A REPORT ON SOLUTION OF PARTIAL DIFFERENCE EQUATIONS USING FINITE DIFFERENCE APPROXIMATIONS

BY

IBEMORAH ARINZE EMMANUEL

Table of Contents

ABSTRACT
This report outlines the use of finite difference approximation for the numerical solution of partial difference equations with a focus on linear advection equation. Finite difference methods are widely utilised in solving partial difference equations in different industries, they provided a discretised representation of a continuous problem. This report details the use of a type of finite difference method which is finite difference approximation, it delves into the fundamentals and how they are applied to solve the linear advection partial differential equation (PDE).

INTRODUCTION
Partial Differential equations (PDEs) are mathematical equations that includes various variables and their partial derivatives and are used to model a range of different physical conditions in different scientific and engineering fields (Mainardi & Carpinteri, 1997). PDEs can be used to describe how a physical quantity like current varies with respect to multiple variables and the rate at which they change (differential) (Diethelm & Freed, 1999).
This paper tackles the linear advection problem using the numerical method of finite difference approximation, emphasising the difficulties posed by various finite-difference techniques. Improving our understanding of the behaviour, precision, and stability of various numerical techniques in estimating solutions to the linear advection problem is the goal.

AIM
This paper aims to
1. Analyse and understand partial difference equations using the one-dimensional linear advection and ways to solve them.
2. To analyse and compare the performance of different finite-difference schemes in solving the linear advection equation.
3. To analyse the stability characteristics of the different finite-difference schemes and assess their accuracy based on the courant number.
4. To assess how consistent each finite-difference scheme is with the main linear advection equation.

OBJECTIVES
The objective of the paper includes:
1. Examine the practical applications of linear advection equation and present the exact solution.
2. Implement various finite difference schemes for solving linear advection equation and compare the solutions with the exact solution to evaluate accuracy.
3. Evaluate the stability and convergence of the finite difference schemes under different courant numbers.

Background
Linear advection equation

The linear advection equation is one of the fundamental PDEs that models the transportation of a physical quantity by constant fluid velocity. It is an important PDE in various scientific and engineering fields like that of numerical solution.
The one-dimensional linear advection is expressed as:

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} = 0 \qquad (1)$$

Where:
u = is the quantity being transported
A = is the constant for the advection velocity

t = is the time of advection
x = is the spatial coordinate

The equation basically explains how u is conserved when travelling across space at a fixed speed of A.
Different ways can be used to solve the linear advection equation, but this paper focuses on finite difference methods.

FINITE DIFFERENCE EQUATIONS

Finite Difference method is a numerical approach used to solve PDEs through finite difference approximation. These methods involve the discretization of space and time domains to make them easier to solve, then applying the finite difference approximation after that then solving for the resulting algebraic equations.
Finite Difference approximation involve replacing the derivates in a mathematical equation with expressions of finite difference expressions. These expressions are basically discretized differential operators making them algebraic expressions which can be solved.

Finite Difference Approximation Methods

- Forward Difference method:
  This method estimates the derivative of a function at a point slightly ahead of the variable with respect to the variable at the given position.
  In context of time, it estimates the derivative at a time t_n by considering the function values at t_n and t_{n+1}, it is given by:
  $$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^{n}}{\Delta t} \quad (2)$$
  For a spatial derivative it is given by:
  $$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i}}{\Delta x} \quad (3)$$
  An example of this scheme is the Forward in time centred in space Method:
  $$u_i^{n+1} = u_i^n - c\frac{\Delta t}{2\Delta x}(u_i^n - u_{i-1}^n) \quad (4)$$

- Backward Difference:
  This method estimates the derivative of a function at a point slightly behind the variable with respect to the variable at a given position.
  In context of time it, it estimates the derivative at a time t_n by considering the function values at t_n and t_{n-1}, it is given by:
  $$\frac{\partial u}{\partial t} \approx \frac{u^{n} - u^{n-1}}{\Delta t} \quad (5)$$
  For a spatial derivative it is given by:
  $$\frac{\partial u}{\partial x} \approx \frac{u_{i} - u_{i-1}}{\Delta x} \quad (6)$$
  An example of the backward difference method is the Implicit Euler Method:
  $$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^{n+1}) \quad (7)$$
  and the Beam warming scheme:
  $$u_i^{n+1} = u_i^n - v(u_i^n - u_{i-1}^n) + \frac{v}{2}(v-1)(u_i^n - 2u_{i-1}^n + u_{i-2}^n) \quad (8)$$

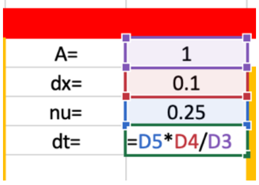
- Centred Difference:
  The centred difference approximation estimates the derivative of a function by evaluating points on both sides of a given point at which the derivative is being estimated.
  In context of time it, it estimates the derivative at a time t_n by considering the function values at t_n and t_{n-1}, it is given by:
  $$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^{n-1}}{2\Delta t} \quad (9)$$
  For a spatial derivative it is given by:
  $$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (10)$$
  Example Scheme: Crank-Nicolson Method:
  $$u_i^{n+1} = u_i^n - \frac{c\Delta t}{2\Delta x}(u_{i+1}^{n+1} - u_{i-1}^{n+1} + u_{i+1}^n - u_{i-1}^n) \quad (11)$$
  and Lax Scheme:
  $$u_i^{n+1} = \frac{u_{i+1}^n + u_{i-1}^n}{2} - \frac{v}{2}(u_{i+1}^n + u_{i-1}^n) \quad (12)$$
- u_i represents the function value at the i-th grid point.
- u_{i+1} represents the function value at the grid point immediately to the right of u_i
- u_{i-1} represents the function value at the grid point immediately to the left of u_i.
- Δx is the spacing between grid points in the spatial domain.
- Δt is the spacing between grid points in the time domain.
- v is the courant number.

After the finite difference scheme is solved, checks are carried out to check its consistency, convergency, stability and accuracy.
Consistency from

METHODOLOGY

The methodology used for this study was a Microsoft spreadsheet program that was used to analyse the different methods and compare their convergence and stability.
The following steps were followed to populate the spreadsheet.

1. First the constants were written out on different cells and the change in space calculated using the courant number formula $\Delta t = \frac{v\Delta x}{A}$ so we can see that the courant number influences the amount of time.



A=	1
dx=	0.1
nu=	0.25
dt=	=D5*D4/D3

Figure 1: Constants in excel.

2. The grid points are taken as 21 and a time of 15 steps is taken.

3. Then the boundary conditions of n = 1 at all i = 1 positions then the cells at i = 21 were equated to the cells at i = 20 which is the Neumann boundary condition (Mazumder, 2016).

i/n	0
1	1
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	=F23

Figure 2: Neumann Boundary conditions

4. Then the lax Scheme equation $u_i^{n+1} = \frac{u_{i+1}^n + u_{i-1}^n}{2} - \frac{v}{2}(u_{i+1}^n + u_{i-1}^n)$(13) as it is at n+1, n is assumed to be zero so all values are taken at n = 0 and the equation is taken as a formula and used to populate the individual cells in the table .

A=	1	i/n	0	1	
dx=	0.1	1	1	1	
nu=	0.25	2	=((F6+F4)/2)-(($D$5/2)*(F6-F4))		
dt=	0.025	3	0	0	0.3

Figure 3: Lax scheme in excel.

5. The courant number is increased to 0.5, 0.75 and 1 and the new values populated for the new tables.
6. The graphs for the various courant number at time steps of 1, 5 10, and 15 were plotted to visualize the data progression.
7. Then steps 4 to 6 were repeated for the Beam warming scheme:
$u_i^{n+1} = u_i^n - v(u_i^n - u_{i-1}^n) + \frac{v}{2}(v-1)(u_i^n - 2u_{i-1}^n + u_{i-2}^n)$ (14)

A=	1	x	i/n	0	1	2	3
dx=	0.1	0	1	1	1	1	
nu=	0.25	0.1	=(F65-($C$65*(F65-F64)))+(($C$65/2)*($C$65-1)*(F65-F64))				
dt=	0.025	0.2	3	0	-0.09375	-0.0048828	0.152

Figure 4: Grid point 2 for beam warming scheme.

Since the formula can't be used at i = 2 as the last value is at point i – 2
$\frac{v}{2}(v-1)(u_i^n - 2u_{i-1}^n + u_{i-2}^n)$ is replaced with $\frac{v}{2}(v-1)(u_i^n - u_{i-1}^n)$(15)

A=	1	x	i/n	0	1	2	3
dx=	0.1	0	1	1	1	1	1
nu=	0.25	0.1	2	0	0.34375	0.56933594	0.7173767
dt=	0.025	=(F66-($C$65*(F66-F65)))+(($C$65/2)*($C$65-1)*(F66-(2*F65)+(F64)))					

Figure 5: Grid point values for i>2

8. Steps 4 to 6 was repeated for the leapfrog scheme $u_i^{n+1} = u_i^n - v(u_{i+1}^n - u_{i-1}^n)$(16) the leapfrog scheme cannot be used to pass from n = 1 to n = 2 so the Forward in time backward in space (FTBS) approximation is used $u_i^{n+1} = u_i^n - v(u_i^n - u_{i-1}^n)$(17).

A=	1	x	i/n	0	1
dx=	0.1	0	1	1	1
nu=	0.25	0.1	2	=F129-($C$129*(F129-F128))	
dt=	0.025	0.2	3	0	0

Figure 6: Leapfrog scheme at n=1

A=	1	x	i/n	0	1	2
dx=	0.1	0	1	1	1	1
nu=	0.25	0.1	2	0	=F129-($C$129*(G130-G128))	
dt=	0.025	0.2	3	0	0	0.0625

Figure 7: Leapfrog scheme at n>=2

9. The exact solution according to (Khongar, 2018), the exact equation for a 1D linear advection equation is given by its initial value assuming the initial value to be $u_0 = u(x, 0)$, the exact solution will be $u(x, t) = u_0(x - ct)$.
10. Then the exact solution was plotted using MATLAB to derive the values of U_0 which is the initial point on the grid and these values were plotted against the time values gotten from the courant formula $\Delta t = \frac{v\Delta x}{A}$ at the different courant increments in step 4 above.

RESULTS

Lax Scheme

x	A=	1	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	dx=	0.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.1	nu=	0.25	2	0	0.625	0.625	0.77148438	0.77148438	0.84014893	0.84014893	0.88038206	0.88038206	0.90678506	0.90678506	0.92534966	0.92534966	0.93902448	0.93902448	0.94944085
0.2	dt=	0.025	3	0	0	0.390625	0.390625	0.57373047	0.57373047	0.68101883	0.68101883	0.75142682	0.75142682	0.80093243	0.80093243	0.83739862	0.83739862	0.8651756	0.8651756
0.3			4	0	0	0	0.24414063	0.24414063	0.415802	0.415802	0.53650141	0.53650141	0.62451139	0.62451139	0.69081355	0.69081355	0.74209413	0.74209413	0.78260223
0.4			5	0	0	0	0	0.15258789	0.15258789	0.29563904	0.29563904	0.41298568	0.41298568	0.50728209	0.50728209	0.58325332	0.58325332	0.64497995	0.64497995
0.5			6	0	0	0	0	0	0.09536743	0.09536743	0.20712614	0.20712614	0.31189993	0.31189993	0.40398627	0.40398627	0.48312297	0.48312297	0.55063646
0.6			7	0	0	0	0	0	0	0.05960464	0.05960464	0.14342368	0.14342368	0.23182656	0.23182656	0.31623904	0.31623904	0.39339732	0.39339732
0.7			8	0	0	0	0	0	0	0	0.0372529	0.0372529	0.09837095	0.09837095	0.16999365	0.16999365	0.24385457	0.24385457	0.31545979
0.8			9	0	0	0	0	0	0	0	0	0.02328306	0.02328306	0.06693881	0.06693881	0.12321379	0.12321379	0.18556392	0.18556392
0.9			10	0	0	0	0	0	0	0	0	0	0.01455192	0.01455192	0.04524736	0.04524736	0.08841283	0.08841283	0.13955942
1			11	0	0	0	0	0	0	0	0	0	0	0.00909495	0.00909495	0.03041123	0.03041123	0.06288525	0.06288525
1.1			12	0	0	0	0	0	0	0	0	0	0	0	0.00568434	0.00568434	0.02033929	0.02033929	0.04438255
1.2			13	0	0	0	0	0	0	0	0	0	0	0	0	0.00355271	0.00355271	0.01354472	0.01354472
1.3			14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00222045	0.00222045	0.00898587
1.4			15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00138778	0.00138778
1.5			16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00086736
1.6			17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.7			18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.8			19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.9			20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2			21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			t	0	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225	0.25	0.275	0.3	0.325	0.35	0.375

Figure 8: Values of the Lax Scheme at courant number of 0.25

The formula of lax scheme was used to populate all the grid positions as shown in figure 8 the real time values t is shown below, and the maximum is 0.375 due to the courant value of 0.25.

