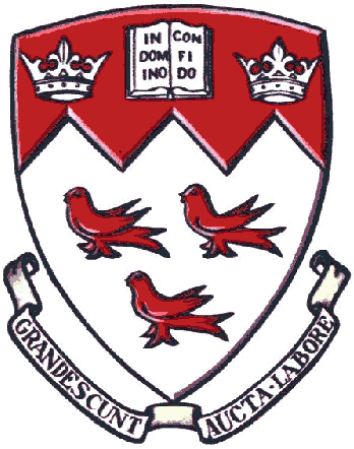
MECH 539: Computational Aerodynamics

Project #1: Numerical Methods for One-Dimensional Wave Equation



Justin ChanWoo Yang
260368098
chanwoo.yang@mail.mcgill.ca

February 19th, 2013

Q1.

Upwind Scheme:

$$u_j^{n+1} = u_j^n - C\frac{\Delta t}{\Delta x}\left(u_j^n - u_{j-1}^n\right) \qquad * C\frac{\Delta t}{\Delta x} = v$$

$$\therefore\ u_j^{n+1} = (1-v)u_j^n + v\left(u_{j-1}^n\right)$$

```
function [ u ] = Upwind( dx, dt )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain

%OUTPUT
%u: Array of values of wave equation at each grid point

x=0;    %Spatial domain boundary (start)
X=40;   %Spatial domain boundary (end)
t=0;    %Time domain boundary (start)
T=10;   %Time domain boundary (end)
C=1/2;  %C = wave speed
v=C*dt/dx;

% Initialize the exact Dirichlet boundary conditions
for n=1:((T/dt)+1)
    u(1,n)=0;
    u(((X/dx)+1),n)=1;
end

% Initialize the initial condition
for j=1:((X/dx)+1)
    u(j,1)=(1/2)*(1+tanh(250*(x-20)));
    x=x+dx;
end

%Upwind Scheme
for n=1:(T/dt)
    for j=2:((X/dx)+1)
        u(j,n+1)=(1-v)*u(j,n)+v*(u(j-1,n));
    end
end

end
```

Upwind.m | Lax.m | LaxWendroff.m | LeapFrog.m | MacCormack.m

Upwind Ln 15 Col 1

Lax Scheme:

$$\frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\Delta t} + \frac{C}{2}\left(\frac{u_{j+1}^n - u_{j-1}^n}{\Delta x}\right) = 0$$

$$u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) + \frac{C\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) = 0 \qquad * \, C\frac{\Delta t}{\Delta x} = \nu$$

$$\therefore \; u_j^{n+1} = (1-\nu)\frac{1}{2}u_{j+1}^n + (1+\nu)\frac{1}{2}u_{j-1}^n$$

```
function [ u ] = Lax( dx, dt )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain

%OUTPUT
%u: Array of values of wave equation at each grid point


x=0;    %Spatial domain boundary (start)
X=40;   %Spatial domain boundary (end)
t=0;    %Time domain boundary (start)
T=10;   %Time domain boundary (end)
C=1/2;  %C = wave speed
v=C*dt/dx;

% Initialize the exact Dirichlet boundary conditions
for n=1:((T/dt)+1)
    u(1,n)=0;
    u(((X/dx)+1),n)=1;
end

% Initialize the initial condition
for j=1:((X/dx)+1)
    u(j,1)=(1/2)*(1+tanh(250*(x-20)));
    x=x+dx;
end

%Lax Method
for n=1:(T/dt)
    for j=2:(X/dx)
        u(j,n+1)=(1-v)*(1/2)*u(j+1,n)+(1+v)*(1/2)*(u(j-1,n));
    end
end

end
```

Upwind.m | Lax.m | LaxWendroff.m | LeapFrog.m | MacCormack.m

Lax Ln 36 Col 4

Lax-Wendroff:

$$u_j^{n+1} = u_j^n - \frac{C\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{C^2\Delta t^2}{2\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) \qquad * C\frac{\Delta t}{\Delta x} = \nu$$

$$\therefore\ u_j^{n+1} = (1-\nu)u_j^n + \nu(\nu-1)\frac{1}{2}u_{j+1}^n + \nu(\nu+1)\frac{1}{2}u_{j-1}^n$$

```
function [ u ] = LaxWendroff( dx, dt )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain

%OUTPUT
%u: Array of values of wave equation at each grid point


x=0;    %Spatial domain boundary (start)
X=40;   %Spatial domain boundary (end)
t=0;    %Time domain boundary (start)
T=10;   %Time domain boundary (end)
C=1/2;  %C = wave speed
v=C*dt/dx;

% Initialize the exact Dirichlet boundary conditions
for n=1:((T/dt)+1)
    u(1,n)=0;
    u(((X/dx)+1),n)=1;
end

% Initialize the initial condition
for j=1:((X/dx)+1)
    u(j,1)=(1/2)*(1+tanh(250*(x-20)));
    x=x+dx;
end

%Lax-Wendroff Method
for n=1:(T/dt)
    for j=2:(X/dx)
        u(j,n+1)=(1-(v^2))*u(j,n)+(v-1)*(1/2)*v*(u(j+1,n))+(v+1)*(1/2)*v*u((j-1),n);
    end
end

end
```

Upwind.m | Lax.m | LaxWendroff.m | LeapFrog.m | MacCormack.m

LaxWendroff Ln 20 Col 17

Leap-Frog Scheme:

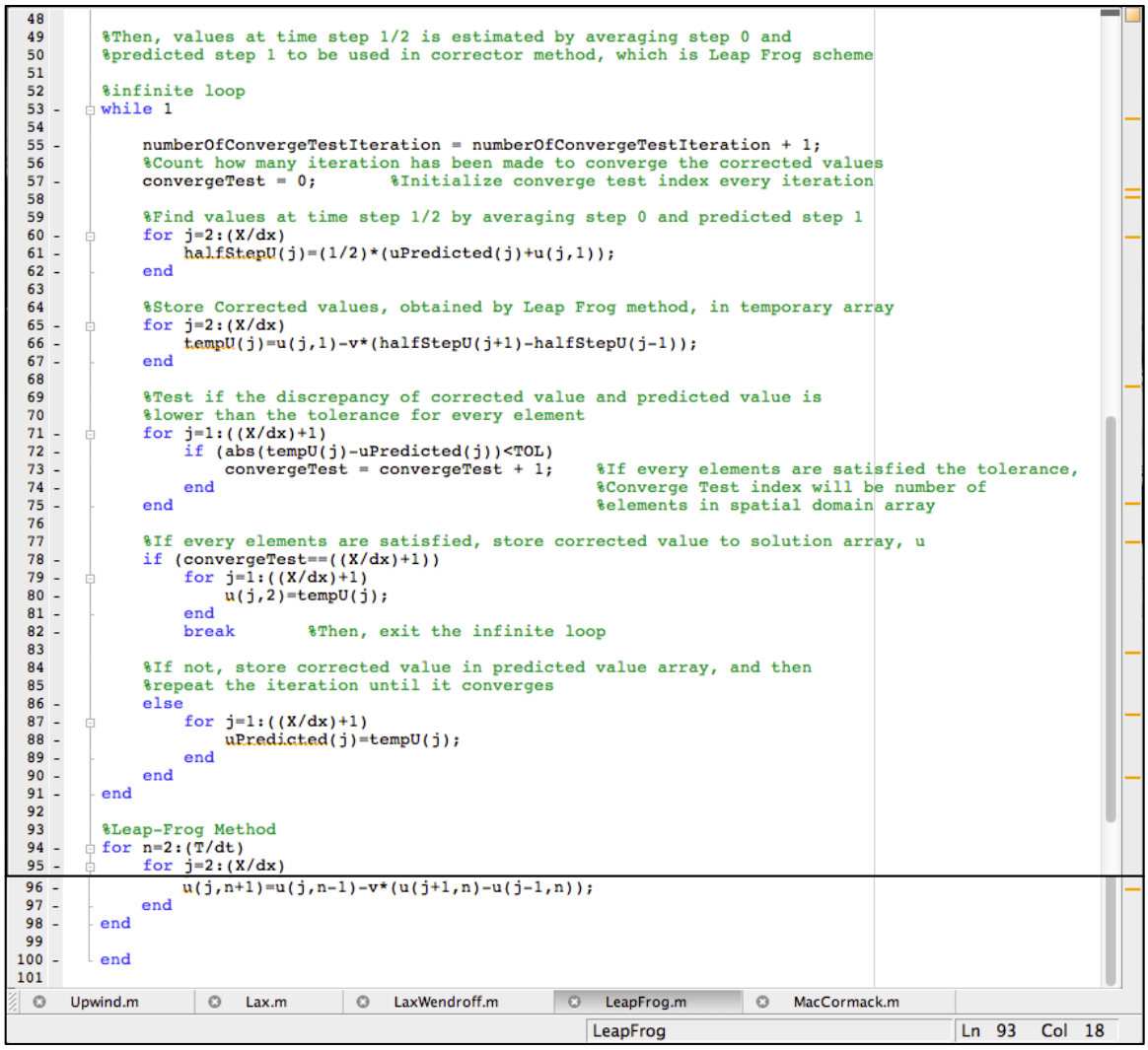
$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + C\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right) = 0$$

$$u_j^{n+1} - u_j^{n-1} + C\frac{2\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) = 0 \qquad * C\frac{\Delta t}{\Delta x} = \nu$$

$$\therefore u_j^{n+1} = u_j^{n-1} - \nu(u_{j+1}^n - u_{j-1}^n)$$

Since Leap-Frog scheme is a two-level scheme, it requires two previous time steps. So 'Predictor/Corrector' method was used to obtain values at time step 1 using the values at time step 0.

```
function [ u , numberOfConvergeTestIteration] = LeapFrog( dx, dt )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain

%OUTPUT
%u: Array of values of wave equation at each grid point

TOL = 10^(-5); %Tolerance of discrepancy between predicted value and corrected value
x=0;    %Spatial domain boundary (start)
X=40;   %Spatial domain boundary (end)
t=0;    %Time domain boundary (start)
T=10;   %Time domain boundary (end)
C=1/2;  %C = wave speed
v=C*dt/dx;

numberOfConvergeTestIteration=0;    %Initialize number of iteration of converge test

% Initialize the exact Dirichlet boundary conditions
for n=1:((T/dt)+1)
    u(1,n)=0;
    u(((X/dx)+1),n)=1;
end

% Initialize the initial condition
for j=1:((X/dx)+1)
    u(j,1)=( 1/2)*(1+tanh(250*(x-20)));
    x=x+dx;
end


%Since leap-frog scheme is a two-level scheme, requiring two previous time
%steps,'Predictor/Corrector' method is used to obtain values at time step 1
%from the values at time step 0

%Initialize the boundary contitions
halfStepU(1)=0;
tempU(1)=0;
uPredicted(1)=0;
halfStepU(((X/dx)+1))=1;
tempU(((X/dx)+1))=1;
uPredicted(((X/dx)+1))=1;

%Values at time step 1 is predicted using 'Upwind' scheme (Predictor)
for j=2:((X/dx)+1)
    uPredicted(j)=(1-v)*u(j,1)+v*(u(j-1,1));
end

```



```
48
49      %Then, values at time step 1/2 is estimated by averaging step 0 and
50      %predicted step 1 to be used in corrector method, which is Leap Frog scheme
51
52      %infinite loop
53 -    while 1
54
55 -        numberOfConvergeTestIteration = numberOfConvergeTestIteration + 1;
56          %Count how many iteration has been made to converge the corrected values
57 -        convergeTest = 0;       %Initialize converge test index every iteration
58
59          %Find values at time step 1/2 by averaging step 0 and predicted step 1
60 -        for j=2:(X/dx)
61 -            halfStepU(j)=(1/2)*(uPredicted(j)+u(j,1));
62 -        end
63
64          %Store Corrected values, obtained by Leap Frog method, in temporary array
65 -        for j=2:(X/dx)
66 -            tempU(j)=u(j,1)-v*(halfStepU(j+1)-halfStepU(j-1));
67 -        end
68
69          %Test if the discrepancy of corrected value and predicted value is
70          %lower than the tolerance for every element
71 -        for j=1:((X/dx)+1)
72 -            if (abs(tempU(j)-uPredicted(j))<TOL)
73 -                convergeTest = convergeTest + 1;    %If every elements are satisfied the tolerance,
74 -            end                                     %Converge Test index will be number of
75 -        end                                         %elements in spatial domain array
76
77          %If every elements are satisfied, store corrected value to solution array, u
78 -        if (convergeTest==((X/dx)+1))
79 -            for j=1:((X/dx)+1)
80 -                u(j,2)=tempU(j);
81 -            end
82 -            break       %Then, exit the infinite loop
83
84          %If not, store corrected value in predicted value array, and then
85          %repeat the iteration until it converges
86 -        else
87 -            for j=1:((X/dx)+1)
88 -                uPredicted(j)=tempU(j);
89 -            end
90 -        end
91 -    end
92
93      %Leap-Frog Method
94 -    for n=2:(T/dt)
95 -        for j=2:(X/dx)
96 -            u(j,n+1)=u(j,n-1)-v*(u(j+1,n)-u(j-1,n));
97 -        end
98 -    end
99
100 -   end
101
```

Upwind.m | Lax.m | LaxWendroff.m | LeapFrog.m | MacCormack.m

LeapFrog | Ln 93 Col 18

MacCormack Scheme:

Predictor:

$$\overline{u_j^{n+1}} = u_j^n - C\frac{\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) \qquad * C\frac{\Delta t}{\Delta x} = \nu$$

$$\therefore \ \overline{u_j^{n+1}} = (1+\nu)u_j^n - \nu \cdot u_{j+1}^n$$

Corrector:

$$u_j^{n+1} = \frac{1}{2}\left[u_j^n + \overline{u_j^{n+1}} - C\frac{\Delta t}{\Delta x}(\overline{u_j^{n+1}} - \overline{u_{j-1}^{n+1}})\right] \qquad * C\frac{\Delta t}{\Delta x} = \nu$$

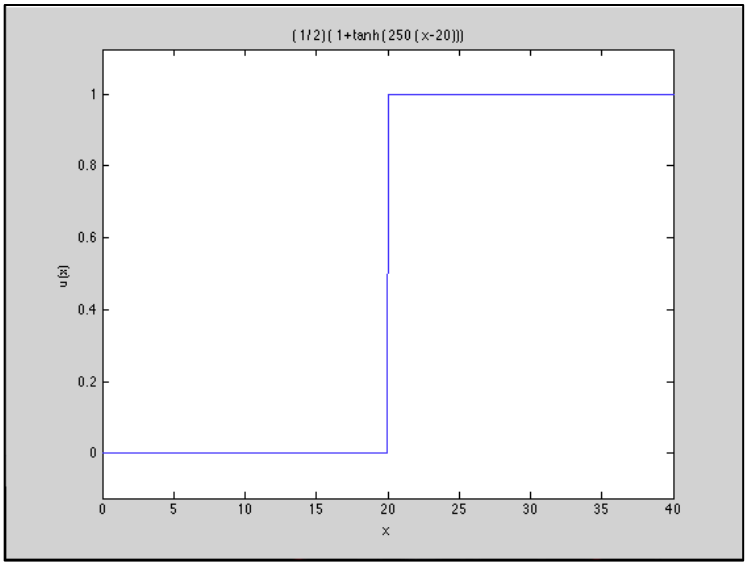
$$\therefore u_j^{n+1} = \frac{1}{2}\left[u_j^n + (1-\nu)\overline{u_j^{n+1}} + \nu \cdot \overline{u_{j-1}^{n+1}})\right]$$

```
function [ u ] = MacCormack( dx, dt )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain

%OUTPUT
%u: Array of values of wave equation at each grid point


x=0;    %Spatial domain boundary (start)
X=40;   %Spatial domain boundary (end)
t=0;    %Time domain boundary (start)
T=10;   %Time domain boundary (end)
C=1/2;  %C = wave speed
v=C*dt/dx;

% Initialize the exact Dirichlet boundary conditions
for n=1:((T/dt)+1)
    uPredicted(1,n)=0;
    uPredicted(((X/dx)+1),n)=1;
    u(1,n)=0;
    u(((X/dx)+1),n)=1;
end

% Initialize the initial condition
for j=1:((X/dx)+1)
    u(j,1)=(1/2)*(1+tanh(250*(x-20)));
    x=x+dx;
end

for n=1:(T/dt)
    for j=2:(X/dx)
        uPredicted(j,n+1)=(1+v)*u(j,n)-v*u(j+1,n);      %Predictor Method
        u(j,n+1)=(1/2)*(u(j,n)+(1-v)*uPredicted(j,n+1)+v*uPredicted(j-1,n+1));  %Corrector Method
    end
end


end
```

