Numerical Relativity

Exercise-1

Demetry Alfalahat

ID: 922365

Exercise 1-1

Note for the plots: Time is the time step (The function running period). $c_f = \frac{a\Delta t}{\Delta x}$ is the Courant factor. j is the resolution (number of points in the domain [Δx] as $\Delta x = \frac{L}{j-1}$)

Advection Equation → $\frac{\partial u}{\partial t} + \frac{\partial y}{\partial x} = 0 , x \in [0,10]$

Solving the advection equation using 4 schemes (FTCS, Lax-Friedrich, Lax-Wendroff, and Leapfrog) on Jupyter Lab using Python.

Initial data: $u(x, t = 0) = 10 * e^{-(x-5)^2}$

1- FTCS scheme: $u_j^{n+1} = u_j^n - a\frac{\Delta t}{2\Delta x} u_j^n [u_{j+1}^n - u_{j-1}^n]$

Theoretically, this scheme is unstable because it does not satisfy the CFL "a derivation was done in the class which shows how CFL condition is not satisfied with all the parameters that affect the stability". Using numerical solution, the theoretical result was confirmed as Figure 1 shows:

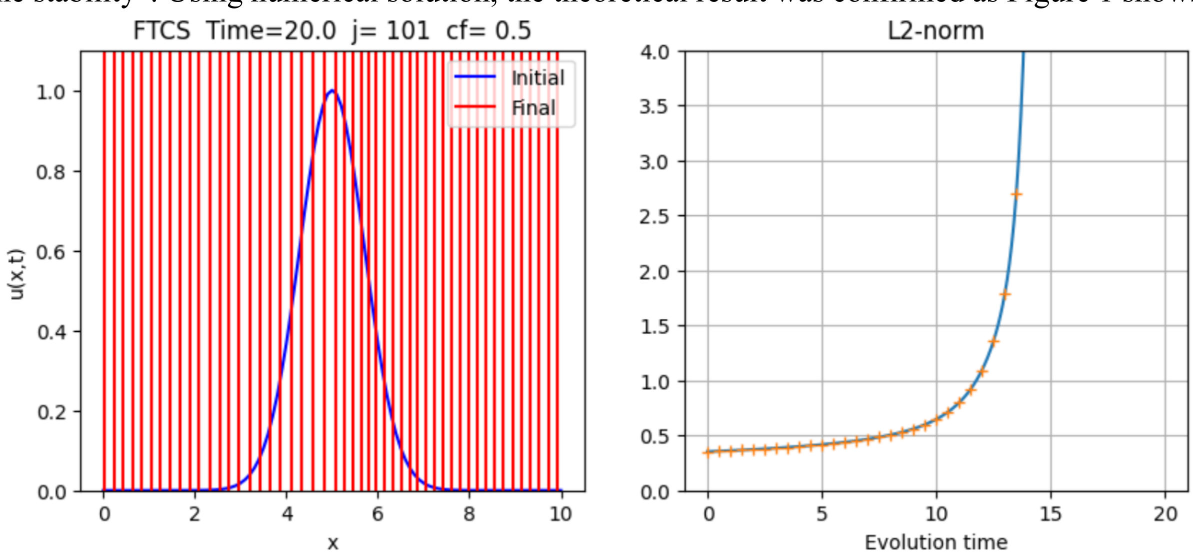
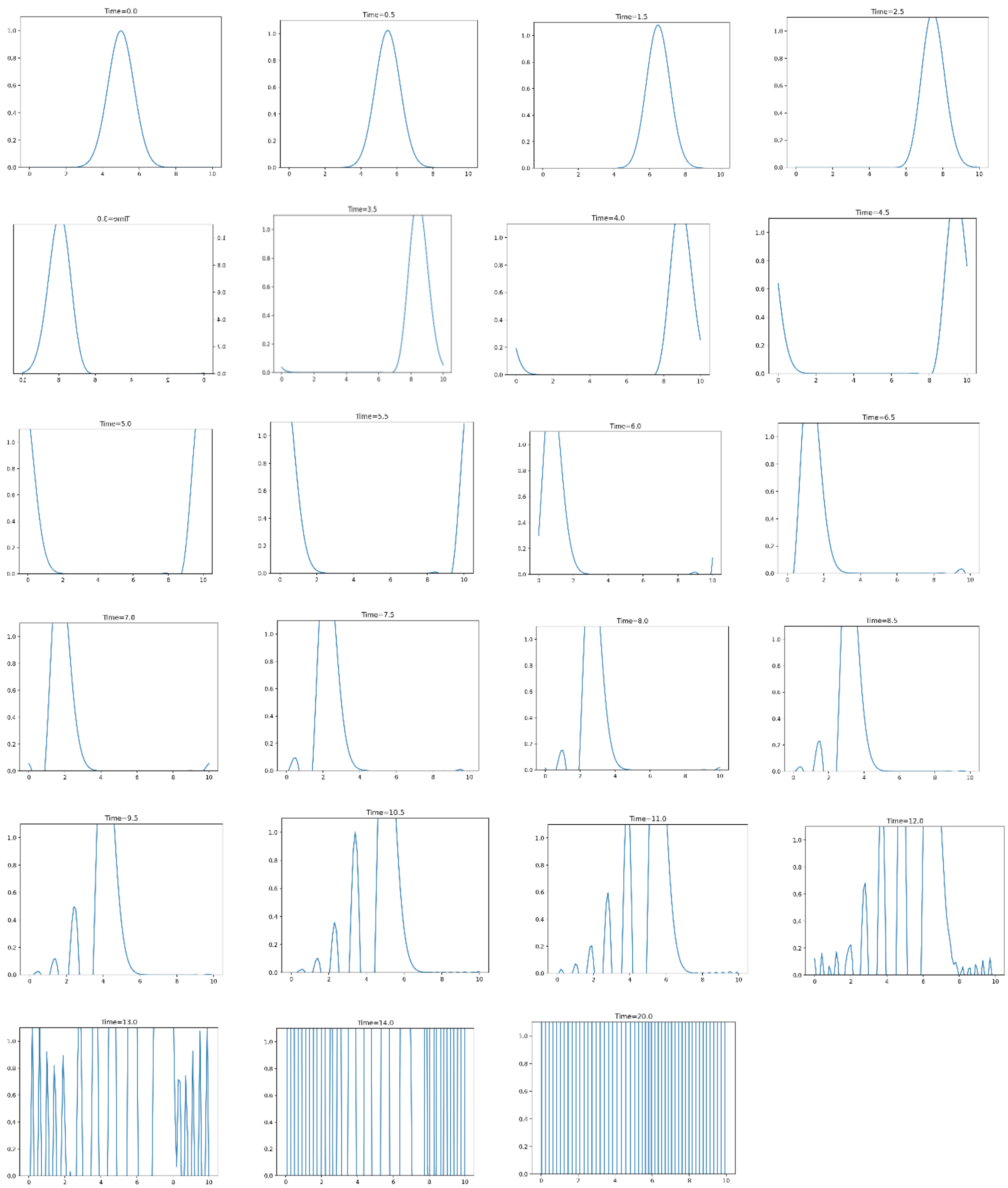


Figure 1: The plot on the left shows the initial data in blue while the data after solving the function with the FTCS scheme are in red. On the right, the plot shows the scheme state where it diverges because the function is unstable. The + sign in the plot on the right are saved values to be compared next after changing the cf, j and the time period.

Here are the snapshots of the function evolution through time for the same initial conditions described in Figure 1:



The FTCS scheme increases the maximum value of u(x,t) as we increase the evolution time and the Courant factor (Figure 2). While reducing the Courant factor, it would prevent the function from growing for a longer time (see Figure 3 and compare the norms in Figures 1,2 & 3).

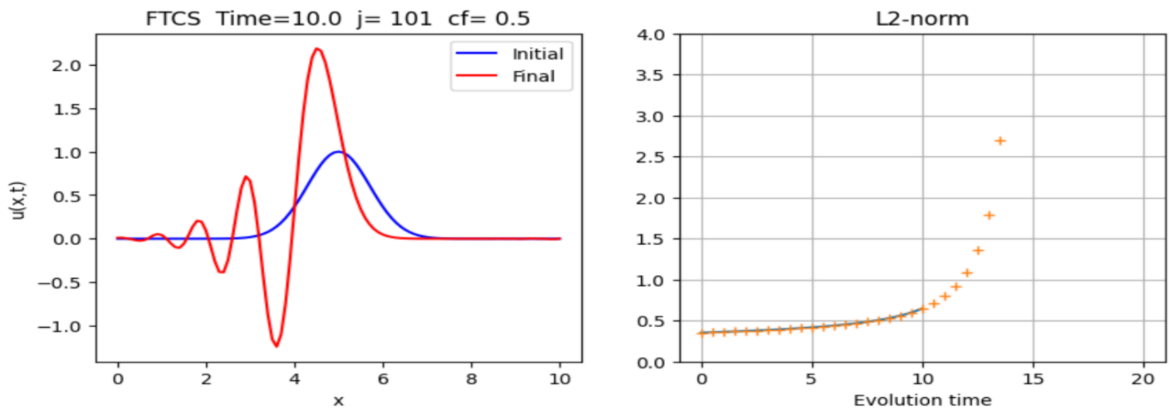


Figure 2: The plot starts perturbation but it stops due to the small-time interval without diverging.