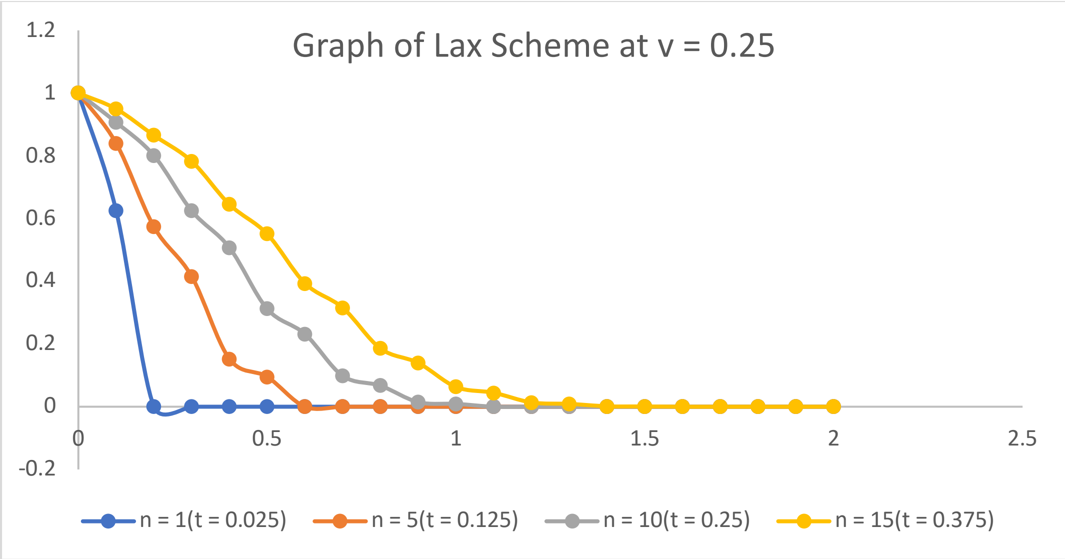


Figure 9: Graph of Lax Scheme @ courant No: 0.25

A graph of the values of the lax scheme at courant number of 0.25 is plotted as shown in figure 9 and it can be seen from the graph that at the courant number of 0.25 the graph seems clustered to the left due to the short time step that is at a result of the dependence of the total time on the change in time which is influenced by the courant number.

A staircase behavior can also be seen from the graph which indicates the presence of errors in the graph.

The lax scheme is unconditionally stable as the errors are not amplified during the iterative process and the errors are bounded.

A=	1	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.5	2	0	0.75	0.75	0.890625	0.890625	0.94335938	0.94335938	0.96807861	0.96807861	0.98105621	0.98105621	0.98835611	0.98835611	0.99265784	0.99265784	0.99527921
dt=	0.05	3	0	0	0.5625	0.5625	0.7734375	0.7734375	0.87231445	0.87231445	0.92422485	0.92422485	0.95342445	0.95342445	0.97063136	0.97063136	0.98111682	0.98111682
		4	0	0	0	0.421875	0.421875	0.65917969	0.65917969	0.79266357	0.79266357	0.87052917	0.87052917	0.9174571	0.9174571	0.94649376	0.94649376	0.96484331
		5	0	0	0	0	0.31640625	0.31640625	0.55371094	0.55371094	0.70944214	0.70944214	0.80955505	0.80955505	0.87408096	0.87408096	0.91602279	0.91602279
		6	0	0	0	0	0	0.23730469	0.23730469	0.45977783	0.45977783	0.62663269	0.62663269	0.74395251	0.74395251	0.82460989	0.82460989	0.87965855
		7	0	0	0	0	0	0	0.17797852	0.17797852	0.37820435	0.37820435	0.54714489	0.54714489	0.67619669	0.67619669	0.77056583	0.77056583
		8	0	0	0	0	0	0	0	0.13348389	0.13348389	0.30868149	0.30868149	0.47292924	0.47292924	0.60843363	0.60843363	0.71352653
		9	0	0	0	0	0	0	0	0	0.10011292	0.10011292	0.25028229	0.25028229	0.40514445	0.40514445	0.54240865	0.54240865
		10	0	0	0	0	0	0	0	0	0	0.07508469	0.07508469	0.20179009	0.20179009	0.34433368	0.34433368	0.47945312
		11	0	0	0	0	0	0	0	0	0	0	0.05631351	0.05631351	0.16190135	0.16190135	0.29058653	0.29058653
		12	0	0	0	0	0	0	0	0	0	0	0	0.04223514	0.04223514	0.1293451	0.1293451	0.24367694
		13	0	0	0	0	0	0	0	0	0	0	0	0	0.03167635	0.03167635	0.10294814	0.10294814
		14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02375726	0.02375726	0.0816656
		15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01781795	0.01781795
		16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01336346
		17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		t	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75

Figure 10: Values for Lax Scheme at courant number of 0.5

The lax scheme formula at courant number 0.5 was used to populate the grid points as shown in figure 10. The total physical time is 0.75 an increase is seen from the previous 0.375 in figure 8.

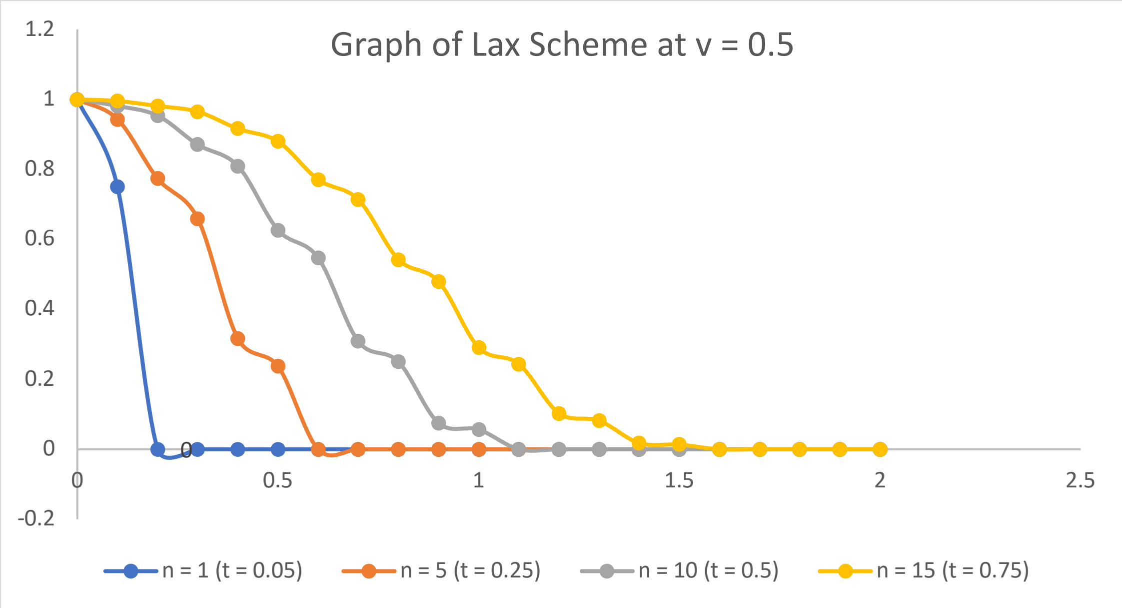


Figure 11: Graph of Lax scheme at Courant No: 0.5

The graph of the values from figure 10 was plotted as shown in figure 11, the graph spreads out a bit due to an increase in the total time because of the increase in courant number.
The staircase sequence is seen in the graph due to errors in the scheme.
The lax scheme at courant number 0.5 is stable as the errors are not amplified as the time increases it remains bounded within the boundary conditions.

A=	1	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.75	2	0	0.875	0.875	0.97070313	0.97070313	0.99163818	0.99163818	0.99736261	0.99736261	0.99911572	0.99911572	0.99969096	0.99969096	0.9998887	0.9998887	0.99995899
dt=	0.075	3	0	0	0.765625	0.765625	0.93310547	0.93310547	0.97890091	0.97890091	0.99292576	0.99292576	0.99752767	0.99752767	0.99910957	0.99910957	0.99967189	0.99967189
		4	0	0	0	0.66992188	0.66992188	0.88973999	0.88973999	0.96186781	0.96186781	0.9864113	0.9864113	0.99503988	0.99503988	0.99815425	0.99815425	0.99930232
		5	0	0	0	0	0.58618164	0.58618164	0.84263611	0.84263611	0.94081008	0.94081008	0.97762533	0.97762533	0.99146699	0.99146699	0.99671529	0.99671529
		6	0	0	0	0	0	0.51290894	0.51290894	0.79340601	0.79340601	0.91612348	0.91612348	0.96645682	0.96645682	0.98664258	0.98664258	0.99467904
		7	0	0	0	0	0	0	0.44879532	0.44879532	0.74331725	0.74331725	0.88827726	0.88827726	0.95287171	0.95287171	0.98042528	0.98042528
		8	0	0	0	0	0	0	0	0.3926959	0.3926959	0.69335371	0.69335371	0.85777594	0.85777594	0.93690414	0.93690414	0.97270291
		9	0	0	0	0	0	0	0	0	0.34360892	0.34360892	0.64426672	0.64426672	0.82513118	0.82513118	0.91864632	0.91864632
		10	0	0	0	0	0	0	0	0	0	0.3006578	0.3006578	0.59661782	0.59661782	0.79084159	0.79084159	0.89823789
		11	0	0	0	0	0	0	0	0	0	0	0.26307558	0.26307558	0.55081449	0.55081449	0.75537887	0.75537887
		12	0	0	0	0	0	0	0	0	0	0	0	0.23019113	0.23019113	0.50713983	0.50713983	0.71917868
		13	0	0	0	0	0	0	0	0	0	0	0	0	0.20141724	0.20141724	0.46577736	0.46577736
		14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.17624008	0.17624008	0.42683145
		15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15421007	0.15421007
		16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.13493381
		17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		t	0	0.075	0.15	0.225	0.3	0.375	0.45	0.525	0.6	0.675	0.75	0.825	0.9	0.975	1.05	1.125

Figure 12: Values of Lax Scheme at 0.75 Courant Number

The courant number was changed to 0.75 for the lax scheme and the new values were inputted into the grid points as shown in figure 12. The total physical time increases to 1.125.

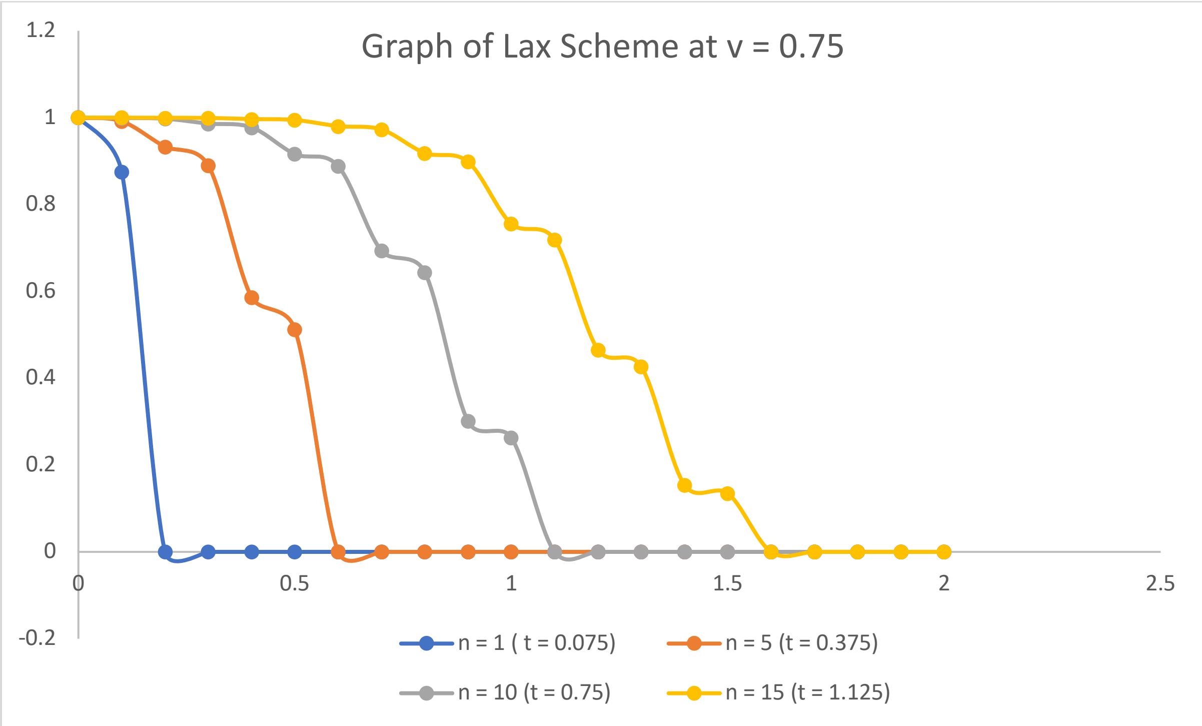


Figure 13: Graph of Lax Scheme at courant No: 0.75

The graph of the values from figure 12 were plotted as shown in figure 13, the graph was used to visualize the stability and shows that at the 0.75 courant number the scheme was stable as the errors are bounded as the time increases.
The staircase behavior is seen in the graph which is due to the errors present in the scheme.