Upwind.m | Lax.m | LaxWendroff.m | LeapFrog.m | MacCormack.m

MacCormack Ln 27 Col 39

```
Command Window
>> u = @(x) (1/2)*(1+tanh(250*(x-20)));
>> ezplot(u,[0,40])
xlabel('x')
ylabel('u(x)')
>>
```



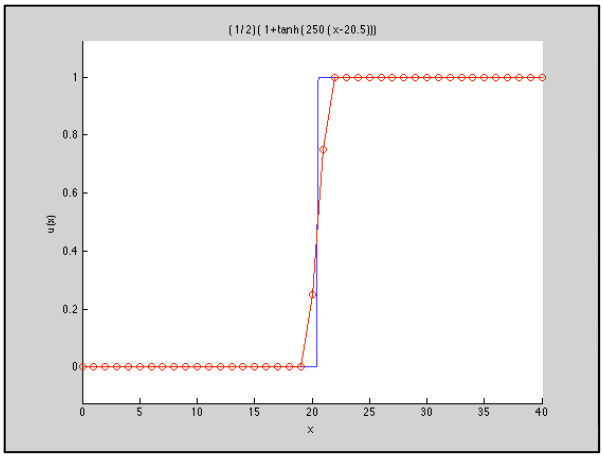
This is the graph of exact solution of wave equation at t = 0. And it is known that as time goes on, the shape of graph stays the same, but the discontinuity occurs at different space. In other words, the wave equation graph shifts to the right as the time goes. Moreover, from this wave equation, since C is equal to $\frac{1}{2}$, which is the wave speed, and the wave equation is computed to t = 10, the discontinuity would occur at x = 25 at t = 10 of time frame,

First, problem is solved with all five schemes for $\Delta t = 1.0$

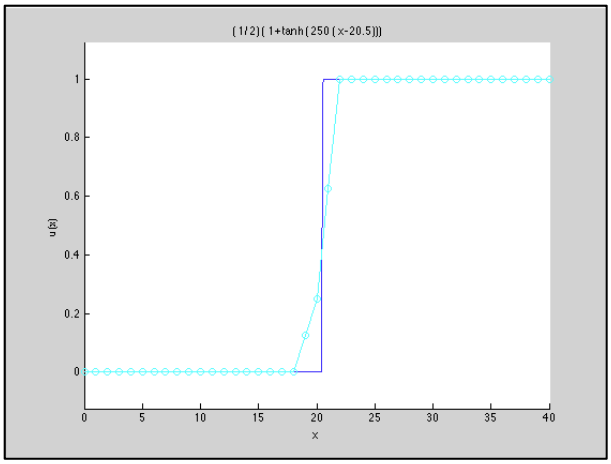
```
Command Window
>> a=Upwind(1,1);
>> b=Lax(1,1);
>> c=LaxWendroff(1,1);
>> d=LeapFrog(1,1);
>> e=MacCormack(1,1);
>> hold on,
ezplot(u,[0,40])
plot(0:40,a(:,2),'-ro')
xlabel('x')
ylabel('u(x)')
>>
```

Commands for setting arrays for each scheme and example of plotting commands.

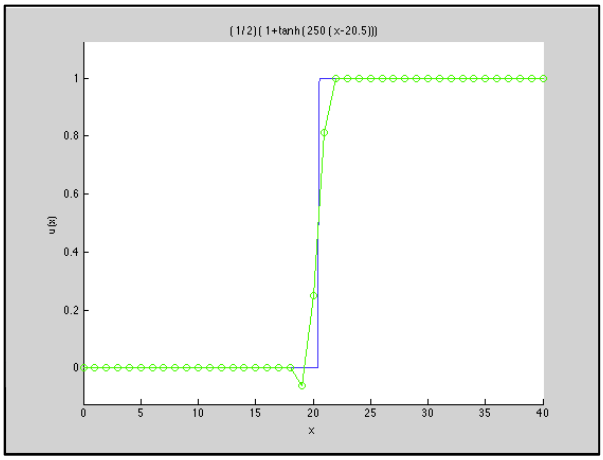
Comparison between each scheme and exact solution at t = 1, (u(x,2) in MATLAB)



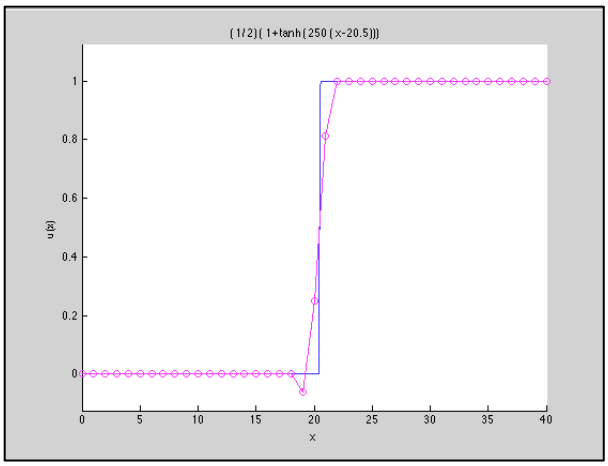
Upwind Scheme



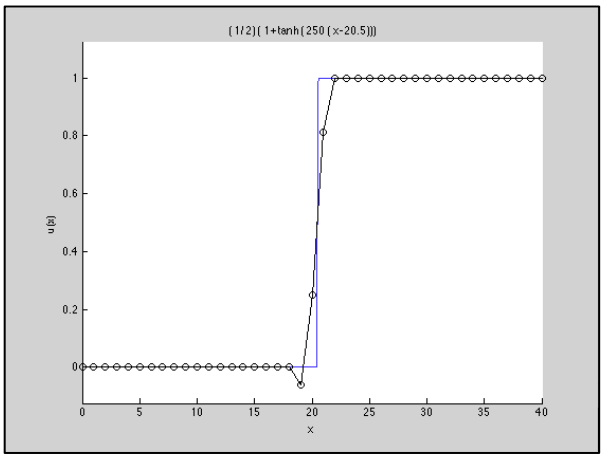
Lax Scheme



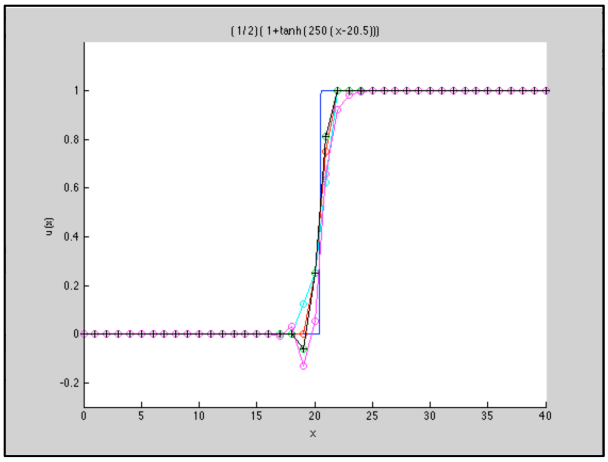
Lax-Wendroff Scheme



Leap-Frog Scheme

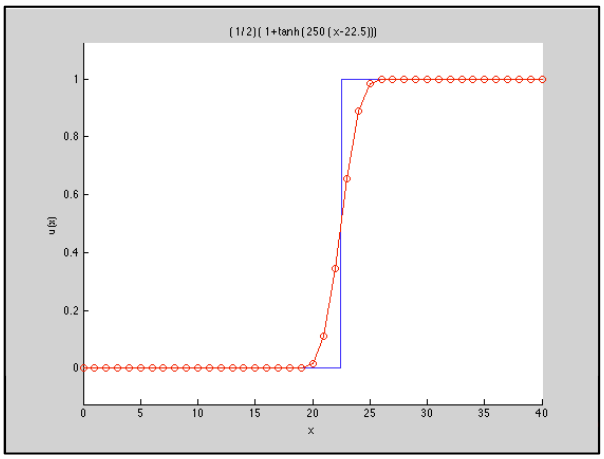


MacCormack Scheme

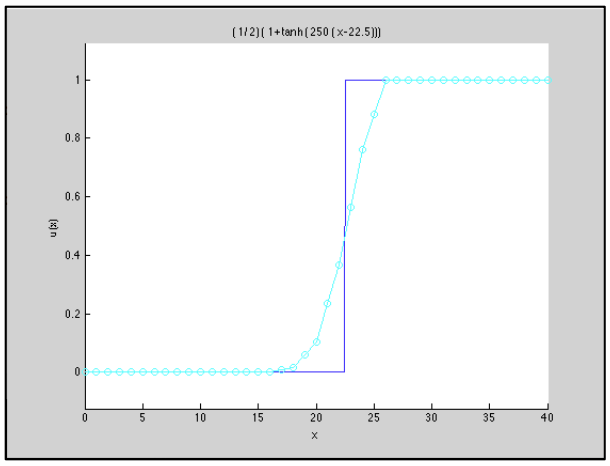


All Five Schemes

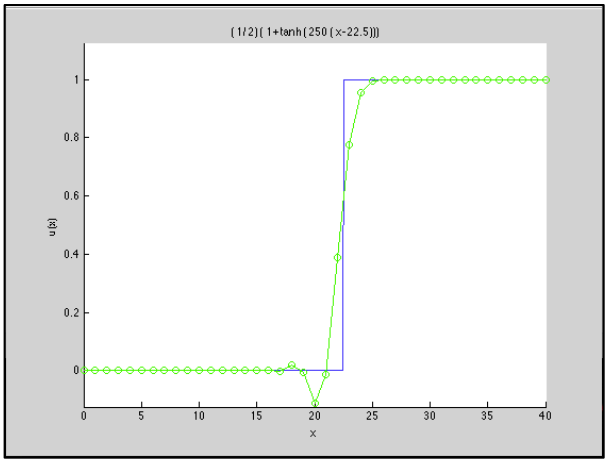
Comparison between each scheme and exact solution at t = 5, (u(x,6) in MATLAB)



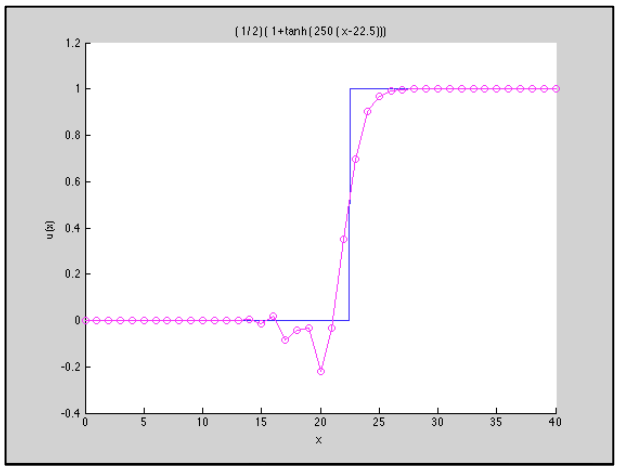
Upwind Scheme



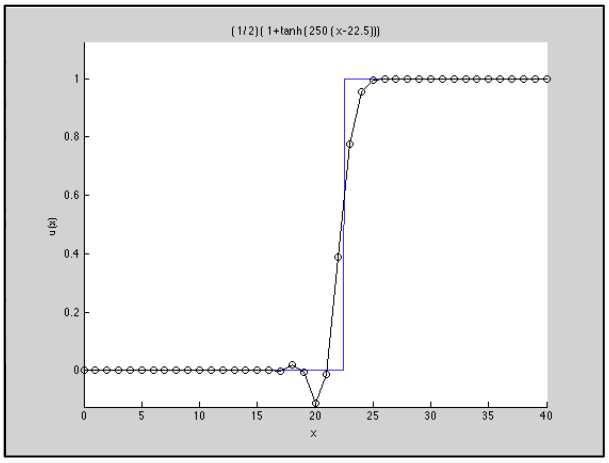
Lax Scheme



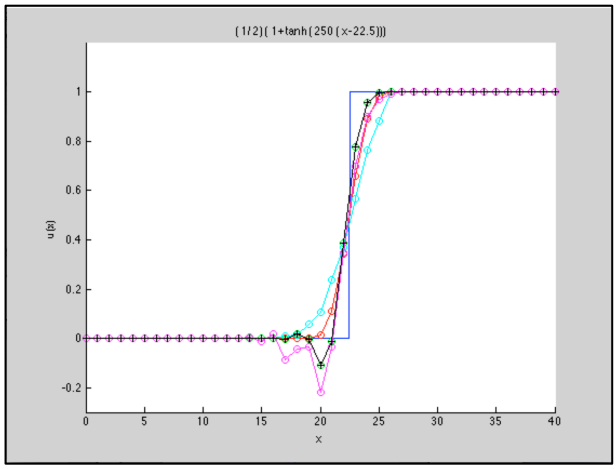
Lax-Wendroff Scheme



Leap-Frog Scheme

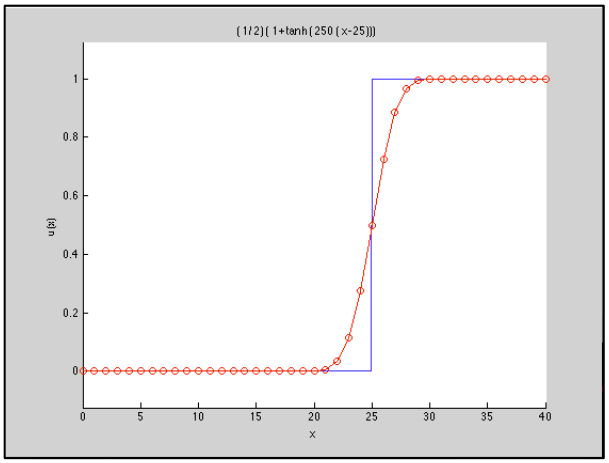


MacCormack Scheme

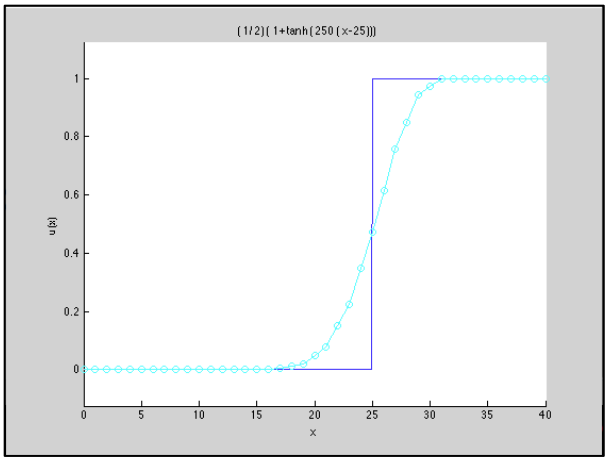


All Five Schemes

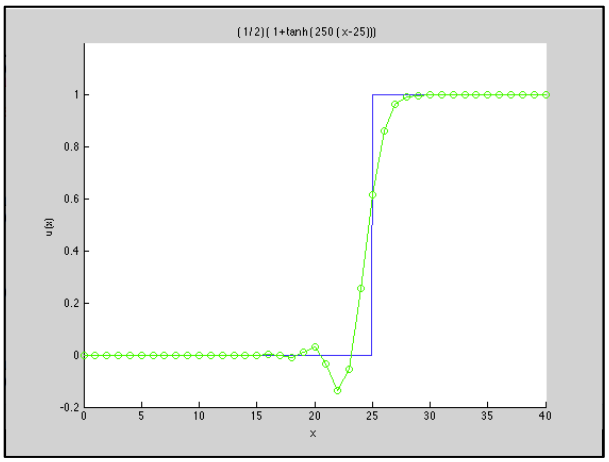
Comparison between each scheme and exact solution at t = 10, (u(x,11) in MATLAB)



