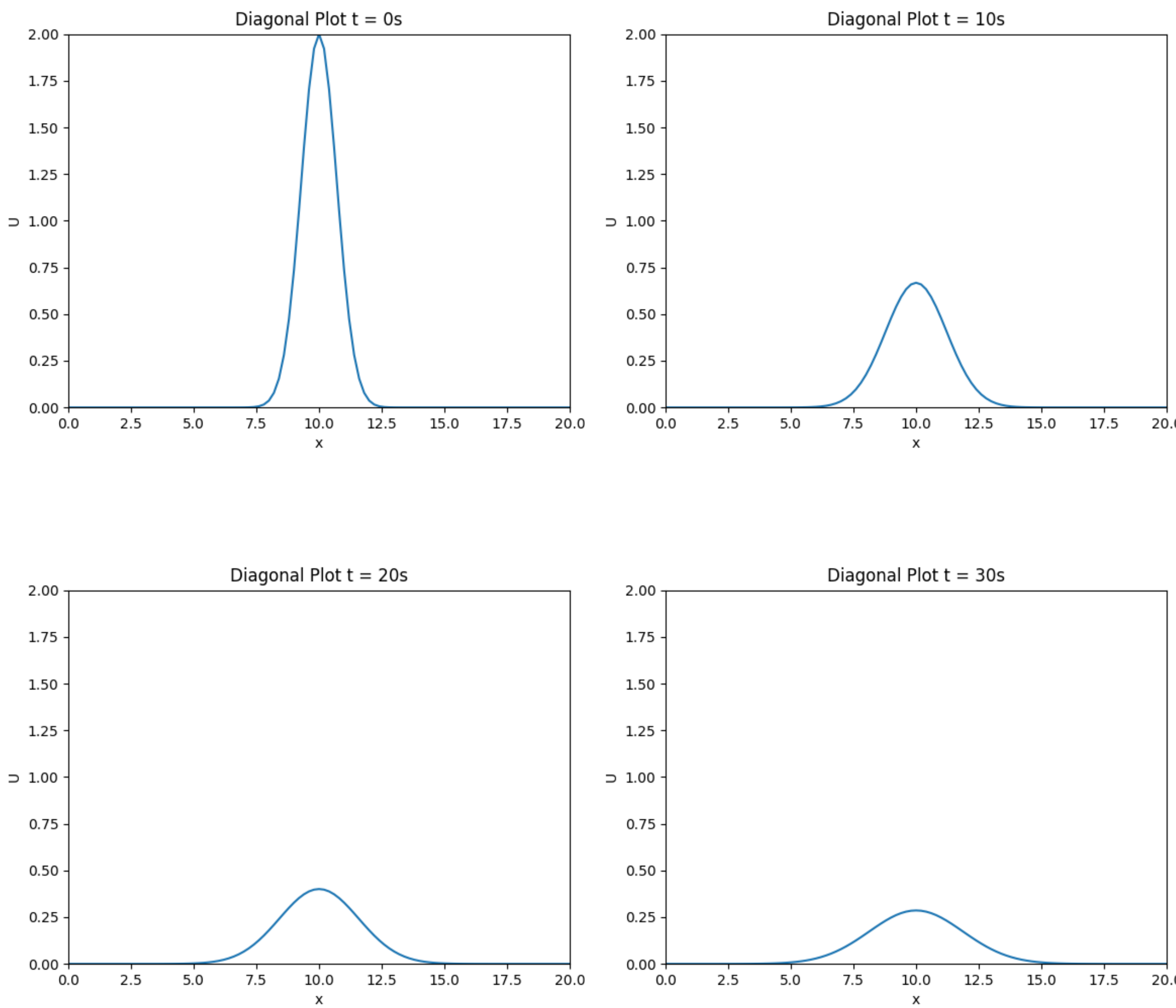


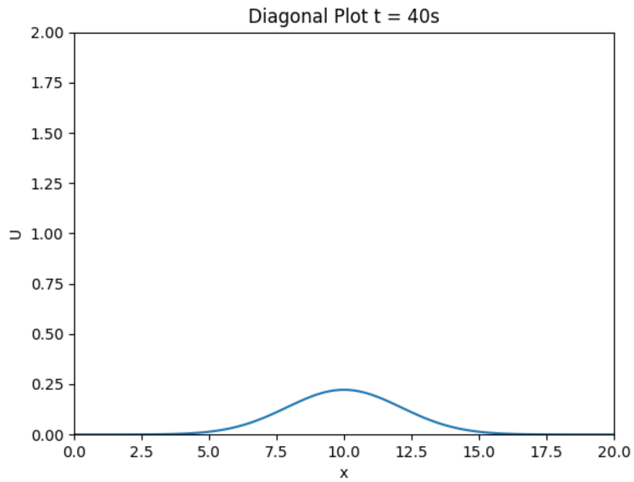







3D Plot after t = 40s:
U
X
Y
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Diagonal Plot t = 0s
Diagonal Plot t = 10s
Diagonal Plot t = 20s
Diagonal Plot t = 30s
U
x
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

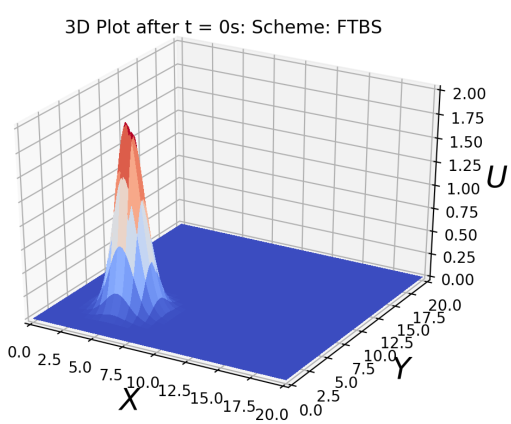


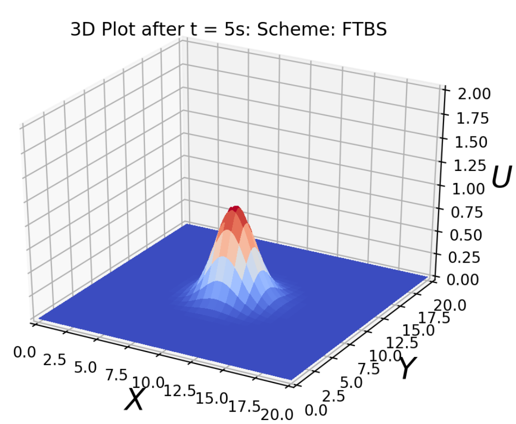
5.1.2 Conclusions

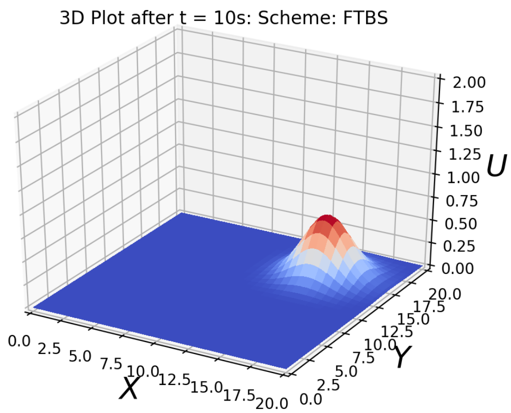
From the following figures, we can see that the peak of the initial waveform reduces and spread of the wave increases with time, or in other words, the wave diffuses outwards in x and y directions. Due to presence of negative second derivative term, the peak reduces and wave spreads out in both x and y directions which is manifested in the form of diffusing of wave. This is a case of diffusion without advection as we can see that the peak of the wave always remains in the middle of the grid.