A=	1	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	1	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
dt=	0.1	3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		5	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
		6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
		7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
		8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
		9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
		10	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
		11	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
		12	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
		13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
		14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
		15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
		16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5

Figure 14: Values of Lax Scheme at courant number 1.

The figure 14 shows the values of using the lax scheme to populate the grid at the courant number of 1 with the physical total time value of 1.5.

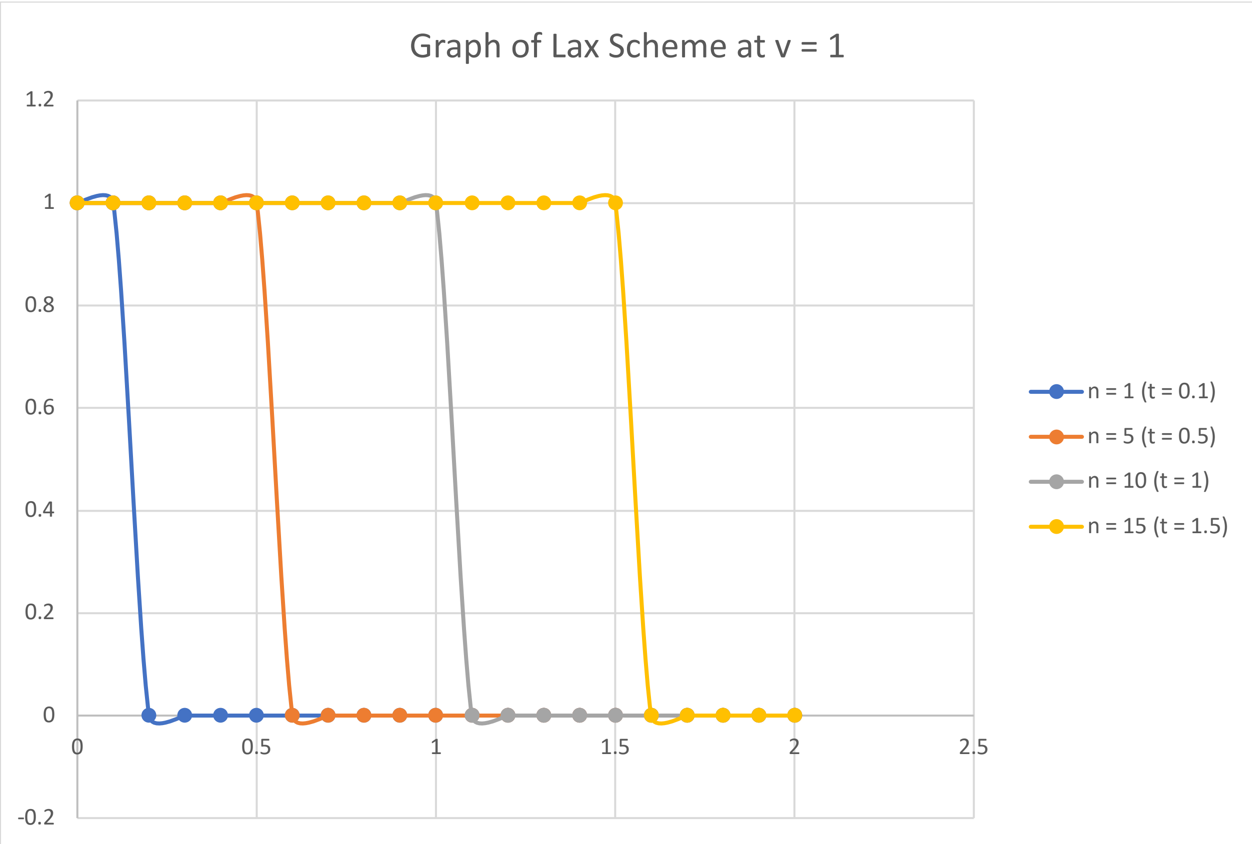


Figure 15: Graph of Lax Scheme at courant No: 1

The graph of the values in figure 7 was plotted as shown in figure 8 and it can be visualized that the scheme is stable as the errors do not increase in magnitude and can be seen that the graph is evenly spread out as the total time is higher and there is no staircase behavior so at the courant number of 1 the lax scheme is accurate and converges with the exact solution which is backed by the formula $E_T = \frac{(1-v)}{2!}\frac{\partial u}{\partial x}| + \cdots$ so when the courant number becomes 1 the error is zero.

Beam Warming Scheme

			t	0	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225	0.25	0.275	0.3	0.325	0.35	0.375
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.25	0.1	2	0	0.34375	0.56933594	0.71737671	0.81452847	0.87828431	0.92012408	0.94758142	0.96560031	0.9774252	0.98518529	0.99027785	0.99361984	0.99581302	0.99725229	0.99819682
dt=	0.025	0.2	3	0	-0.09375	-0.0048828	0.15213013	0.31993771	0.47256532	0.60062038	0.70296141	0.7821353	0.84197642	0.88642055	0.91898205	0.94257853	0.95952584	0.97160703	0.98016499
		0.3	4	0	0	-0.0732422	-0.1035767	-0.0686693	0.01854646	0.13657929	0.26613994	0.39336419	0.50980441	0.61129022	0.69660708	0.7663645	0.82215295	0.86597296	0.89988042
		0.4	5	0	0	0.00878906	-0.0258179	-0.07652	-0.1102532	-0.1085426	-0.0677858	0.00604916	0.10274141	0.21152819	0.32315292	0.43068013	0.52955157	0.61725458	0.69284833
		0.5	6	0	0	0	0.01071167	0.00544453	-0.0234668	-0.0653746	-0.1031938	-0.1223278	-0.114509	-0.0779913	-0.0159467	0.06560748	0.15963079	0.25935968	0.35906871
		0.6	7	0	0	0	-0.000824	0.00656605	0.0138647	0.00916824	-0.0124089	-0.0469357	-0.0848871	-0.1154368	-0.1297074	-0.1223927	-0.0919932	-0.0401775	0.02923573
		0.7	8	0	0	0	0	-0.0013647	0.00146663	0.00922829	0.01619604	0.01487419	0.00069507	-0.0259467	-0.0602195	-0.094771	-0.121891	-0.1352034	-0.1306199
		0.8	9	0	0	0	0	7.7248E-05	-0.0011619	-0.0014207	0.00224553	0.00972273	0.01728822	0.01960764	0.01233803	-0.0060891	-0.033984	-0.0670049	-0.0993568
		0.9	10	0	0	0	0	0	0.00016174	-0.0005397	-0.0018409	-0.001744	0.00171471	0.00862371	0.01667016	0.02198325	0.0206473	0.01010908	-0.0100053
		1	11	0	0	0	0	0	-7.242E-06	0.00017494	1.1873E-05	-0.0010081	-0.0023361	-0.0024036	0.00035727	0.00637096	0.01436947	0.02164916	0.0249117
		1.1	12	0	0	0	0	0	0	-1.833E-05	0.0001151	0.00025331	-0.0001113	-0.0012558	-0.0026842	-0.003168	-0.0013527	0.00346328	0.01079656
		1.2	13	0	0	0	0	0	0	6.7893E-07	-2.397E-05	3.3511E-05	0.00022733	0.00031949	-0.0001144	-0.0012829	-0.0028252	-0.003793	-0.0030036
		1.3	14	0	0	0	0	0	0	0	2.0156E-06	-1.996E-05	-2.218E-05	9.5332E-05	0.00032008	0.00041164	5.8592E-06	-0.0011054	-0.0027095
		1.4	15	0	0	0	0	0	0	0	-6.365E-08	3.0877E-06	-9.847E-06	-3.748E-05	-1.284E-05	0.00014233	0.00039377	0.00052584	0.00021707
		1.5	16	0	0	0	0	0	0	0	0	-2.168E-07	3.0796E-06	-2.072E-07	-2.547E-05	-5.234E-05	-1.067E-05	0.00016472	0.00044178
		1.6	17	0	0	0	0	0	0	0	0	5.9672E-09	-3.804E-07	2.0208E-06	4.7492E-06	-6.823E-06	-4.072E-05	-6.831E-05	-2.206E-05
		1.7	18	0	0	0	0	0	0	0	0	0	2.2936E-08	-4.401E-07	6.1471E-07	4.869E-06	5.1171E-06	-1.346E-05	-5.416E-05
		1.8	19	0	0	0	0	0	0	0	0	0	-5.594E-10	4.5331E-08	-3.522E-07	-4.075E-07	2.5025E-06	7.6985E-06	5.5685E-06
		1.9	20	0	0	0	0	0	0	0	0	0	0	-2.395E-09	5.9519E-08	-1.727E-07	-7.481E-07	1.2419E-07	4.7112E-06
		2	21	0	0	0	0	0	0	0	0	0	0	-2.395E-09	5.9519E-08	-1.727E-07	-7.481E-07	1.2419E-07	4.7112E-06

Figure 16: Values for Beam Warming Scheme at 0.25

The formula of beam warming scheme was used to populate the grid points as shown in figure 16.

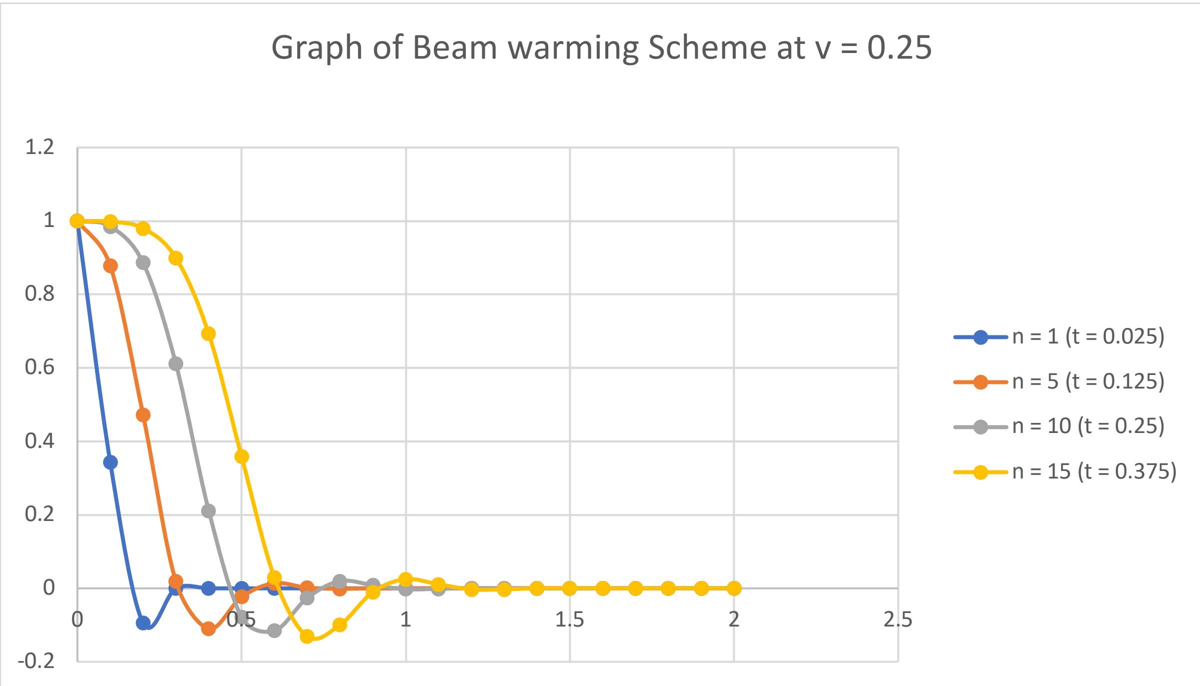


Figure 17: Graph of Beam warming Scheme at courant number: 0.25.