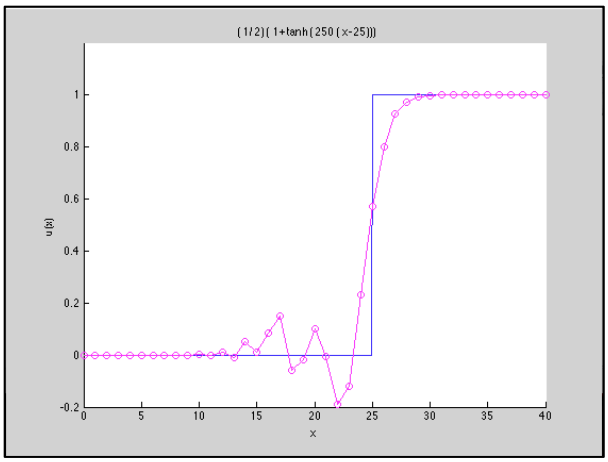
Upwind Scheme



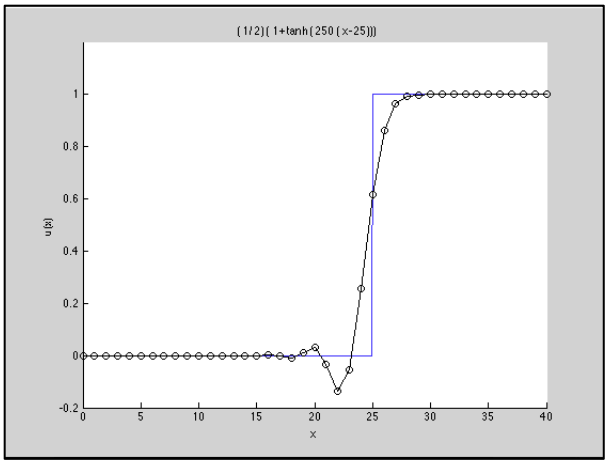
Lax Scheme



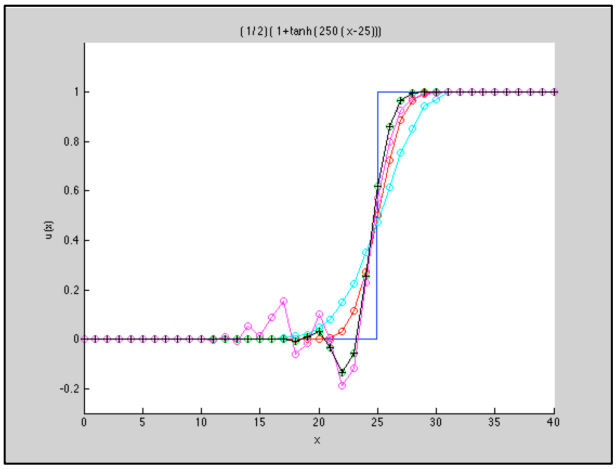
Lax-Wendroff Scheme



Leap-Frog Scheme

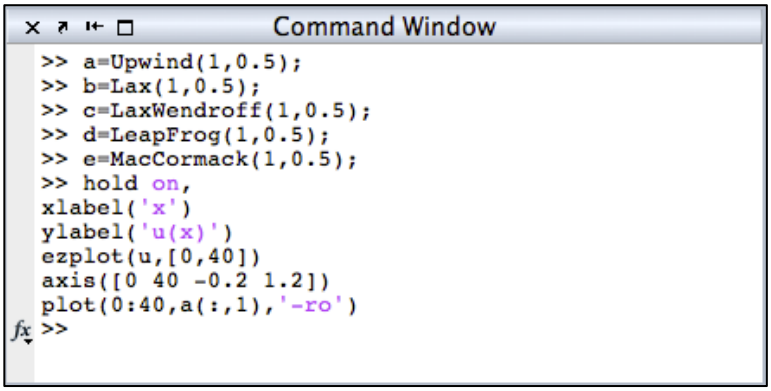


MacCormack Scheme



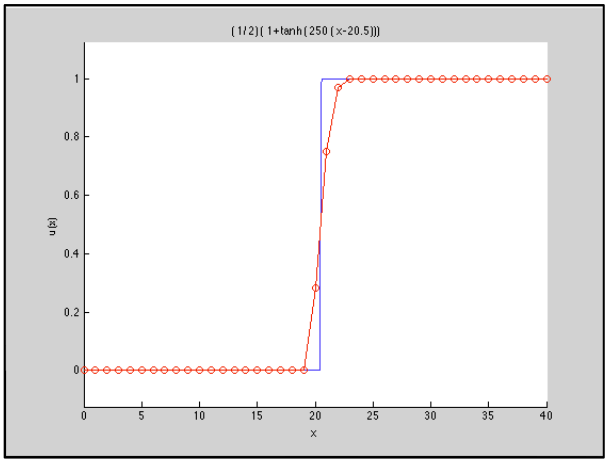
All Five Schemes

Next, problem is solved with all five schemes for $\Delta t = 0.5$

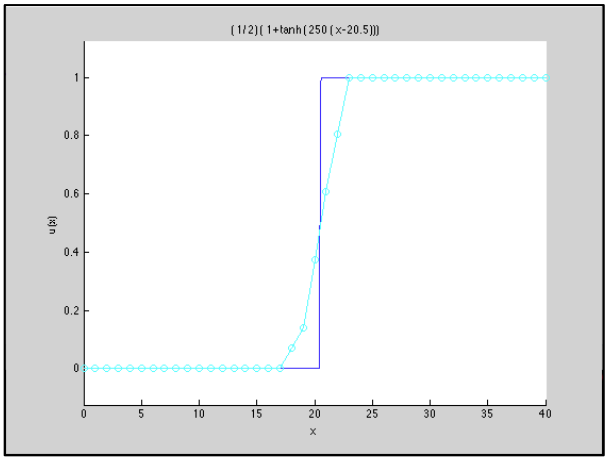


Commands for setting arrays for each scheme and example of plotting command.

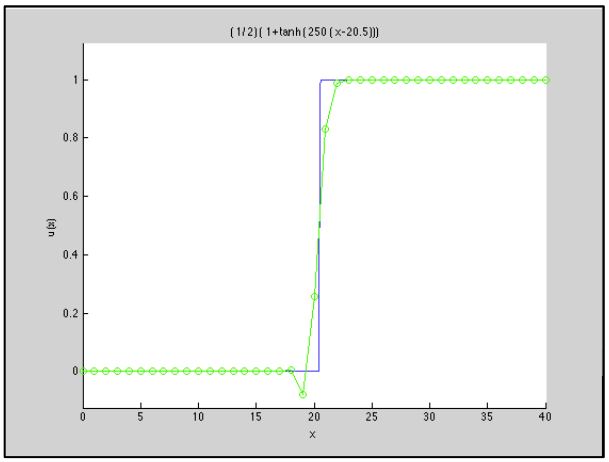
Comparison between each scheme and exact solution at t = 1, (u(x,3) in MATLAB)



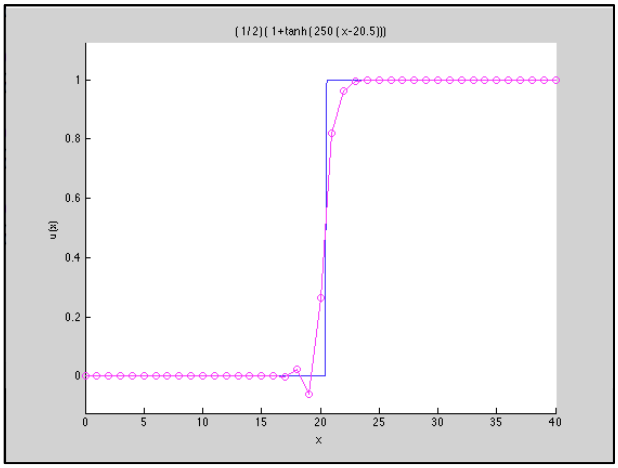
Upwind Scheme



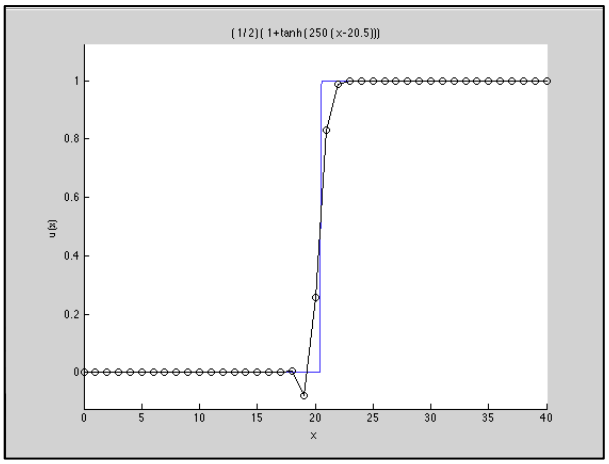
Lax Scheme



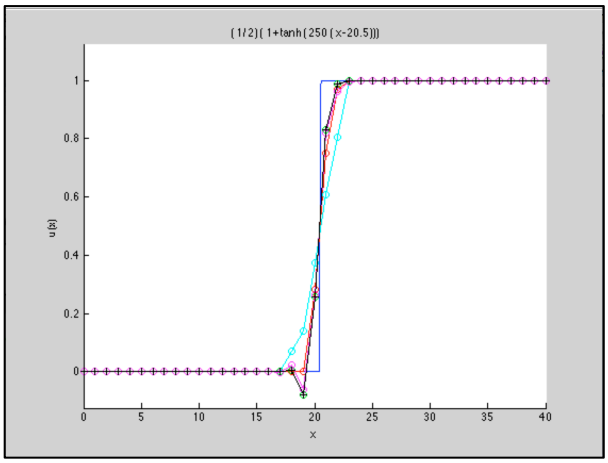
Lax-Wendroff Scheme



Leap-Frog Scheme

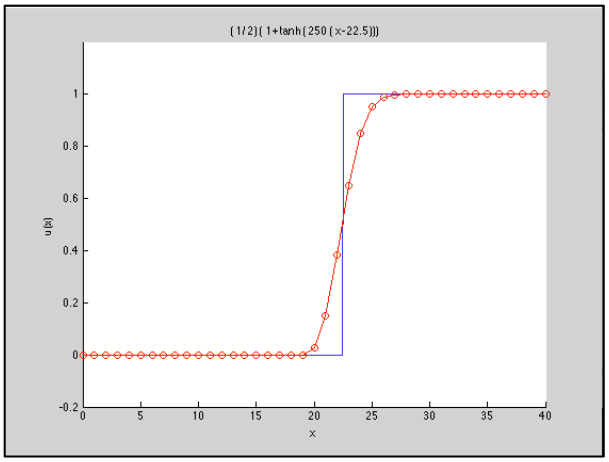


MacCormack Scheme

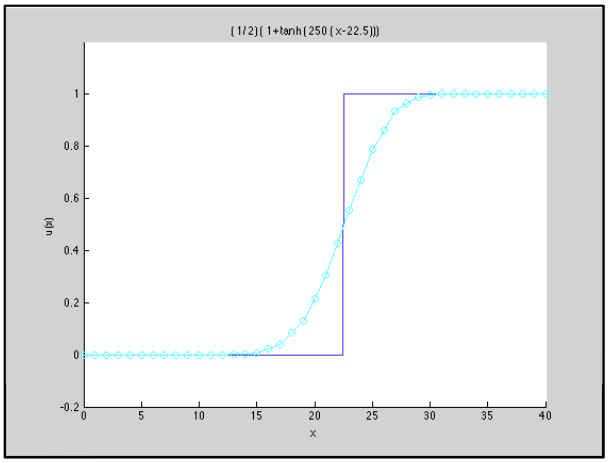


All Five Schemes

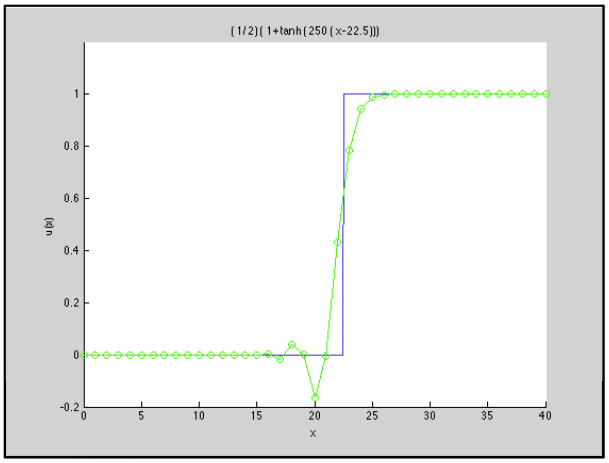
Comparison between each scheme and exact solution at t = 5, (u(x,11) in MATLAB)



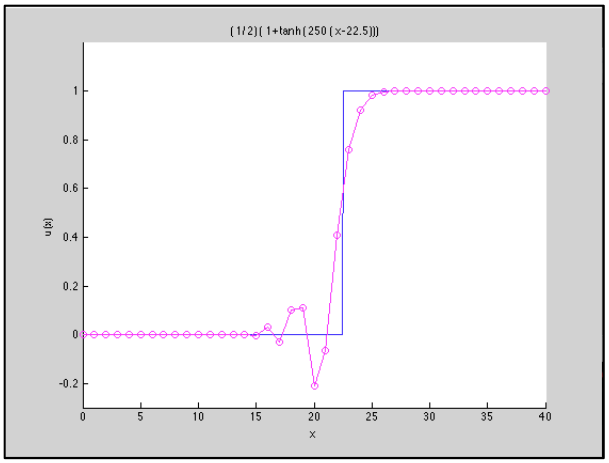
Upwind Scheme



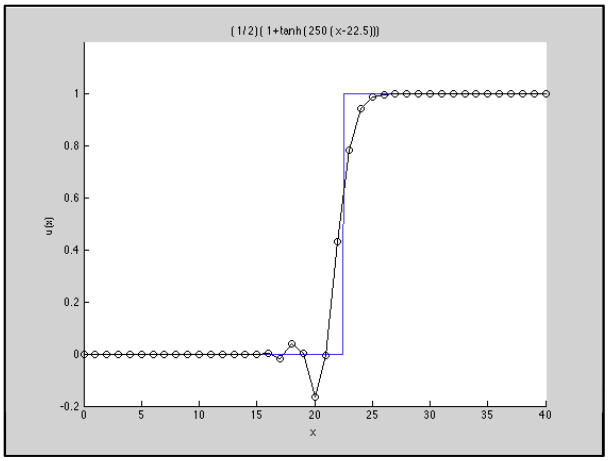
Lax Scheme



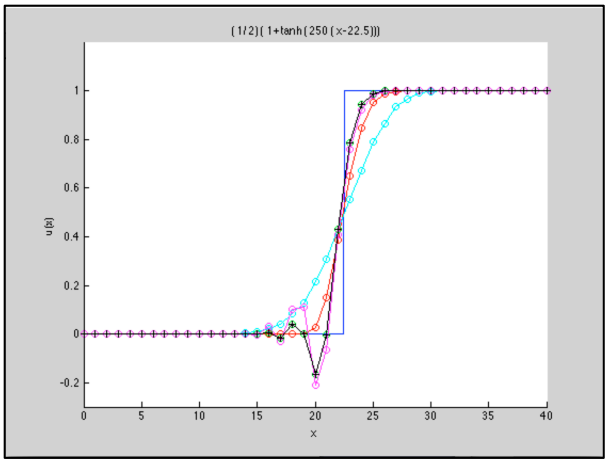
Lax-Wendroff Scheme



Leap-Frog Scheme

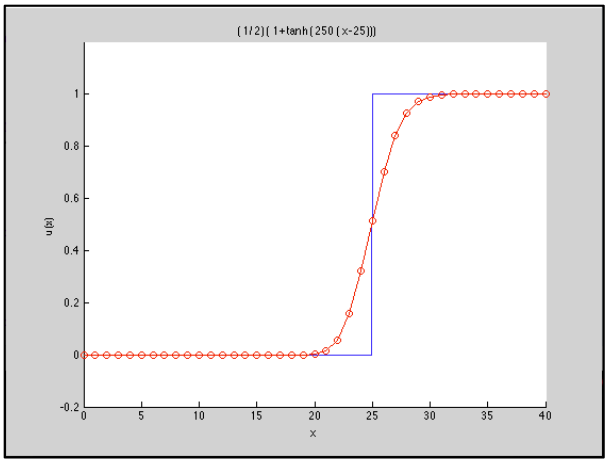


MacCormack Scheme

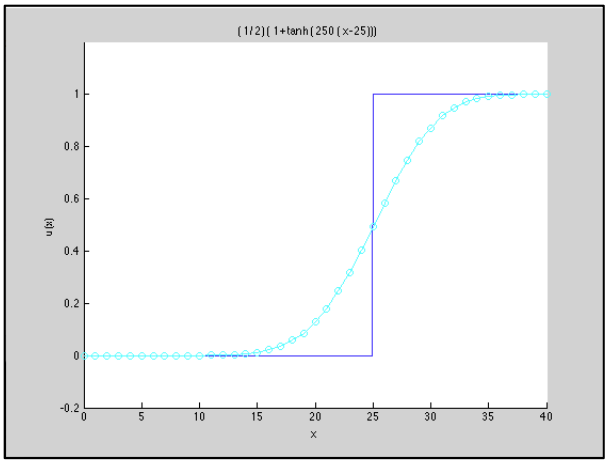


All Five Schemes

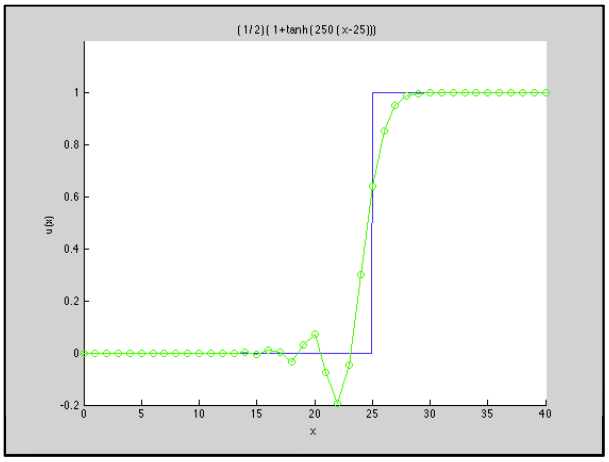
Comparison between each scheme and exact solution at t = 10, (u(x,21) in MATLAB)