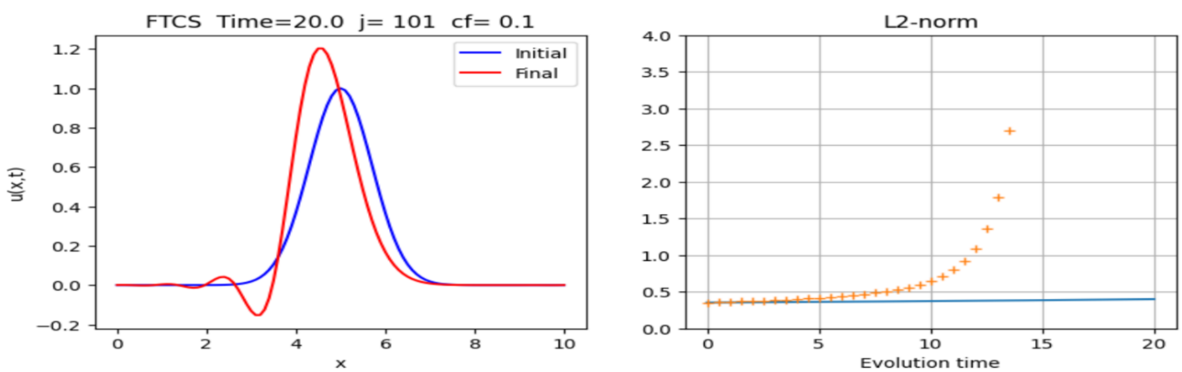


Figure 3: The plot now is more stable due to reducing the cf. On the right, the norm grows slowly.

Increasing the resolution (j) would reduce the perturbations as shown in Figure 4. Although, if we increased the cf or Time for a certain value, the perturbations will be the same as in Figure 1.

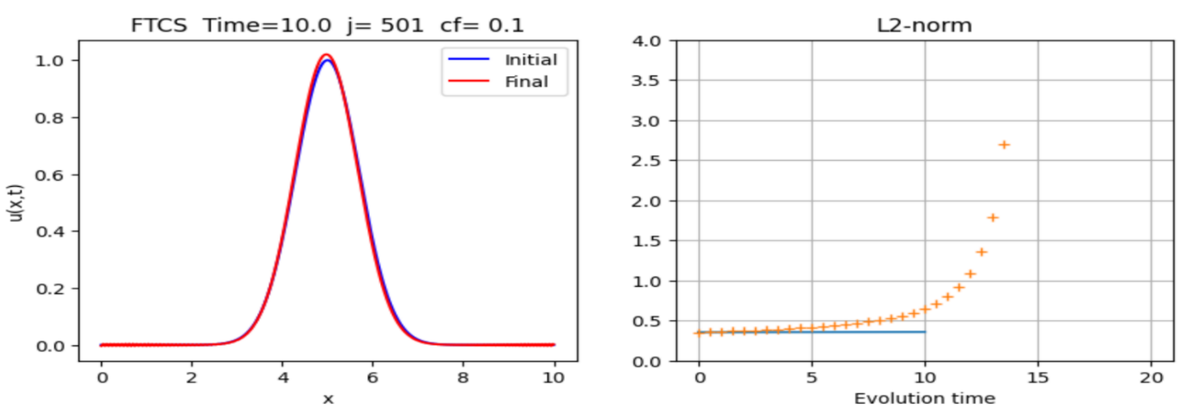


Figure 4: Perturbations vanished after increasing the resolution(j) and in a relatively short period.

2- Lax-Fredrich scheme: $$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - a\frac{\Delta t}{2\Delta x}[u_{j+1}^n - u_{j-1}^n]$$

Unlike the FTCS method, Lax-Fredrich's scheme is stable both theoretically and confirmed by numerical simulations and converges as the L2-norm shows in Figure 5:

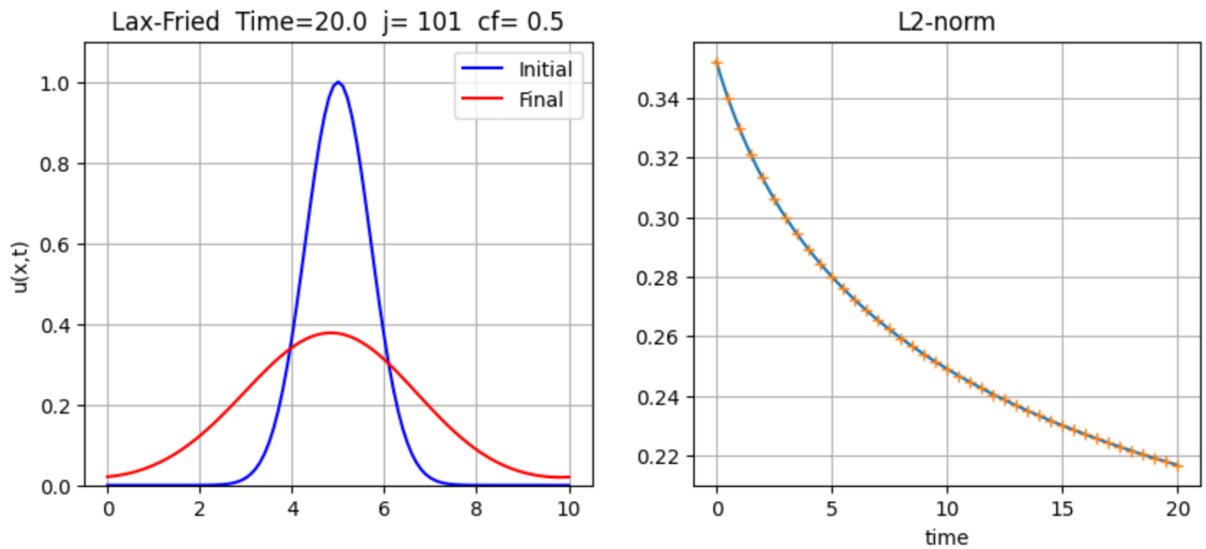
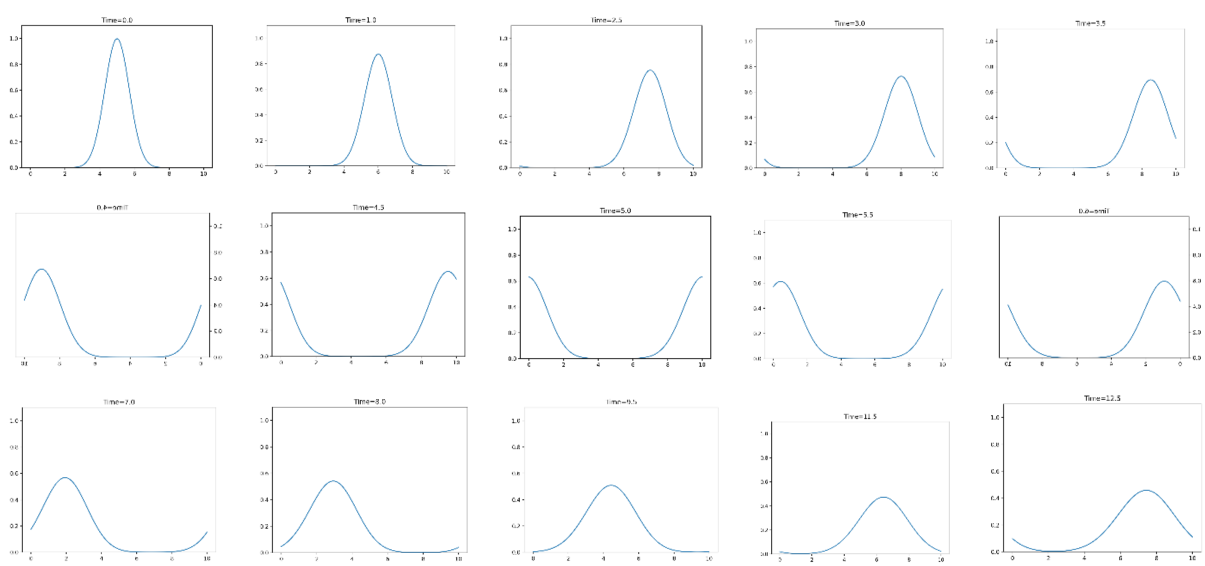


Figure 5: Lax-Friedrich's initial data in blue and the Final data after the numerical computation in red.

The function dissipates through time (u becomes less) as we can see in these snapshots. The function starts to move from left to right as you go with the pictures from top left.



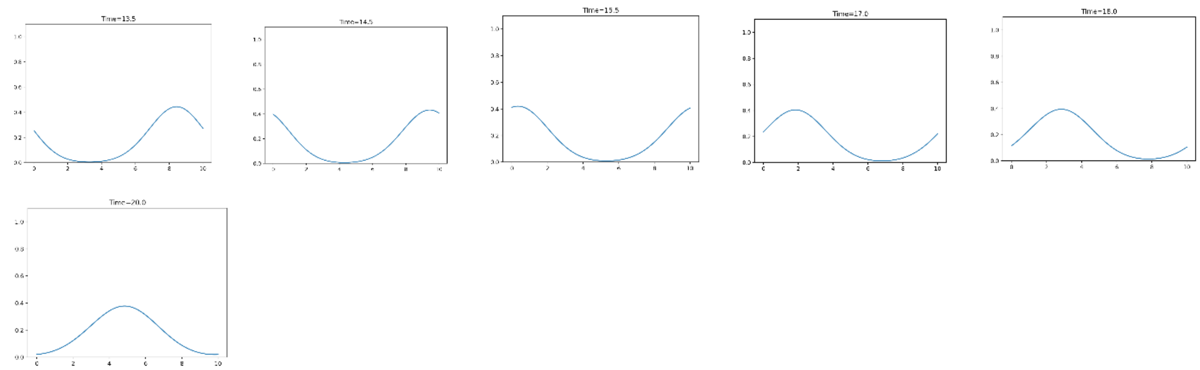


Figure 6: Lax-Friedrichs numerical simulation snapshots.