The graph of values from figure 16 was plotted as shown in figure 17 and can be visualized that the beam warming scheme is stable at the courant number of 0.25. It can be visualized that the errors do not increase over the iterative process.
The graphs can also be seen to cluster towards the left. And some oscillations is seen below the graphs due to some errors in the scheme.

			t	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.5	0.1	2	0	0.625	0.859375	0.94726563	0.98022461	0.99258423	0.99721909	0.99895716	0.99960893	0.99985335	0.99994501	0.99997938	0.99999227	0.9999971	0.99999891	0.99999959
dt=	0.05	0.2	3	0	-0.125	0.296875	0.63085938	0.82202148	0.91842651	0.96384811	0.98435736	0.99335188	0.99721365	0.99884513	0.99952568	0.99980666	0.9999217	0.99996846	0.99998736
		0.3	4	0	0	-0.171875	0.05078125	0.3737793	0.63415527	0.80255508	0.89919186	0.95059532	0.97653604	0.98912959	0.99506432	0.99779596	0.99902945	0.99957768	0.99981811
		0.4	5	0	0	0.015625	-0.1601563	-0.1008301	0.13977051	0.41322708	0.63639545	0.78999752	0.88502657	0.93963529	0.96935478	0.98486557	0.99269572	0.99654277	0.99839074
		0.5	6	0	0	0	0.03320313	-0.1140137	-0.1651001	-0.0363541	0.19596815	0.43838567	0.63806835	0.78097856	0.87395222	0.93036513	0.96281161	0.98069746	0.99022142
		0.6	7	0	0	0	-0.0019531	0.04418945	-0.0563354	-0.1624222	-0.1398273	0.01499146	0.23566136	0.45629595	0.63939049	0.77406625	0.8649405	0.92237443	0.95684566
		0.7	8	0	0	0	0	-0.0056152	0.04528809	-0.004631	-0.119009	-0.1739948	-0.1088027	0.05618647	0.26566957	0.46992493	0.6404759	0.76853238	0.85739328
		0.8	9	0	0	0	0	0.00024414	-0.0096436	0.03739166	0.03085136	-0.0602091	-0.1549485	-0.1691654	-0.0783341	0.08995306	0.28941781	0.48077104	0.64139162
		0.9	10	0	0	0	0	0	0.00088501	-0.0125618	0.02391195	0.04698163	-0.0057894	-0.104782	-0.1731906	-0.1569058	-0.0501155	0.11821056	0.30884069
		1	11	0	0	0	0	0	-3.052E-05	0.00185776	-0.0133986	0.00905305	0.04615726	0.03233551	-0.045315	-0.1370943	-0.1803338	-0.141389	-0.0244593
		1.1	12	0	0	0	0	0	0	-0.0001335	0.00291348	-0.0119454	-0.0035625	0.03400569	0.05010152	0.00645062	-0.0807885	-0.1592816	-0.1805487
		1.2	13	0	0	0	0	0	0	3.8147E-06	-0.0003309	0.00373584	-0.0086898	-0.0117002	0.01707477	0.04964356	0.04059109	-0.022828	-0.1103481
		1.3	14	0	0	0	0	0	0	0	1.955E-05	-0.000605	0.00406817	-0.0045465	-0.0147307	0.00101936	0.0368086	0.05434511	0.02316861
		1.4	15	0	0	0	0	0	0	0	-4.768E-07	5.585E-05	-0.0008998	0.00379991	-0.0005224	-0.0133783	-0.0104578	0.0186109	0.05059142
		1.5	16	0	0	0	0	0	0	0	0	-2.801E-06	0.00011647	-0.0011397	0.00299085	0.00257115	-0.009197	-0.0158933	0.00120506
		1.6	17	0	0	0	0	0	0	0	0	5.9605E-08	-9.06E-06	0.00019643	-0.0012561	0.00183739	0.00428967	-0.0039819	-0.0157395
		1.7	18	0	0	0	0	0	0	0	0	0	3.9488E-07	-2.121E-05	0.00028184	-0.0012103	0.0006028	0.00459292	0.0007226
		1.8	19	0	0	0	0	0	0	0	0	0	-7.451E-09	1.4259E-06	-3.992E-05	0.00035342	-0.0010048	-0.0004609	0.00376958
		1.9	20	0	0	0	0	0	0	0	0	0	0	-5.495E-08	3.6994E-06	-6.378E-05	0.00039243	-0.0006818	-0.0011755
		2	21	0	0	0	0	0	0	0	0	0	0	-5.495E-08	3.6994E-06	-6.378E-05	0.00039243	-0.0006818	-0.0011755

Figure 18: Values of Beam Warming Scheme at courant number 0.5

Figure 18 displays the values of the populated grid points using beam warming scheme and a courant number of 0.5.

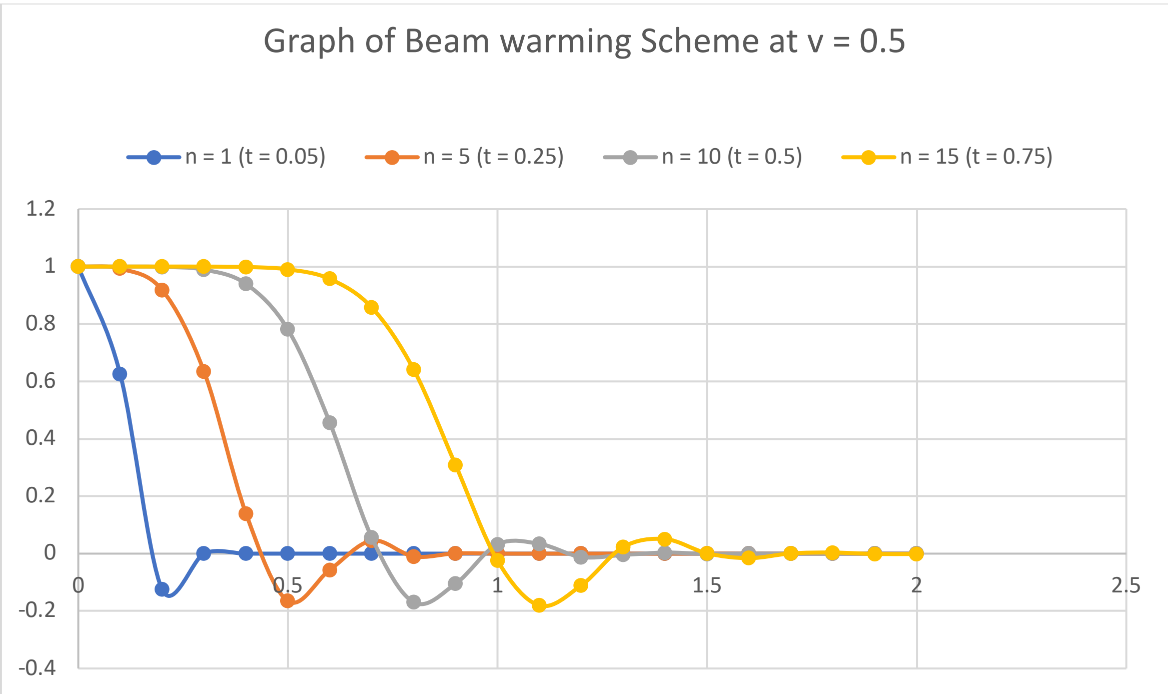


Figure 19: Graph of Beam Warming Scheme at courant number: 0.5.

The graph of the values of figure 11 was plotted in figure 12, from the graph it can be visualized that the beam warming scheme is stable at courant number 0.5 as the errors are bounded within the boundary conditions.

			t	0	0.075	0.15	0.225	0.3	0.375	0.45	0.525	0.6	0.675	0.75	0.825	0.9	0.975	1.05	1.125
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.75	0.1	2	0	0.84375	0.97558594	0.9961853	0.99940395	0.99990687	0.99998545	0.99999773	0.99999964	0.99999994	0.99999999	1	1	1	1	1
dt=	0.075	0.2	3	0	-0.09375	0.68261719	0.92752075	0.98509884	0.9971129	0.99946158	0.99990223	0.99998259	0.99999695	0.99999947	0.99999991	0.99999998	1	1	1
		0.3	4	0	0	-0.1669922	0.5223999	0.85778332	0.96386468	0.99165593	0.99819283	0.99962618	0.9999253	0.99998547	0.99999723	0.99999948	0.9999999	0.99999998	1
		0.4	5	0	0	0.00878906	-0.2191772	0.36854839	0.76940453	0.93036326	0.98134717	0.99540044	0.9989325	0.99976346	0.99994947	0.99998952	0.99999788	0.99999958	0.99999992
		0.5	6	0	0	0	0.02389526	-0.25072	0.22592193	0.66625473	0.88335012	0.96445585	0.99016919	0.99747016	0.99938432	0.99985669	0.99996783	0.99999299	0.99999851
		0.6	7	0	0	0	-0.000824	0.04282093	-0.2629107	0.09859034	0.552797	0.82251397	0.93937638	0.98141125	0.99474595	0.99860659	0.99964891	0.99991518	0.99998022
		0.7	8	0	0	0	0	-0.0030127	0.0631789	-0.2577872	-0.0103122	0.43382183	0.74847377	0.90478602	0.96793304	0.99012159	0.99716361	0.99923068	0.99980093
		0.8	9	0	0	0	0	7.7248E-05	-0.0068268	0.08281141	-0.2379791	-0.0986766	0.31417906	0.6627181	0.85977929	0.9485203	0.98282592	0.99469035	0.99845708
		0.9	10	0	0	0	0	0	0.00035486	-0.0122677	0.09988643	-0.2065314	-0.1654507	0.19852179	0.56749356	0.80397023	0.92203423	0.97198306	0.99071668
		1	11	0	0	0	0	0	-7.242E-06	0.00097155	-0.0191127	0.11296771	-0.166721	-0.2106144	0.09107585	0.46565151	0.73755636	0.88751035	0.95665539
		1.1	12	0	0	0	0	0	0	-4.006E-05	0.00205467	-0.0269615	0.12105681	-0.1218748	-0.2351054	-0.0045541	0.3604645	0.66134096	0.84425206
		1.2	13	0	0	0	0	0	0	6.7893E-07	-0.0001285	0.00369798	-0.0352893	0.12360691	-0.075199	-0.2406995	-0.0855336	0.25542493	0.5767132
		1.3	14	0	0	0	0	0	0	0	4.3919E-06	-0.0003124	0.00594568	-0.0435038	0.12050978	-0.0296283	-0.2298583	-0.1498967	0.1540388
		1.4	15	0	0	0	0	0	0	0	-6.365E-08	1.6157E-05	-0.0006371	0.0087829	-0.0510006	0.11205897	0.01229831	-0.2055517	-0.1965917
		1.5	16	0	0	0	0	0	0	0	0	-4.714E-07	4.4365E-05	-0.0011477	0.01213312	-0.0572151	0.09889308	0.04853092	-0.171069
		1.6	17	0	0	0	0	0	0	0	0	5.9672E-09	-1.956E-06	0.00010101	-0.0018836	0.01586179	-0.0616662	0.08192395	0.07756883
		1.7	18	0	0	0	0	0	0	0	0	0	4.9789E-08	-5.985E-06	0.00020136	-0.0028719	0.0197856	-0.0639918	0.06225521
		1.8	19	0	0	0	0	0	0	0	0	0	-5.594E-10	2.2994E-07	-1.504E-05	0.00036302	-0.0041227	0.02368604	-0.0639718
		1.9	20	0	0	0	0	0	0	0	0	0	0	-5.192E-09	7.7584E-07	-3.286E-05	0.00060443	-0.0056255	0.02732591
		2	21	0	0	0	0	0	0	0	0	0	0	-5.192E-09	7.7584E-07	-3.286E-05	0.00060443	-0.0056255	0.02732591

Figure 20: Values of Beam warming Scheme at courant number 0.75.

Figure 20 displays the populated grid points of the beam warming scheme at the courant number of 0.75.

