ME 254 Computational Fluid Dynamics
Homework 4

Due on 03/07/19 at 11:59 pm (through Catcourses)
Maximum points: 160

Jose A. Millan Higuera
March 10, 2019

1. Consider the first order wave equation

$$u_t + u_x = 0 \tag{1}$$

The initial conditions are given by

$$u(x,0) = \sin(2\pi x) \qquad 0 \leq x \leq 1 \tag{2}$$
$$u(x,0) = 0 \qquad 1 < x \leq 5 \tag{3}$$

Assume periodic boundary conditions. To implement this in a finite difference formulation using M grid points, you could force

$$u_1^{n+1} = u_M^{n+1} \tag{4}$$

Remember that the boundary condition at grid point M for the wave equation is given by

$$u_M^{n+1} = u_{M-1}^n \tag{5}$$

(a) (**60 points**) Write a code to solve the above problem numerically using the first-order Upwind, second-order Upwind, Lax-Wendroff and Rusanov schemes. Make sure that your code can take in the value of ν and Δx to determine a suitable timestep.

(b) (**30 points**) Plot the exact solution and the numerical solutions (from all schemes on the same plot) obtained using $\Delta x = 0.1$ for $\nu = 1$ and $\nu = 0.75$ at $t = 1$, $t = 2$, $t = 3$.

(c) (**15 points**) Define the L_∞, L^1, L^2 norms of the numerical solution as follows

$$L_\infty = \max(|u_i - u_{\text{exact,i}}|) \tag{6}$$
$$L^1 = \frac{1}{M}\sum_{i=1}^{M} |u_i - u_{\text{exact,i}}| \tag{7}$$
$$L^2 = \sqrt{\frac{\sum_{i=1}^{M} |u_i - u_{\text{exact,i}}|^2}{M}} \tag{8}$$

where i is the grid point index. Compute the L_∞, L^1, L^2 norms of the numerical solution for $\Delta x = 0.1$, $\Delta x = 0.05$, $\Delta x = 10^{-2}$ at $t = 3$ and tabulate (for all schemes considered above). Use $\nu = 0.75$ for this part.

(d) (**5 points**) Determine the order of each of the schemes based on your table above (and not based on prior information provided in class).

2. Now modify the initial condition to a square wave

$$u(x,0) = 1 \qquad 0 \leq x \leq 1 \tag{9}$$
$$u(x,0) = 0 \qquad 1 < x \leq 5 \tag{10}$$

(a) (**30 points**) Plot the exact solution and the numerical solutions (from all schemes on the same plot) obtained using $\Delta x = 0.1$ for $\nu = 1$ and $\nu = 0.75$ at $t = 1$, $t = 2$, $t = 3$.

(b) (**15 points**) Define the L_∞, L^1, L^2 norms of the numerical solution as follows

$$
\begin{aligned}
L_\infty &= \max(|u_i - u_{\mathrm{exact,i}}|) && (11)\\
L^1 &= \frac{1}{M}\sum_{i=1}^{M} |u_i - u_{\mathrm{exact,i}}| && (12)\\
L^2 &= \sqrt{\frac{\sum_{i=1}^{M} |u_i - u_{\mathrm{exact,i}}|^2}{M}} && (13)
\end{aligned}
$$

where i is the grid point index. Compute the L_∞, L^1, L^2 norms of the numerical solution for $\Delta x = 0.1$, $\Delta x = 0.05$, $\Delta x = 10^{-2}$ at $t = 3$ and tabulate (for all schemes considered above). Use $\nu = 0.75$ for this part.