As we increase the resolution or the Courant factor, the final data becomes more aligned with the initial one and the function more stable as L2-norm represents in Figures 7&8:

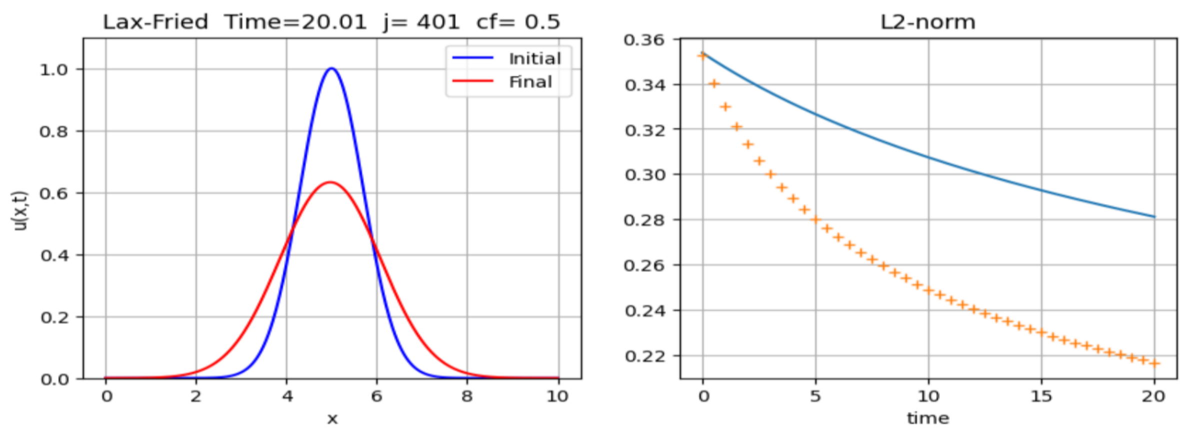


Figure 7: Increasing j gives the final data in red higher value (compare the highest value in figures 5,6 & 7).

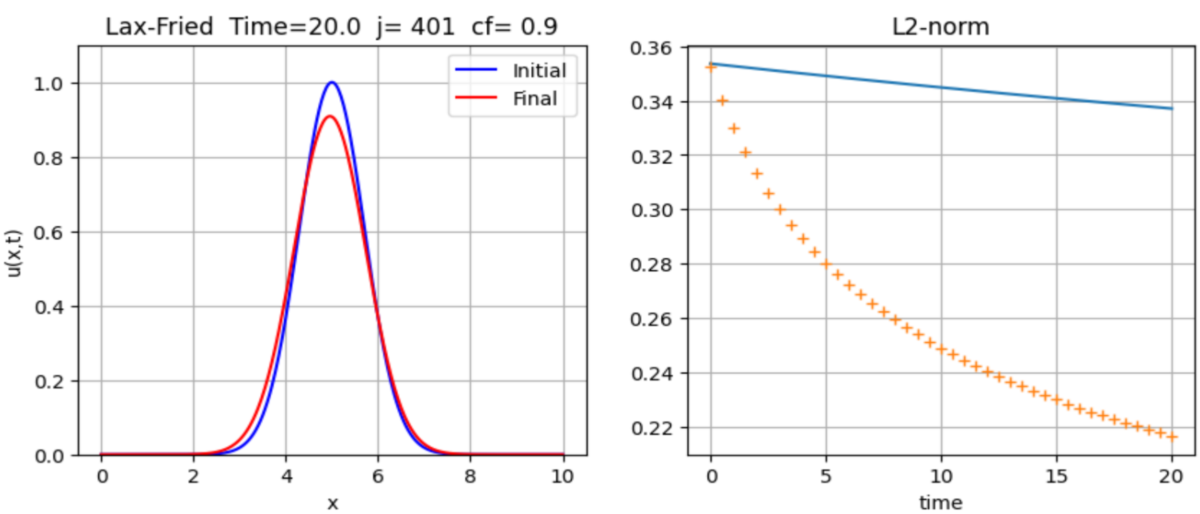


Figure 8: The function becomes stable more and more as we increase the cf or j (compare j and cf values with figure 6.

No need to show another numerical simulation with increasing time as it is obvious that the function becomes more stable with time (also, time is proportional to the cf).

3- Leapfrog scheme: $u_j^{n+1} = u_j^n - a\frac{\Delta t}{\Delta x}[u_{j+1}^n - u_{j-1}^n]$

The plot in Figure 9 shows the stability of the Leapfrog scheme compared to the FTCS and Lax-Friedrich schemes with the same values of j, cf and time (compare Figures 1,5 & 9):
The special property for leapfrog that it is very stable with small j as in figure 10. However, there is a notch around x = 3 that disappears by increasing j.

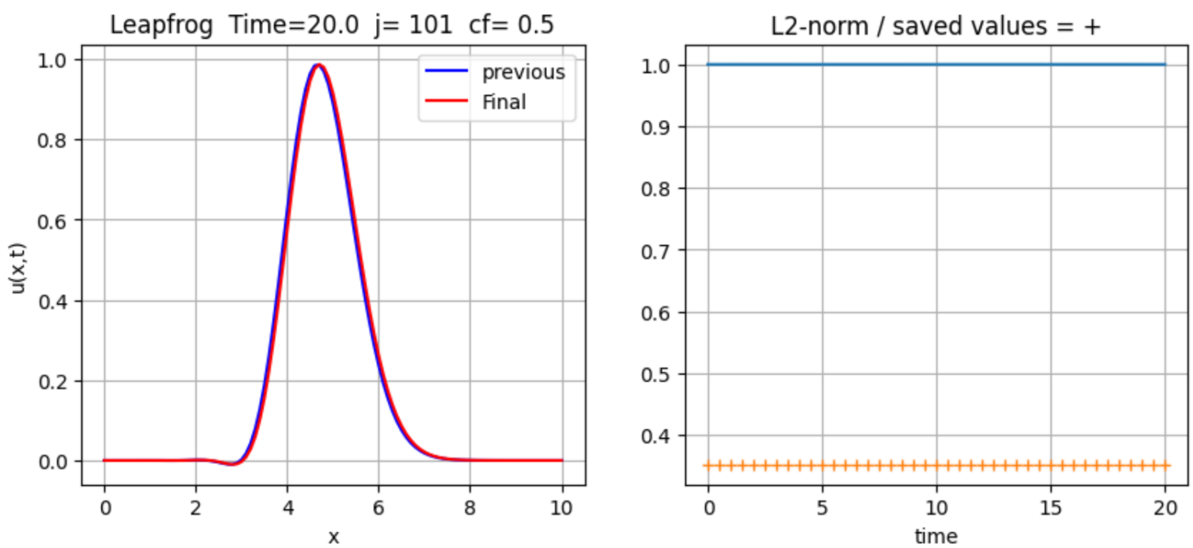


Figure 9: Leapfrog Plot, the final plot is perfectly aligned with the initial one (blue plot) except for the Notch around x = 3. For the L2-norm, we can't see the values due to the difference between the values. Check Figure 11 for better clarification.

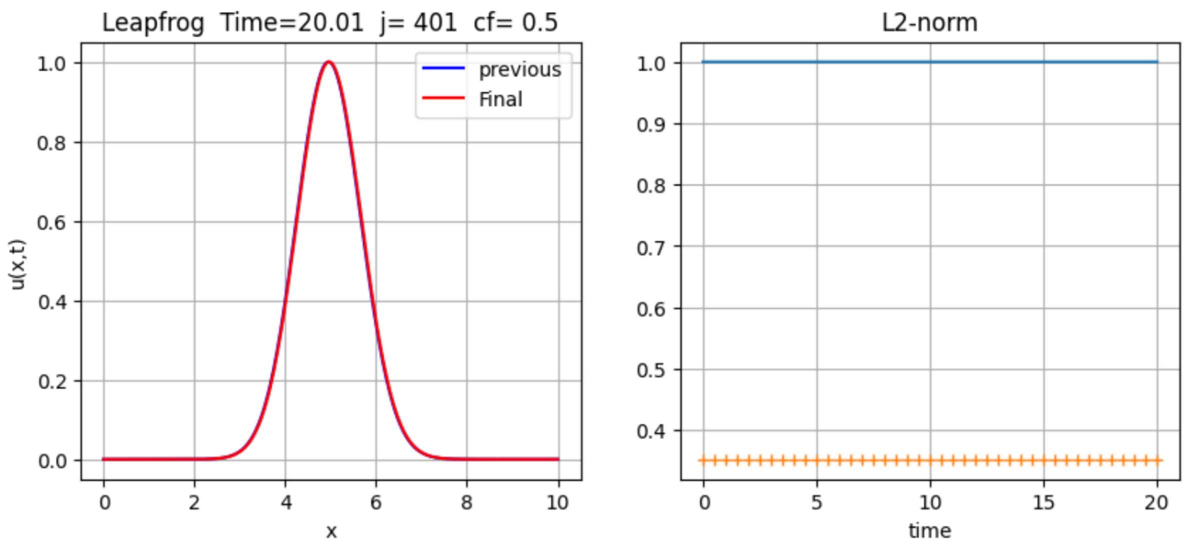


Figure 10: The notch disappeared and the plot is perfectly aligned with initial values (The blue plot behind the red one).