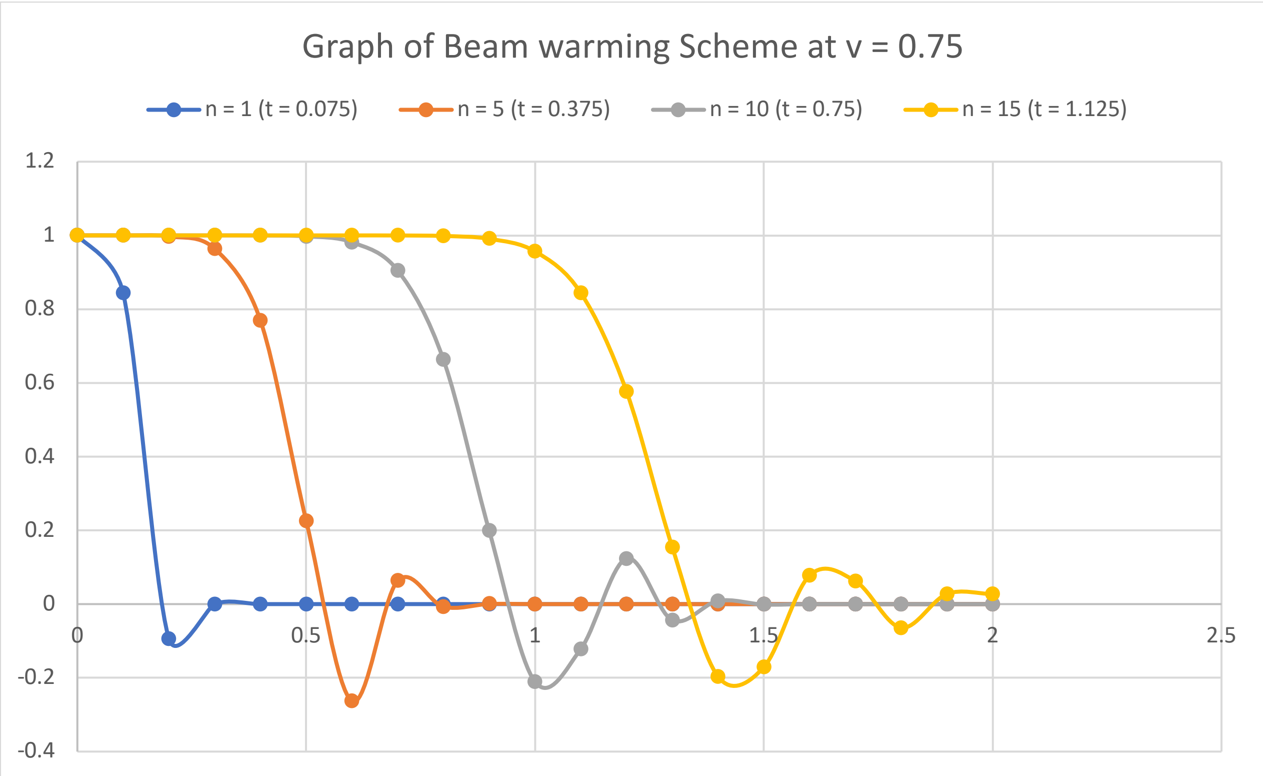


Figure 21: Beam Warming Scheme for courant number at 0.75.

Figure 21 displays the graph of values from figure 13 and from it we can visualize that at the courant number of 0.75 the beam warming scheme is stable at all time steps as the errors do not increase over the iterative process.

			t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	1	0.1	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
dt=	0.1	0.2	3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0.3	4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		0.4	5	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
		0.5	6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
		0.6	7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
		0.7	8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
		0.8	9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
		0.9	10	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
		1	11	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
		1.1	12	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
		1.2	13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
		1.3	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
		1.4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
		1.5	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1.6	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.7	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.8	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.9	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 22: Values of Beam Warming Scheme at Courant Number of 1

Figure 22 displays the grid points populated using the beam warming scheme at the courant values of 1.

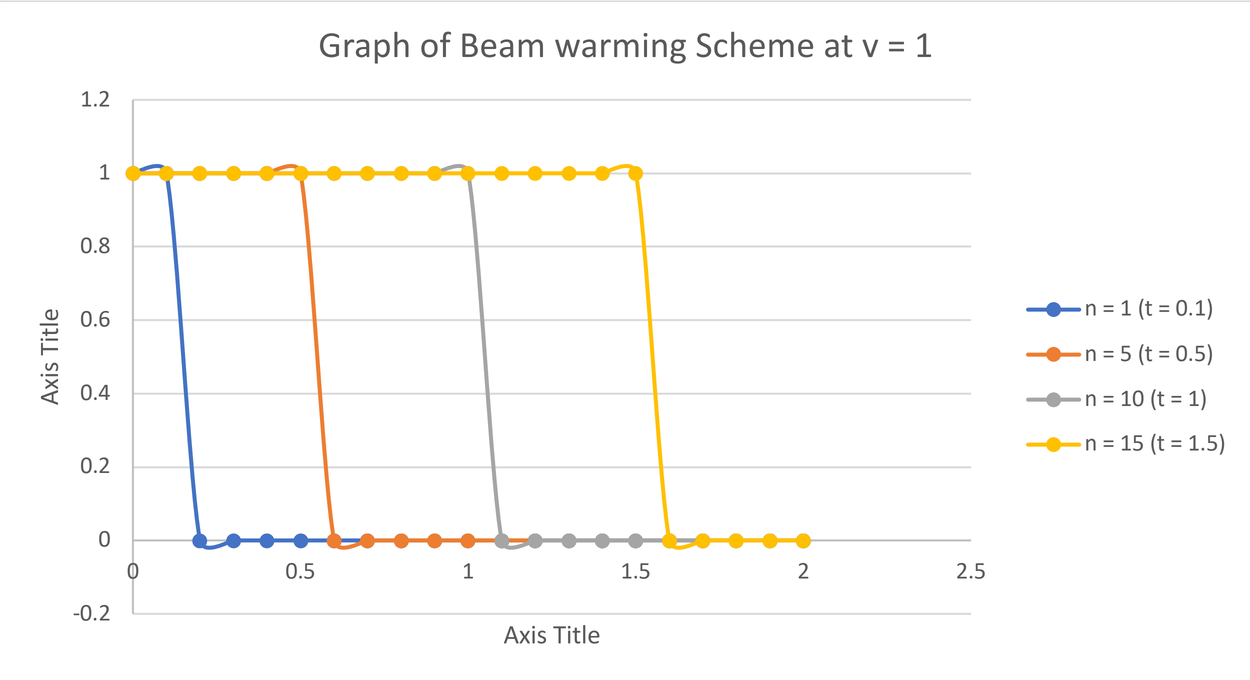


Figure 23: Beam Warming Scheme at courant number 1.

The figure 16 displays a graph of the values from figure 15, from the figure 16 it is seen that the beam warming scheme is stable at all time steps for the courant number of 1.

Leapfrog Scheme

			t	0	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225	0.25	0.275	0.3	0.325	0.35	0.375
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.25	0.1	2	0	0.25	0.25	0.484375	0.484375	0.68945313	0.68945313	0.85516357	0.85516357	0.97612762	0.97612762	1.05203772	1.05203772	1.08729178	1.08729178	1.08997484
dt=	0.025	0.2	3	0	0	0.0625	0.0625	0.1796875	0.1796875	0.3371582	0.3371582	0.5161438	0.5161438	0.69635963	0.69635963	0.85898376	0.85898376	0.98926776	0.98926776
		0.3	4	0	0	0	0.015625	0.015625	0.05957031	0.05957031	0.13922119	0.13922119	0.25526428	0.25526428	0.40154123	0.40154123	0.56615578	0.56615578	0.7336073
		0.4	5	0	0	0	0	0.00390625	0.00390625	0.01855469	0.01855469	0.05197144	0.05197144	0.11125183	0.11125183	0.20052558	0.20052558	0.31946166	0.31946166
		0.5	6	0	0	0	0	0	0.00097656	0.00097656	0.0055542	0.0055542	0.0181427	0.0181427	0.04444623	0.04444623	0.09041147	0.09041147	0.1608321
		0.6	7	0	0	0	0	0	0	0.00024414	0.00024414	0.00161743	0.00161743	0.00603771	0.00603771	0.01666462	0.01666462	0.03777914	0.03777914
		0.7	8	0	0	0	0	0	0	0	6.1035E-05	6.1035E-05	0.00046158	0.00046158	0.00193858	0.00193858	0.0059534	0.0059534	0.01488632
		0.8	9	0	0	0	0	0	0	0	0	1.5259E-05	1.5259E-05	0.0001297	0.0001297	0.00060534	0.00060534	0.00204748	0.00204748
		0.9	10	0	0	0	0	0	0	0	0	0	3.8147E-06	3.8147E-06	3.6001E-05	3.6001E-05	0.00018486	0.00018486	0.00068287
		1	11	0	0	0	0	0	0	0	0	0	0	9.5367E-07	9.5367E-07	9.8944E-06	9.8944E-06	5.5436E-05	5.5436E-05
		1.1	12	0	0	0	0	0	0	0	0	0	0	0	2.3842E-07	2.3842E-07	2.6971E-06	2.6971E-06	1.6374E-05
		1.2	13	0	0	0	0	0	0	0	0	0	0	0	0	5.9605E-08	5.9605E-08	7.3016E-07	7.3016E-07
		1.3	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1.4901E-08	1.4901E-08	1.9651E-07
		1.4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.7253E-09	3.7253E-09
		1.5	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.3132E-10
		1.6	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.7	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.8	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.9	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 24: Values of Leapfrog Scheme for courant number 0.25

Figure 24 shows the populated grid points for the leapfrog scheme at the courant number of 0.25.

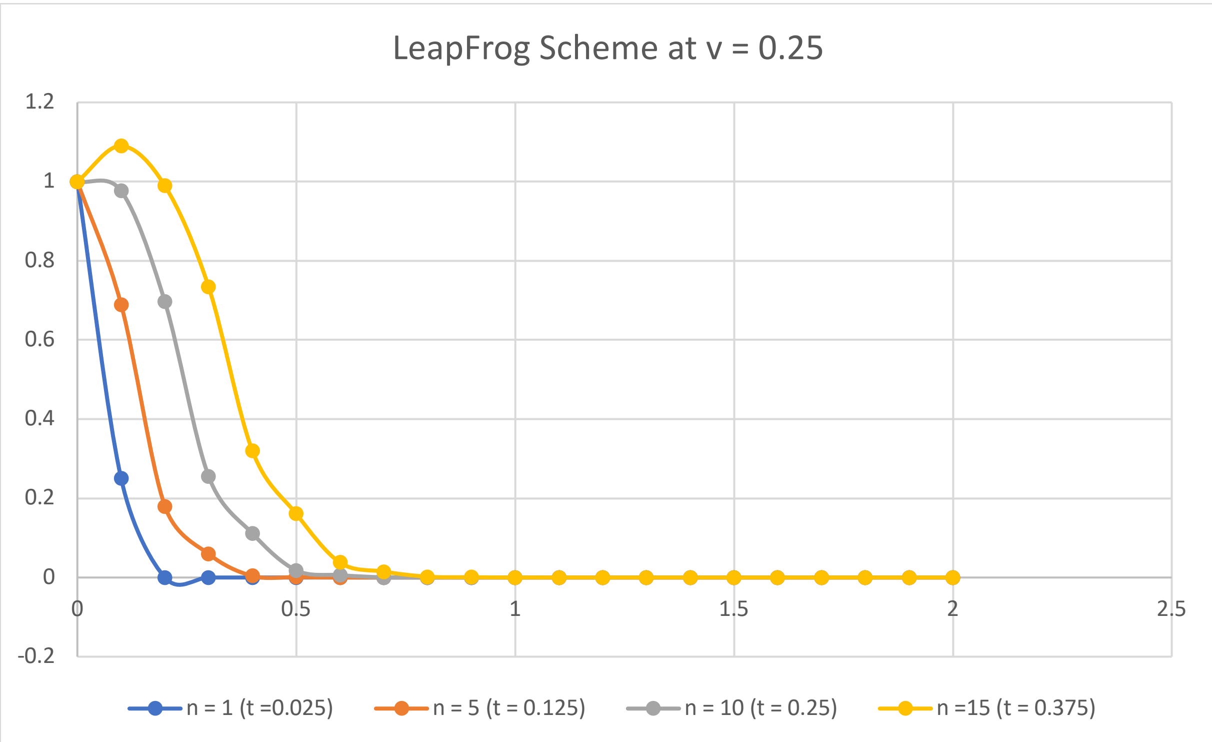


Figure 25: Leapfrog Scheme for courant Number 0.25

The graph of the values from figure 24 is plotted as seen in figure 25, it can be seen from figure 25 that the leapfrog scheme is stable for all time steps before that of 0.25, but at the timestep of 0.375 the errors are above the boundary conditions making it unstable. Therefore, for the courant number of 0.25 the leapfrog is conditionally stable and the condition is if the timestep is less than 0.25, from 0.375 the scheme is unstable.