(c) (**5 points**) Determine the order of each of the schemes based on your table above (and not based on their names or prior information provided in class).

```
% The following code is for the wave equation
% using 1st order upwind scheme

function wave_methods(dx,Y,time,n)

dt = Y*dx;

% rows in the matrix
t = 0:dt:time;
% columns in the matrix
L = 0:dx:5;

%initial condition in wavelenght
IC1 = sin(2*pi*(0:dx:1));
%secondary initial condition in wavelength
IC2 = 0*((1+dx):dx:5);

u1 = zeros(length(t),length(L));
u1(size(u1,1),:) = [IC1,IC2];

% This is the 1st order upwind scheme
for w = length(t)-1:-1:1

    for i=2:1:length(L)
       u1(w,i) = (Y)*(-u1(w+1,i)+u1(w+1,i-1))+u1(w+1,i);
    end

    u1(w,length(L)) = u1(w+1,length(L)-1);
    u1(w,1) = u1(w+1,length(L)-1);

end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using lax-wendroff method

u2 = zeros(length(t),length(L));
u2(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1
        for i=2:1:length(L)-1
            u2(w,i) = u2(w+1,i)-(1/2)*(Y)*(u2(w+1,i+1)-u2(w
+1,i-1))+...
                (1/2)*(Y)^2*(u2(w+1,i+1)-2*u2(w+1,i)+u2(w+1,i-1));
        end

    u2(w,length(L)) = u2(w+1,length(L)-1);
    u2(w,1) = u2(w+1,length(L)-1);
end
```

```
%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using second order upwind
% a predictor is used and then the other values are iterated
% using a 1 step algorithm

u3 = zeros(length(t),length(L));
u3(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    u3(w,2) = u3(w+1,2)-(Y)*(u3(w+1,2)-u3(w+1,1));
        for i=3:1:length(L)
            u3(w,i) = u3(w+1,i)-(dt/dx)*(u3(w+1,i)-u3(w+1,i-1))+...
                (1/2)*(dt/dx)*((dt/dx)-1)*(u3(w+1,i)-...
                2*u3(w+1,i-1)+u3(w+1,i-2));
        end

    u3(w,length(L)) = u3(w+1,length(L)-1);
    u3(w,1) = u3(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using Rusanov method

u4 = zeros(length(t),length(L));
u4(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    %for the 1st step before n+1
    for i=1:1:length(L)-1
        u4_1(1,i) = (1/2)*(u4(w+1,i+1)+u4(w+1,i))-...
            (1/3)*Y*(u4(w+1,i+1)-u4(w+1,i));
    end

    %for the 2nd step before n+1
    for i=2:1:length(L)-1
        u4_2(1,i) = u4(w+1,i)-(2/3)*Y*(u4_1(1,i)-u4_1(1,i-1));
    end

    for i=3:1:length(L)-2
        u4(w,i)=u4(w+1,i)-(1/24)*Y*(-2*u4(w+1,i+2)+7*u4(w+1,i+1)-...
            7*u4(w+1,i-1)+2*u4(w+1,i-2))-...
            (6/8)*Y*(u4_2(1,i-1)-u4_2(1,i-2))-...
            ((4*Y^2-Y^4)/24)*(u4(w+1,i+2)-4*u4(w+1,i+1)+6*u4(w+1,i)...
            -4*u4(w+1,i-1)+u4(w+1,i-2));
    end

    u4(w,length(L)-1)=u4(w+1,length(L)-1)-(1/24)*Y*(-2*u4(w
+1,2)+7*u4(w+1,length(L))-...
```