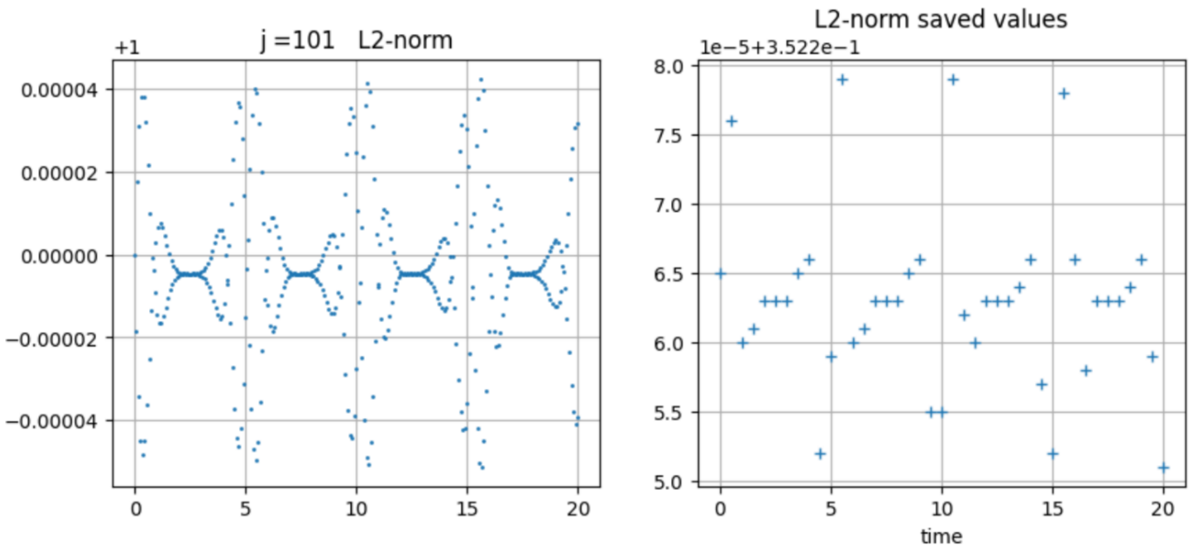


Figure 11: L2-norm for Figure 9. The left one is for the values regarding j = 101. Note the jumps around x = 5,10,15 and 20 between the 2 plots. The first value of L2-norm only is normalized in Figure 12

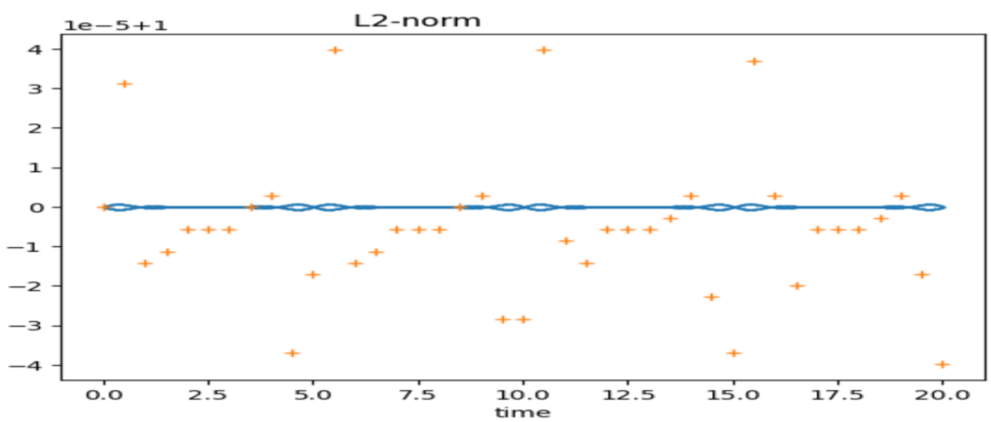
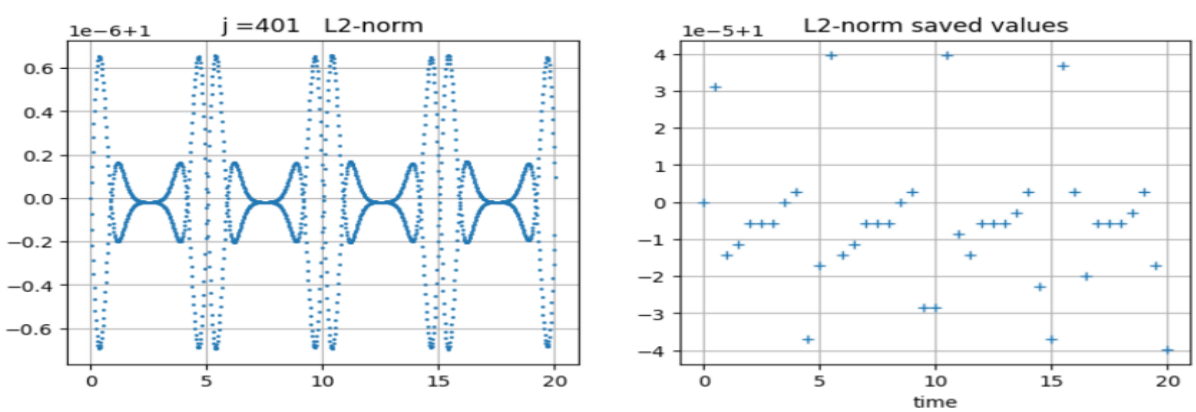
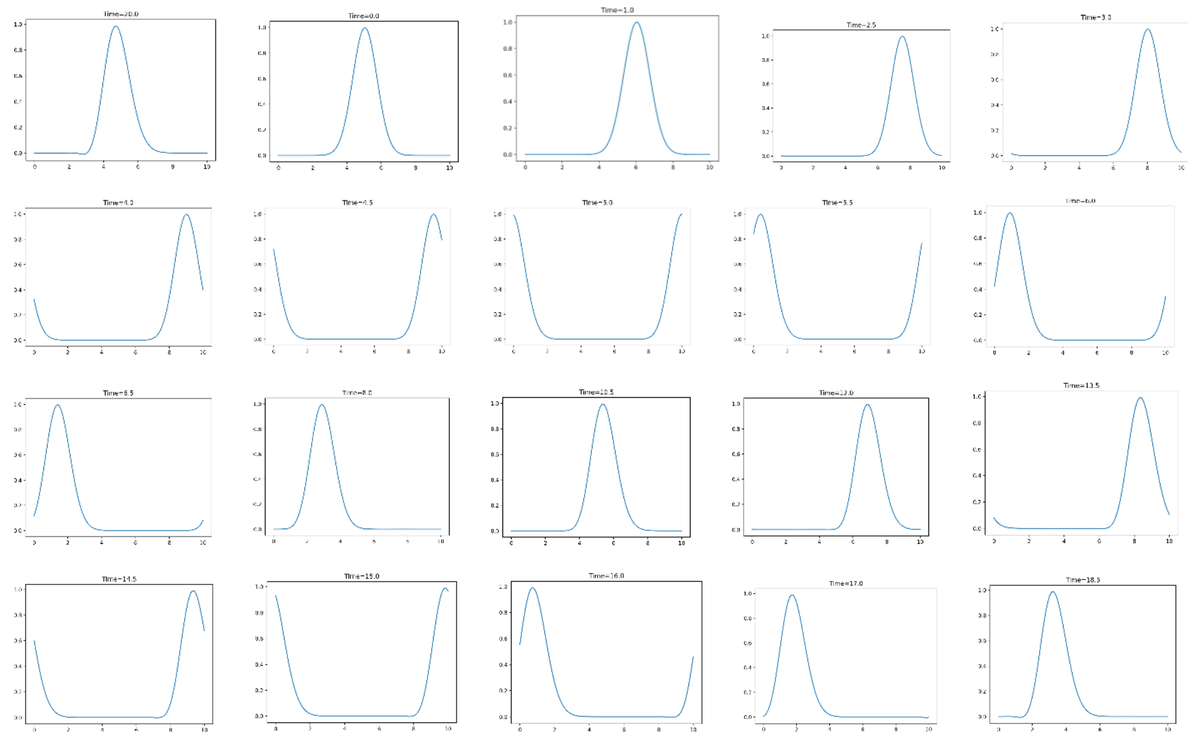


Figure 12: Only the first value of L2-norm which is at x = 0 is normalized.

One last Note for changing j or cf, it doesn't make a noticeable change. Although, increasing j will give closer values for a very high-resolution simulation which is the one from the professor. Note the difference in the y-axis for these plots and the one in Figure 11, "check the last plot j=4001 on the lab":



Here are the snapshots for Leapfrog simulation regarding the first case from figure 9:



4- Lax-Wendroff scheme: $u_j^{n+1} = u_j^n - \left(a\frac{\Delta t}{2\Delta x}\right)[u_{j+1}^n - u_{j-1}^n] - \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)^2[u_{j+1}^n - 2u_j^n + u_{j-1}^n]$

The results for Lax-Wendroff doesn't differ much from the leapfrog (Figure 13), as we increase the resolution or the cf, the solution will be more stable as in Figure 14.

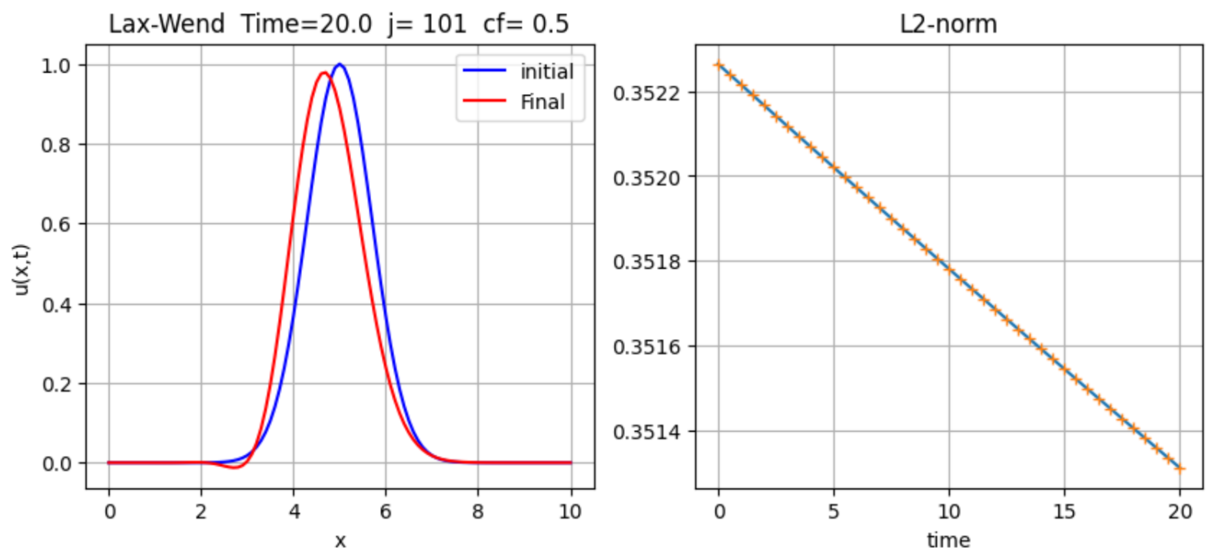


Figure 13: Lax-Wendroff initial data are not aligned with the final solution. The L2-norm is linear.