			t	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.5	0.1	2	0	0.5	0.5	0.875	0.875	1.0625	1.0625	1.0859375	1.0859375	1.02734375	1.02734375	0.97167969	0.97167969	0.96142578	0.96142578	0.98788452
dt=	0.05	0.2	3	0	0	0.25	0.25	0.625	0.625	0.953125	0.953125	1.1171875	1.1171875	1.11132813	1.11132813	1.02050781	1.02050781	0.94708252	0.94708252
		0.3	4	0	0	0	0.125	0.125	0.40625	0.40625	0.7578125	0.7578125	1.0390625	1.0390625	1.15332031	1.15332031	1.10827637	1.10827637	0.99758911
		0.4	5	0	0	0	0	0.0625	0.0625	0.25	0.25	0.5546875	0.5546875	0.8828125	0.8828125	1.1105957	1.1105957	1.16845703	1.16845703
		0.5	6	0	0	0	0	0	0.03125	0.03125	0.1484375	0.1484375	0.3828125	0.3828125	0.69775391	0.69775391	0.99255371	0.99255371	1.16110229
		0.6	7	0	0	0	0	0	0	0.015625	0.015625	0.0859375	0.0859375	0.25292969	0.25292969	0.52099609	0.52099609	0.83135986	0.83135986
		0.7	8	0	0	0	0	0	0	0	0.0078125	0.0078125	0.04882813	0.04882813	0.16162109	0.16162109	0.37182617	0.37182617	0.65957642
		0.8	9	0	0	0	0	0	0	0	0	0.00390625	0.00390625	0.02734375	0.02734375	0.10058594	0.10058594	0.25585938	0.25585938
		0.9	10	0	0	0	0	0	0	0	0	0	0.00195313	0.00195313	0.01513672	0.01513672	0.0612793	0.0612793	0.17086792
		1	11	0	0	0	0	0	0	0	0	0	0	0.00097656	0.00097656	0.00830078	0.00830078	0.03668213	0.03668213
		1.1	12	0	0	0	0	0	0	0	0	0	0	0	0.00048828	0.00048828	0.0045166	0.0045166	0.02163696
		1.2	13	0	0	0	0	0	0	0	0	0	0	0	0	0.00024414	0.00024414	0.00244141	0.00244141
		1.3	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00012207	0.00012207	0.00131226
		1.4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.1035E-05	6.1035E-05
		1.5	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.0518E-05
		1.6	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.7	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.8	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.9	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 26: Values of Leapfrog Scheme at 0.5

Figure 26 shows the populated grid points for the leapfrog scheme at the courant number of 0.5.

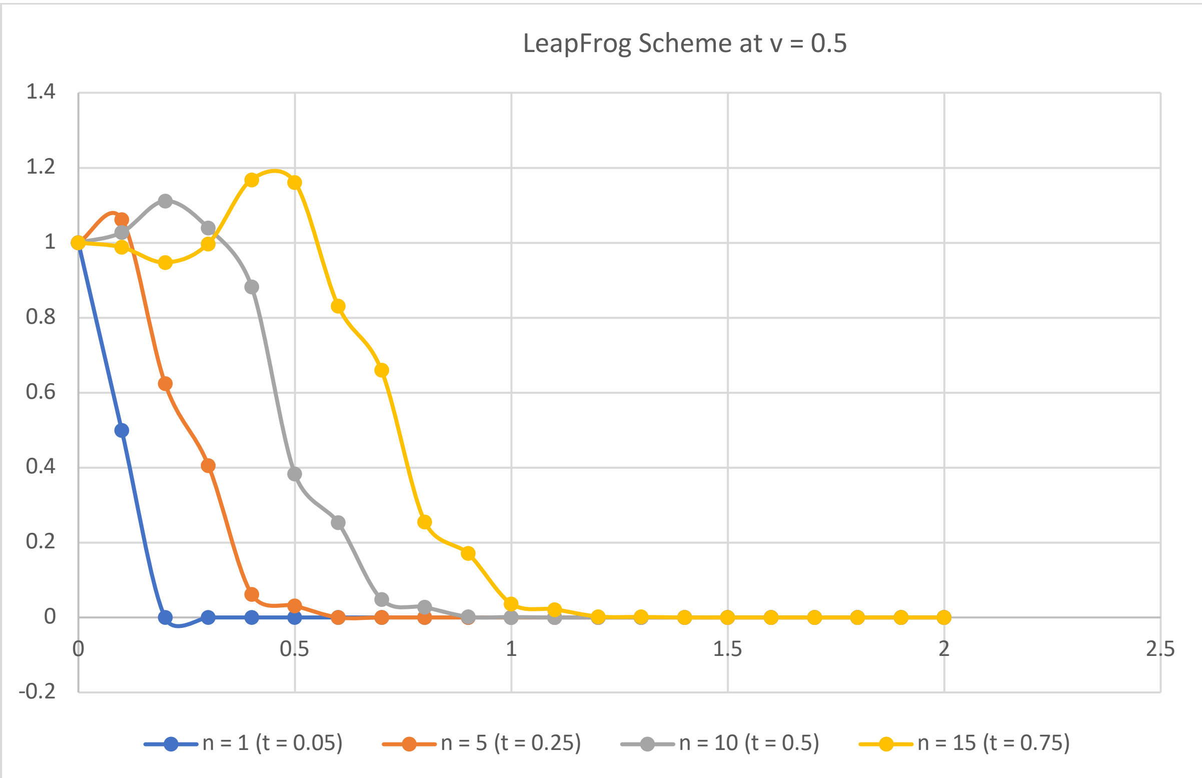


Figure 27: Leapfrog Scheme at courant number of 0.5

Figure 27 displays the graph of the values from figure 19, and it is seen from the graph that at timestep of 0.05 the system is stable, but from the time step of 0.25 the errors exceed the boundary and increase over the iterative process, so the system is unstable for any time step above 0.05.

			t	0	0.075	0.15	0.225	0.3	0.375	0.45	0.525	0.6	0.675	0.75	0.825	0.9	0.975	1.05	1.125
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	0.75	0.1	2	0	0.75	0.75	1.078125	1.078125	1.03710938	1.03710938	0.96148682	0.96148682	0.99112701	0.99112701	1.02338076	1.02338076	1.00010794	1.00010794	0.98467665
dt=	0.075	0.2	3	0	0	0.5625	0.5625	1.0546875	1.0546875	1.10083008	1.10083008	0.96047974	0.96047974	0.95699501	0.95699501	1.03103042	1.03103042	1.02057507	1.02057507
		0.3	4	0	0	0	0.421875	0.421875	0.97558594	0.97558594	1.14862061	1.14862061	0.99577332	0.99577332	0.92466688	0.92466688	1.01404841	1.01404841	1.04540795
		0.4	5	0	0	0	0	0.31640625	0.31640625	0.87011719	0.87011719	1.16427612	1.16427612	1.05180359	1.05180359	0.91185504	0.91185504	0.97876234	0.97876234
		0.5	6	0	0	0	0	0	0.23730469	0.23730469	0.75640869	0.75640869	1.14573669	1.14573669	1.11126494	1.11126494	0.92483868	0.92483868	0.938084
		0.6	7	0	0	0	0	0	0	0.17797852	0.17797852	0.64517212	0.64517212	1.09776592	1.09776592	1.1604234	1.1604234	0.96110191	0.96110191
		0.7	8	0	0	0	0	0	0	0	0.13348389	0.13348389	0.54227829	0.54227829	1.02772164	1.02772164	1.19060066	1.19060066	1.01305056
		0.8	9	0	0	0	0	0	0	0	0	0.10011292	0.10011292	0.45050812	0.45050812	0.94325137	0.94325137	1.19783539	1.19783539
		0.9	10	0	0	0	0	0	0	0	0	0	0.07508469	0.07508469	0.37073064	0.37073064	0.85115531	0.85115531	1.18183223
		1	11	0	0	0	0	0	0	0	0	0	0	0.05631351	0.05631351	0.30268514	0.30268514	0.75693283	0.75693283
		1.1	12	0	0	0	0	0	0	0	0	0	0	0	0.04223514	0.04223514	0.24549173	0.24549173	0.66470845
		1.2	13	0	0	0	0	0	0	0	0	0	0	0	0	0.03167635	0.03167635	0.1979772	0.1979772
		1.3	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02375726	0.02375726	0.1588767
		1.4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01781795	0.01781795
		1.5	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01336346
		1.6	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.7	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.8	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.9	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 28: Values of the Leapfrog Scheme at courant number of 0.75

Figure 28 shows the populated grid points using the leapfrog scheme at a courant number of 0.75.

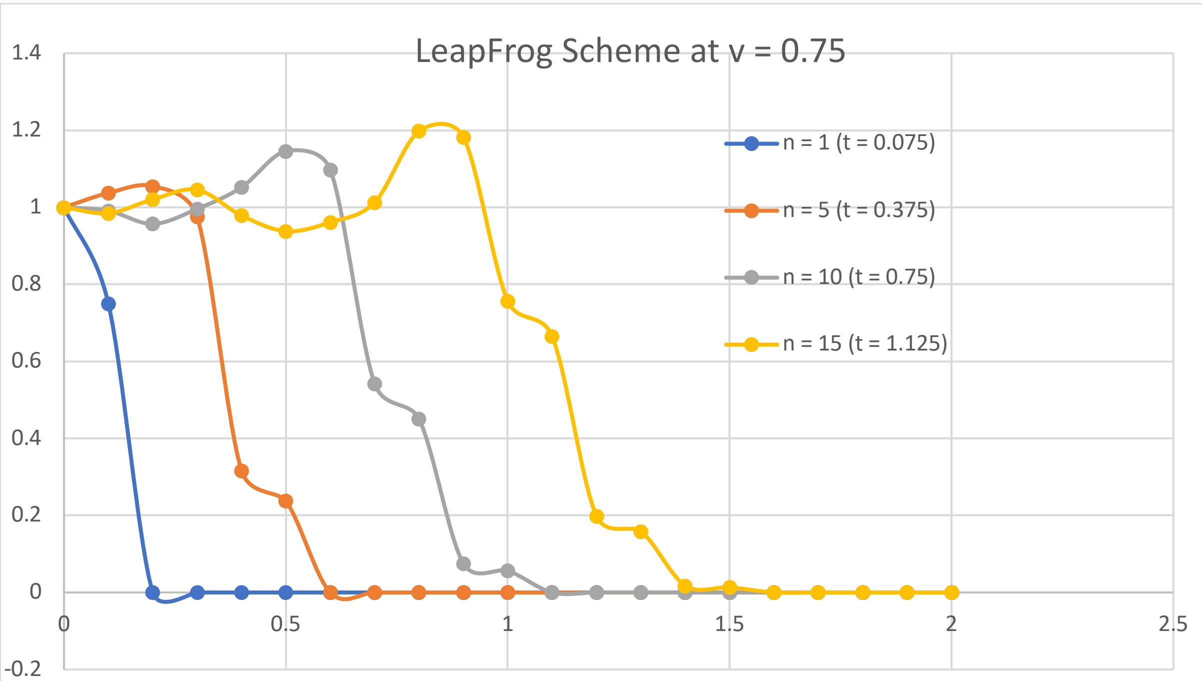


Figure 29: Leapfrog Scheme at courant number of 0.75

Figure 29 shows the graph of the values from figure 21, this shows the scheme at courant number of 0.75 is unstable from timestep 0.075 as the errors exceed the boundary conditions and they increase over the iterative process at timestep 1.125 the error grows past 1.2.

			t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
A=	1	x	i/n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
dx=	0.1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
nu=	1	0.1	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
dt=	0.1	0.2	3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0.3	4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		0.4	5	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
		0.5	6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
		0.6	7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
		0.7	8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
		0.8	9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
		0.9	10	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
		1	11	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
		1.1	12	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
6		1.2	13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
		1.3	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
		1.4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
		1.5	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		1.6	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.7	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.8	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1.9	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 30: Values of Leapfrog Scheme at courant number of 1

Figure 30 shows the populated grid points using the leapfrog scheme at a courant number of 1.