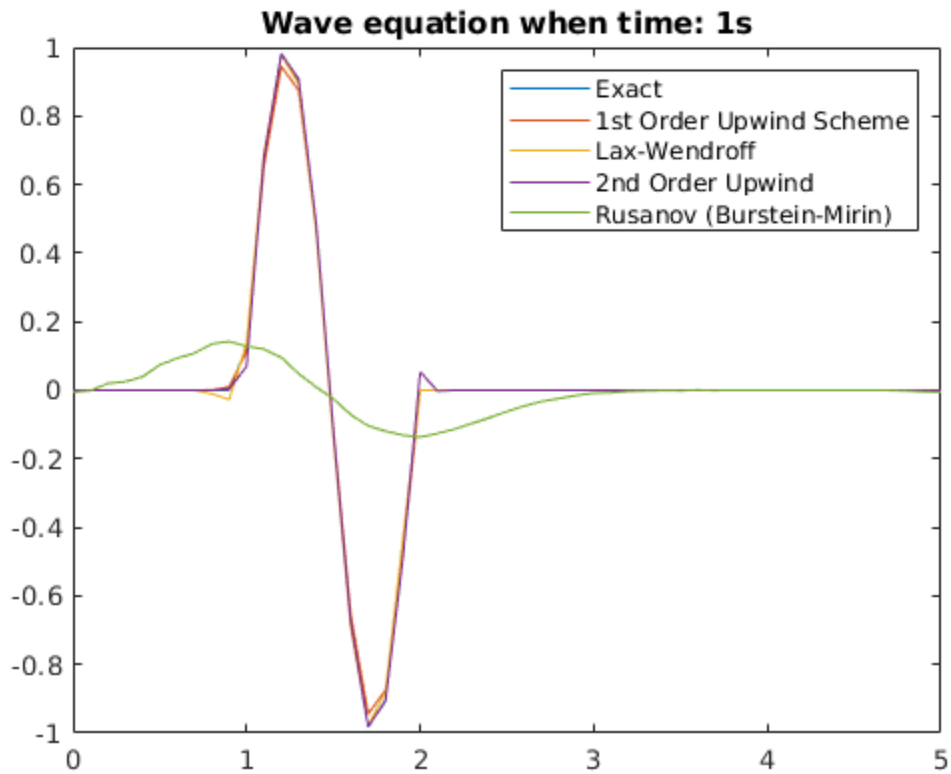
```
    7*u4(w+1,length(L)-2)+2*u4(w+1,length(L)-3))-...
    (3/8)*Y*(u4_2(1,length(L)-2)-u4_2(1,length(L)-3))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2)-4*u4(w+1,length(L))+6*u4(w
+1,length(L)-1)...
    -4*u4(w+1,length(L)-2)+u4(w+1,length(L)-3));

    u4(w,2)=u4(w+1,2)-(1/24)*Y*(-2*u4(w+1,2+2)+7*u4(w+1,2+1)-...
    7*u4(w+1,2-1)+2*u4(w+1,length(L)-1))-...
    (3/8)*Y*(u4_2(1,2-1)-u4_2(1,length(u4_2)))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2+2)-4*u4(w+1,2+1)+6*u4(w+1,2)...
    -4*u4(w+1,2-1)+u4(w+1,length(L)-1));

    u4(w,length(L)) = u4(w+1,length(L)-1);
    u4(w,1) = u4(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using the exact solution
% so u(x) = sin(2*pi*x) from 0,x,1
% u(x) = 0 for 1,x,5
% hence, for the exact solution, u(x,t) = F(x-ct)

u_exact = zeros(length(t),length(L));

for w = length(t):-1:1
    for i=1:1:length(L)
        if ((L(i)-t(w))>=0) && (((L(i)-t(w)))<=1)
            u_exact(w,i) = sin(2*pi*(L(i)-t(w)));
        else
            u_exact(w,i) = 0;
        end
    end
end

figure(n)
plot(L,u_exact(length(t),:),L,u1(1,:),L,u2(1,:),L,u3(1,:),L,u4(1,:));
title(['Wave equation when time: ', num2str(time),'s'])
legend('Exact','1st Order Upwind Scheme','Lax-Wendroff','2nd Order
 Upwind',...
    'Rusanov (Burstein-Mirin)')


end
```

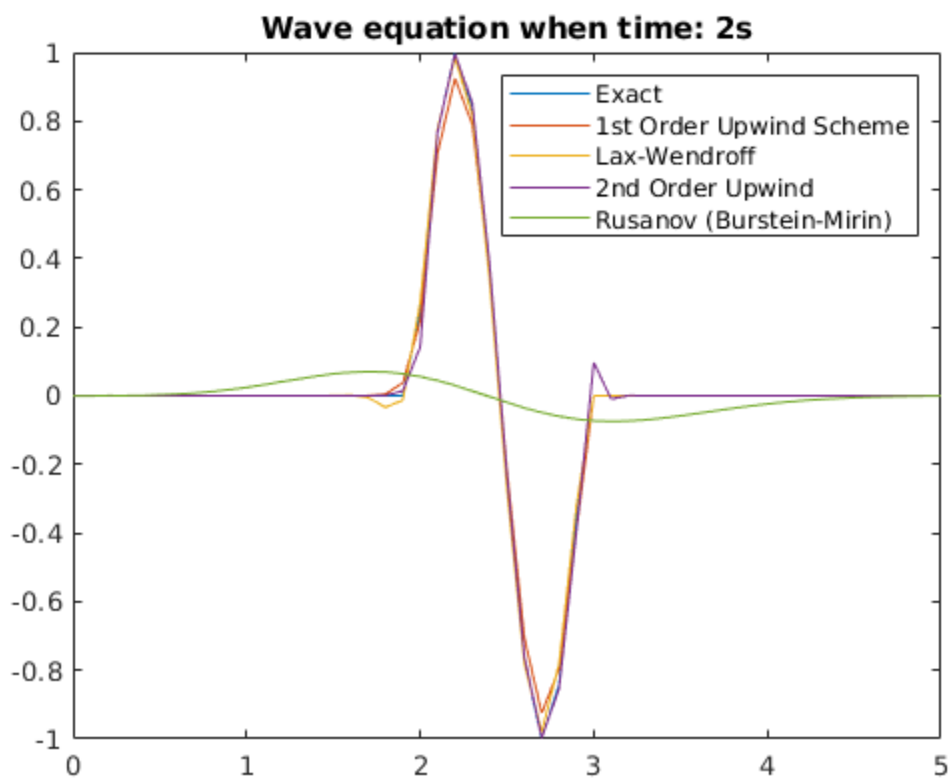
Not enough input arguments.