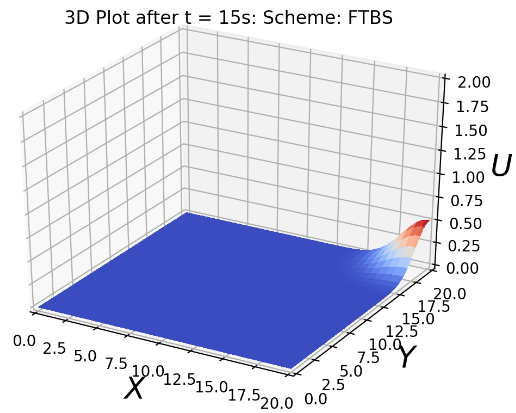
5.2 Advection Equation

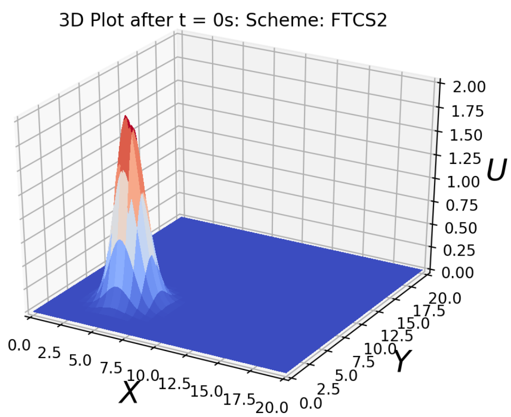
5.2.1 Surface Plots

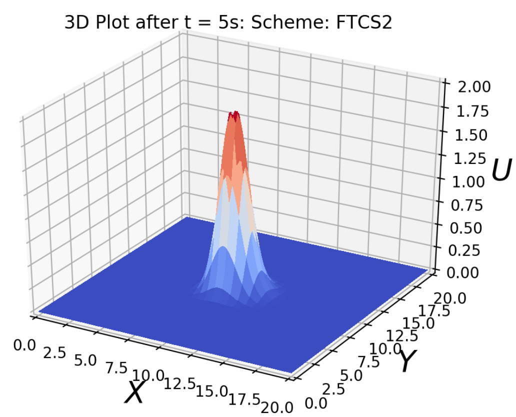


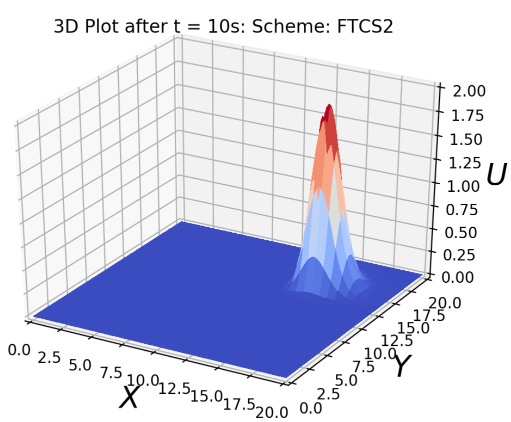


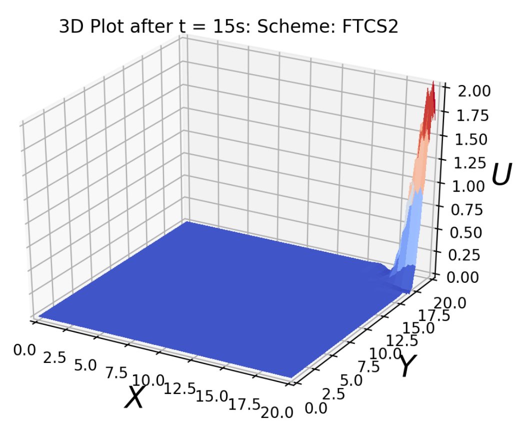

3D Plot after t = 10s: Scheme: FTBS
X
Y
U


3D Plot after t = 15s: Scheme: FTBS
X
Y
U

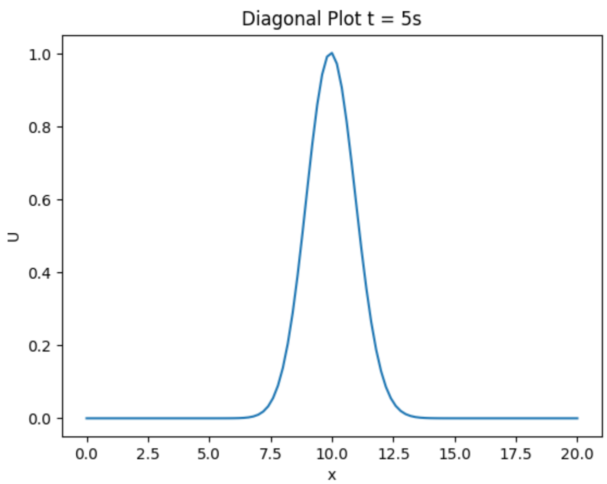

3D Plot after t = 0s: Scheme: FTCS2
X
Y
U


3D Plot after t = 5s: Scheme: FTCS2
X
Y
U

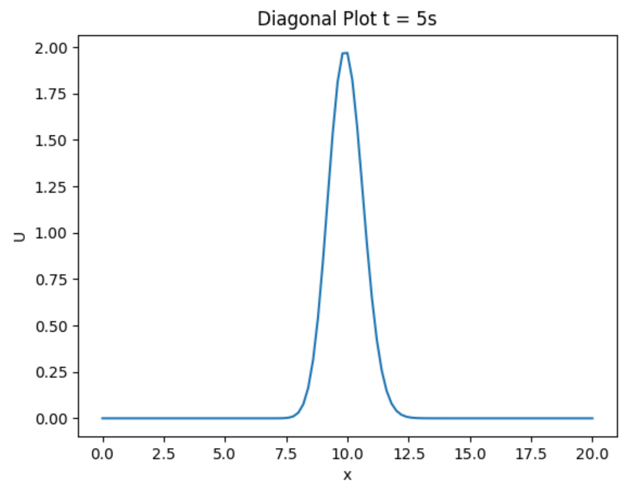

3D Plot after t = 10s: Scheme: FTCS2
X
Y
U


3D Plot after t = 15s: Scheme: FTCS2
X
Y
U

5.2.2 Diagonal Plots



(a) Scheme: FTBS

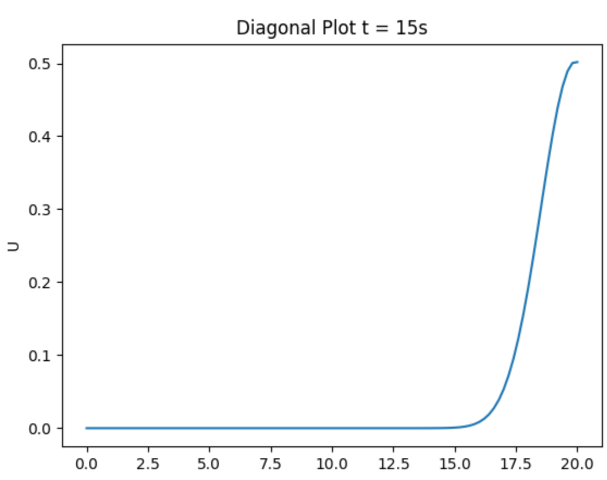


(b) Scheme: FTCS2

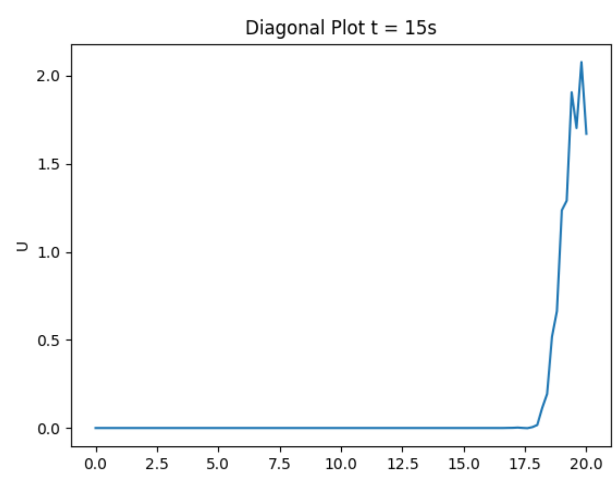
5.2.3 Conclusions

From the diagonal plots, we can see that the peak of FTBS solution has undergone diffusion while that of FTCS2 retains its shape better. This is because of dissipation in FTBS due to negative second derivative term in the modified equation for the FTBS scheme. In FTCS2, there is no dissipation so the wave retains its peak. We don't see dispersion as such for FTCS2 for t=5s in the diagonal plots. But in the surface plots, we can see that FTBS solution is smoother than FTCS2 solution. Moreover if we observe the diagonal plots at t = 25s, then we can see that the dispersion is starting to show its effects in case of FTCS2 in the form of irregularities in waveform as compared to the smooth FTBS solution.

Lax-Friedrich scheme is unstable for the given problem as we require $\lambda < 1$ for stability of Lax-Friedrich scheme as per Von-Neumann stability analysis but λ is close to 200 in the given problem statement as dx/dt is very high. Due to massive blowup of u because of this instability, I got an error in generation of 3D plots for this scheme.