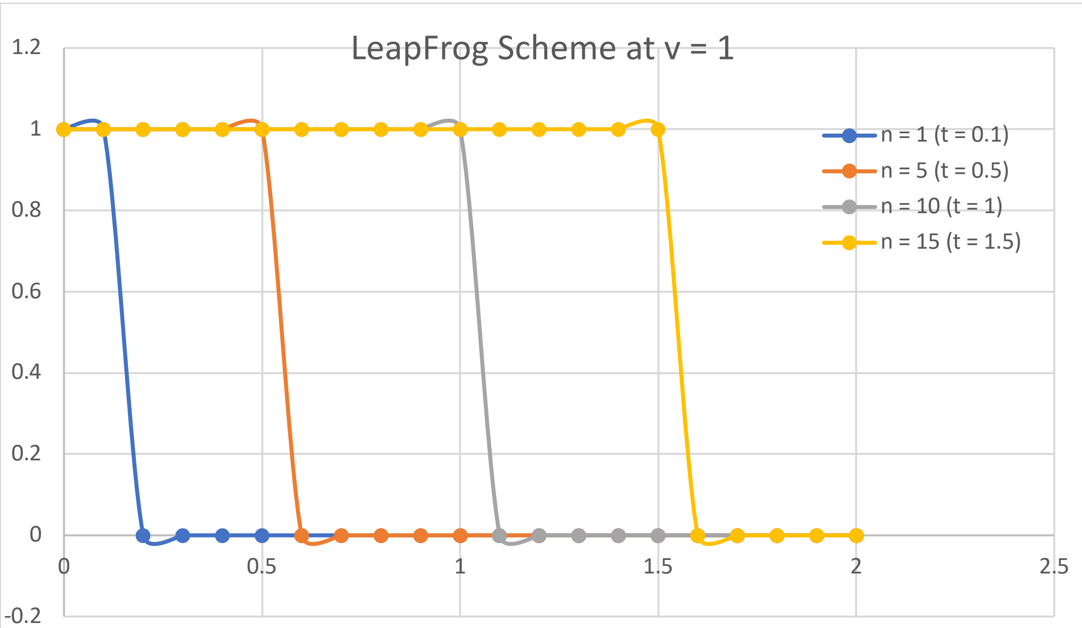


Figure 31: Leapfrog scheme at courant number of 1

Figure 24 shows a graph of values gotten from figure 23 and it shows that the leapfrog scheme is unconditionally stable at all timesteps at courant number 1.

Exact Solution

```
% Defining of the Constants
A = 1; % A is defined as Advection coefficient
g_p = 21; % Number of grid points represent by g_p
x = linspace(0, 2, g_p); % Discretization of Spatial domain
% Initialize the plot using figure command
figure;
% Solve and plot the solution for individual time step
for t = 0:0.01:1.5
    U = zeros(1, g_p);
    for i = 1:g_p
        if x(i)-A*t <= x(1)
            U(i) = 1; % Initial condition at x = 0
        end
    end

    plot(x, U, 'LineWidth', 1.5);
    ylim([-0.5, 1.5]); % Set y-axis limits
    title(['Linear Advection Equation Solution at t = ' num2str(t)]);
    xlabel('x');
    ylabel('U(x, t)');
    grid on;
    pause(0.5); % Pause for visualization
end
U = U';
```

Figure 32: MATLAB program for the Exact solution

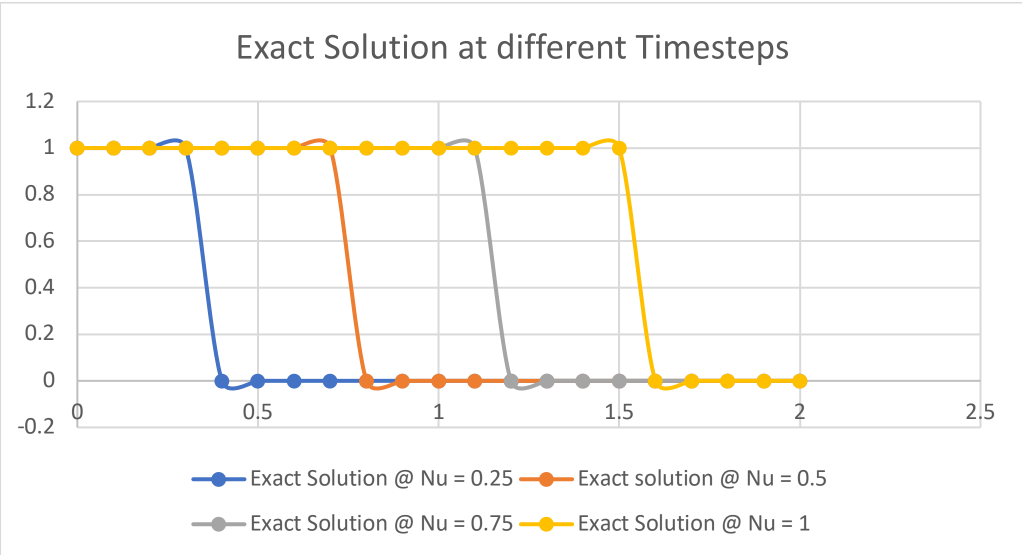


Figure 33: Graph of Exact solution at different Timesteps

Figure 25 shows the exact solution of the linear advection equation at different time steps.

DISCUSSION

It can be observed for all the graphs above at lower courant numbers the graphs at different time steps cluster to the left this is due to the dependence on the total time and this is influenced by the courant number as it affects the time step so at 0.25 the total time was 0.375, at 0.5 the total time increases to 0.75 which in turns increases the various time steps, at 0.75 courant number the total time increases to 1.125, and for courant number of 1 the physical total time is 1.5secs so at higher courant number we have higher time steps thereby making the graphs no longer clustered to the left.
The Lax scheme is also observed to have staircase behavior due to errors, but it remains unconditionally stable for every courant number investigated.

The beam warming scheme is seen to be stable but there is presence of oscillations on the lower side of the graph due to errors in the scheme.
The leapfrog scheme is conditionally stable but there is presence of errors which can be visualized by the oscillations at the top part of the graph.

Convergence Check

Nu = 0.25				
LEAP FROG	BW	LAX SCHEME	Exact solutio	x
1	1	1	1	0
1.089975	0.998197	0.949441	1	0.1
0.989268	0.980165	0.865176	1	0.2
0.733607	0.89988	0.782602	1	0.3
0.319462	0.692848	0.64498	0	0.4
0.160832	0.359069	0.550636	0	0.5
0.037779	0.029236	0.393397	0	0.6
0.014886	-0.13062	0.31546	0	0.7
0.002047	-0.099357	0.185564	0	0.8
0.000683	-0.010005	0.139559	0	0.9
5.54E-05	0.024912	0.062885	0	1
1.64E-05	0.010797	0.044383	0	1.1
7.3E-07	-0.003004	0.013545	0	1.2
1.97E-07	-0.00271	0.008986	0	1.3
3.73E-09	0.000217	0.001388	0	1.4
9.31E-10	0.000442	0.000867	0	1.5
0	-2.21E-05	0	0	1.6
0	-5.42E-05	0	0	1.7
0	5.57E-06	0	0	1.8
0	4.71E-06	0	0	1.9
0	4.71E-06	0	0	2

Nu = 0.5				
Leapfrog	BW	Lax Scheme	Exact solutio	x
1	1	1	1	0
0.987885	1	0.995279	1	0.1
0.947083	0.999987	0.981117	1	0.2
0.997589	0.999818	0.964843	1	0.3
1.168457	0.998391	0.916023	1	0.4
1.161102	0.990221	0.879659	1	0.5
0.83136	0.956846	0.770566	1	0.6
0.659576	0.857393	0.713527	1	0.7
0.255859	0.641392	0.542409	0	0.8
0.170868	0.308841	0.479453	0	0.9
0.036682	-0.024459	0.290587	0	1
0.021637	-0.180549	0.243677	0	1.1
0.002441	-0.110348	0.102948	0	1.2
0.001312	0.023169	0.081666	0	1.3
6.1E-05	0.050591	0.017818	0	1.4
3.05E-05	0.001205	0.013363	0	1.5
0	-0.01574	0	0	1.6
0	0.000723	0	0	1.7
0	0.00377	0	0	1.8
0	-0.001175	0	0	1.9
0	-0.001175	0	0	2

Nu = 0.75				
Leapfrog	BW	Lax Scheme	Exact solutio	x
1	1	1	1	0
0.984677	1	0.999959	1	0.1
1.020575	1	0.999672	1	0.2
1.045408	1	0.999302	1	0.3
0.978762	1	0.996715	1	0.4
0.938084	0.999999	0.994679	1	0.5
0.961102	0.99998	0.980425	1	0.6
1.013051	0.999801	0.972703	1	0.7
1.197835	0.998457	0.918646	1	0.8
1.181832	0.990717	0.898238	1	0.9
0.756933	0.956655	0.755379	1	1
0.664708	0.844252	0.719179	1	1.1
0.197977	0.576713	0.465777	0	1.2
0.158877	0.154039	0.426831	0	1.3
0.017818	-0.196592	0.15421	0	1.4
0.013363	-0.171069	0.134934	0	1.5
0	0.077569	0	0	1.6
0	0.062255	0	0	1.7
0	-0.063972	0	0	1.8
0	0.027326	0	0	1.9
0	0.027326	0	0	2

Nu = 1				
LeapFrog	BW	Lax Scheme	Exact solutio	x
1	1	1	1	0
1	1	1	1	0.1
1	1	1	1	0.2
1	1	1	1	0.3
1	1	1	1	0.4
1	1	1	1	0.5
1	1	1	1	0.6
1	1	1	1	0.7
1	1	1	1	0.8
1	1	1	1	0.9
1	1	1	1	1
1	1	1	1	1.1
1	1	1	1	1.2
1	1	1	1	1.3
1	1	1	1	1.4
1	1	1	1	1.5
0	0	0	0	1.6
0	0	0	0	1.7
0	0	0	0	1.8
0	0	0	0	1.9
0	0	0	0	2

Figure 34: Spreadsheet showing values of different schemes and exact solution at different courant numbers.

Figure 26 shows the values for different schemes at the final timestep of different courant numbers.

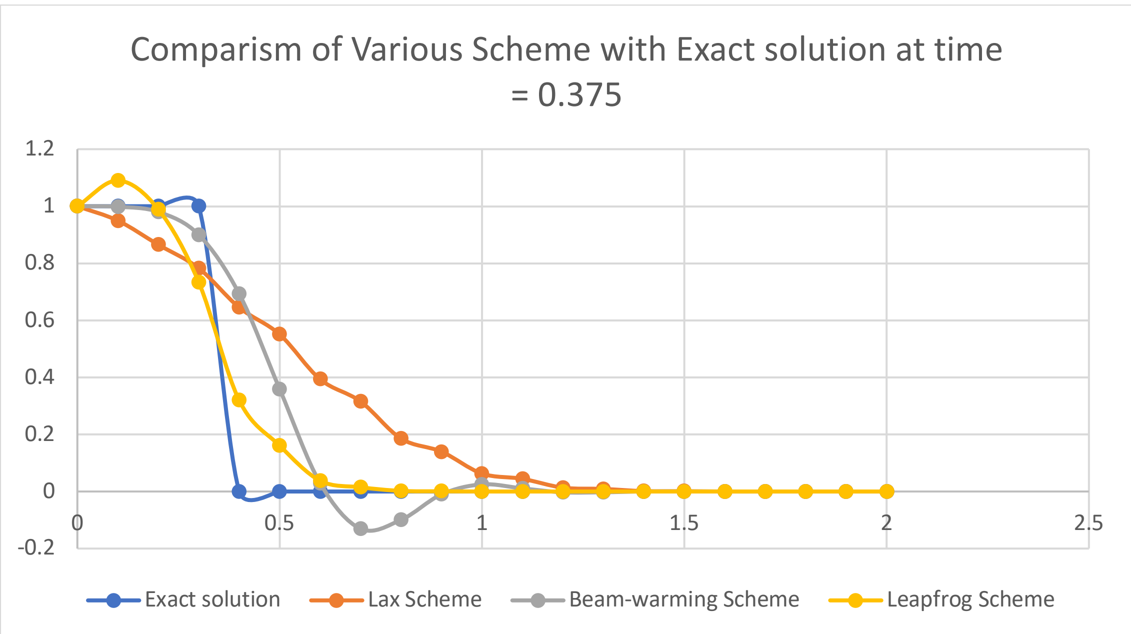


Figure 35: Graph comparing various schemes at v=0.25 to the exact solution

Figure 27 displays the graph of the various schemes and exact solution at timestep 0.375 the schemes do not converge with the exact solution at courant number 0.25. And the leap frog scheme is observed to be the closest to the exact solution in terms of accuracy while the lax scheme is the most stable it is the one that is farthest from the accuracy.