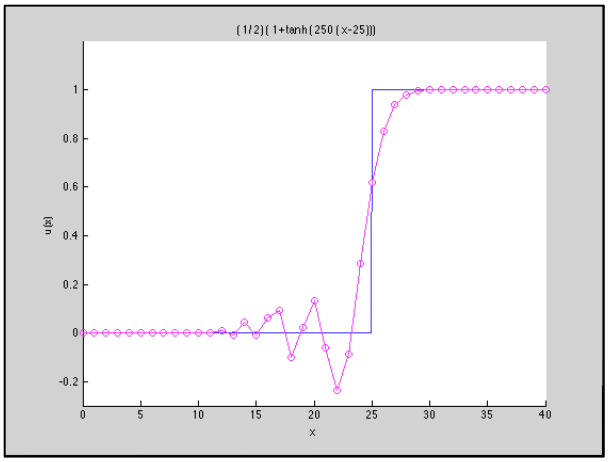
Upwind Scheme



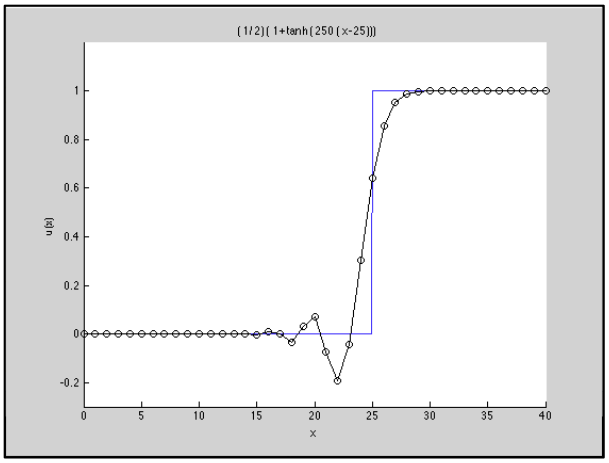
Lax Scheme



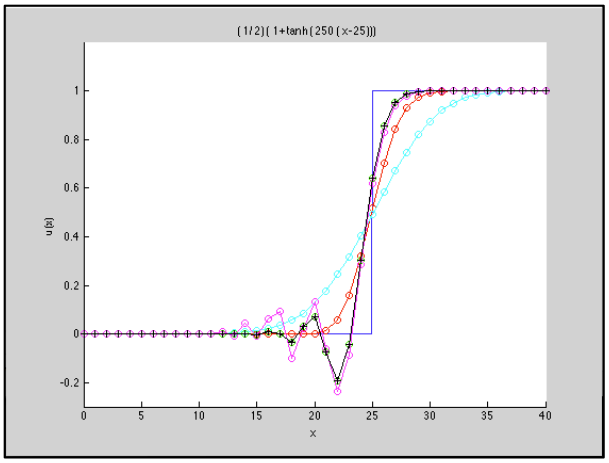
Lax-Wendroff Scheme



Leap-Frog Scheme



MacCormack Scheme



All Five Schemes

For both cases of $\Delta t = 1.0$ and $\Delta t = 0.5$, Upwind and Lax schemes do not introduce the dispersive errors, but they suffer from the dissipation as they iterate more to further time frames. The dissipation of Lax scheme is greater than that of Upwind scheme. And the dissipation of two schemes gets larger as Δt is changed from 1.0 to 0.5, with fixed $\Delta x = 1.0$. Specifically, in case of $\Delta t = 0.5$ at t = 10, the Lax scheme diffuses quite significantly.

Meanwhile, Lax-Wendroff, Leap-Frog, and MacCormack schemes do not suffer from large dissipation, but they introduce the dispersive errors as they iterate more for both cases of $\Delta t = 1.0$ and $\Delta t = 0.5$. The dispersive error of Leap-Frog scheme is larger than Lax-Wendroff and MacCormack schemes. The dispersive errors of these three schemes also get larger as Δt is changed from 1.0 to 0.5. For the case of $\Delta t = 0.5$ at t = 10, the dispersive error of Leap-Frog scheme is quite significant. Additionally, it is noticeable that the values produced by Lax-Wendroff scheme and MacCormack scheme are exactly the same, according to the superimposed plots of all five schemes for both cases of $\Delta t = 1.0$ and $\Delta t = 0.5$. Thus, from this analysis, a scheme with finer grid does not always give the better and accurate results.

Q2.

Upwind and Lax scheme are chosen to perform the grid study. The new functions for each scheme are written to perform the test easier. Then, the study is started with the 41 grid point mesh as the coarsest grid: $\Delta x = 1.0, \Delta t = 1.0$

i) Upwind Scheme

```
function [ u ] = GridStudyUpwind( dx, dt, t )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain
%t: Specific time frame at which the values will be plotted (0? t ?10)

%OUTPUT
%u: Array of values of wave equation

%Compute the wave equation using Upwind scheme
u = Upwind(dx,dt);

T=10;   %Time domain boundary (end)
X=40;   %Spatial domain boundary (end)

dn = ((T/dt))/10;   %Calculate the discrepancy between each time fame
n = 1+dn*t;         %Calculate appropriate index that corresponds to specified time frame

hold on,
xlabel('x')
ylabel('u(x)')
y=linspace(0,40,((X/dx)+1));
plot(y,u(:,n),'-ro');       %Plot the values at specified time frame
hold off

end
```

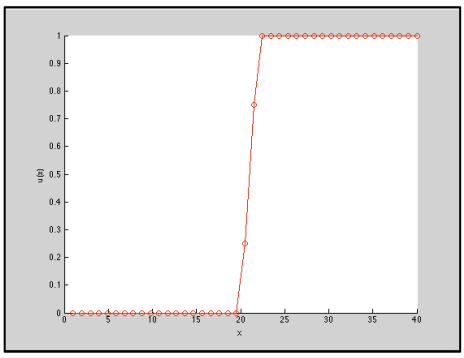
GridStudyLax.m GridStudyUpwind.m

ii) Lax Scheme

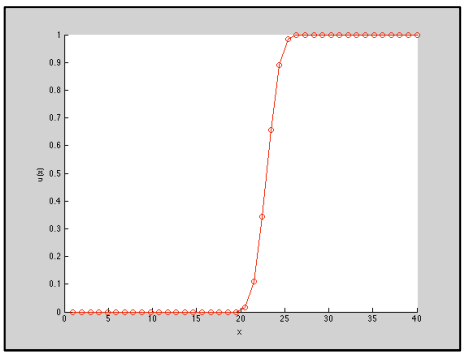
```
function [ u ] = GridStudyLax( dx,dt,t )
%INPUT
%dt: Discretization size of time domain
%dx: Discretization size of spatial domain
%t: Specific time frame at which the values will be plotted (0? t ?10)

%OUTPUT
%u: Array of values of wave equation

%Compute the wave equation using Lax scheme
u = Lax(dx,dt);

T=10;   %Time domain boundary (end)
X=40;   %Spatial domain boundary (end)

dn = ((T/dt))/10;   %Calculate the discrepancy between each time fame
n = 1+dn*t;         %Calculate appropriate index that corresponds to specified time frame

hold on,
xlabel('x')
ylabel('u(x)')
y=linspace(0,40,((X/dx)+1));
plot(y,u(:,n),'-bo');       %Plot the values at specified time frame
hold off

end
```

GridStudyLax.m GridStudyUpwind.m

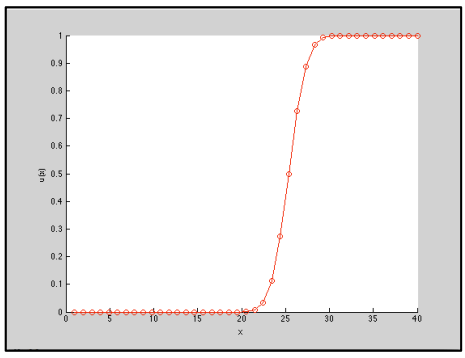
i) Upwind Scheme



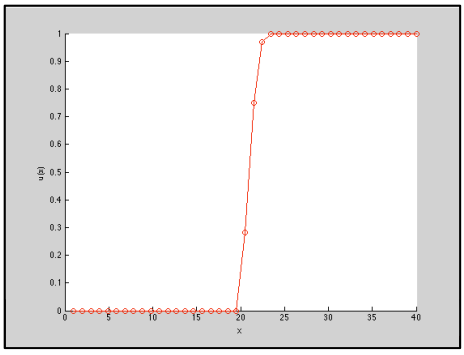
(1) $\Delta x = 1.0, \Delta t = 1.0$ at t = 1



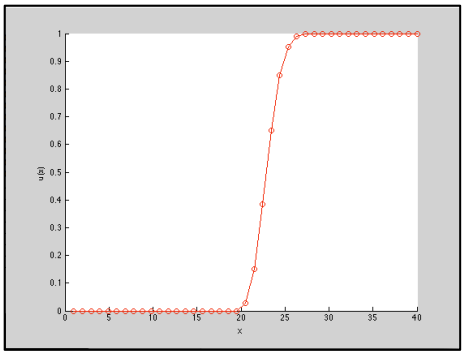
(2) $\Delta x = 1.0, \Delta t = 1.0$ at t = 5



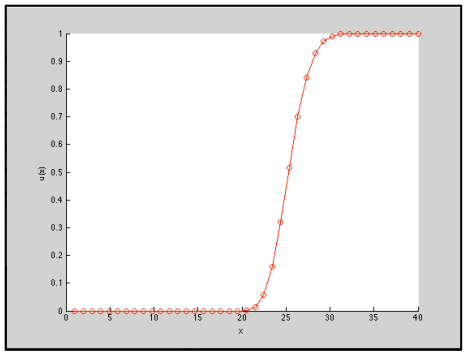
(3) $\Delta x = 1.0, \Delta t = 1.0$ at t = 10



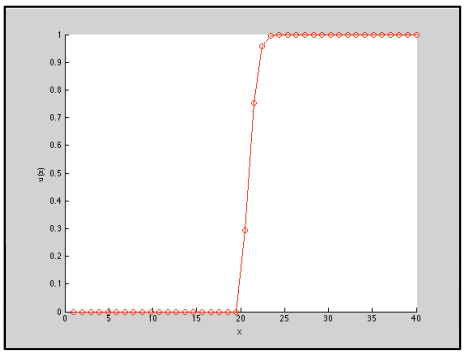
(1) $\Delta x = 1.0, \Delta t = 0.5$ at t = 1



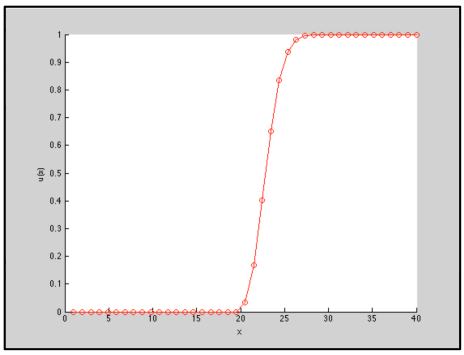
(2) $\Delta x = 1.0, \Delta t = 0.5$ at t = 5



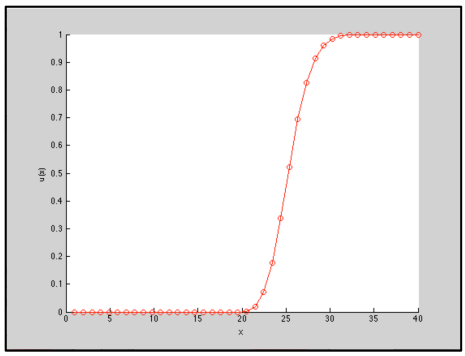
(3) $\Delta x = 1.0, \Delta t = 0.5$ at t = 10



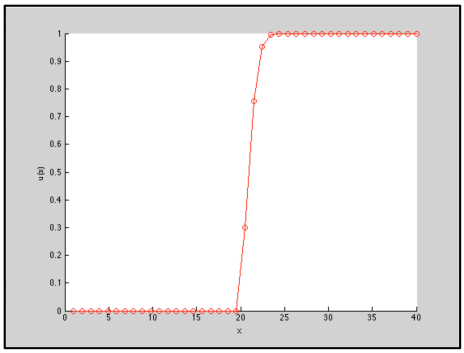
(1) $\Delta x = 1.0, \Delta t = 0.25$ at t = 1



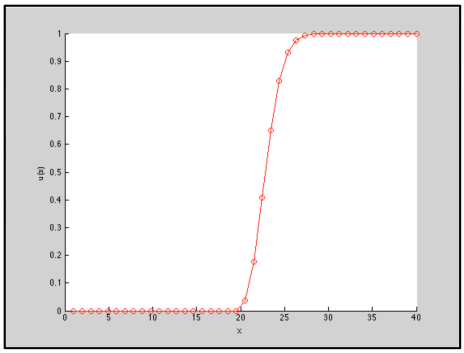
(2) $\Delta x = 1.0, \Delta t = 0.25$ at t = 5



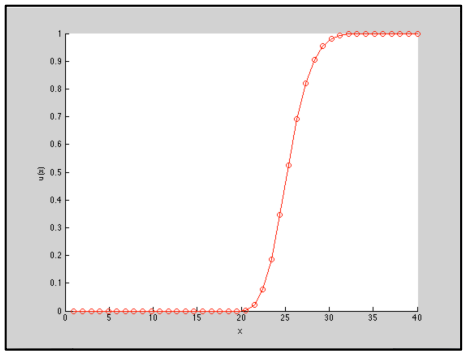
(3) $\Delta x = 1.0, \Delta t = 0.25$ at t = 10



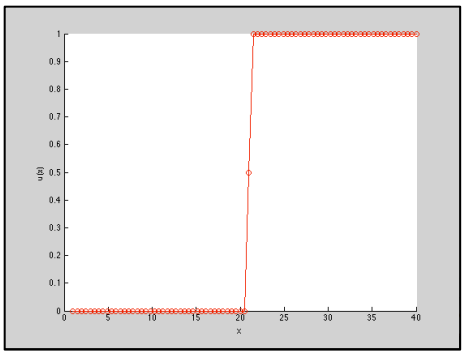
(1) $\Delta x = 1.0, \Delta t = 0.125$ at t = 1



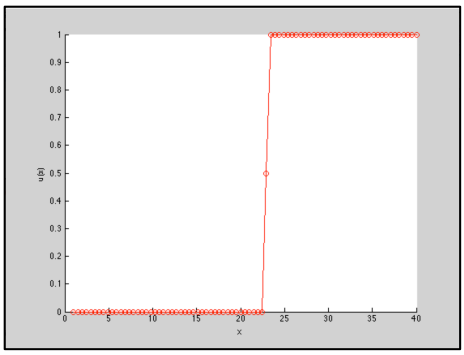
(2) $\Delta x = 1.0, \Delta t = 0.125$ at t = 5



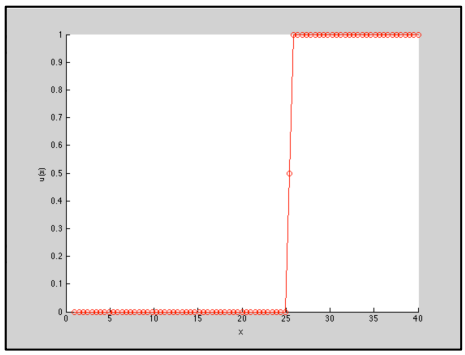
(3) $\Delta x = 1.0, \Delta t = 0.125$ at t = 10