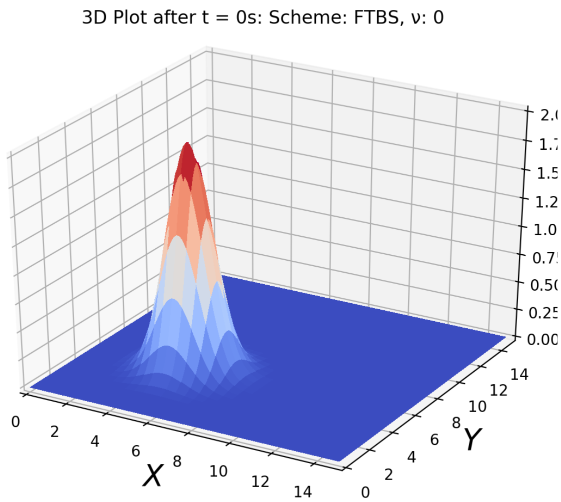
(a) Scheme: FTBS

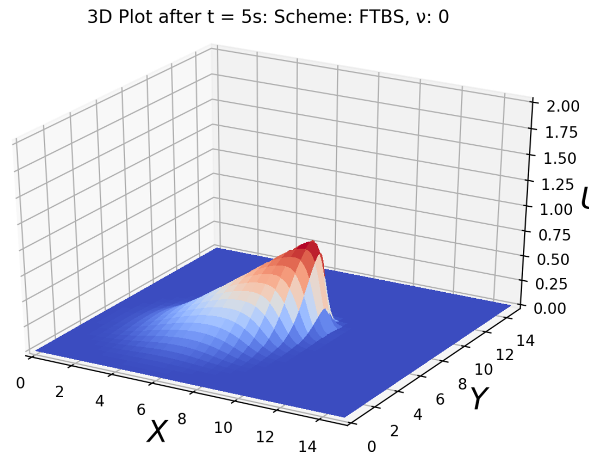


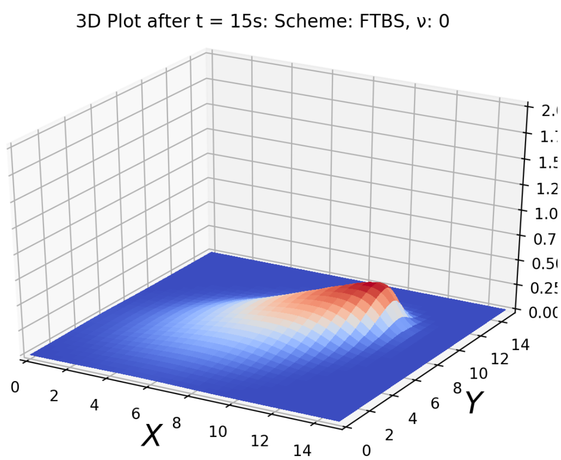
(b) Scheme: FTCS2

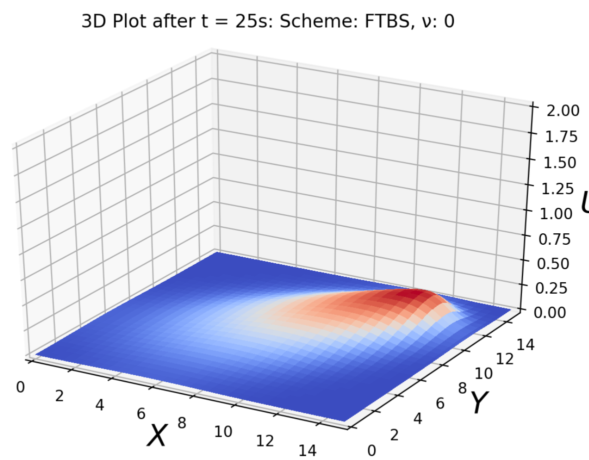
5.3 Burger's Equation

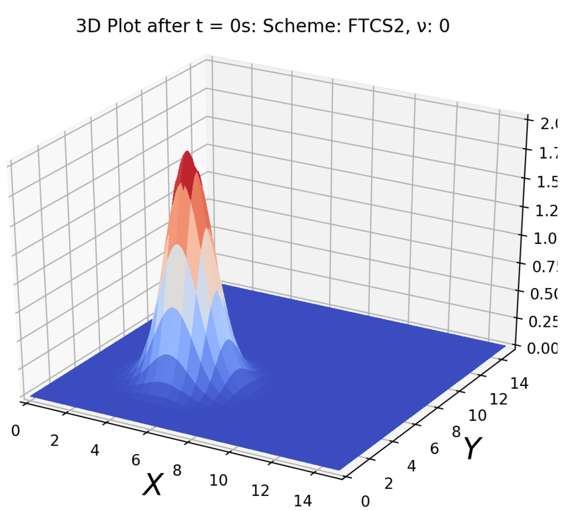
5.3.1 $\nu = 0.0$

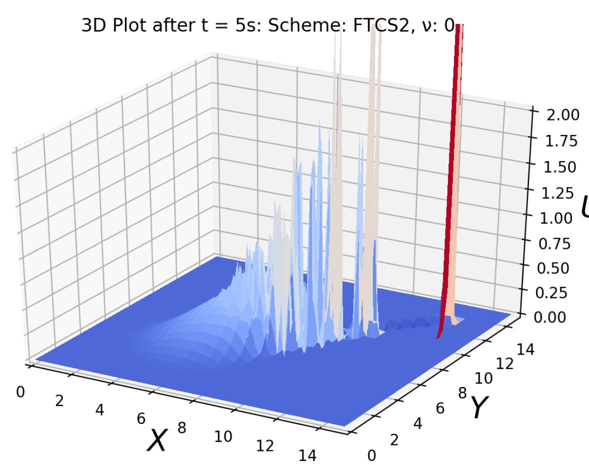




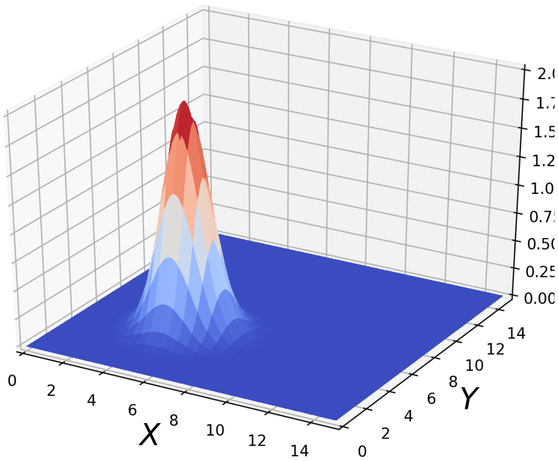




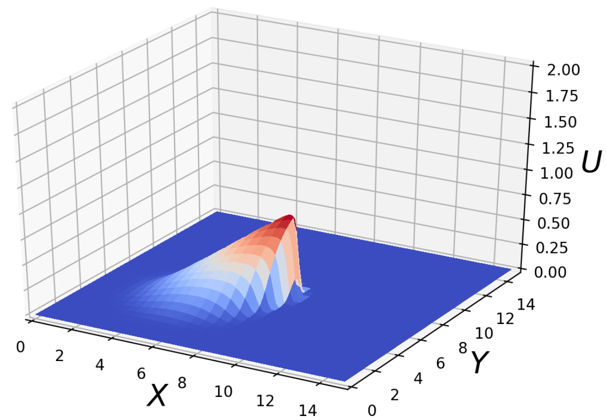




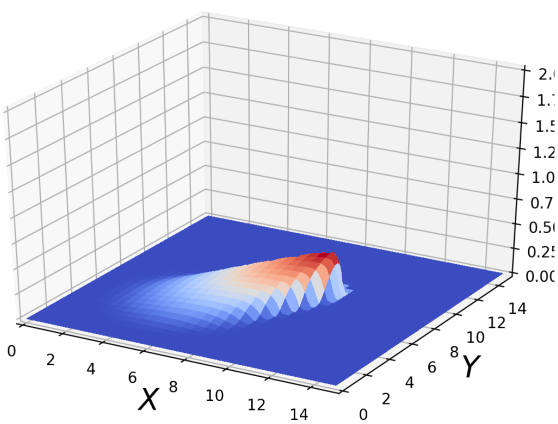
3D Plot after t = 0s: Scheme: Lax-Friedrich, ν: 0



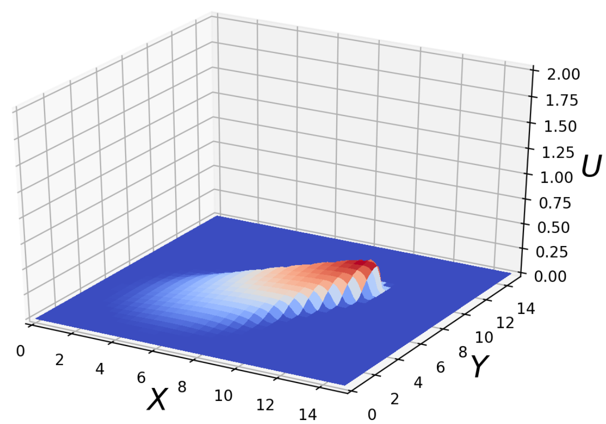
3D Plot after t = 5s: Scheme: Lax-Friedrich, ν: 0



