# Linear Convection

## Given Data:

1. Length L=1m
2. Convection velocity, C = 1 and  time = 0.4seconds
3. Number of nodes n=20,40,80,160
4. The initial velocity profile is a step function. It is equal to 2m/s between x= 0.1 and 0.3 and 1m/s everywhere else
5. Time step = 0.01

## Objective:

For the given data:

1. Compare the original and final velocity profiles.
2. Use first-order forward differencing for the time derivative and first-order backward/rearward differencing for the space term
3. Make the comparison for the values of 'n'
4. Explanation of the plot.
5. Detailed explanation for your results, plots or codes without explanation will be awarded zero marks.

## Procedure:

## 1D Linear convection equation

$$\frac{\partial u}{\partial t} + c * \frac{\partial u}{\partial x} = 0$$

Here,c - convection velocity

Applying first-order forward differencing for the time derivative and first-order backward/rearward differencing for the space term,the equation is as follows

$$u_i^{n+1} = u_i^n - c * \frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right)$$

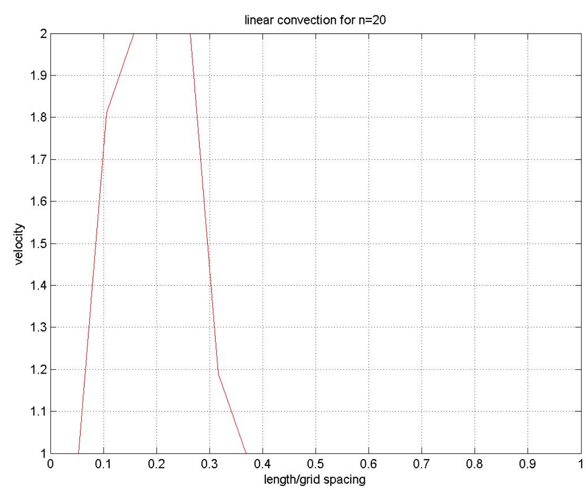
## Program:

```
clear all
close all
clc
%initial variables
L=1;
n=20;
c=1;
dt=0.01;
x=linspace(0,L,n);
dx=L/(n-1);
%start & end of wave
x_start=0.1;
x_stop=0.3;
%range of velocity u=2
a= (x_start/dx)+1;
b=(x_stop/dx)+1;
%initiating arrays & bc
u=1*ones(1,n);
u(a:b)=2;
%bc
u(1)=1;
uold=u;
u_initial=u;
%time fn
t=0.4;
for k=1:(n*t)
%space fn
for i=2:n
```

```
    u(i)=uold(i)-(c*dt/dx)*(uold(i)-uold(i-1));
end
%updating new velocity
uold=u;
 %plotting
h=figure;
 plot(x,u,'r');
 title('linear convection for n=20')
 xlabel('length/grid spacing');
 ylabel('velocity');
 saveas(h,sprintf('FIG%d.png',k));
end
```

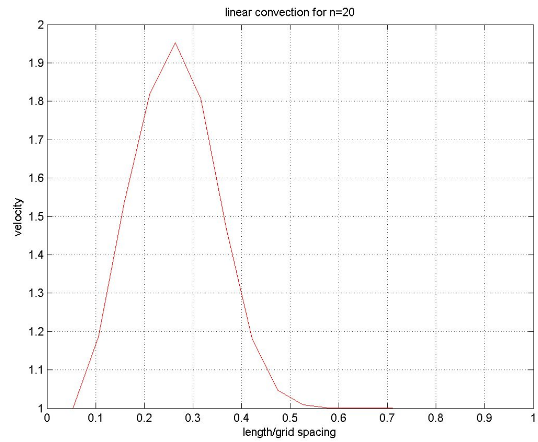
# Output:

## 1. When n=20, Initial profile



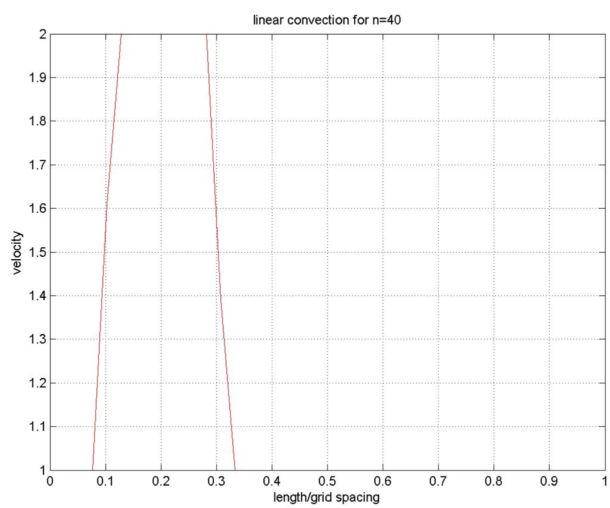

Number of nodes =20, this means that the number of nodes is less,hence the velocity drops to 1.5m/s at 0.3 and starts to behave like a sine function instead of square. The abrupt profile at the top is also due to a phenomenon called artificial diffusion.

## 2. When n=20, Final profile



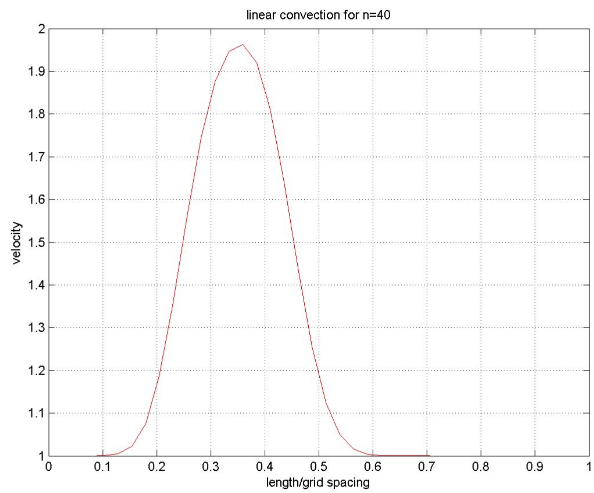

Since the number of nodes is less, the final profile looks sharper and distorted from the initial profile.

## 3. When n=40, Initial profile



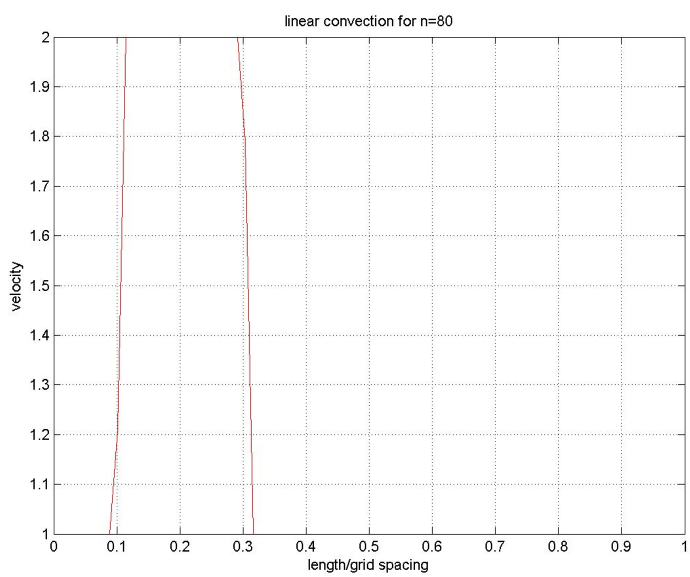

As the number of nodes is increased, the graph looks much sharper and squarer and the value of velocity drops at a slower rate.