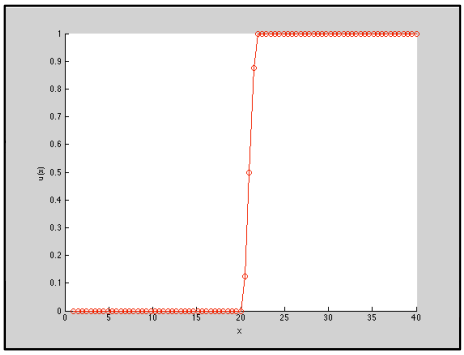
(1) $\Delta x = 0.5, \Delta t = 1.0$ at t = 1



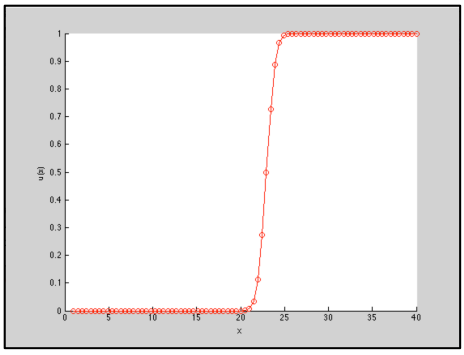
(2) $\Delta x = 0.5, \Delta t = 1.0$ at t = 5



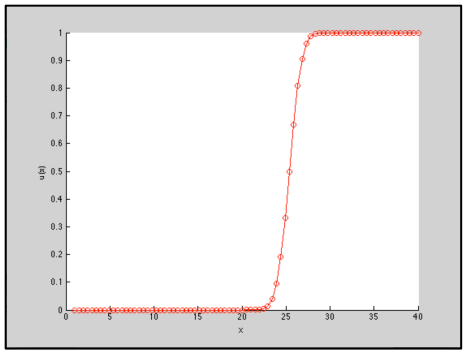
(3) $\Delta x = 0.5, \Delta t = 1.0$ at t = 10



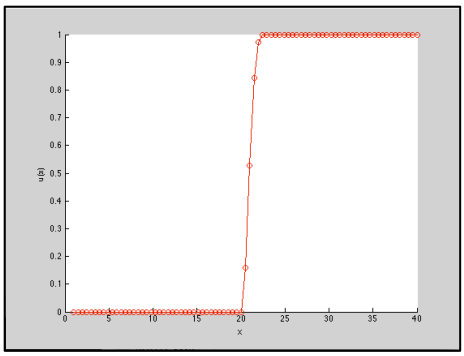
(1) $\Delta x = 0.5, \Delta t = 0.5$ at t = 1



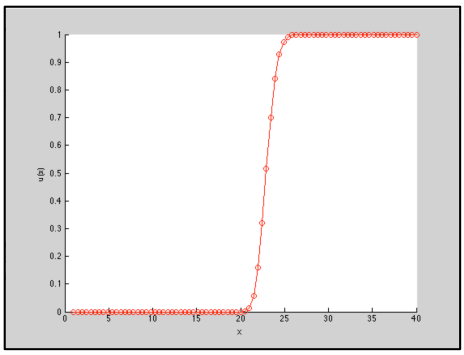
(2) $\Delta x = 0.5, \Delta t = 0.5$ at t = 5



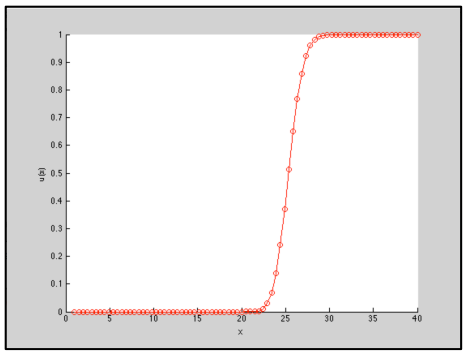
(3) $\Delta x = 0.5, \Delta t = 0.5$ at t = 10



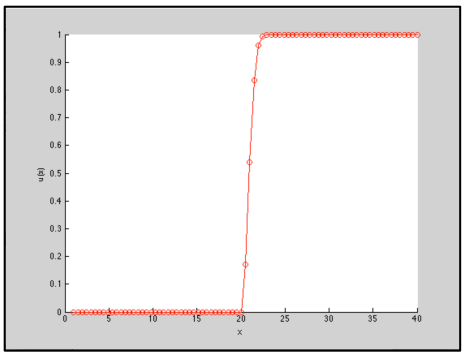
(1) $\Delta x = 0.5, \Delta t = 0.25$ at t = 1



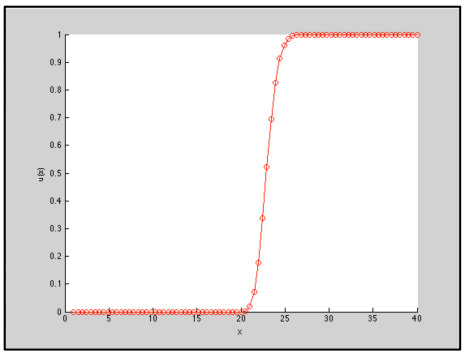
(2) $\Delta x = 0.5, \Delta t = 0.25$ at t = 5



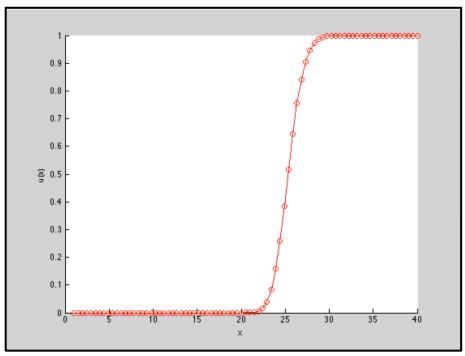
(3) $\Delta x = 0.5, \Delta t = 0.25$ at t = 10



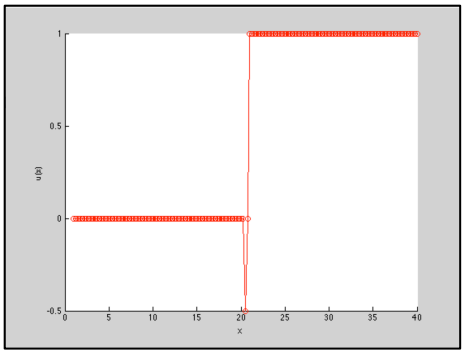
(1) $\Delta x = 0.5, \Delta t = 0.125$ at t = 1



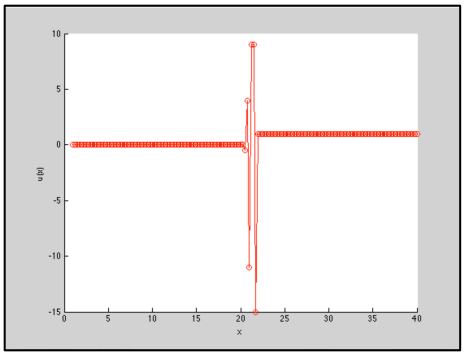
(2) $\Delta x = 0.5, \Delta t = 0.125$ at t = 5



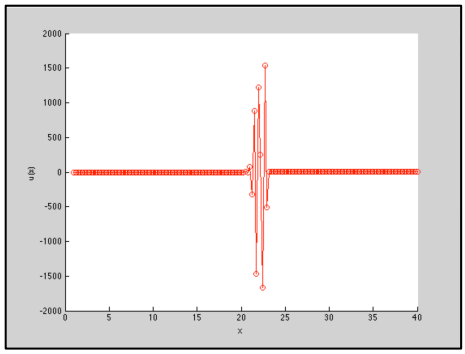
(3) $\Delta x = 0.5, \Delta t = 0.125$ at t = 10



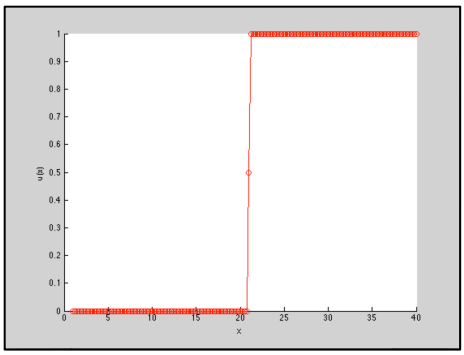
(1) $\Delta x = 0.25, \Delta t = 1.0$ at t = 1



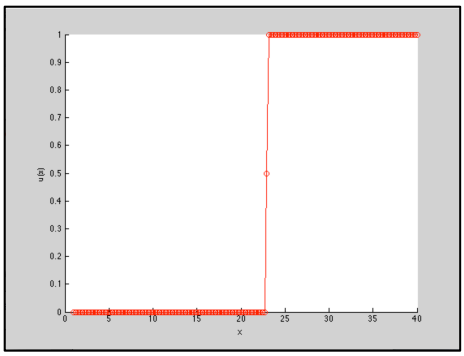
(2) $\Delta x = 0.25, \Delta t = 1.0$ at t = 5



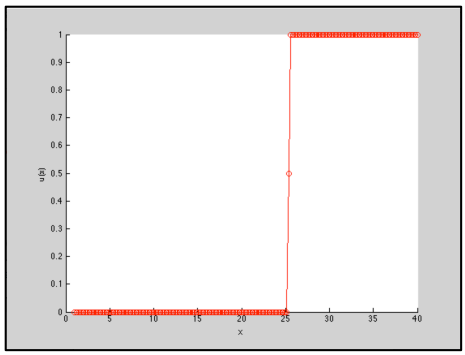
(3) $\Delta x = 0.25, \Delta t = 1.0$ at t = 10



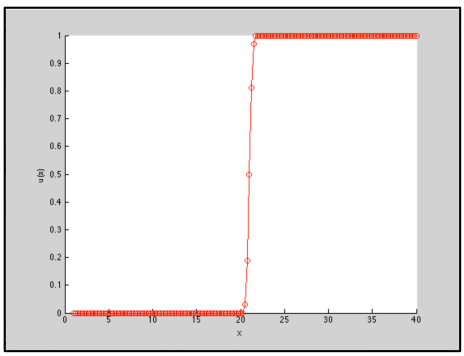
(1) $\Delta x = 0.25, \Delta t = 0.5$ at t = 1



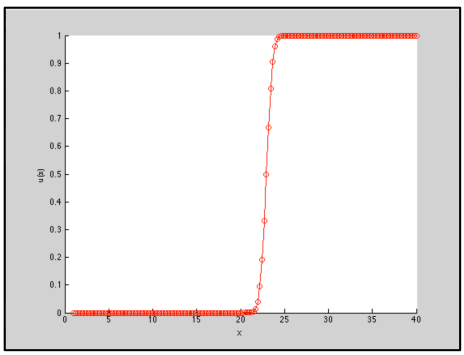
(2) $\Delta x = 0.25, \Delta t = 0.5$ at t = 5



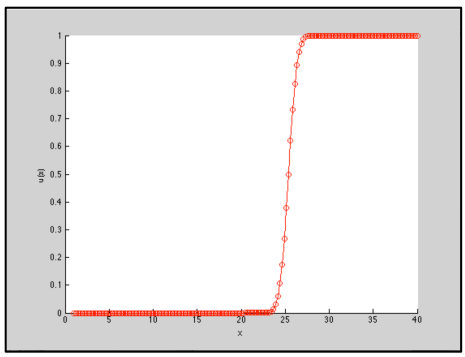
(3) $\Delta x = 0.25, \Delta t = 0.5$ at t = 10



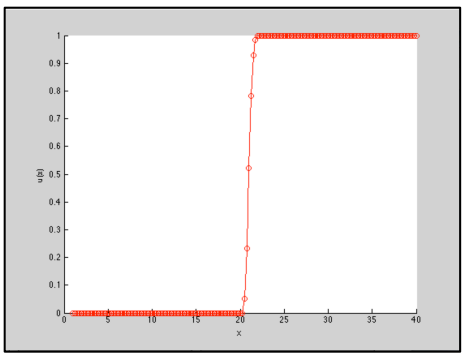
(1) $\Delta x = 0.25, \Delta t = 0.25$ at t = 1



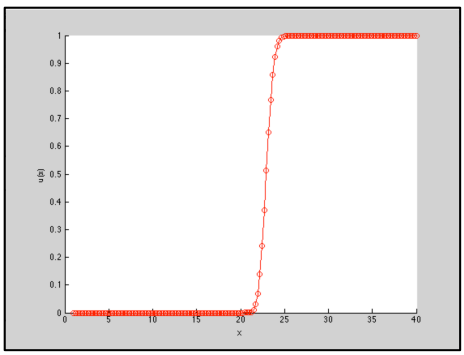
(2) $\Delta x = 0.25, \Delta t = 0.25$ at t = 5



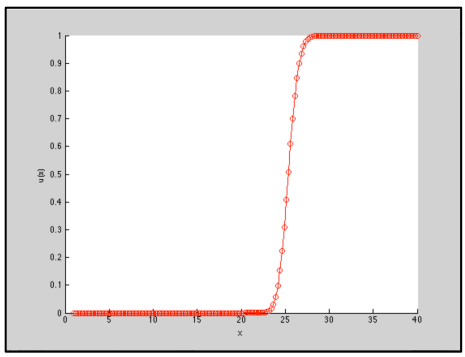
(3) $\Delta x = 0.25, \Delta t = 0.25$ at t = 10



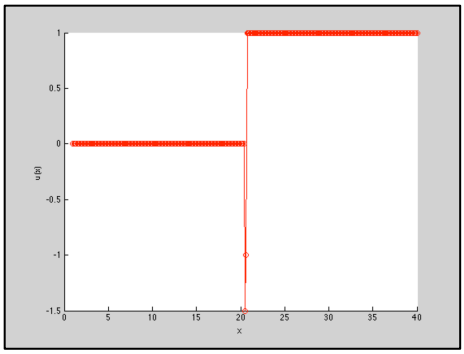
(1) $\Delta x = 0.25, \Delta t = 0.125$ at t = 1



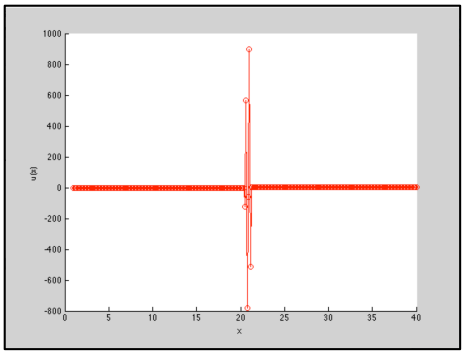
(2) $\Delta x = 0.25, \Delta t = 0.125$ at t = 5



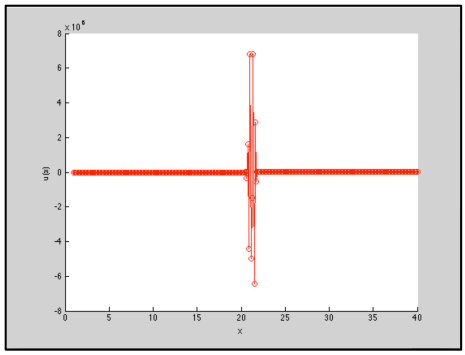
(3) $\Delta x = 0.25, \Delta t = 0.125$ at t = 10



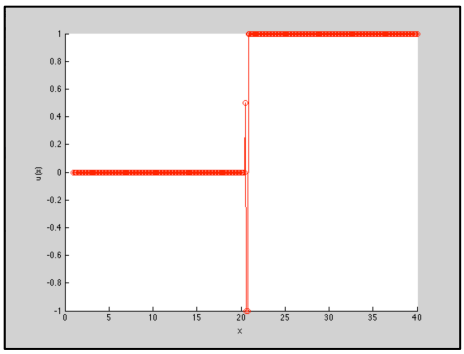
(1) $\Delta x = 0.125, \Delta t = 1.0$ at t = 1



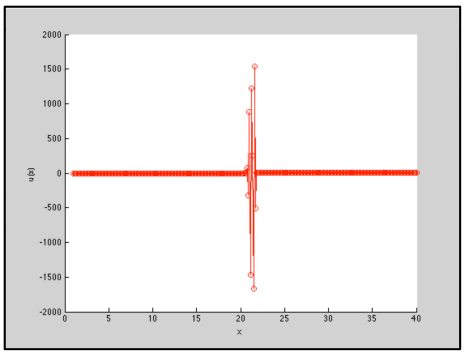
(2) $\Delta x = 0.125, \Delta t = 1.0$ at t = 5



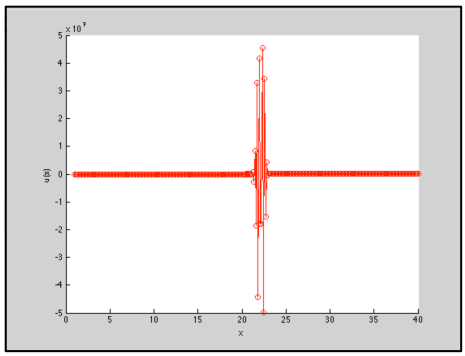
(3) $\Delta x = 0.125, \Delta t = 1.0$ at t = 10