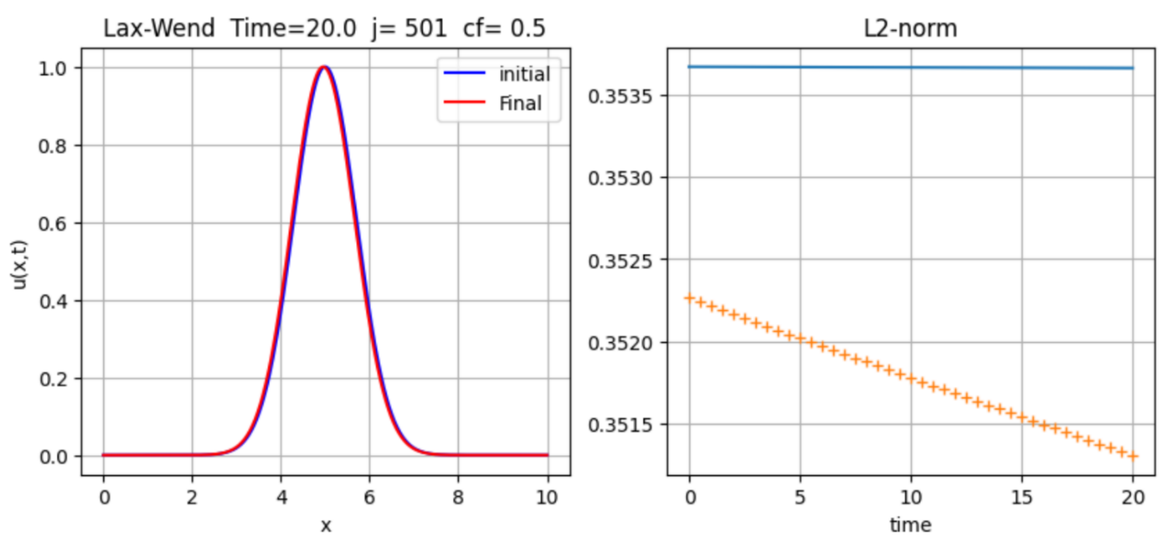
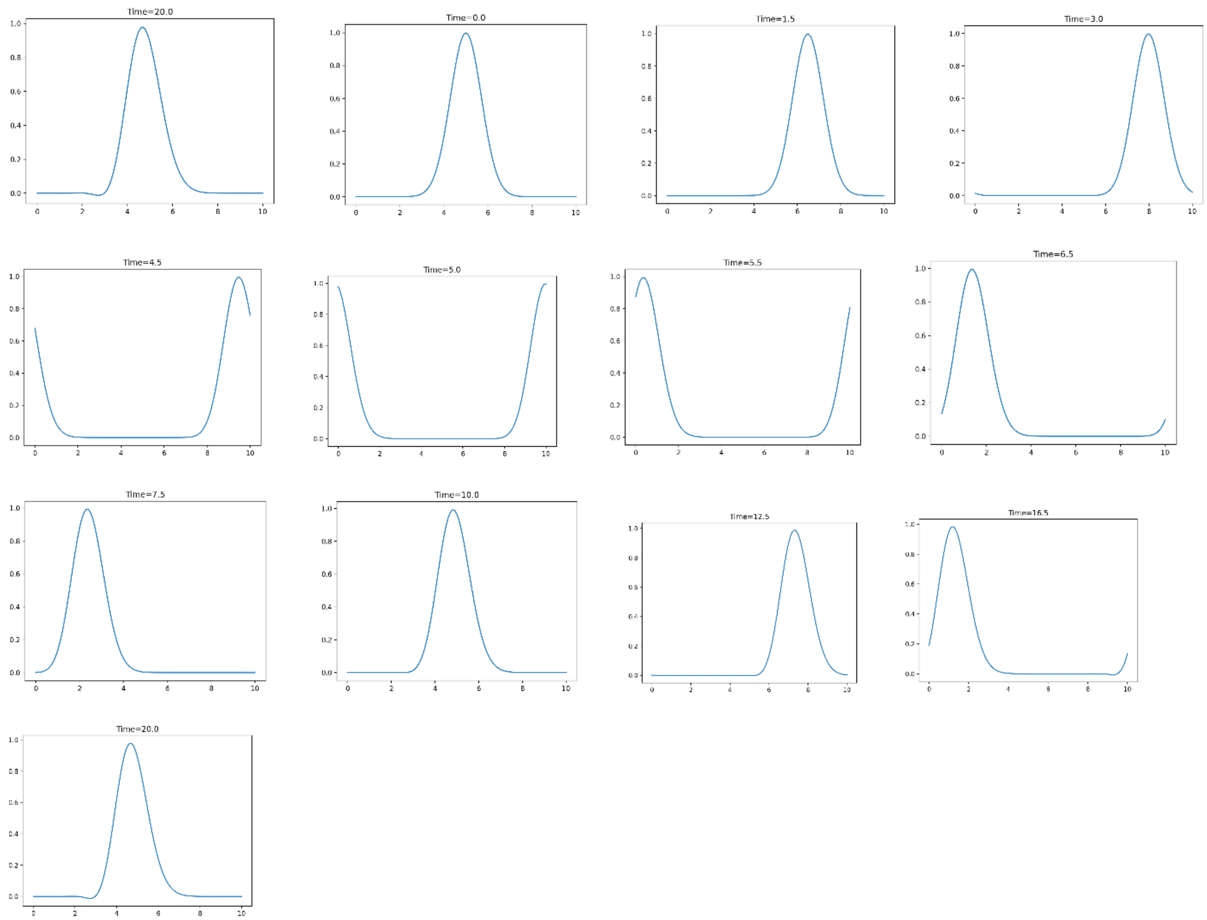


Figure 14: Maybe it is shaded, but the blue plot is behind the red one.

Some snapshots illustrating the function evolution through time for the final solution in Figure 13:



Exercise 1-2

Solving the advection equation using 2 schemes (Lax-Friedrich and Lax-Wendroff) on Jupyter Lab using Python for a step function.

Initial data: $u(x, t = 0) = 1$

1- Lax-Fredrich scheme: $$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - a\frac{\Delta t}{2\Delta x}\left[u_{j+1}^n - u_{j-1}^n\right]$$

Using numerical solution for the step function, Figure 1 shows oscillations and dissipation in the function as well as the L2-norm goes to zero:

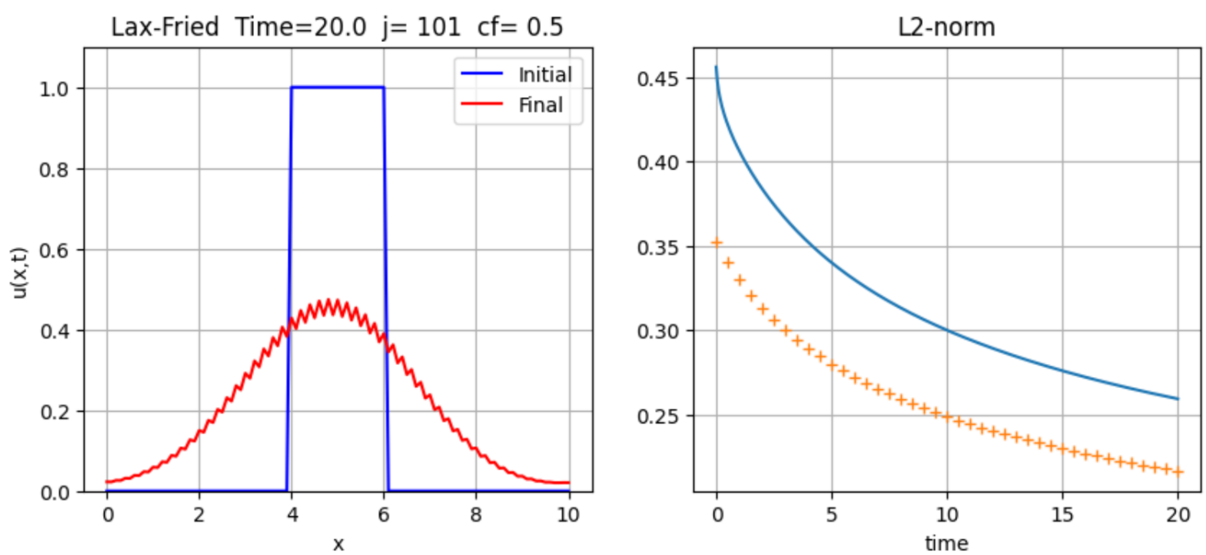
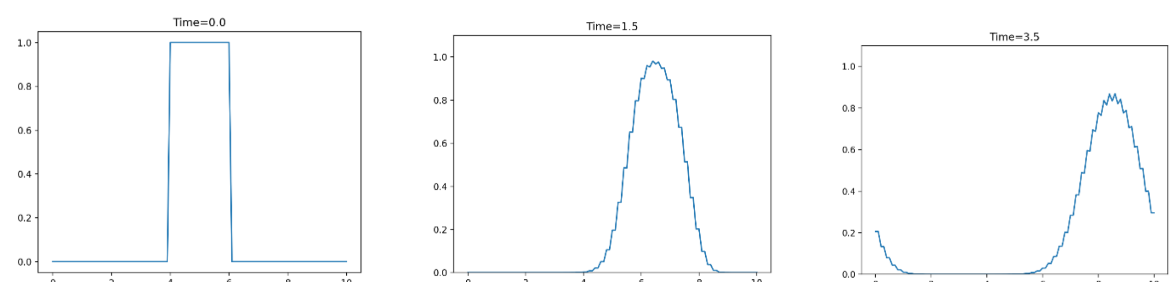
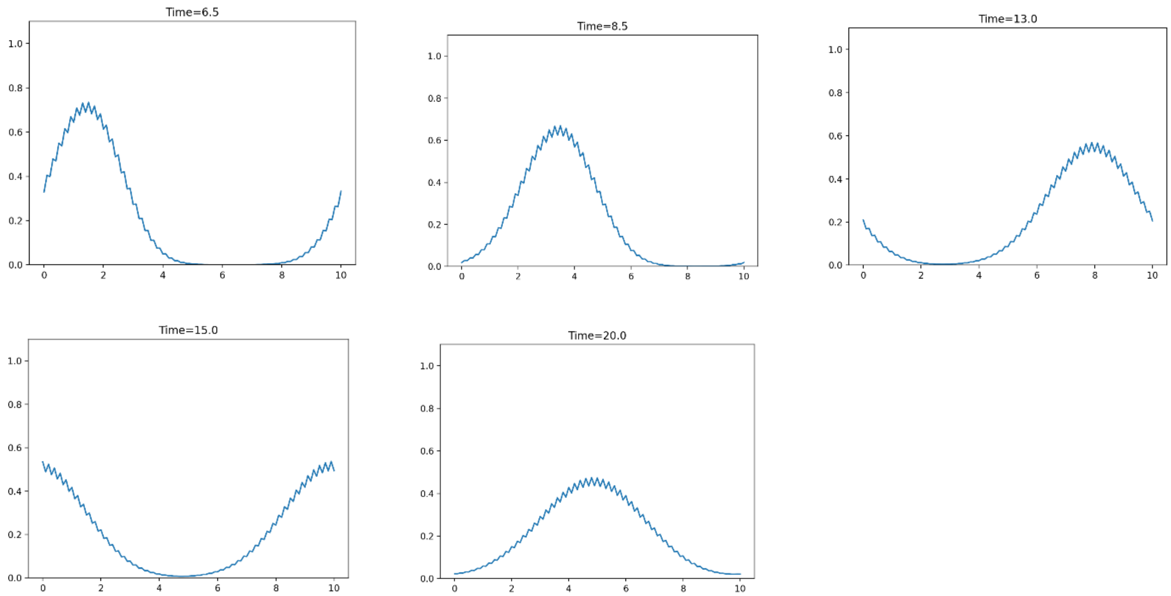