## 4. When n=40, Final profile



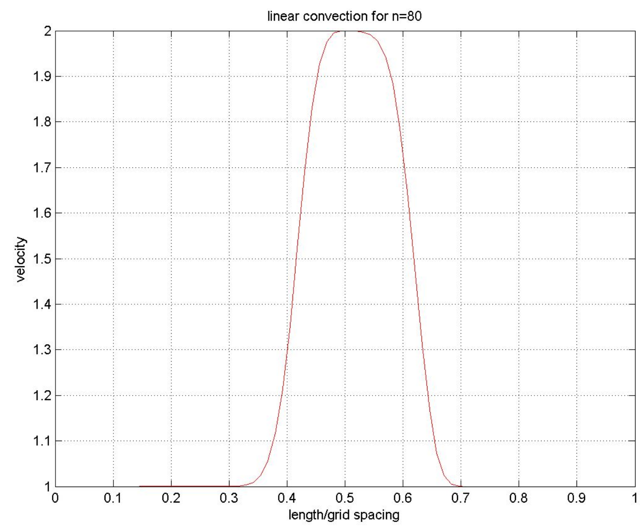

The final profile looks much smoother than the n=20 profile due to increased nodes.

## 5. When n=80, Initial profile



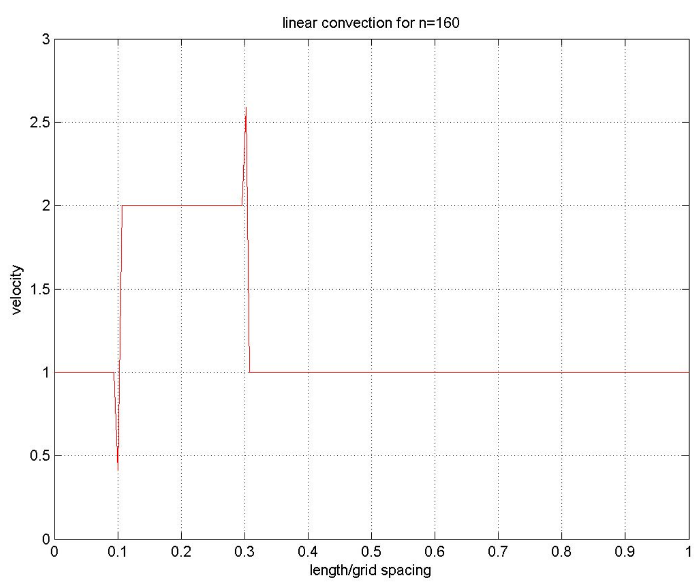

The plot looks squarer and more closer to velocity=2m/s than the initial plot when n=40. This looks almost a perfect solution due to the increased number of nodes.

## 6. <u>When n=80, Final profile</u>



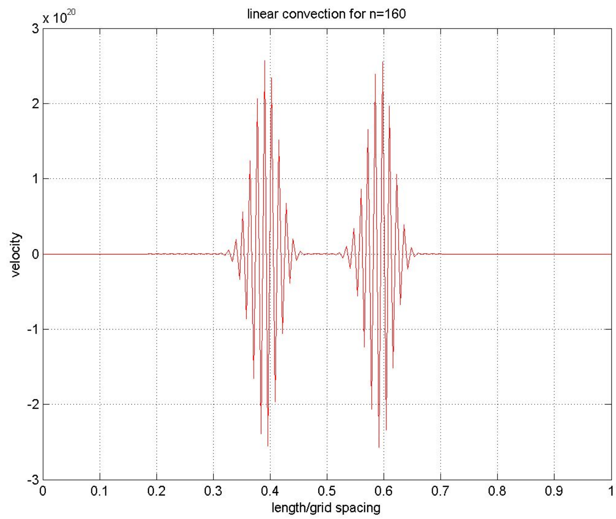

Due to increased nodes to 80, the final profile looks much more smoother and uniform.

## 7. <u>When n=160, Initial profile</u>




The profile has more number of nodes which blows up the solution. There are more wiggles in the plot which does not provide a stable solution.

## 8. When n=160, Final Profile




Since the number of nodes is increased by a huge margin, the final plot does not have a proper profile as the final solution is being blown up. This shows that the solution gets converged at some specific number of nodes lower than this value.(i.e.,160)