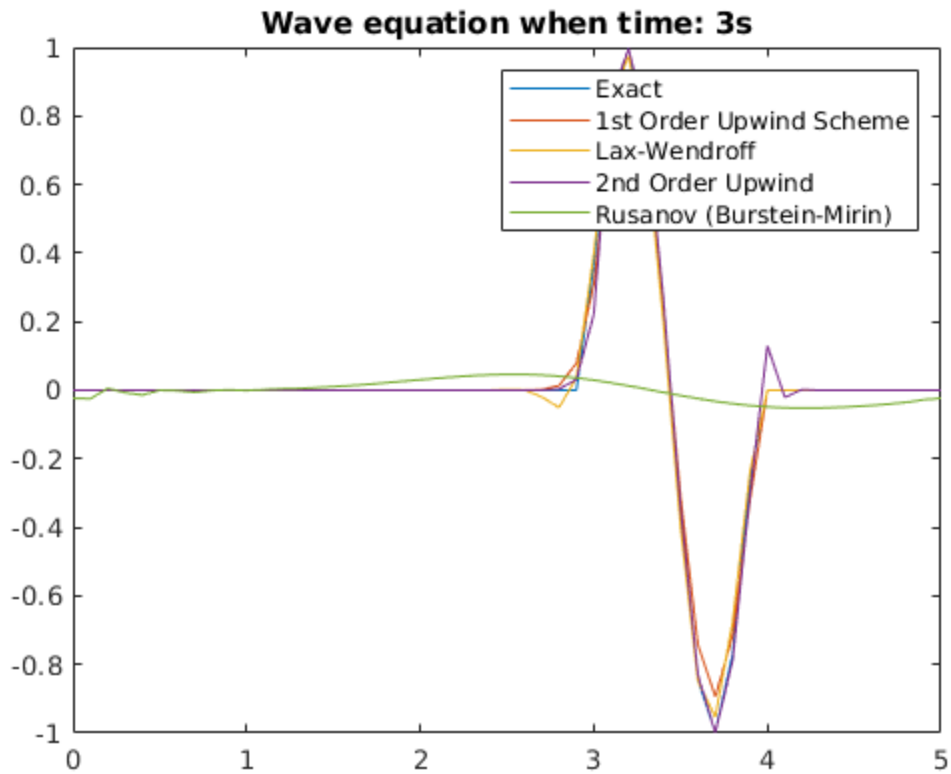
Error in wave_methods (line 6)
*dt = Y*dx;*