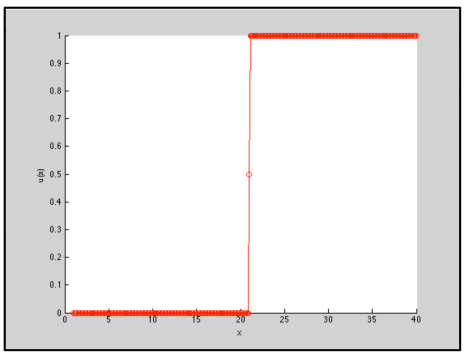
(1) $\Delta x = 0.125, \Delta t = 0.5$ at t = 1



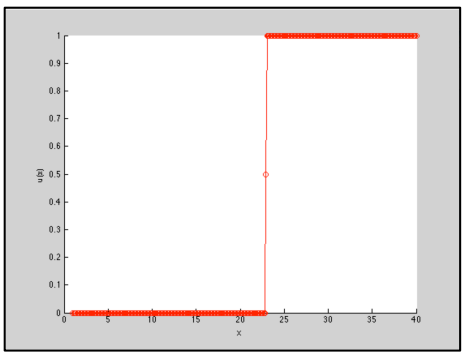
(2) $\Delta x = 0.125, \Delta t = 0.5$ at t = 5



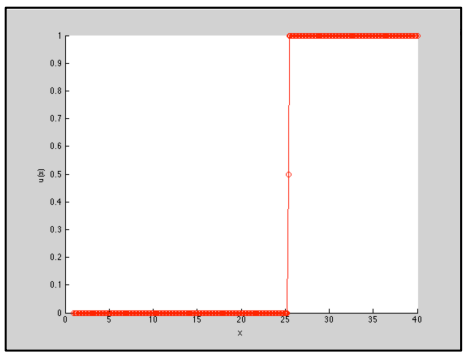
(3) $\Delta x = 0.125, \Delta t = 0.5$ at t = 10



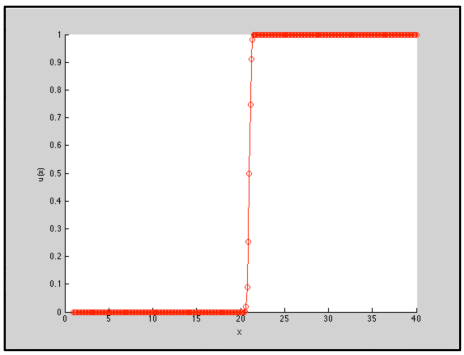
(1) $\Delta x = 0.125, \Delta t = 0.25$ at t = 1



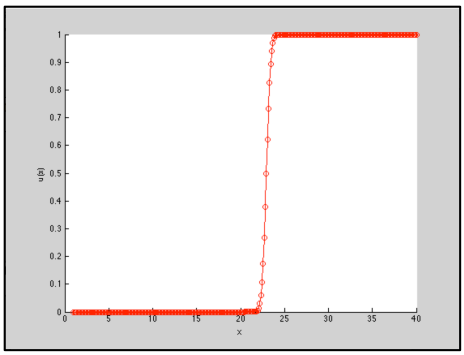
(2) $\Delta x = 0.125, \Delta t = 0.25$ at t = 5



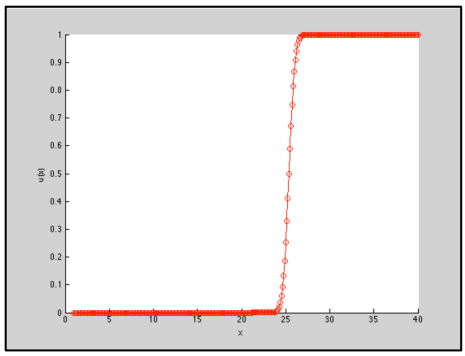
(3) $\Delta x = 0.125, \Delta t = 0.25$ at t = 10



(1) $\Delta x = 0.125, \Delta t = 0.125$ at t = 1



(2) $\Delta x = 0.125, \Delta t = 0.125$ at t = 5

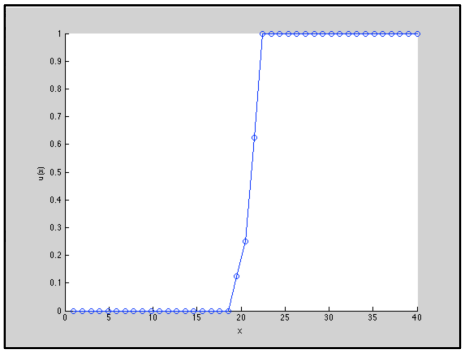


(3) $\Delta x = 0.125, \Delta t = 0.125$ at t = 10

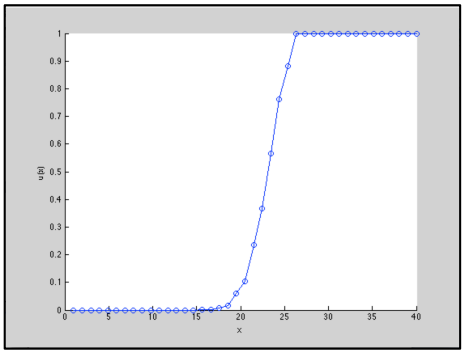
By doing the grid study for the Upwind scheme, it is verified that finer grid does not always result the accurate and exact solution. First, when the grid size of spatial domain is the same with or bigger than the grid size of time domain, reliable values are still computed and the shape of the plot does not change much as the scheme iterates to further time frames. However, it needs to be noted that much smaller grid size of time domain, compared to the grid size of spatial domain, causes the diffusion of the shape of the plot, although its effect to the shape of plot is small. Second, if the grid size of time domain is twice bigger than that of spatial domain, then the values are the closest to the exact solution and they do not diffuse. They only shift to the right in spatial domain as the scheme iterates to next time domain. Moreover, it is observed that if grid size ratio, $\frac{\Delta t}{\Delta x}$, is bigger than 2, the scheme produces completely incorrect results, so those values are no longer valid and the problem cannot be solved.

Also, in general, finer grid size of spatial domain produces values that are closer to the exact solution and the values diffuse less.

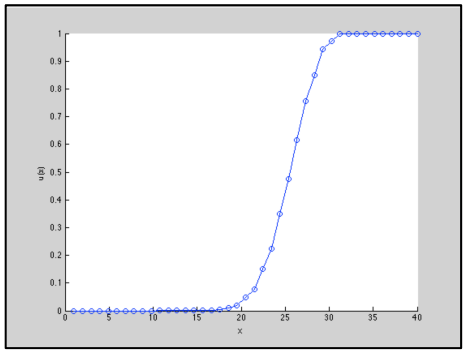
ii) Lax Scheme



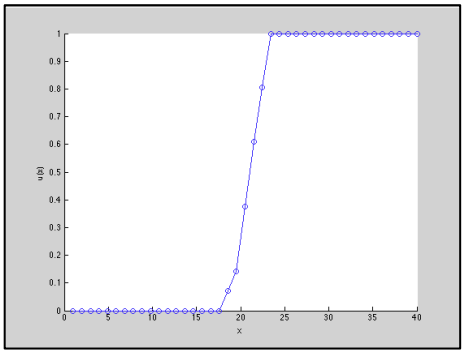
(1) $\Delta x = 1.0, \Delta t = 1.0$ at t = 1



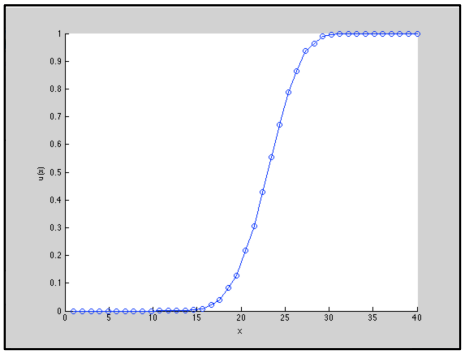
(2) $\Delta x = 1.0, \Delta t = 1.0$ at t = 5



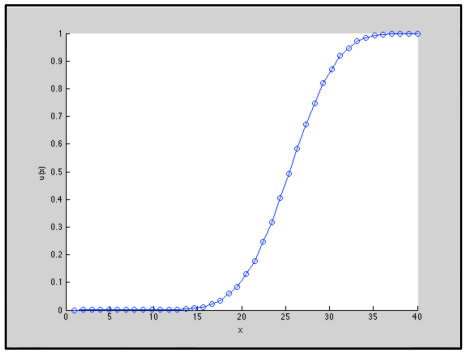
(3) $\Delta x = 1.0, \Delta t = 1.0$ at t = 10



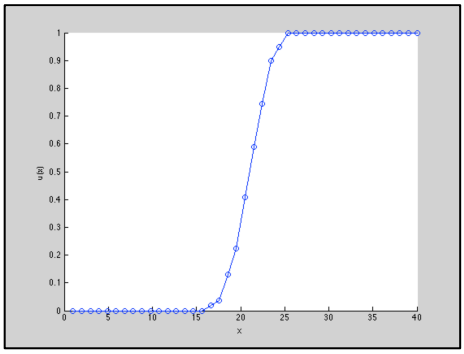
(1) $\Delta x = 1.0, \Delta t = 0.5$ at t = 1



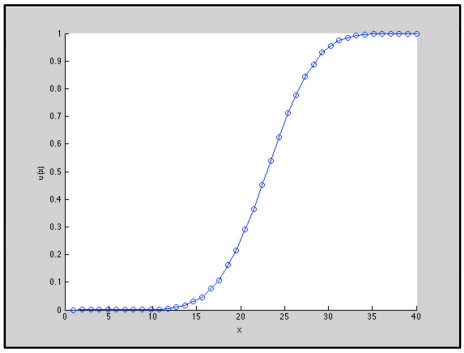
(2) $\Delta x = 1.0, \Delta t = 0.5$ at t = 5



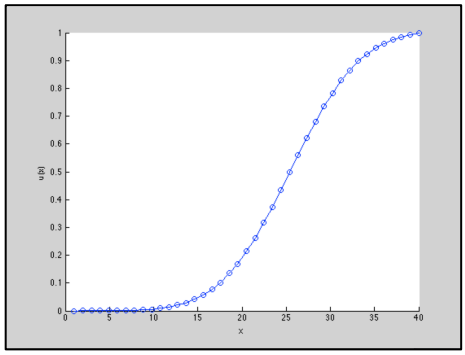
(3) $\Delta x = 1.0, \Delta t = 0.5$ at t = 10



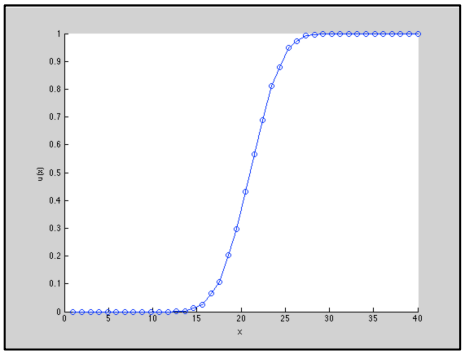
(1) $\Delta x = 1.0, \Delta t = 0.25$ at t = 1



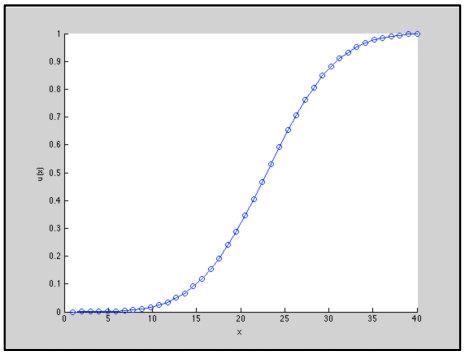
(2) $\Delta x = 1.0, \Delta t = 0.25$ at t = 5



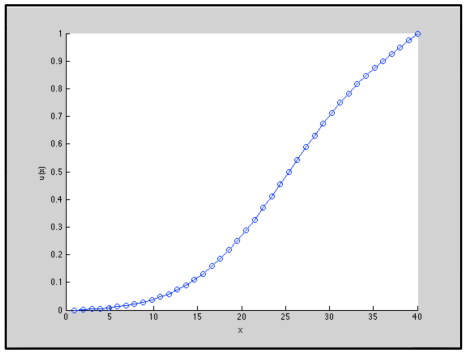
(3) $\Delta x = 1.0, \Delta t = 0.25$ at t = 10



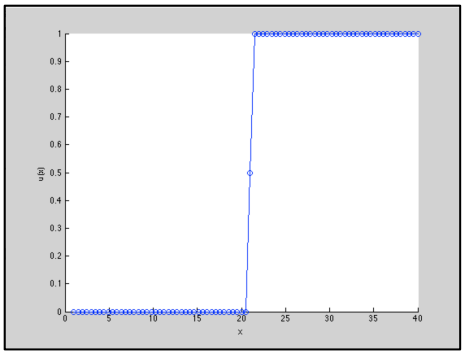
(1) $\Delta x = 1.0, \Delta t = 0.125$ at t = 1



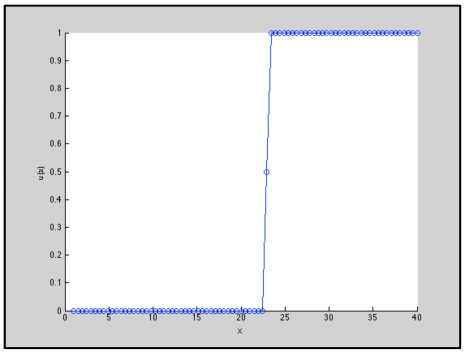
(2) $\Delta x = 1.0, \Delta t = 0.125$ at t = 5



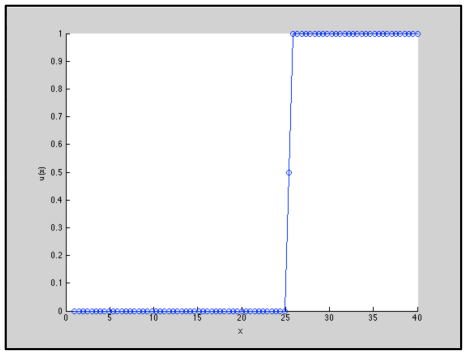
(3) $\Delta x = 1.0, \Delta t = 0.125$ at t = 10



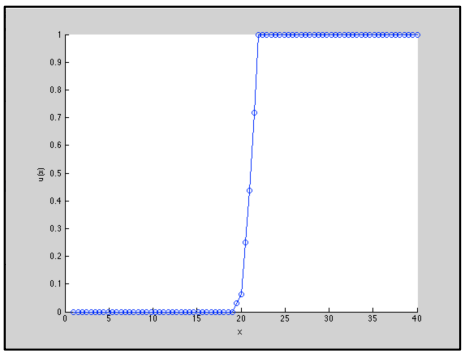
(1) $\Delta x = 0.5, \Delta t = 1.0$ at t = 1



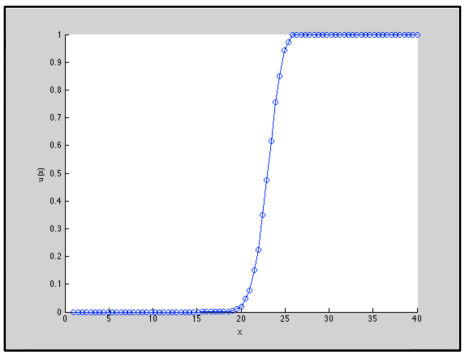
(2) $\Delta x = 0.5, \Delta t = 1.0$ at t = 5



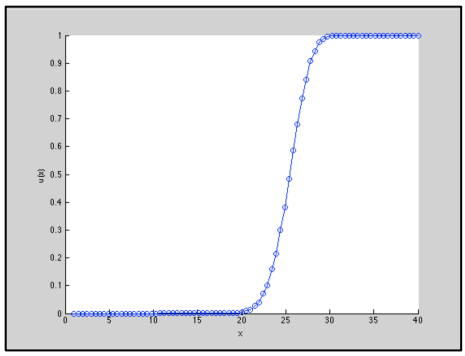
(3) $\Delta x = 0.5, \Delta t = 1.0$ at t = 10