Figure 15: The step function simulation shows perturbations.

Check these snapshots for Figure 1 between the initial function and the final one:





Changing the number of points (j) and cf will make less perturbations and increase the stability of the function as we see in Figure 16 & 17:

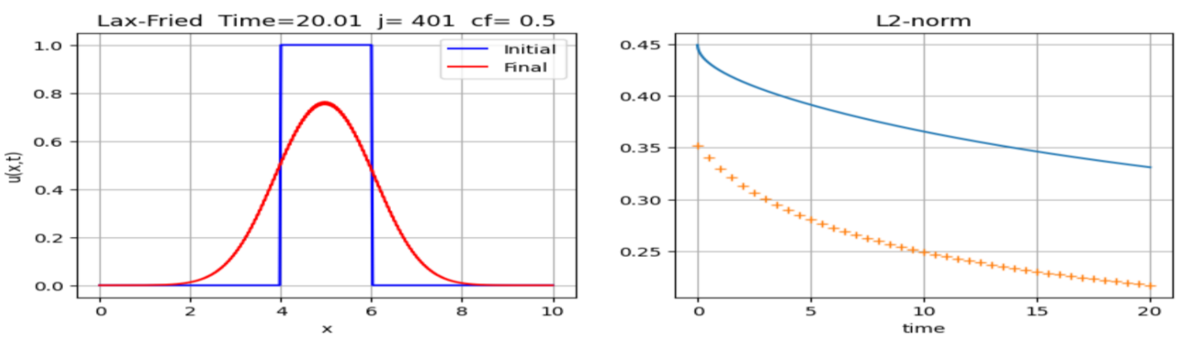


Figure 16: The simulation with higher number of points (j) gives less perturbed function.

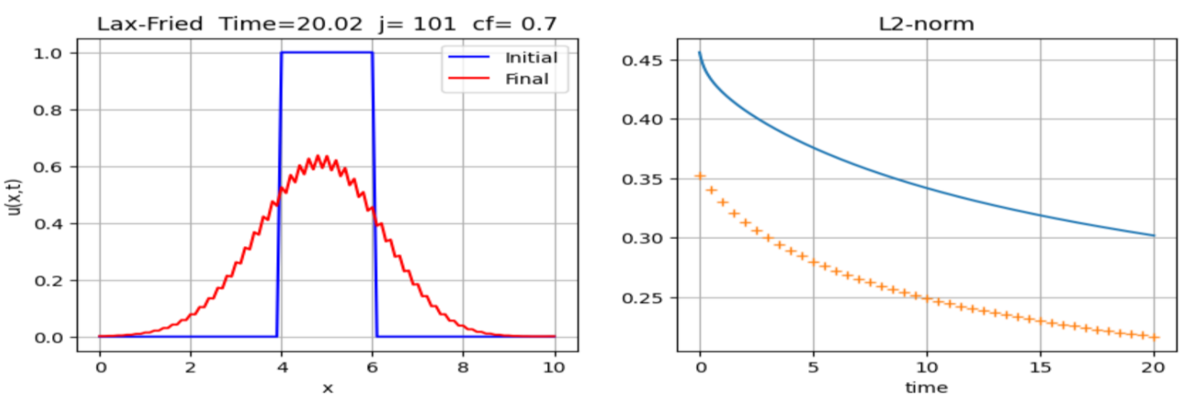


Figure 17: Increasing the cf while j=101. Note the the highest and lowest value of the norm for Figures 1,2&3.

2- Lax-Wendroff scheme: $u_j^{n+1} = u_j^n - \left(a\frac{\Delta t}{2\Delta x}\right)[u_{j+1}^n - u_{j-1}^n] - \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)^2[u_{j+1}^n - 2u_j^n + u_{j-1}^n]$

This scheme shows different behavior as at each jump point the function get disturbed the most, this will be explained after showing Figure 18. For the first simulation to be compared with the Lax-Friedrichs scheme that becomes less with time while for Lax-Wendroff, it has a dispersive behavior as in Figure 18:

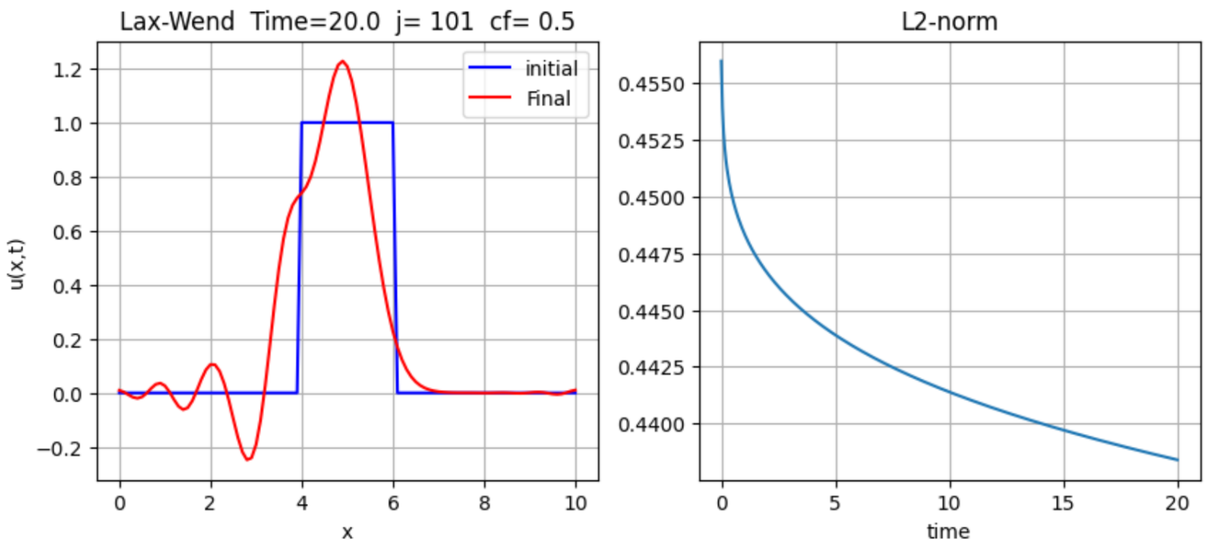


Figure 18: Lax-Wendroff scheme applied for the step function. The L2-norm here is more stable than Lax-Friedrich, compare with Figure 3 & 2

The simulation shows the dispersive behavior before the jump points (at x = 4 & x = 6) as in Figure 19:

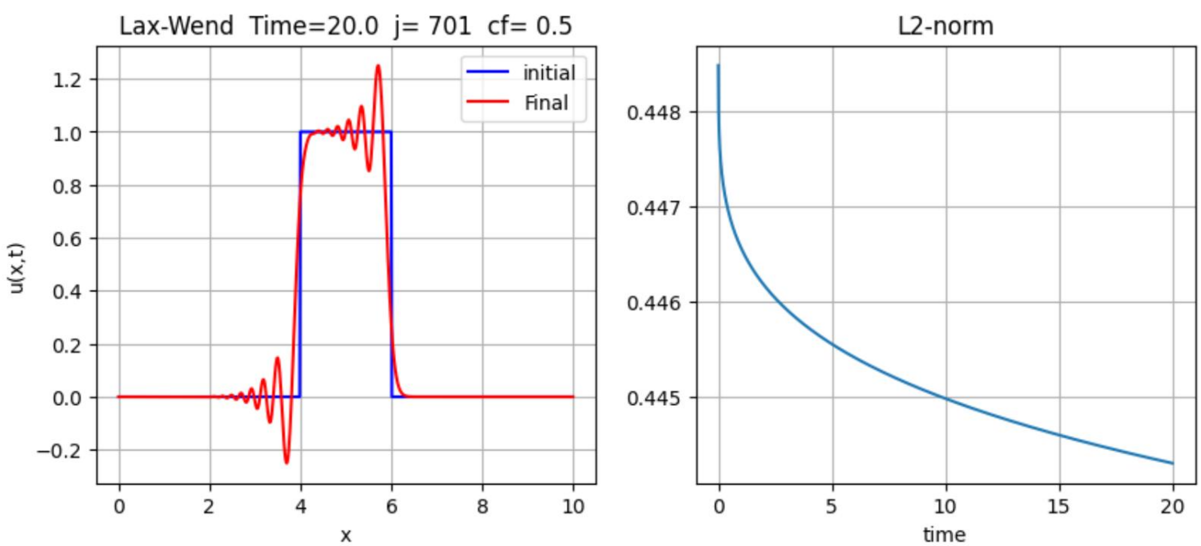


Figure 19: The function is more stable regarding the L2-norm while still shows dispersive behavior in the left plot.