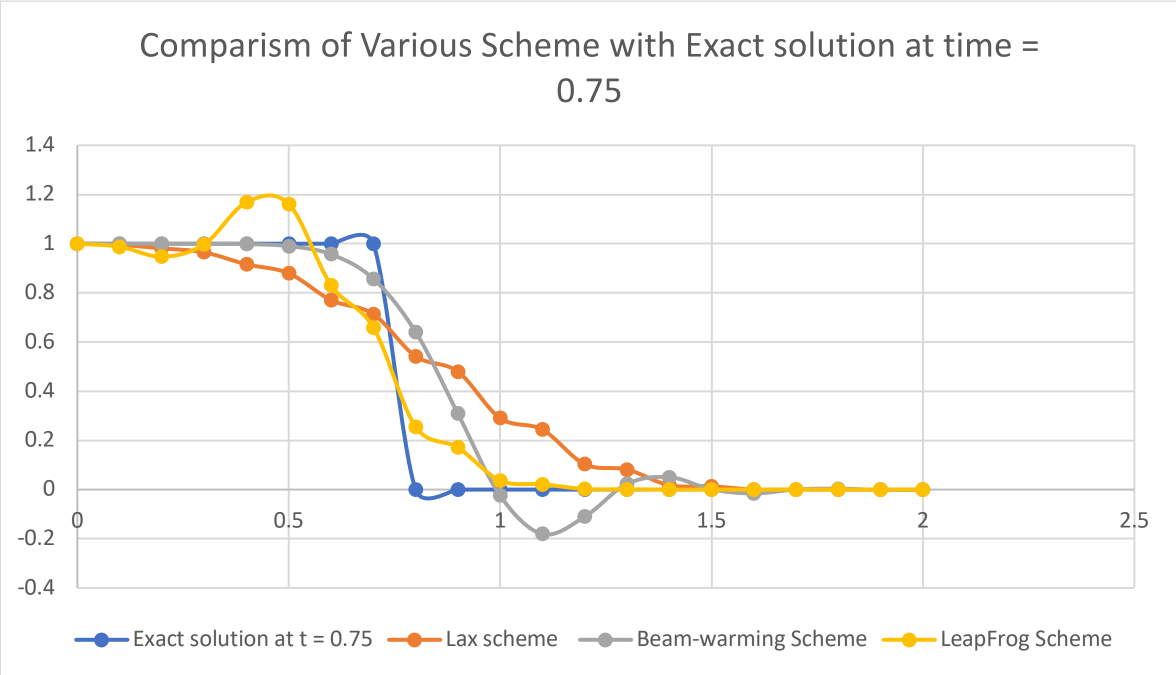


Figure 36: Graph comparing various schemes and exact solution at timestep: 0.75.

Figure 28 shows the three schemes compared to exact solution; the schemes are not converging with the exact solution, but the lax scheme is the least accurate as can be seen from the graph.

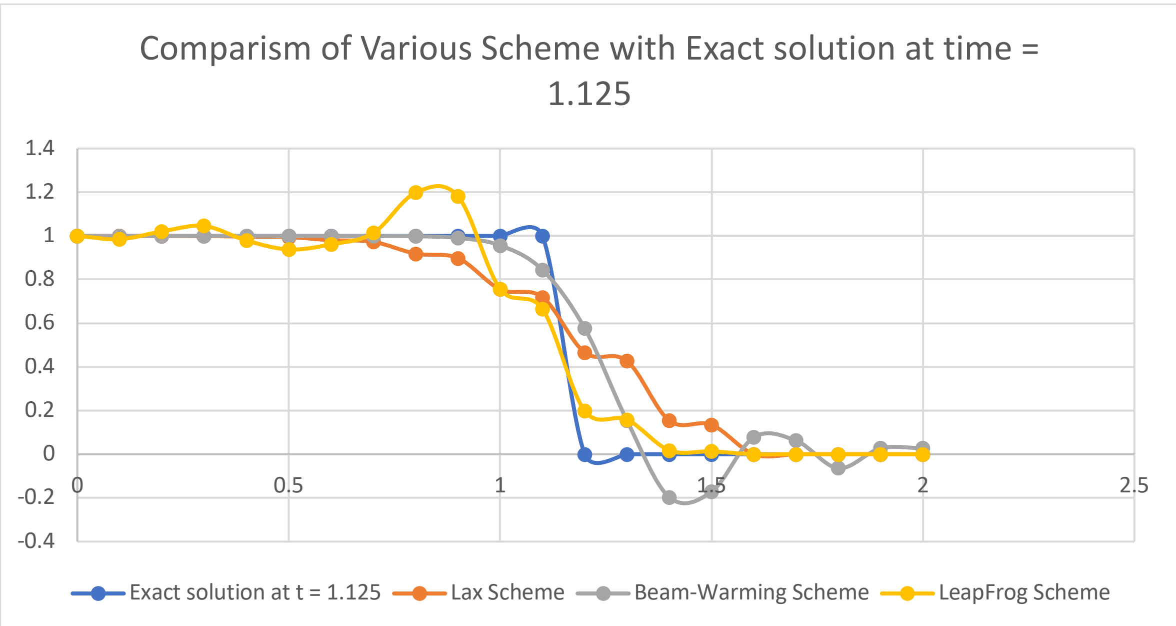


Figure 37: Graph comparing different schemes to exact solution at time step 1.125.

From figure 29 it can be visualized that the schemes are not converging with the exact solution.

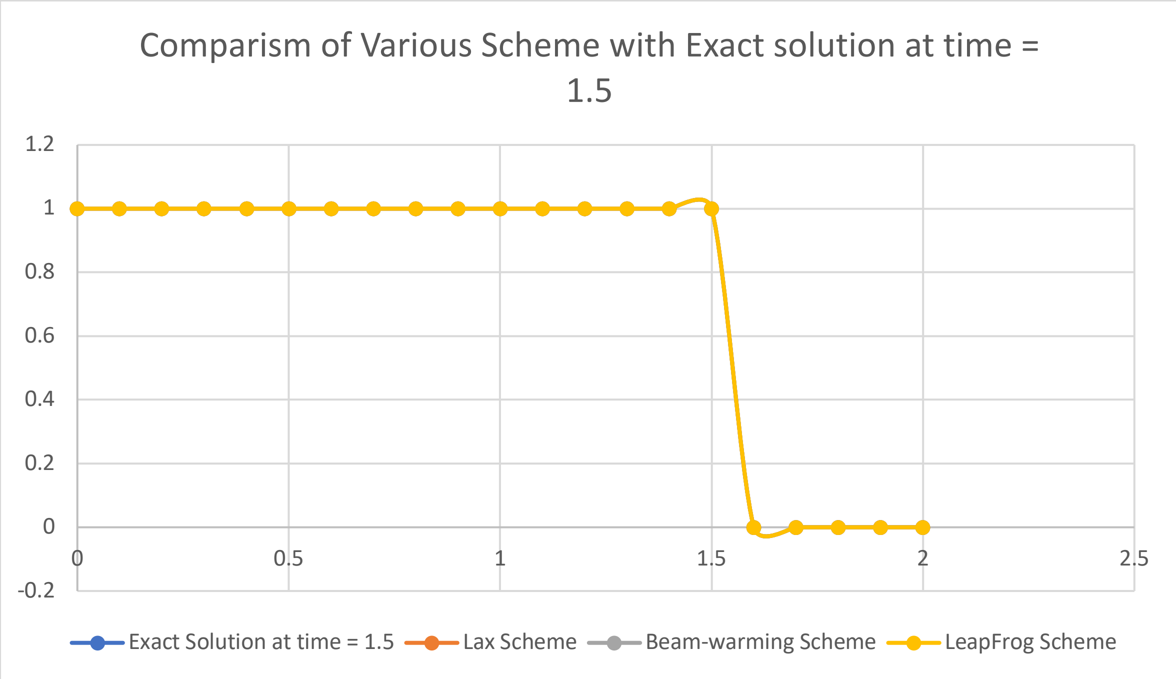


Figure 38: Graph of various Schemes and exact solution at courant number 1

From figure 30 it can be visualized that the schemes converge with the exact solution at the courant number of 1.

Consistency Check

Lax Scheme:

$$U_i^{n+1} = \frac{U_{i+1}^n + U_{i-1}^n}{2} - \frac{A\Delta t}{2\Delta x}(U_{i+1}^n - U_{i-1}^n) \quad (18)$$

$$U_i^{n+1} = U_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{1}{2!}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right) + \frac{1}{3!}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right) \quad (19)$$

$$U_{i+1}^n = U_i^n + \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) + \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right) \quad (20)$$

$$U_{i-1}^n = U_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right) \quad (21)$$

$$\frac{U_i^{n+1} - \frac{1}{2}(U_{i+1}^n + U_{i-1}^n)}{\Delta t} + A\frac{(U_{i+1}^n - U_{i-1}^n)}{2\Delta x} = 0 \quad (22)$$

Substituting equation 19, 20, 21 to equation 22 and rearranging the equation

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} = \frac{1}{3!}\Delta t^2\left(\frac{\partial^3 u}{\partial t^3}\right) - \frac{1}{2!}\Delta t\left(\frac{\partial^2 u}{\partial t^2}\right) + \frac{1}{2!\Delta t}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) + \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right) \quad (23)$$

Tending the $\Delta t, \Delta x \to 0$

We can see that the equation doesn't tend to zero on the right-hand side due to the presence of $\frac{\Delta x^2}{\Delta t}$ which gives undefined therefore it doesn't look like the linear advection equation hence it doesn't converge.

Beam warming Scheme:

$$U_i^{n+1} = U_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{1}{2!}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right) + \frac{1}{3!}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right)(24)$$
$$U_{i+1}^n = U_i^n + \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) + \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)(25)$$
$$U_{i-1}^n = U_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)(26)$$
$$U_{i-2}^n = U_i^n - 2\Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}4\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{1}{3!}8\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)(27)$$

Substituting equation 25,26,27 into equation 24 and rearranging the equation

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} = -\frac{1}{2!}\Delta t\left(\frac{\partial^2 u}{\partial t^2}\right) - \frac{1}{3!}\Delta t^2\left(\frac{\partial^3 u}{\partial t^3}\right) + \frac{A\Delta x}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{A\Delta x^2}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right) + \frac{A^2\Delta t}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{A^2\Delta t\Delta x}{2!}\left(\frac{\partial^3 u}{\partial x^3}\right) - \frac{A\Delta x}{2}\left(\frac{\partial^2 u}{\partial x^2}\right) + \frac{A\Delta x^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)(28)$$

Tending the $\Delta t, \Delta x \to 0$

It can be observed that the right-hand side of the equation tends to zero, so the equation is consistent with the main linear advection equation.

Leapfrog Scheme:

$$U_i^{n+1} = U_i^{n-1} - \frac{A\Delta t}{\Delta x}(U_{i+1}^n - U_{i-1}^n)(29)$$
$$U_i^{n+1} = U_i^n + \Delta t\frac{\partial u}{\partial t} + \frac{1}{2!}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right) + \frac{1}{3!}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right)(30)$$
$$U_i^{n-1} = U_i^n - \Delta t\frac{\partial u}{\partial t} + \frac{1}{2!}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right) - \frac{1}{3!}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right)(31)$$
$$U_{i+1}^n = U_i^n + \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) + \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)(32)$$
$$U_{i-1}^n = U_i^n - \Delta x\frac{\partial u}{\partial x} + \frac{1}{2!}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right) - \frac{1}{3!}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)(33)$$

Substituting equation 30,31,32,33 into equation 29 and rearranging the equation

$$\frac{\partial u}{\partial t} + A\frac{\partial u}{\partial x} = -\frac{1}{3!}\Delta t^2\left(\frac{\partial^3 u}{\partial t^3}\right) - \frac{\Delta x^2}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)(34)$$

It can be observed that the right-hand side of the equation tends to zero, so the equation is consistent with the main linear advection equation.

CONCLUSION

The 1D Linear Advection equation was investigated using different finite difference schemes and then compared with the exact equation and then these schemes were now evaluated for consistency with the linear advection equation, the stability of the schemes, and accuracy. From the analysis the lax scheme is the most stable but the least accurate and it is not consistent with the linear advection equation, the beam warming scheme is a bit accurate, it is also conditionally stable, and it is consistent with the main equation, the leapfrog scheme is the most accurate and it is also conditionally stable, and consistent with the main equation. Therefore lax is the best scheme for stability, while leapfrog is the best in terms of accuracy.

REFERENCES

1. Diethelm, K. & Freed, A. D., 1999. On the solution of nonlinear fractional-order differential equations used in the modeling of viscoplasticity. *Springer Berlin Heidelberg,* pp. 217-224.
2. Khongar, P. D., 2018. *Diff. Academy.* [Online]
   Available at: https://peymandavvalo.github.io/linear_1d_advection_equation.html
   [Accessed 23 November 2023].
3. Mainardi, F. & Carpinteri, A., 1997. Fractals and Fractional Calculus Continuum Mechanics. *Springer Verlag,* pp. 291-48.
4. Mazumder, S., 2016. The Finite Difference Method. In: S. Mazumder, ed. *Numerical Methods for Partial Differential Equations* . s.l.:Academic Press, pp. 51-101.