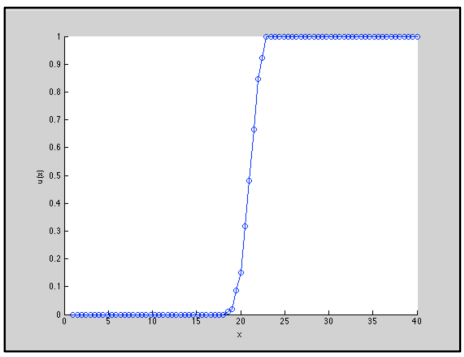
(1) $\Delta x = 0.5, \Delta t = 0.5$ at t = 1



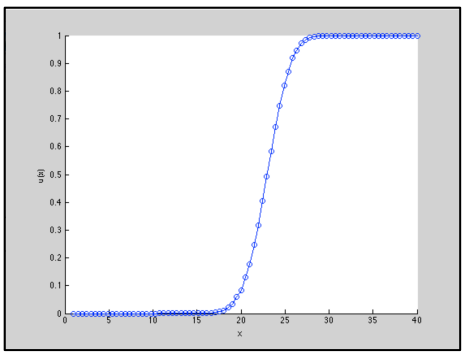
(2) $\Delta x = 0.5, \Delta t = 0.5$ at t = 5



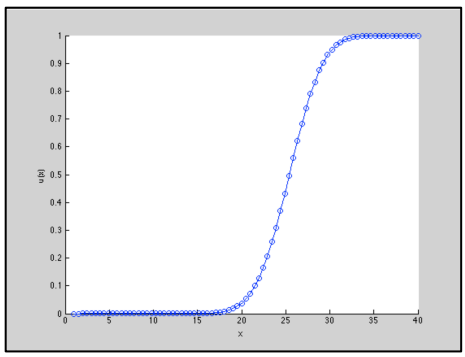
(3) $\Delta x = 0.5, \Delta t = 0.5$ at t = 10



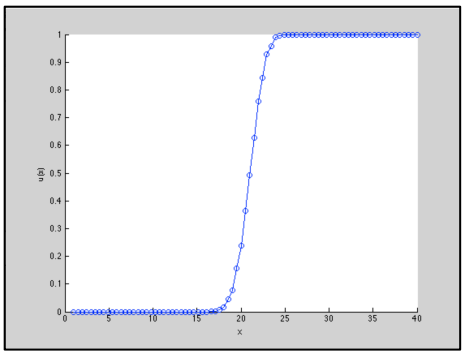
(1) $\Delta x = 0.5, \Delta t = 0.25$ at t = 1



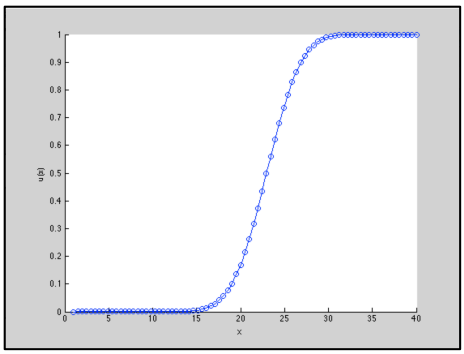
(2) $\Delta x = 0.5, \Delta t = 0.25$ at t = 5



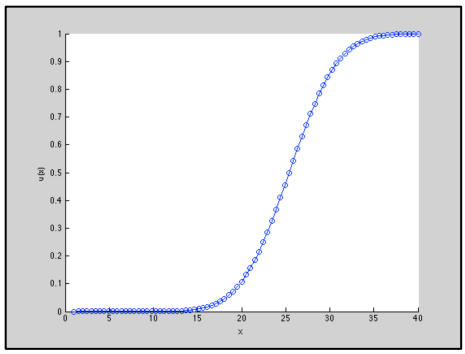
(3) $\Delta x = 0.5, \Delta t = 0.25$ at t = 10



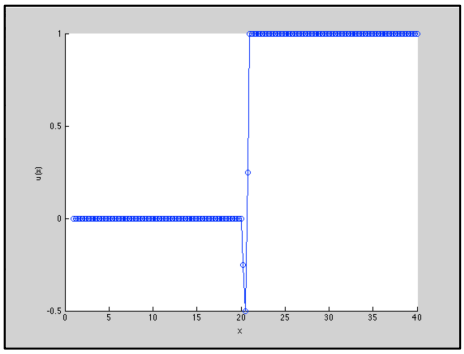
(1) $\Delta x = 0.5, \Delta t = 0.125$ at t = 1



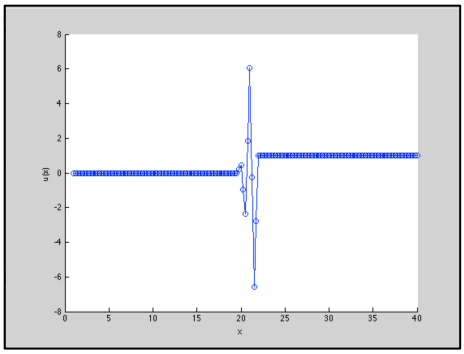
(2) $\Delta x = 0.5, \Delta t = 0.125$ at t = 5



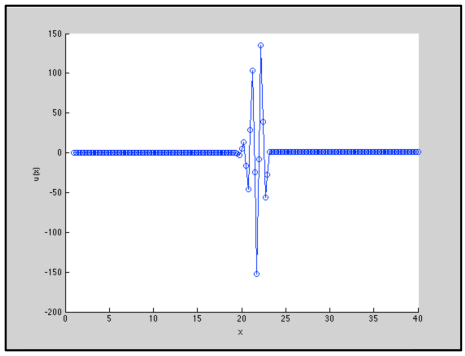
(3) $\Delta x = 0.5, \Delta t = 0.125$ at t = 10



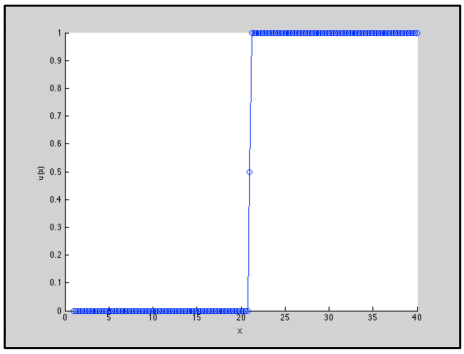
(1) $\Delta x = 0.25, \Delta t = 1.0$ at t = 1



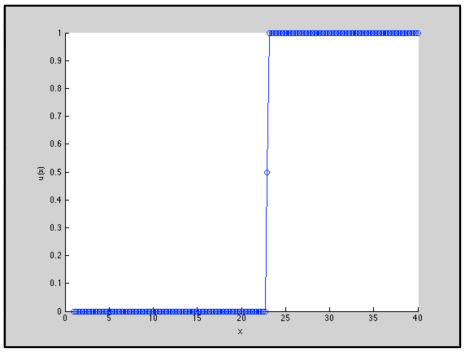
(2) $\Delta x = 0.25, \Delta t = 1.0$ at t = 5



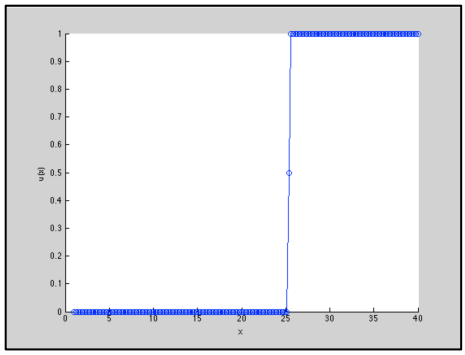
(3) $\Delta x = 0.25, \Delta t = 1.0$ at t = 10



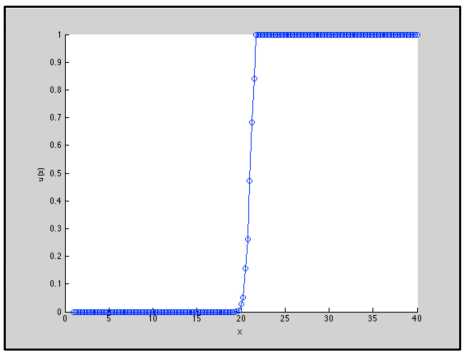
(1) $\Delta x = 0.25, \Delta t = 0.5$ at t = 1



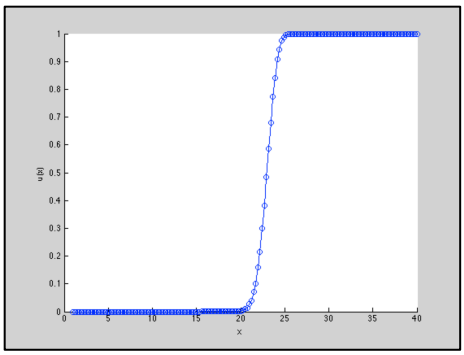
(2) $\Delta x = 0.25, \Delta t = 0.5$ at t = 5



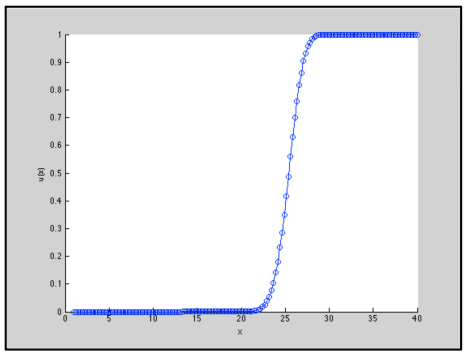
(3) $\Delta x = 0.25, \Delta t = 0.5$ at t = 10



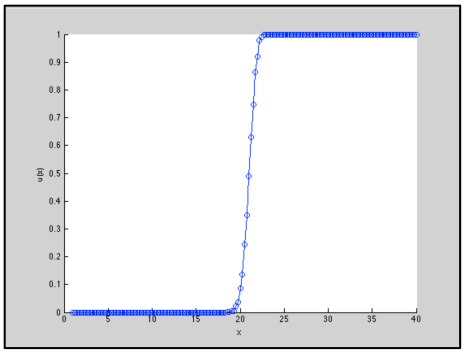
(1) $\Delta x = 0.25, \Delta t = 0.25$ at t = 1



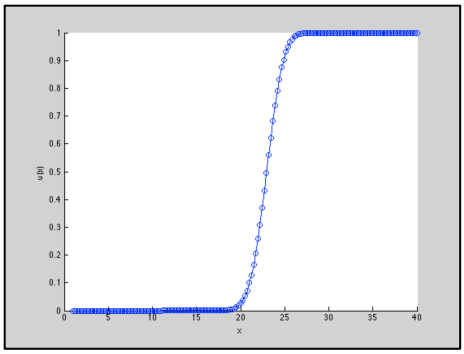
(2) $\Delta x = 0.25, \Delta t = 0.25$ at t = 5



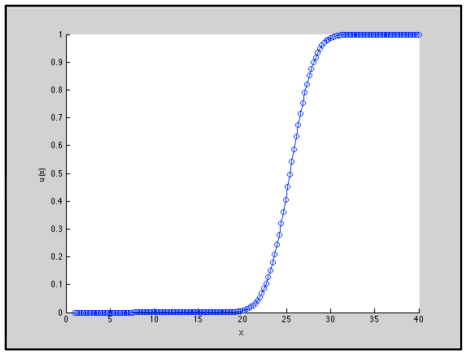
(3) $\Delta x = 0.25, \Delta t = 0.25$ at t = 10



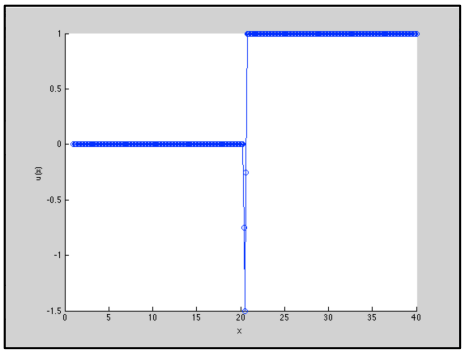
(1) $\Delta x = 0.25, \Delta t = 0.125$ at t = 1



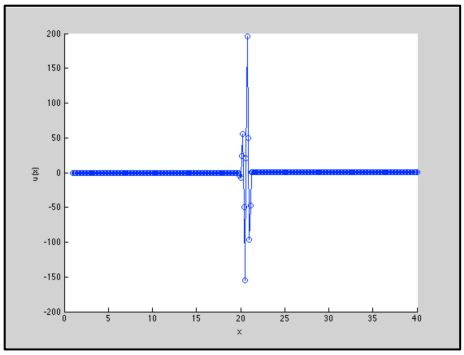
(2) $\Delta x = 0.25, \Delta t = 0.125$ at t = 5



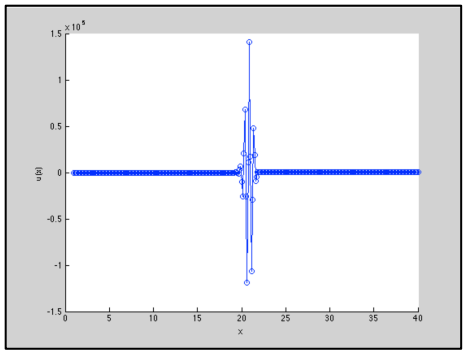
(3) $\Delta x = 0.25, \Delta t = 0.125$at t = 10



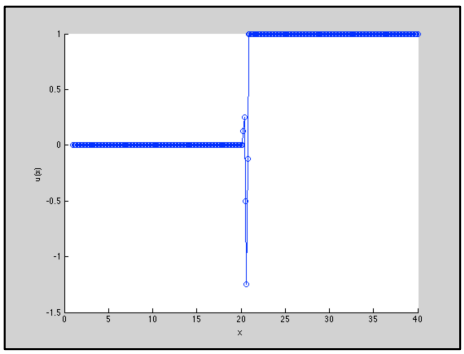
(1) $\Delta x = 0.125, \Delta t = 1.0$ at t = 1



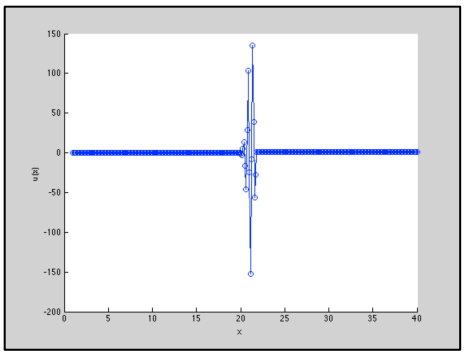
(2) $\Delta x = 0.125, \Delta t = 1.0$ at t = 5



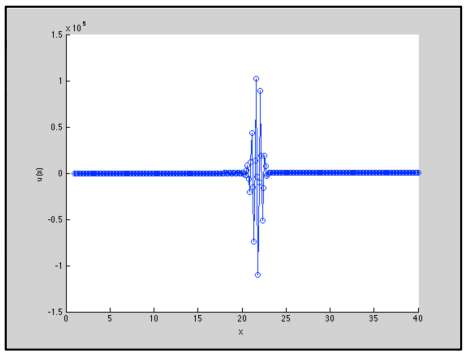
(3) $\Delta x = 0.125, \Delta t = 1.0$ at t = 10



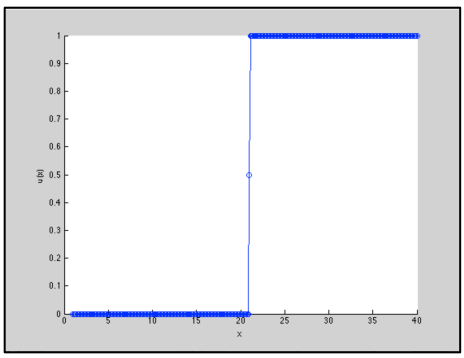
(1) $\Delta x = 0.125, \Delta t = 0.5$ at t = 1



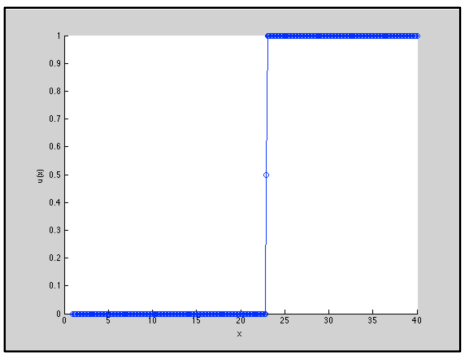
(2) $\Delta x = 0.125, \Delta t = 0.5$ at t = 5



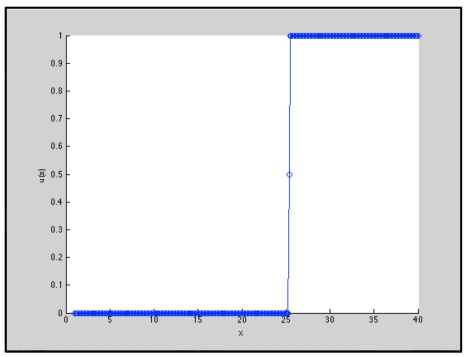
(3) $\Delta x = 0.125, \Delta t = 0.5$ at t = 10