Increasing or decreasing the cf wouldn't change a lot nor in the simulation nor in the L2-norm as in Figure 20:

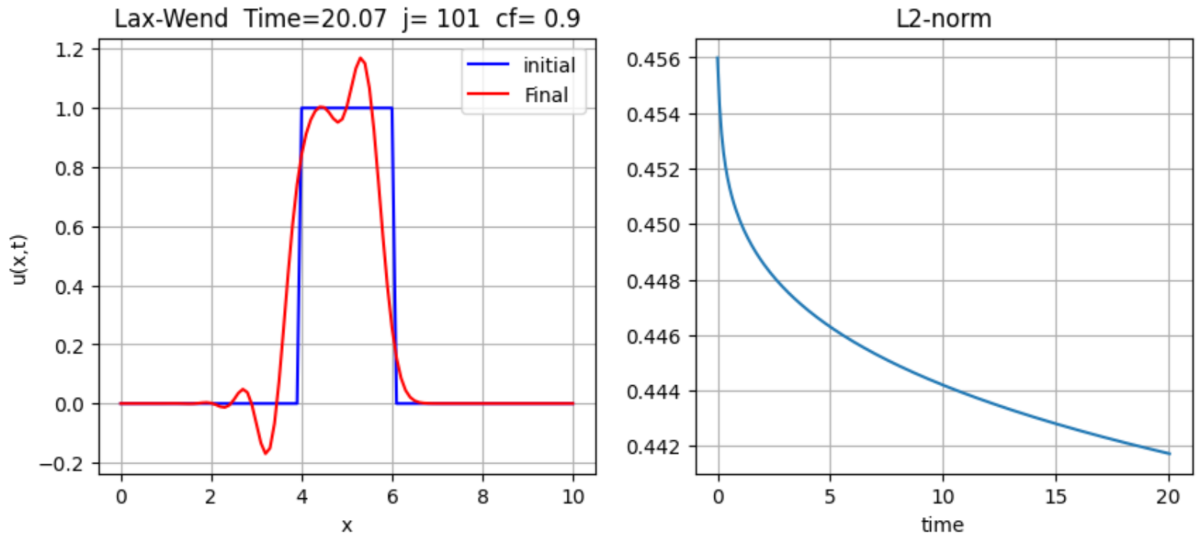
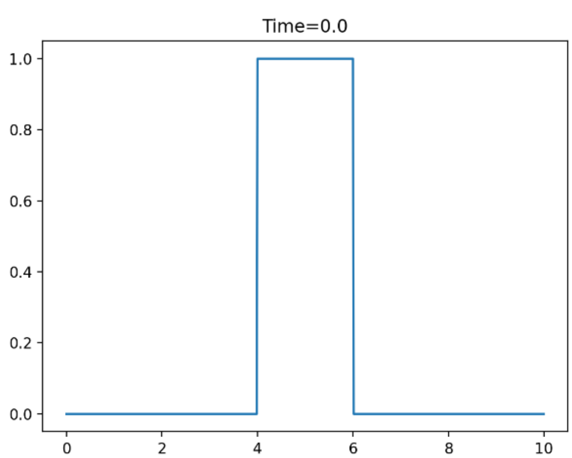
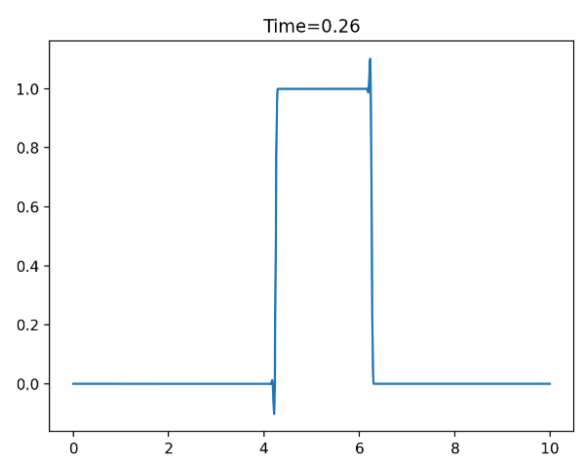
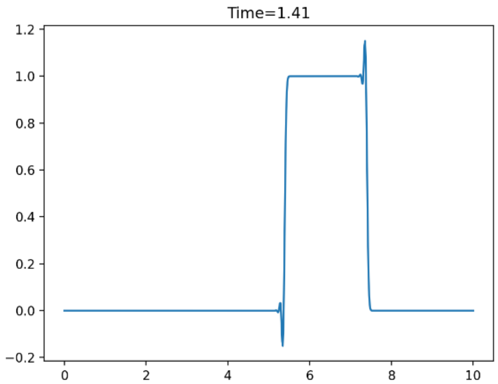


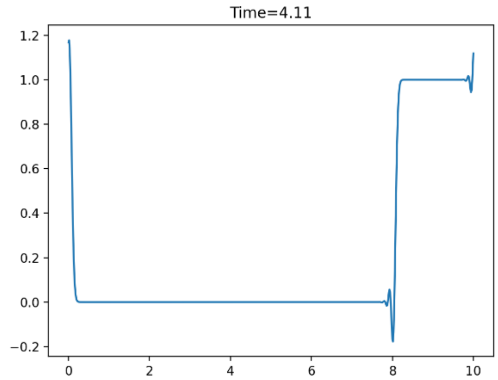
Figure 20: Increasing the cf would make the function more stable but the change is very small in stability as we see in the L2norm.

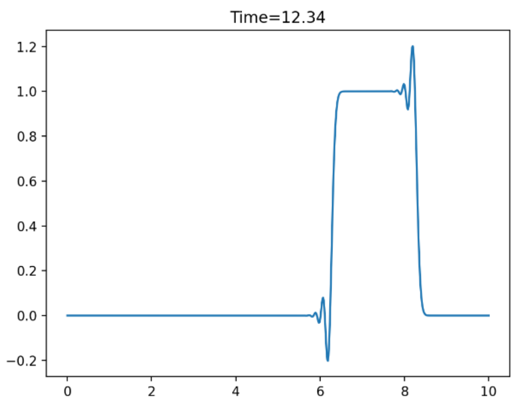
Finally, I will provide snapshots for the behavior of the function with cf = 0.9 and j = 701 to show the most stable case I did:

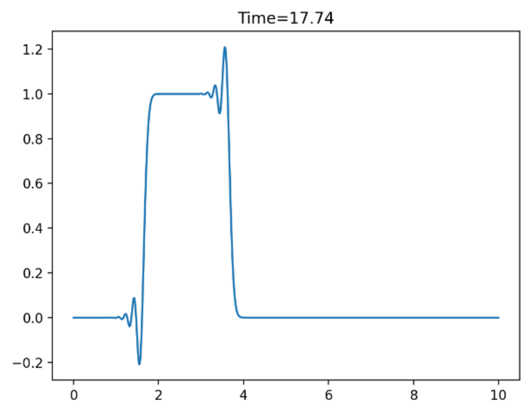


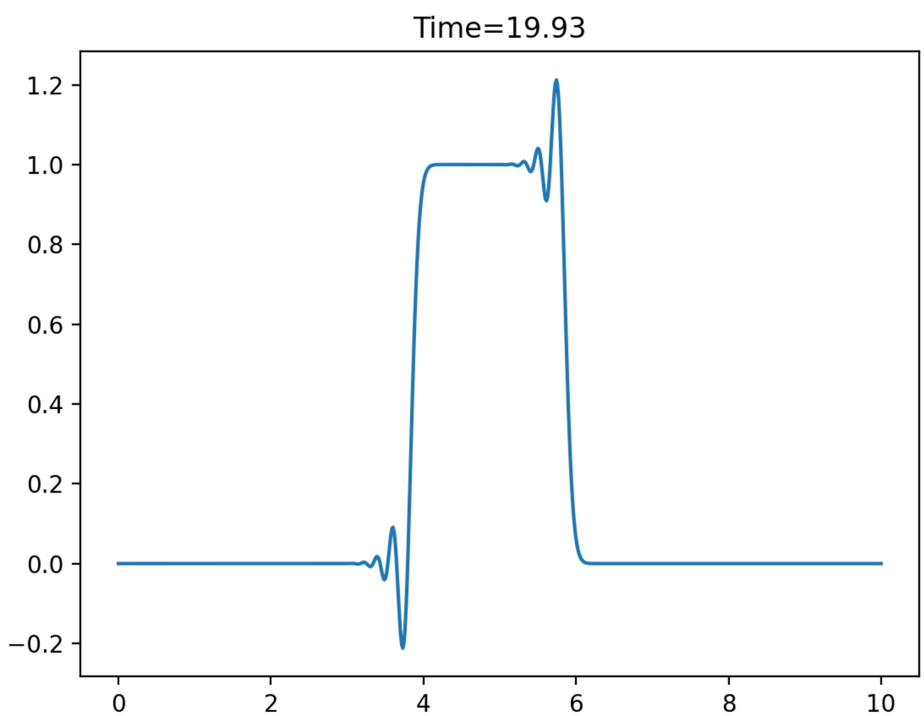



Time=1.41
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10


Time=4.11
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10


Time=12.34
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10


Time=17.74
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10


Time=19.93
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10

Exercise 1-3

Note for the plots: The y-axis is for the velocity u and the x-axis is for space dimension x. Time is the time step Δt (The function running period). c_f is the Courant factor.

Burger's Equation $\quad \frac{du}{dt} + u\frac{du}{dx} = 0\ , x \in [0, 10]$

Initial data: $u(x, t = 0) = 10 * e^{-(x-x_0)^2}$

The initial data develop shock and rarefaction wave in both cases (Conservative and Non-Conservative Flux, CF and NCF respectively).