3D Plot after t = 15s: Scheme: Lax-Friedrich, ν: 0

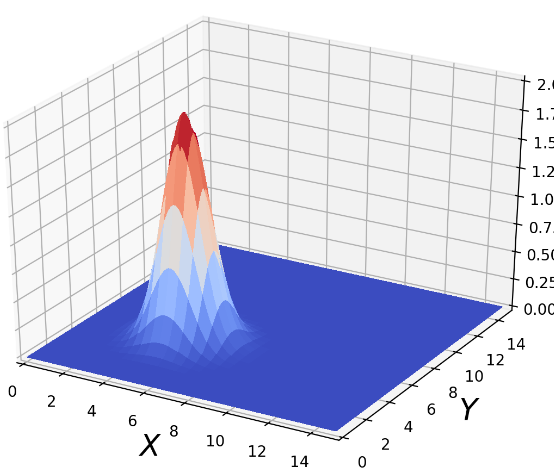


3D Plot after t = 25s: Scheme: Lax-Friedrich, ν: 0

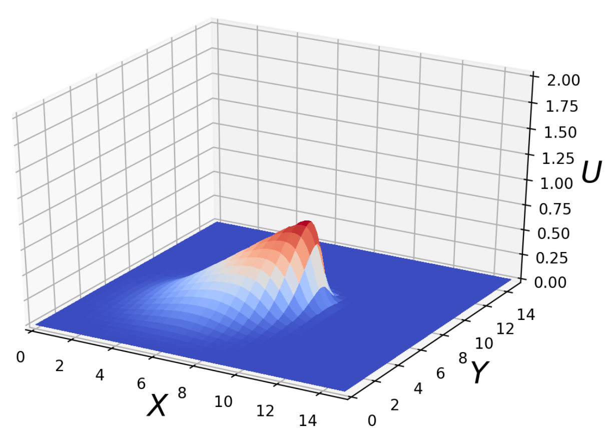


5.3.2 $\nu = 0.02$

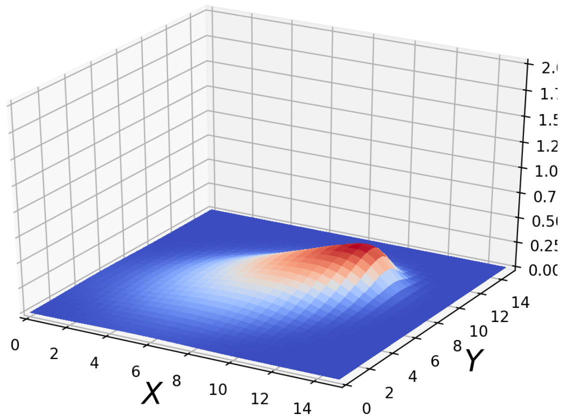
3D Plot after t = 0s: Scheme: FTBS, ν: 0.02



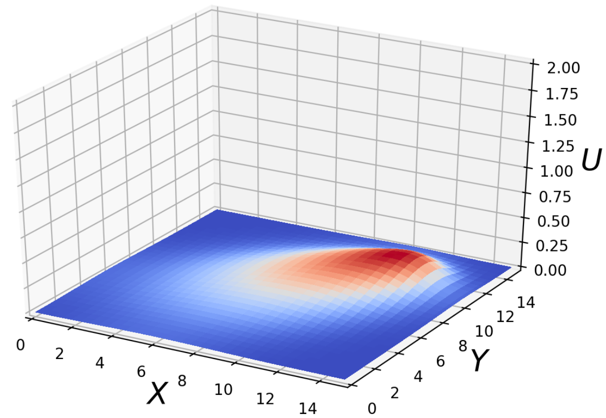
3D Plot after t = 5s: Scheme: FTBS, ν: 0.02



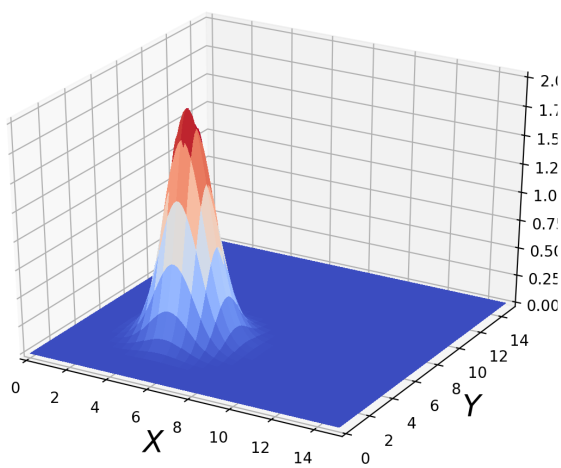
3D Plot after t = 15s: Scheme: FTBS, ν: 0.02



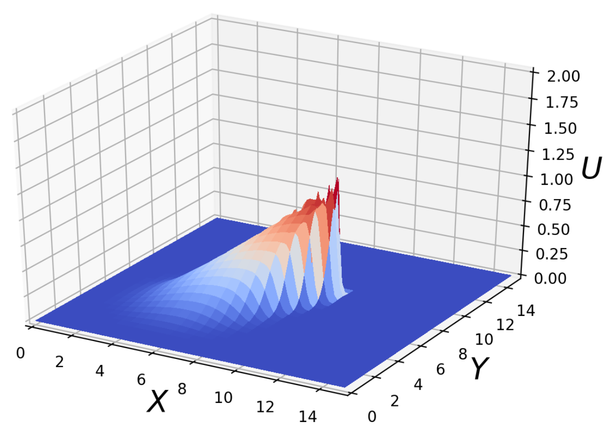
3D Plot after t = 25s: Scheme: FTBS, ν: 0.02



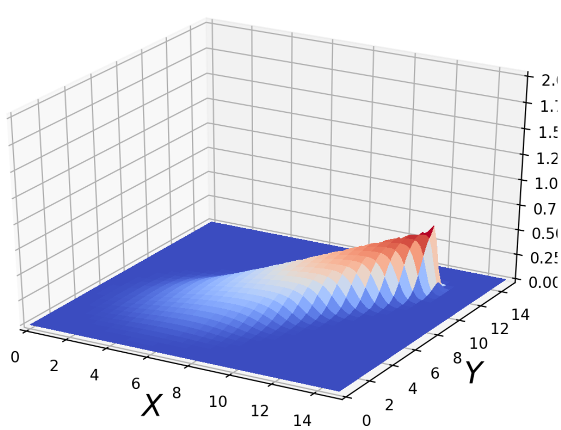
3D Plot after t = 0s: Scheme: FTCS2, ν: 0.02



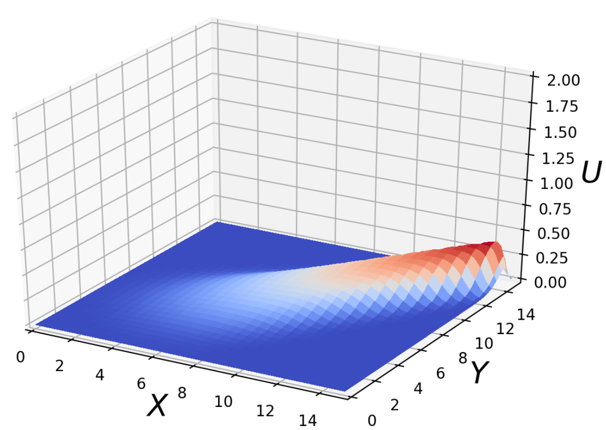
3D Plot after t = 5s: Scheme: FTCS2, ν: 0.02