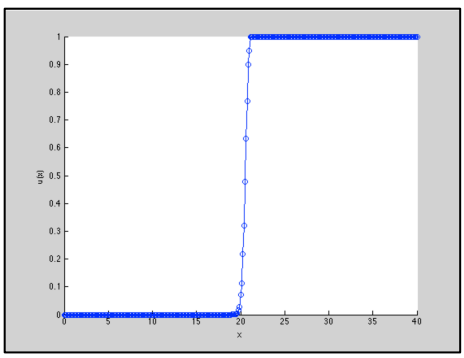
(1) $\Delta x = 0.125, \Delta t = 0.25$ at t = 1



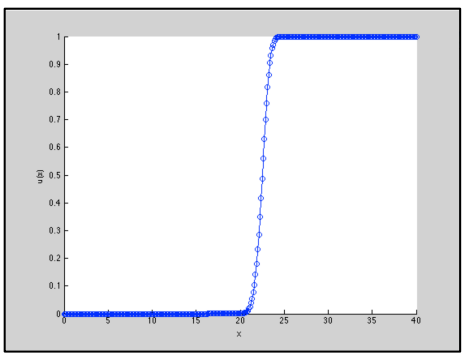
(2) $\Delta x = 0.125, \Delta t = 0.25$ at t = 5



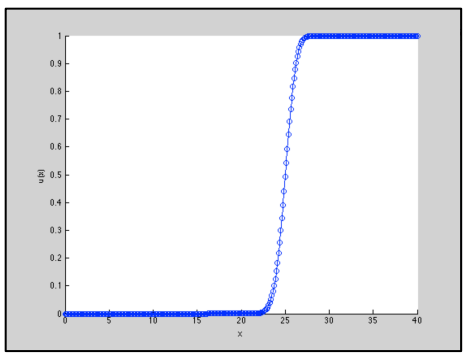
(3) $\Delta x = 0.125, \Delta t = 0.25$ at t = 10



(1) $\Delta x = 0.125, \Delta t = 0.125$ at t = 1



(2) $\Delta x = 0.125, \Delta t = 0.125$ at t = 5



(3) $\Delta x = 0.125, \Delta t = 0.125$ at t = 10

The behavior of values, computed by the Lax scheme, is the same with the behavior of computed values by the Upwind scheme, but the values, computed by Lax method, diffuses more severely. Generally, if the grid size ratio, $\frac{\Delta t}{\Delta x}$, is between 0 and 2, the computed values are still reliable, but they diffuse as the scheme iterates. Specifically, the larger grid size ratio between 0 and 2 produces the values close to the exact solution, while the smaller value of the grid size ratio (Δt is much smaller than Δx), the larger diffusion of values occurs and its magnitude is quite big enough to arise the issue in reliability of the computed values. Compare to the case, computed by Upwind scheme, at the same grid resolution, the computed values by the Lax scheme, more largely diffuse. And, if the grid size ratio, $\frac{\Delta t}{\Delta x}$, is equal to the 2, then the Lax scheme produces the closest values to the exact solution, and the computed values do not diffuse as time goes on. Lastly, if the grid size ratio, $\frac{\Delta t}{\Delta x}$, is bigger than 2, the scheme produces completely wrong results, so the scheme is no longer reliable for such case.

To sum up, the higher resolution of grid of domain does not always give the accurate results. The scheme with a grid size ratio, $\frac{\Delta t}{\Delta x}$, within the specific range gives the reliable results, but the diffusion is still an issue. In detail, large grid size ratio between the ranges produces values close to the exact solution, while small grid size ratio between the ranges results less reliable values and larger diffusion is occurred. Moreover, grid size ratio at the value of boundary of range (0≤ ratio ≤ **N**) produces the most accurate values and grid size ratio beyond that range gives completely incorrect results.

** In this case, $\nu = A\frac{\Delta t}{\Delta x} = \frac{1}{2}\frac{\Delta t}{\Delta x}$, so if $\frac{\Delta t}{\Delta x} = \frac{2}{1}$, then ν becomes equal to 1. It is why the grid size ratio, $\frac{\Delta t}{\Delta x}$, that equals to 2 gives the most accurate results without the diffusion along the time frame.

Q 3.

* Von Neumann Stability Analysis

i) Upwind Scheme

$$U_j^{n+1} - U_j^n = \frac{A\Delta t}{\Delta x}\left(U_{j-1}^n - U_j^n\right)$$

$* U_j^n = D + \epsilon$ (D: exact solution to FDE; ϵ: Error (Round-off))

$$\hookrightarrow D_j^{n+1} + \epsilon_j^{n+1} - D_j^n - \epsilon_j^n = A\frac{\Delta t}{\Delta x}\left(D_{j-1}^n + \epsilon_{j-1}^n - D_j^n - \epsilon_j^n\right)$$

* D's can be canceled because they are the exact solution to the FDE

$$\epsilon_j^{n+1} - \epsilon_j^n = A\frac{\Delta t}{\Delta x}\left(\epsilon_{j-1}^n - \epsilon_j^n\right) \quad \rightarrow \text{Use Fourier transformation}$$

then, $\epsilon(x,t) = \sum_m b_m(t)\, e^{ik_m x}$ $\quad * k_m = \left(\frac{2\pi}{L}\right)m$, $m = 0, 1, \ldots M$

Since $b_m(t)$ is only function of t and $e^{ik_m x}$ is only function of x, it can be assumed that $\epsilon(x,t) = \sum e^{at} e^{ik_m x} = \sum e^{(at + ik_m x)}$ ↳ * Subs *

$$\Rightarrow e^{a(t+\Delta t)} e^{ik_m x} - e^{at} e^{ik_m x} = A\frac{\Delta t}{\Delta x}\left(e^{at} e^{ik_m(x-\Delta x)} - e^{at} e^{ik_m x}\right)$$

* Divide equation by $\left(e^{at} e^{ik_m x}\right)$

$$e^{a\Delta t} - 1 = A\frac{\Delta t}{\Delta x}\left(e^{-ik_m \Delta x} - 1\right) \qquad * \text{Let } A\frac{\Delta t}{\Delta x} = \nu$$

$$e^{a\Delta t} = \nu\left(e^{-ik_m \Delta x} - 1\right) + 1$$

$* \epsilon_j^{n+1} = e^{a\Delta t}\epsilon_j^n \rightarrow \epsilon_j^{n+1} \le e^{a\Delta t}\epsilon_j^n \Rightarrow |e^{a\Delta t}| \le 1$

$$\hookrightarrow \left|1 + \nu\left(e^{-ik_m \Delta x} - 1\right)\right| \le 1$$

$$\left|1 - \nu + \nu \cdot e^{-ik_m \Delta x}\right| \le 1 \qquad \rightarrow * \text{Use complex trig function identity}$$

$$* e^{-ik_m \Delta x} = \cos(k_m \Delta x) - i \sin(k_m \Delta x)$$

$$\rightarrow \left[1-\nu+\nu\cos(k_m\Delta x)\right]^2+\nu^2\sin^2(k_m\Delta x) \le 1$$

$$\left[1-2\nu\sin^2\left(\frac{k_m\Delta x}{2}\right)\right]^2+4\nu^2\sin^2\left(\frac{k_m\Delta x}{2}\right)\cos^2\left(\frac{k_m\Delta x}{2}\right) \le 1$$

$$1-4\nu(1-\nu)\underbrace{\sin^2\left(\frac{k_m\Delta x}{2}\right)}_{0\le\ \le 1} \le 1$$

$$\Rightarrow \quad 1-4\nu(1-\nu)\le 1 \quad \rightarrow \quad 4\nu^2-4\nu+1\le 1$$

$$\rightarrow \quad (2\nu-1)^2\le 1 \quad \rightarrow \quad 2\nu-1\le 1$$

$$2\nu\le 2 \quad \Rightarrow \quad \underline{0\le\nu\le 1}$$

∴ Thus, Upwind scheme is stable as long as $0\le A\frac{\Delta t}{\Delta x}\le 1$

* in our case, since $A=\frac{1}{2}$ $\Rightarrow$ $0\le\frac{\Delta t}{\Delta x}\le 2$

ii) Lax-Wendroff scheme * $U_j^n = D + \epsilon$

$$U_j^{n+1} - U_j^n = \frac{1}{2}\left(A\frac{\Delta t}{\Delta x}\right)^2 \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right) - \frac{1}{2}A\frac{\Delta t}{\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right)$$

↳ $D_j^{n+1} + \epsilon_j^{n+1} - D_j^n - \epsilon_j^n = \frac{1}{2}\left(A\frac{\Delta t}{\Delta x}\right)^2 \left(D_{j+1}^n + \epsilon_{j+1}^n - 2D_j^n - 2\epsilon_j^n + D_{j-1}^n + \epsilon_{j-1}^n\right)$

$$- \frac{1}{2}A\frac{\Delta t}{\Delta x}\left(D_{j+1}^n + \epsilon_{j+1}^n - D_{j-1}^n - \epsilon_{j-1}^n\right)$$

$$\epsilon_j^{n+1} - \epsilon_j^n = \frac{1}{2}\left(A\frac{\Delta t}{\Delta x}\right)^2 \left(\epsilon_{j+1}^n - 2\epsilon_j^n + \epsilon_{j-1}^n\right) - \frac{1}{2}A\frac{\Delta t}{\Delta x}\left(\epsilon_{j+1}^n - \epsilon_{j-1}^n\right)$$

↳ Use Fourier Transformation & Let $A\frac{\Delta t}{\Delta x} = \nu$

↳ $\epsilon(x,t) = \sum_m b_m(t) \cdot e^{ik_m x}$

Assume $b_m(t) = e^{at}$, and then $\epsilon(x,t) = \sum e^{at} e^{ik_m x} = \sum e^{(at + ik_m x)}$

↳ * Subs *

$$\Rightarrow e^{a(t+\Delta t)} e^{ik_m x} - e^{at} e^{ik_m x} = \frac{1}{2}\nu^2 \left(e^{at} e^{ik_m(x+\Delta x)} - 2e^{at} e^{ik_m x} + e^{at} e^{ik_m(x-\Delta x)}\right)$$

$$- \frac{1}{2}\nu\left(e^{at} e^{ik_m(x+\Delta x)} - e^{at} e^{ik_m(x-\Delta x)}\right)$$

* Divide equation by $e^{at} e^{ik_m x}$

$$\rightarrow e^{a\Delta t} - 1 = \frac{1}{2}\nu^2\left(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x}\right) - \frac{1}{2}\nu\left(e^{ik_m\Delta x} - e^{-ik_m\Delta x}\right)$$

* $\epsilon_j^{n+1} = e^{a\Delta t}\epsilon_j^n \rightarrow \epsilon_j^{n+1} \le e^{a\Delta t}\epsilon_j^n \Rightarrow |e^{a\Delta t}| \le 1$

↳ $e^{a\Delta t} = \frac{1}{2}\nu^2\left(e^{ik_m\Delta x} + e^{-ik_m\Delta x} - 2\right) - \frac{1}{2}\nu\left(e^{ik_m\Delta x} - e^{-ik_m\Delta x}\right) + 1$

* Use Complex trig function identity: $\begin{cases} \cos(k_m\Delta x) = \dfrac{e^{ik_m\Delta x} + e^{-ik_m\Delta x}}{2} \\ \sin(k_m\Delta x) = \dfrac{e^{ik_m\Delta x} - e^{-ik_m\Delta x}}{2i} \end{cases}$

$$e^{a\Delta t} = \frac{1}{2}\nu^2\left(2\cos(k_m\Delta x) - 2\right) - \frac{1}{2}\nu\left(2\cdot i\sin(k_m\Delta x)\right) + 1$$

$$\left|e^{a\Delta t}\right| = \left|\nu^2\left[\cos(k_m\Delta x) - 1\right] - \nu\cdot i\sin(k_m\cdot\Delta x) + 1\right|$$

$$= \left|1 - \nu^2\left[1 - \cos(k_m\Delta x)\right] - i\cdot\nu\cdot\sin(k_m\cdot\Delta x)\right|$$

$$= \left|1 - 2\nu^2\sin^2\left(\frac{k_m\cdot\Delta x}{2}\right) - i\nu\sin(k_m\cdot\Delta x)\right|$$

$$\left|e^{a\Delta t}\right| = \left|1 - 2\nu^2\sin^2\left(\frac{k_m\cdot\Delta x}{2}\right) - i\nu\sin(k_m\cdot\Delta x)\right|$$

$$= 1 + 4\nu^4\sin^4\left(\frac{k_m\cdot\Delta x}{2}\right) - 2^2\nu^2\sin^4\left(\frac{k_m\cdot\Delta x}{2}\right)$$

$$= 1 - 4\nu^2\underbrace{\sin^4\left(\frac{k_m\cdot\Delta x}{2}\right)}_{\leq 1}\cdot\left(1-\nu^2\right)$$

$$\left|e^{a\Delta t}\right| = 1 - 4\nu^2(1-\nu^2) \leq 1$$

$$4\nu^4 - 4\nu^2 + 1 \leq 1 \quad \rightarrow \quad (2\nu^2 - 1)^2 \leq 1$$

$$2\nu^2 - 1 \leq 1 \quad \rightarrow \quad 2\nu^2 \leq 2 \quad \rightarrow \quad \nu^2 \leq 1$$

$$\Rightarrow \quad 0 \leq \nu \leq 1$$

∴ Thus, Lax-Wendroff scheme is stable as long as $|\nu| = |A|\frac{\Delta t}{\Delta x} \leq 1$

* In our case, since $A = \frac{1}{2}$ $\Rightarrow$ $0 \leq \frac{\Delta t}{\Delta x} \leq 2$

Q4.

* Consistency Analysis

* Wave equation : $\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u}{\partial x} = 0$

Scheme : Upwind scheme $\rightarrow$ $\frac{u_j^{n+1} - u_j^n}{\Delta t} = -A\left(\frac{u_j^n - u_{j-1}^n}{\Delta x}\right)$

$\rightarrow$ Start with Taylor Series.

Time

$$u(x,t+\Delta t) = u(x,t) + \frac{\partial u}{\partial t}(\Delta t) + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(\Delta t)^2 + \frac{1}{3!}\frac{\partial^3 u}{\partial t^3}(\Delta t)^3 + \cdots$$

$$\hookrightarrow \frac{\partial u}{\partial t} = \underbrace{\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t}}_{FDE} - \underbrace{\frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(\Delta t) - \frac{1}{3!}\frac{\partial^3 u}{\partial t^3}(\Delta t)^2 - \cdots}_{T.E}$$

Space

$$u(x-\Delta x, t) = u(x,t) + \frac{\partial u}{\partial x}(-\Delta x) + \frac{1}{2!}\frac{\partial^2 u}{\partial x^2}(-\Delta x)^2 + \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(-\Delta x)^3 + \cdots$$

$$\hookrightarrow \frac{\partial u}{\partial x} = \underbrace{\frac{u(x,t) - u(x-\Delta x,t)}{\Delta x}}_{} + \underbrace{\frac{1}{2!}\frac{\partial^2 u}{\partial x^2}(\Delta x) - \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}(\Delta x)^2 + \cdots}_{}$$

$$\hookrightarrow \frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u}{\partial x} = \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + \frac{1}{2}\left(\frac{u(x,t) - u(x-\Delta x,t)}{\Delta x}\right) \rightarrow FDE$$

$$= \left[-\underbrace{\frac{\Delta t}{2!}\frac{\partial^2 u}{\partial t^2}} + \frac{1}{2}\cdot\frac{\Delta x}{2!}\frac{\partial^2 u}{\partial x^2} - \underbrace{\frac{(\Delta t)^2}{3!}\frac{\partial^3 u}{\partial t^3}} - \frac{1}{2}\cdot\frac{(\Delta x)^2}{3!}\frac{\partial^3 u}{\partial x^3} + \cdots\right]$$

$\hookrightarrow$ T.E

$$\lim_{\Delta t \to 0}\left[\frac{\Delta t}{2!}\frac{\partial^2 u}{\partial t^2}\right] \rightarrow 0$$

$$\lim_{\Delta x \to 0}\left[\frac{1}{2}\cdot\frac{\Delta x}{2!}\frac{\partial^2 u}{\partial x^2}\right] \rightarrow 0$$

$$\rightarrow \lim_{\Delta t \to 0,\, \Delta x \to 0}[T.E] \rightarrow 0$$

$\therefore$ Upwind scheme is consistent