CF: $u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}[f_j^n - f_{j-1}^n]$ NCF: $u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} u_j^n [u_j^n - u_{j-1}^n]$

For the Flux Conservative form, I got these results (Figure 21):

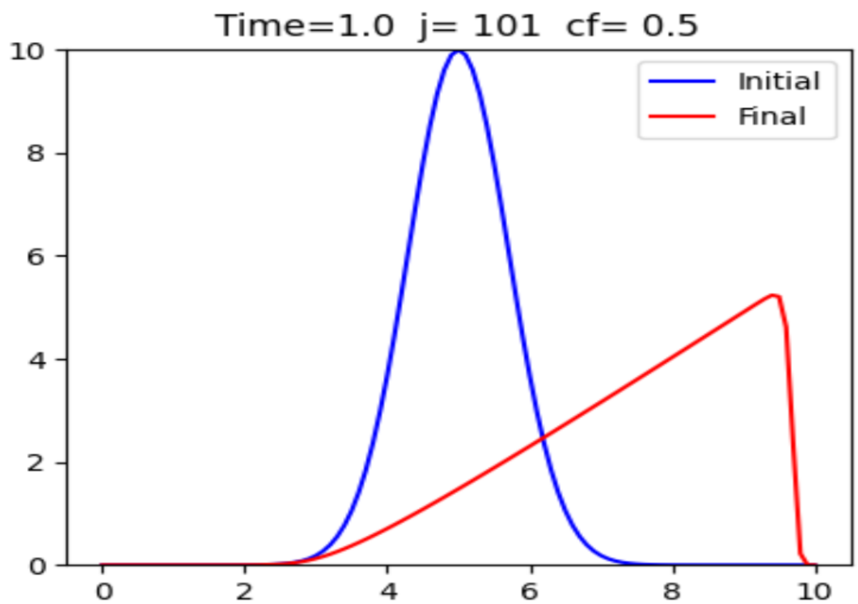


Figure 21: The blue line is the plot for the initial data while the red is for the initial data after a time step (Time = 1).

1- If a is less than 10 the method will be unstable due to the CFL. 10 is the maximum value of the function of the initial data.

2- Increasing the resolution will increase the sharpness of the shock (Compare Figure 21 to Figure 22). But at some point, increasing j does not produce visible change (Figure 22).

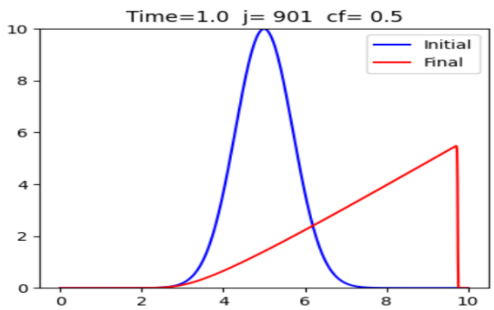


Figure 22: Plot for CF with evolution time=1

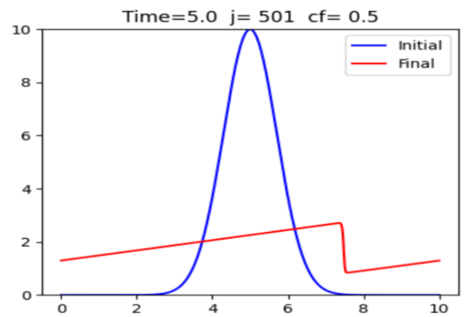
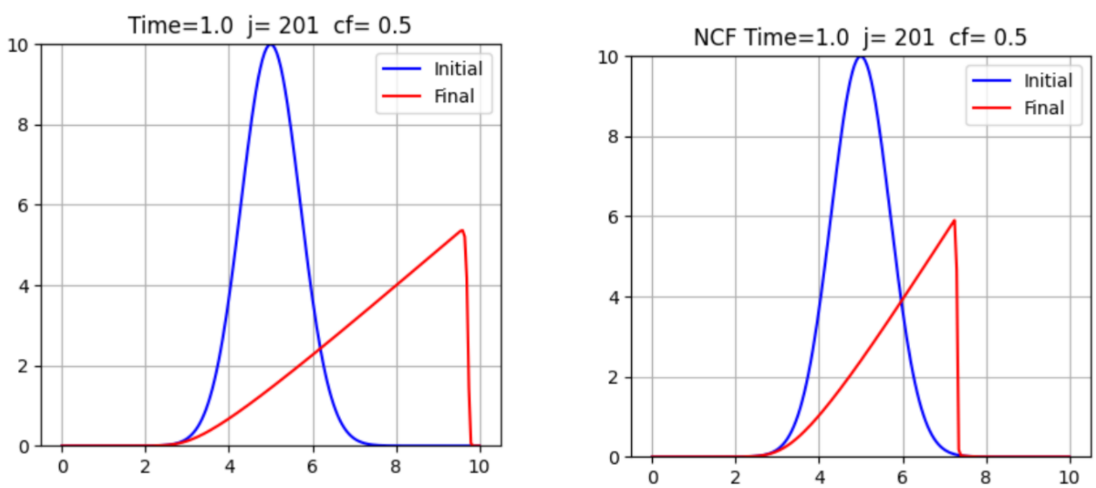


Figure 23: Plot for CF with evolution time=5

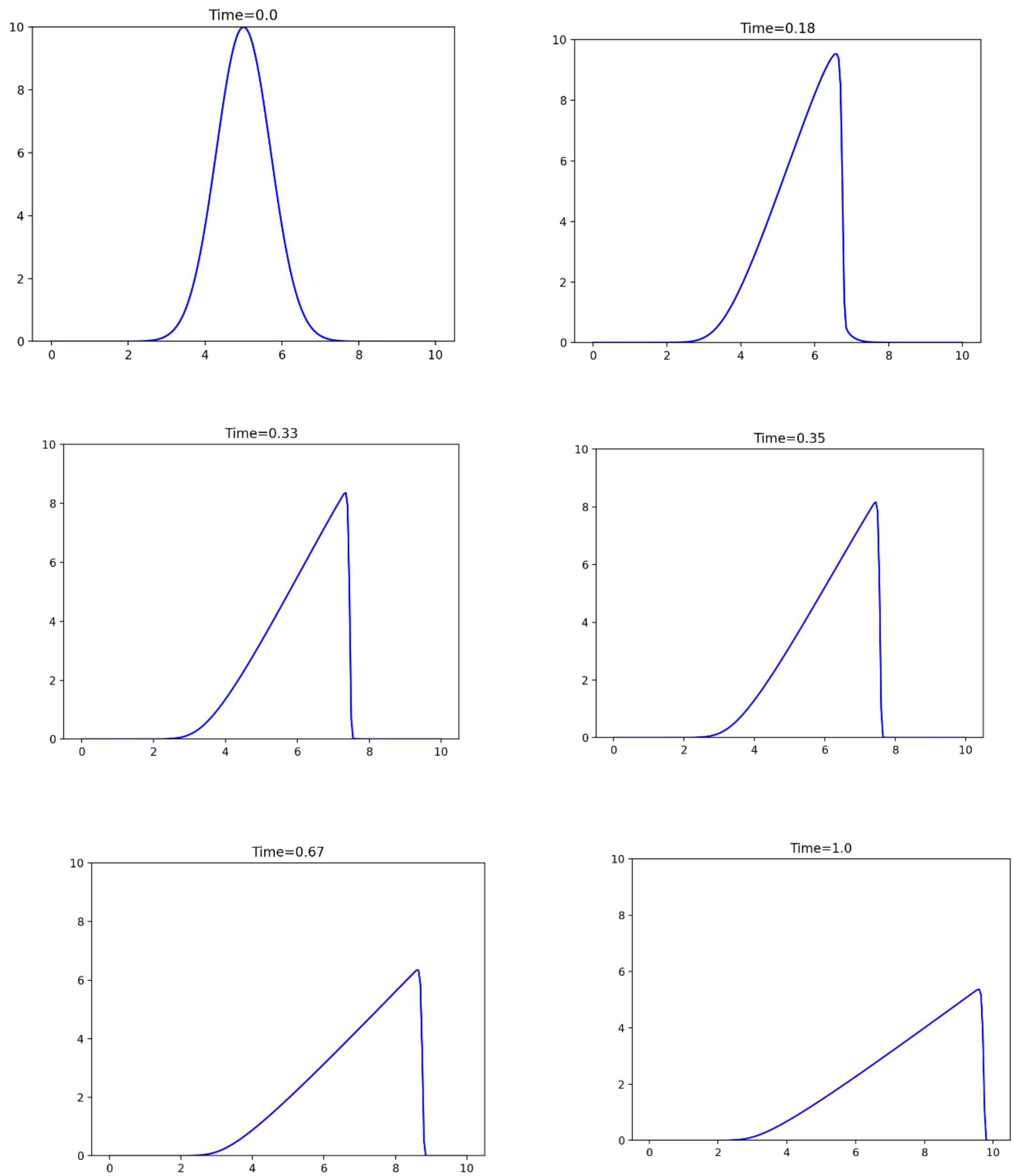
3- The shock maximum value depends on “Time”. Increasing the code run time step Δt, will make dissipation in the shock more and more (The maximum value of the shock will be less). Compare Figure 22 & 23.

4- Both forms CF and NCF act the same under changing the resolution and the time of evolution. The difference is that in NCF gives the wrong solution for the equation as expected from Hou-le Flock Theorem while CF gives the right solution as expected from Lax-Wendroff Theorem. The NCF creates a shock at a different position on the x-axis. (see figure 24)

Figure 24: *The solution for the CF is on the left and for NCF is on the right.*



Here are the snapshots for Burger’s equation simulation (All simulations have been done produce rarefaction and shock waves with same behavior):


Time=0.0
Time=0.18
Time=0.33
Time=0.35
Time=0.67
Time=1.0
0
2
4
6
8
10