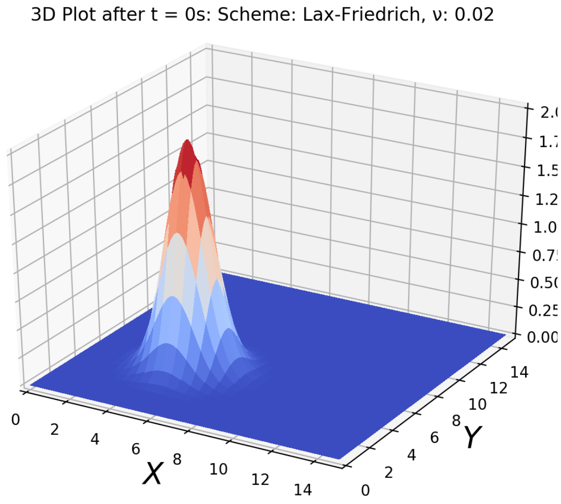


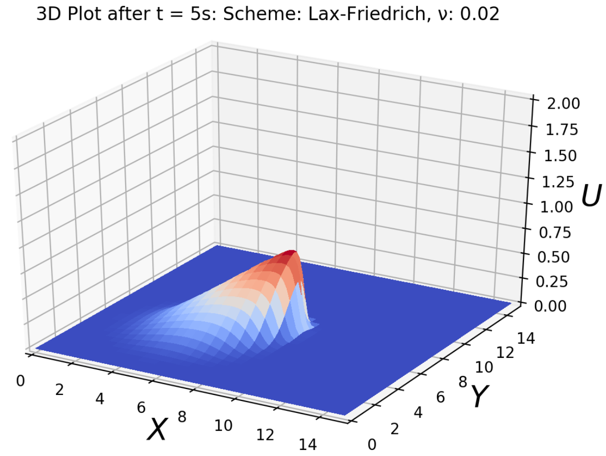
3D Plot after t = 15s: Scheme: FTCS2, ν: 0.02

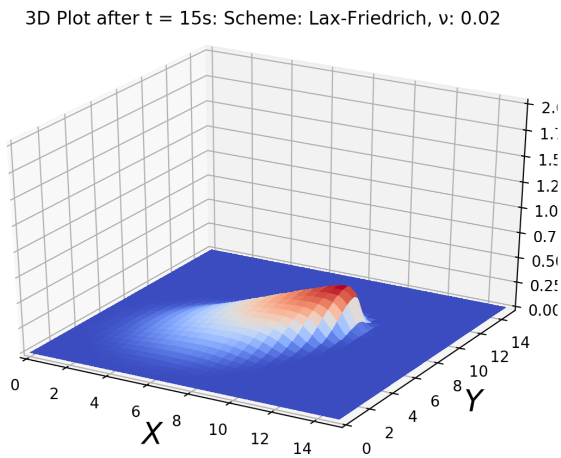


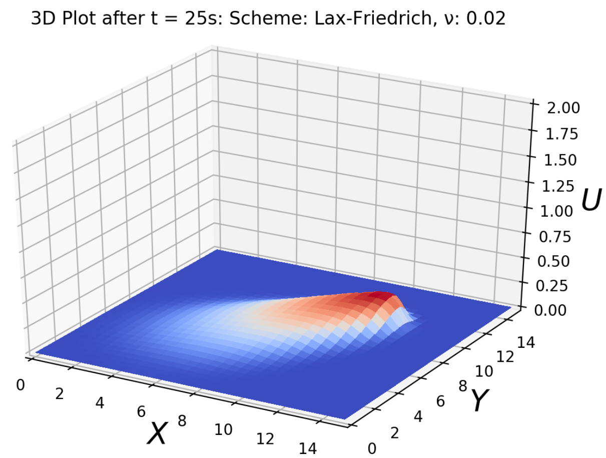
3D Plot after t = 25s: Scheme: FTCS2, ν: 0.02



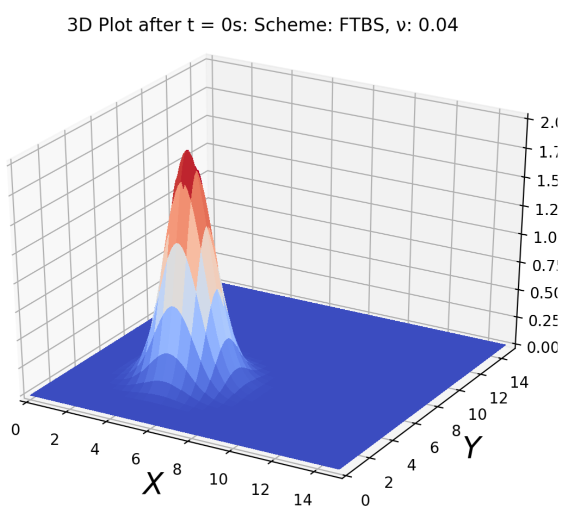

3D Plot after t = 0s: Scheme: Lax-Friedrich, ν: 0.02
X
Y

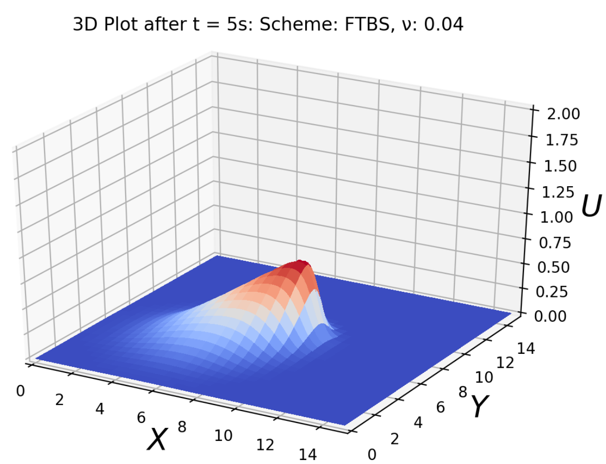

3D Plot after t = 5s: Scheme: Lax-Friedrich, ν: 0.02
X
Y
U

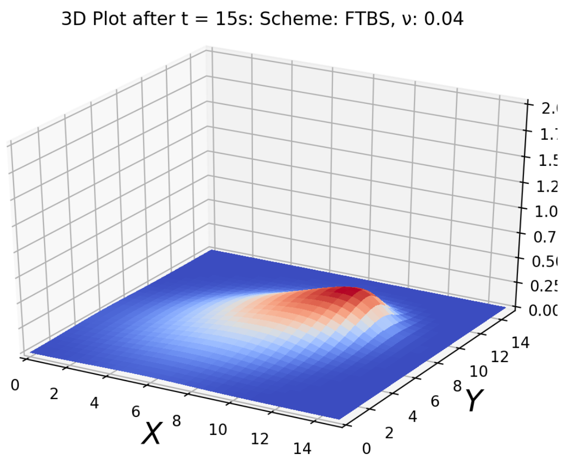

3D Plot after t = 15s: Scheme: Lax-Friedrich, ν: 0.02
X
Y

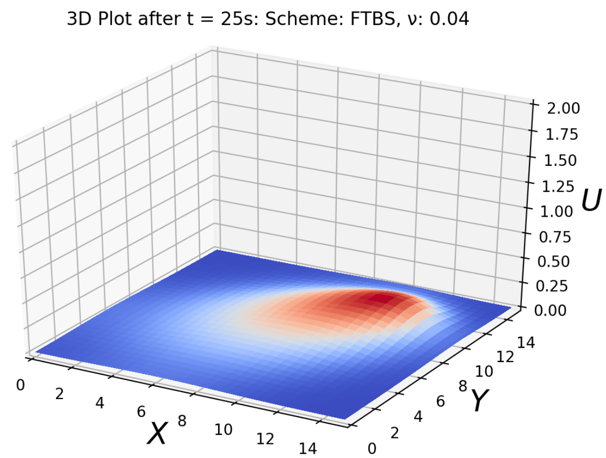

3D Plot after t = 25s: Scheme: Lax-Friedrich, ν: 0.02
X
Y
U

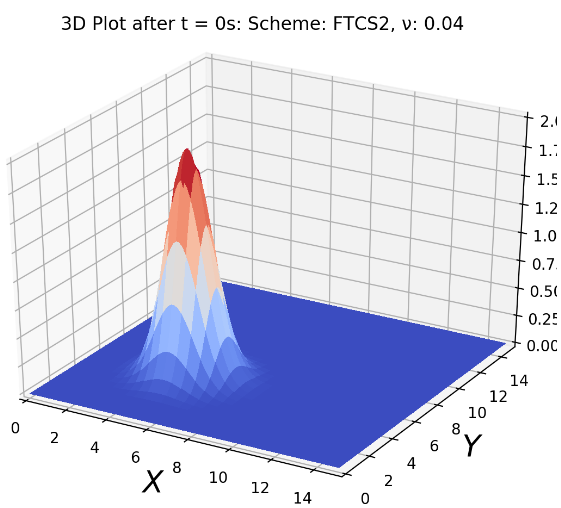
5.3.3 $\nu = 0.04$

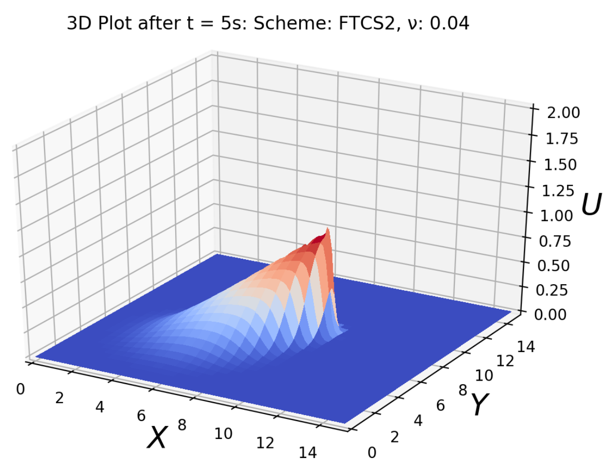

3D Plot after t = 0s: Scheme: FTBS, ν: 0.04
X
Y

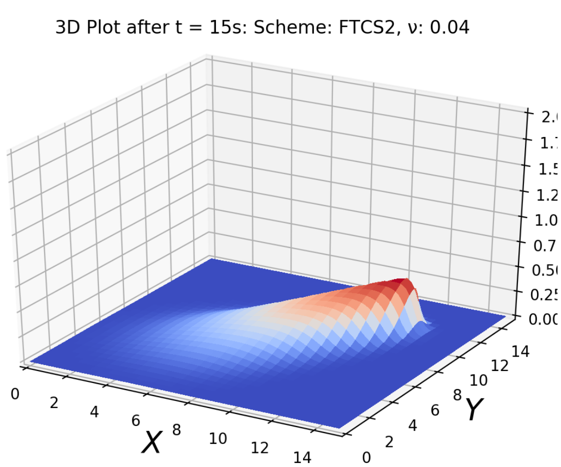

3D Plot after t = 5s: Scheme: FTBS, ν: 0.04
X
Y
U

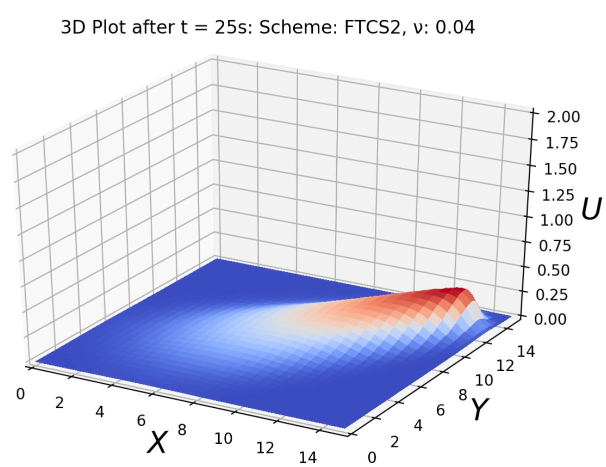

3D Plot after t = 15s: Scheme: FTBS, ν: 0.04
X
Y

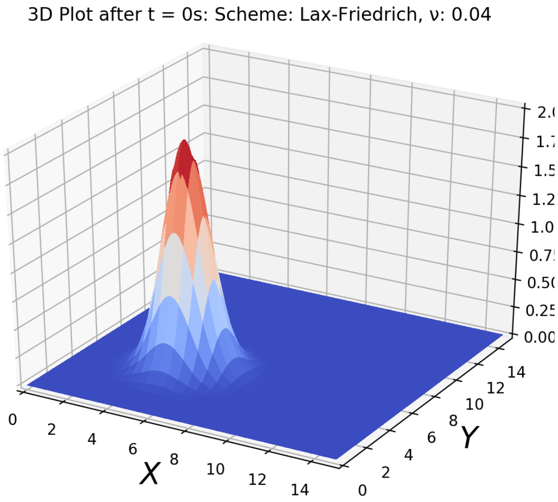

3D Plot after t = 25s: Scheme: FTBS, ν: 0.04
X
Y
U

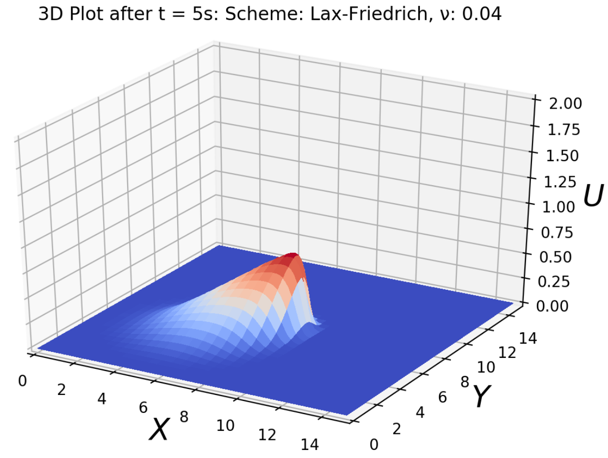

3D Plot after t = 0s: Scheme: FTCS2, ν: 0.04
X
Y

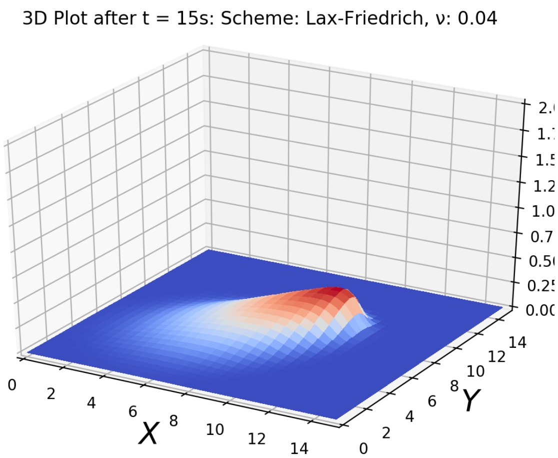

3D Plot after t = 5s: Scheme: FTCS2, ν: 0.04
X
Y
U

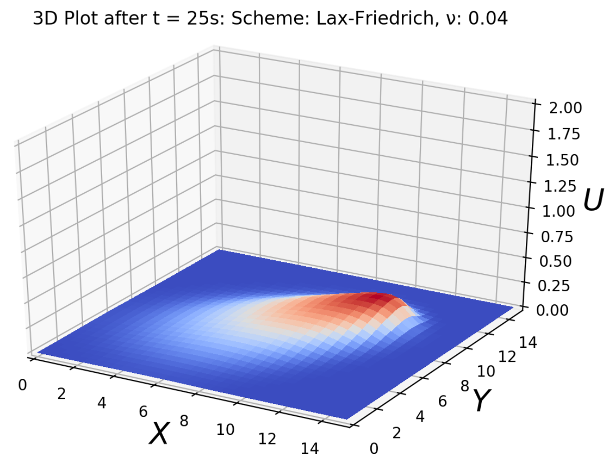

3D Plot after t = 15s: Scheme: FTCS2, ν: 0.04
X
Y


3D Plot after t = 25s: Scheme: FTCS2, ν: 0.04
X
Y
U

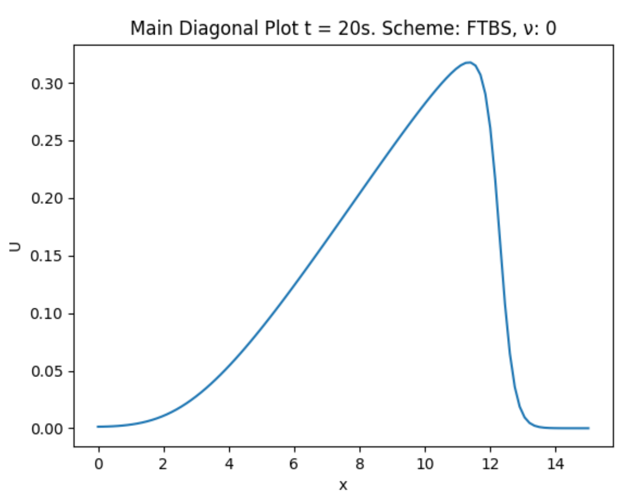


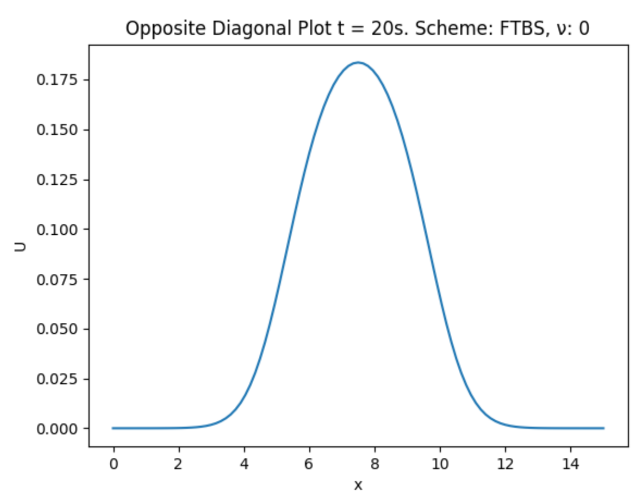


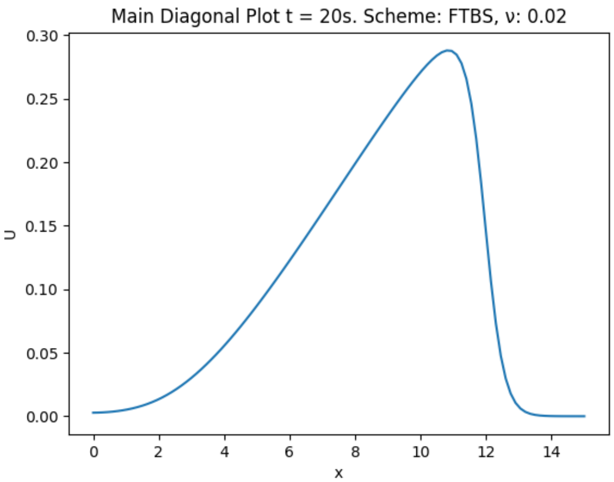


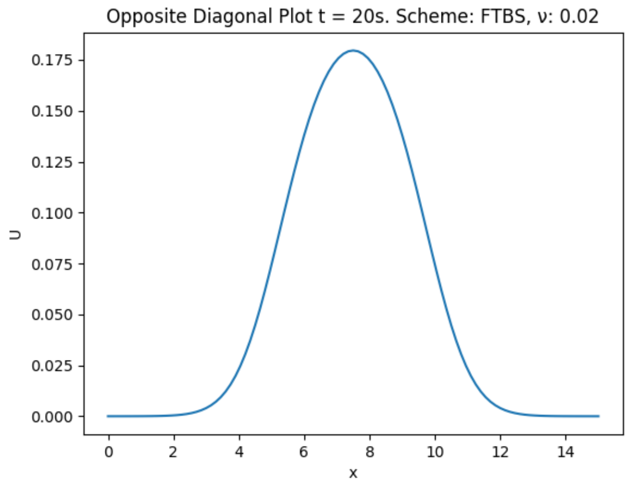


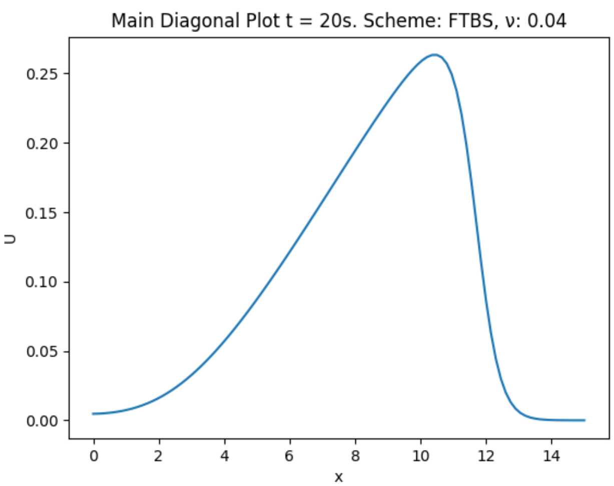
5.3.4 Diagonal Plots

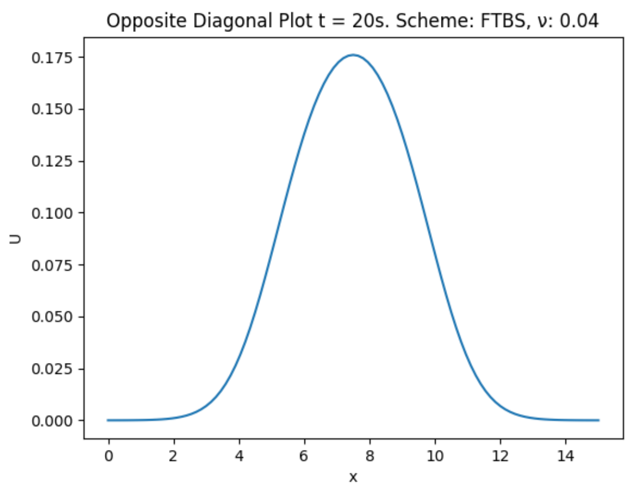


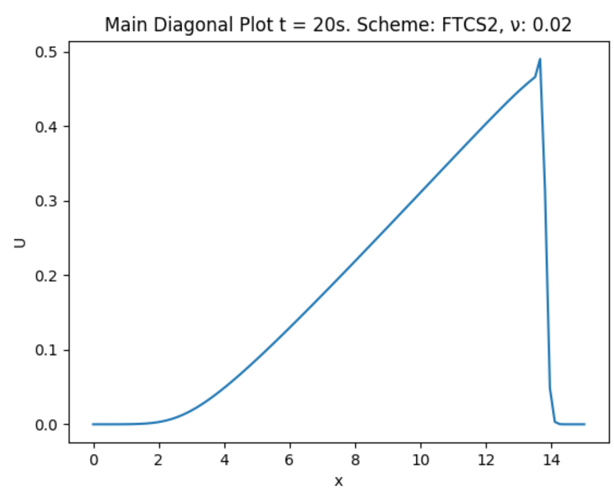


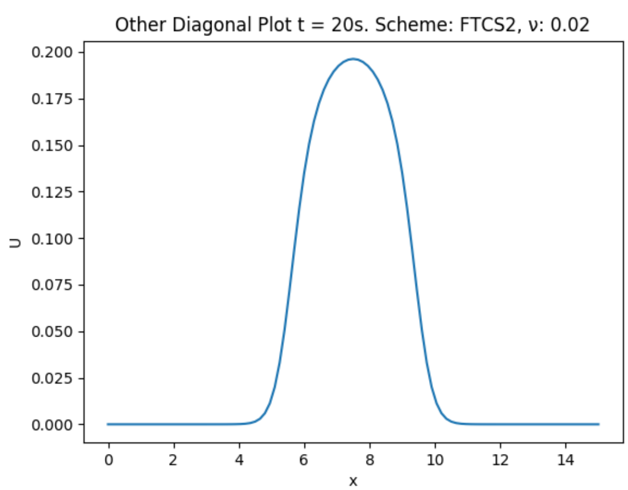

Main Diagonal Plot t = 20s. Scheme: FTBS, ν: 0.02
U
x

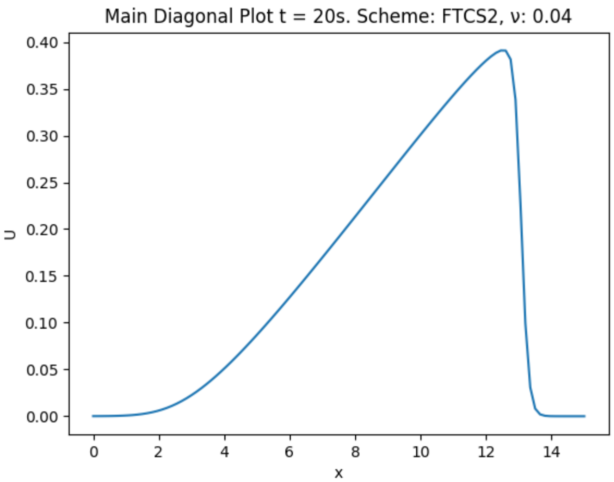

Opposite Diagonal Plot t = 20s. Scheme: FTBS, ν: 0.02
U
x

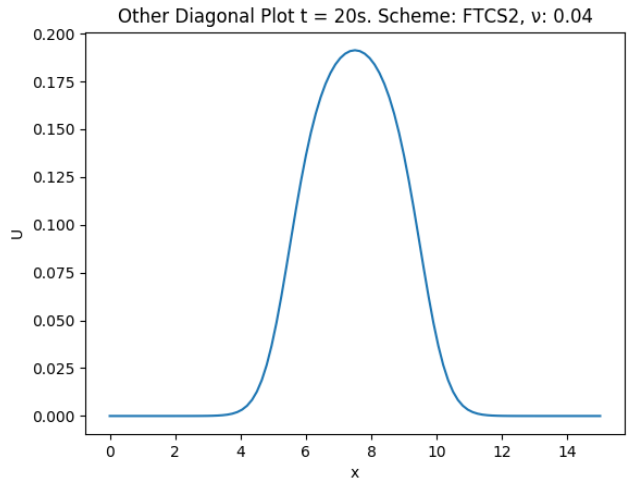

Main Diagonal Plot t = 20s. Scheme: FTBS, ν: 0.04
U
x

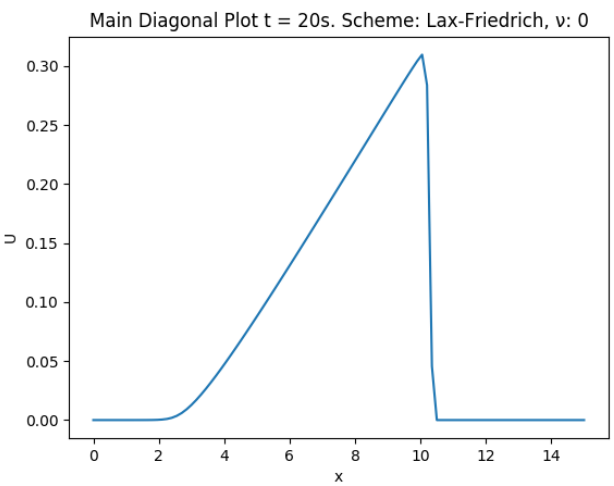

Opposite Diagonal Plot t = 20s. Scheme: FTBS, ν: 0.04
U
x

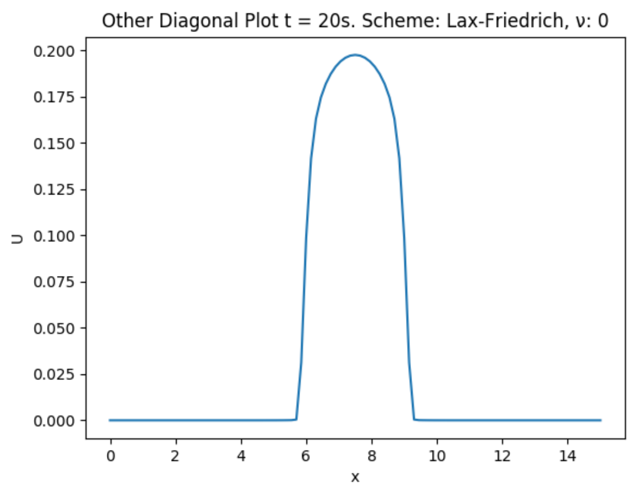

Main Diagonal Plot t = 20s. Scheme: FTCS2, ν: 0.02
U
x

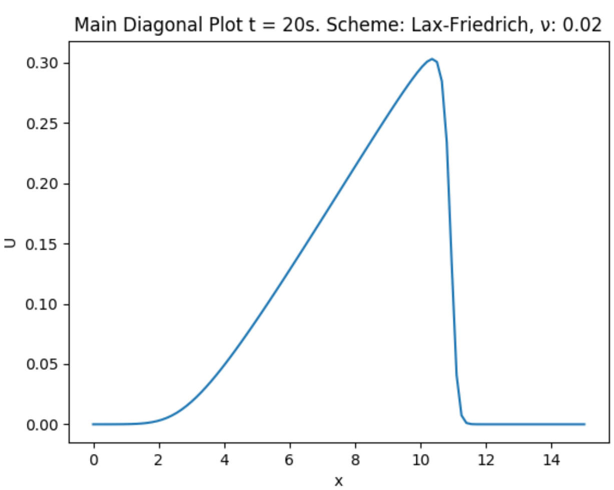

Other Diagonal Plot t = 20s. Scheme: FTCS2, ν: 0.02
U
x

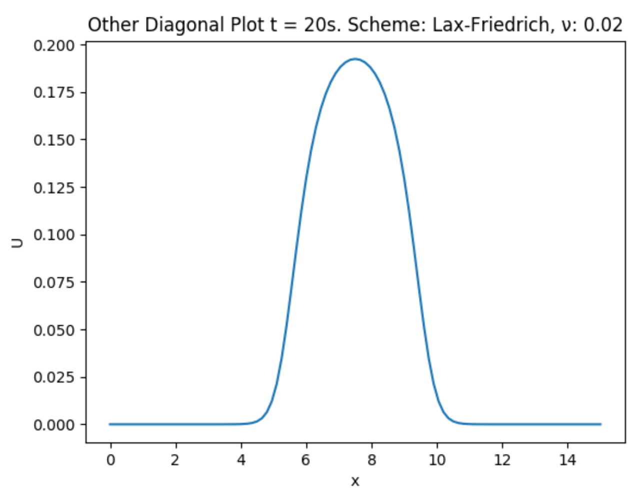

Main Diagonal Plot t = 20s. Scheme: FTCS2, ν: 0.04
U
x

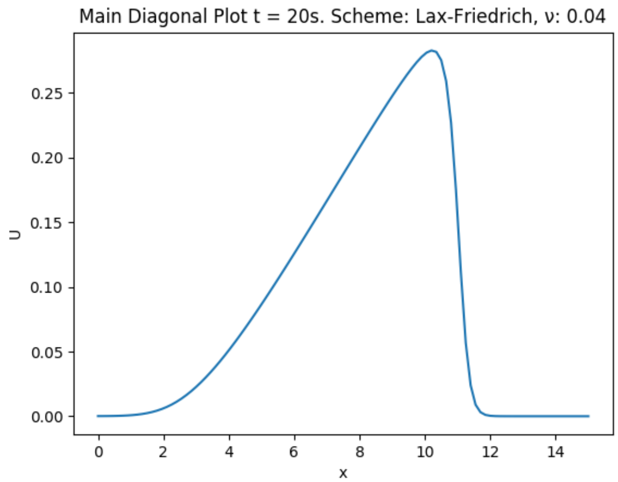

Other Diagonal Plot t = 20s. Scheme: FTCS2, ν: 0.04
U
x

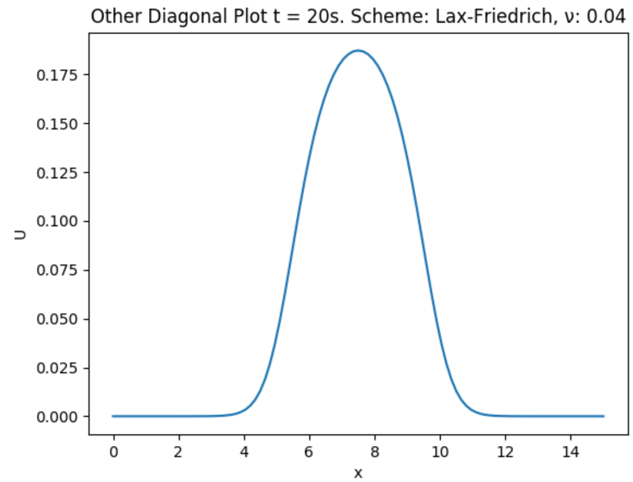

Main Diagonal Plot t = 20s. Scheme: Lax-Friedrich, ν: 0
U
x


Other Diagonal Plot t = 20s. Scheme: Lax-Friedrich, ν: 0
U
x


Main Diagonal Plot t = 20s. Scheme: Lax-Friedrich, ν: 0.02
U
x


Other Diagonal Plot t = 20s. Scheme: Lax-Friedrich, ν: 0.02
U
x





5.3.5 Conclusions

1. For $\nu = 0$, FTCS2 is unstable due to positive second derivative term. We get an error in generating surface plots for t greater than 5s.

2. Dispersion in FTCS2 and Lax Friedrich (due to third derivative term) is quite evident in the surface plots. We can see the irregularities and ripple kind of effect in the forward portion of the wave of FTCS2. The effect of dispersion seems to reduce on increasing ν as one can see that the FTCS2 solution for $\nu = 0.04$ is much smoother than that for $\nu = 0$. Dispersion effect is greater for FTCS2 than Lax-Friedrich solution.

3. Effect of diffusion is highest for FTBS scheme as we get a very smooth out curve for all ν values (negative second derivative term effect is greater).

4. In the diagonal plots, we can see that the spread of diagonal is greatest for FTBS scheme due to greater diffusion effect. The peak is also smooth in the main diagonal plot of FTBS scheme.

5. We see dispersion effects for FTCS2 main diagonal plots for $\nu = 0.02$ around the peak. This effect is also diffused out on increasing ν as we can see in the FTCS2 main diagonal plots for $\nu = 0.04$. The diffusing of peak on increasing ν is also observed in Lax-Friedrich solution.

6. In the diagonal plots for the same ν, we see that the peak is lowest for FTBS, followed by Lax-Friedrich and maximum for FTCS2 due to greater diffusion as mentioned above.

7. The opposite diagonal plots seems to demonstrate diffusion whereas the main diagonal plot seems to show advection direction and propagation.