Published with MATLAB® R2018a

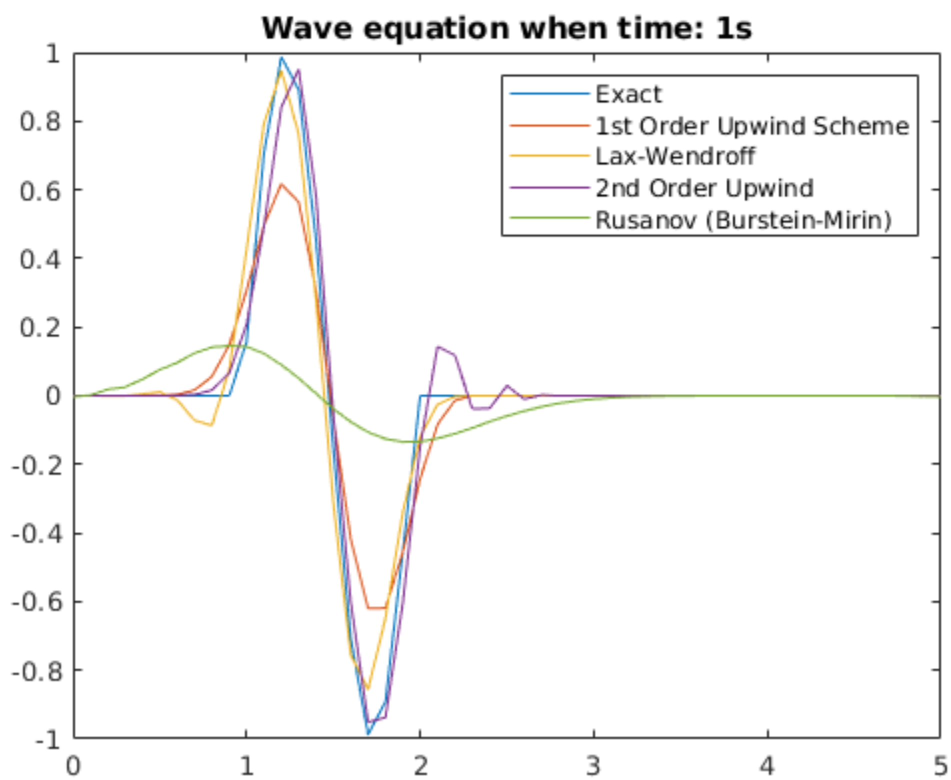
```
clc
clear all

% for Y = 1
% 0.98 was needed in order to be able to distinguish, still 0.98->1
wave_methods(0.1,0.98,1,1);
wave_methods(0.1,0.98,2,2);
wave_methods(0.1,0.98,3,3);

% for Y = 0.75
wave_methods(0.1,0.75,1,4);
wave_methods(0.1,0.75,2,5);
wave_methods(0.1,0.75,3,6);

% tabulate the tables for L's of the numerical solution
% using v = 0.75, dx = 0.1, dx = 0.05, dx = 0.01 at t = 3. (wave)
% wave_methods_tabulation(0.1,0.75,3)
% wave_methods_tabulation(0.05,0.75,3)
% wave_methods_tabulation(0.01,0.75,3)

% % for Y = 1
% 0.98 was needed in order to be able to distinguish, still 0.98->1
% wave_methods_linear(0.1,0.98,1,7);
% wave_methods_linear(0.1,0.98,2,8);
% wave_methods_linear(0.1,0.98,3,9);
%
% % for Y = 0.75
% wave_methods_linear(0.1,0.75,1,10);
% wave_methods_linear(0.1,0.75,2,11);
% wave_methods_linear(0.1,0.75,3,12);

% tabulate the tables for L's of the numerical solution
% using v = 0.75, dx = 0.1, dx = 0.05, dx = 0.01 at t = 3.
% wave_methods_linear_tabulation(0.1,0.75,3)
% wave_methods_linear_tabulation(0.05,0.75,3)
% wave_methods_linear_tabulation(0.01,0.75,3)
```


Wave equation when time: 1s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)

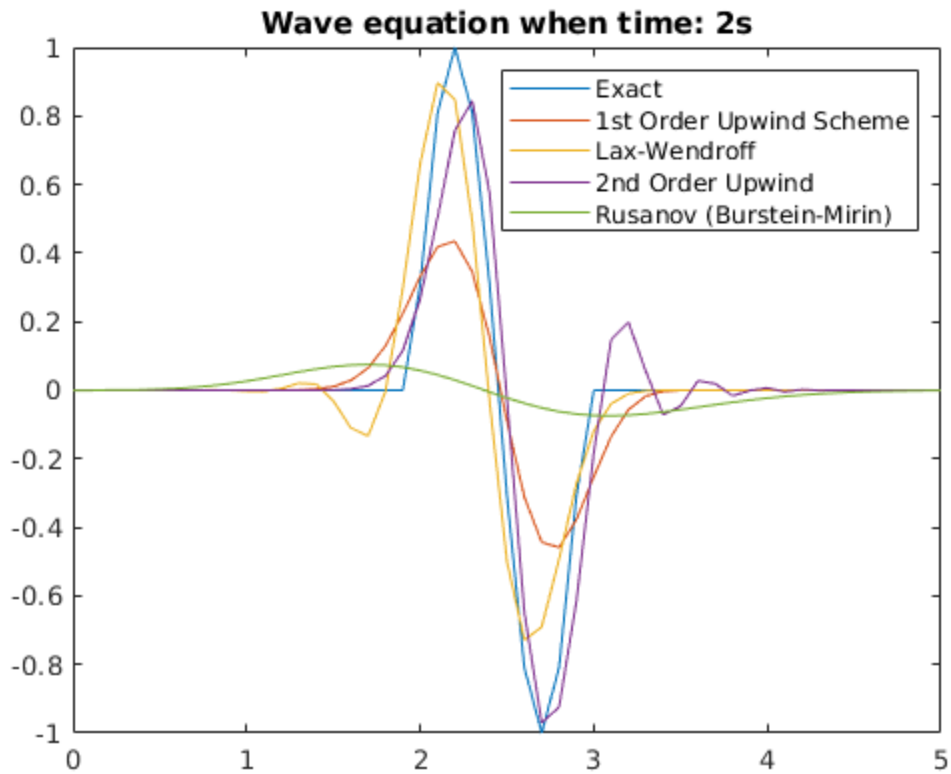

Wave equation when time: 2s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)

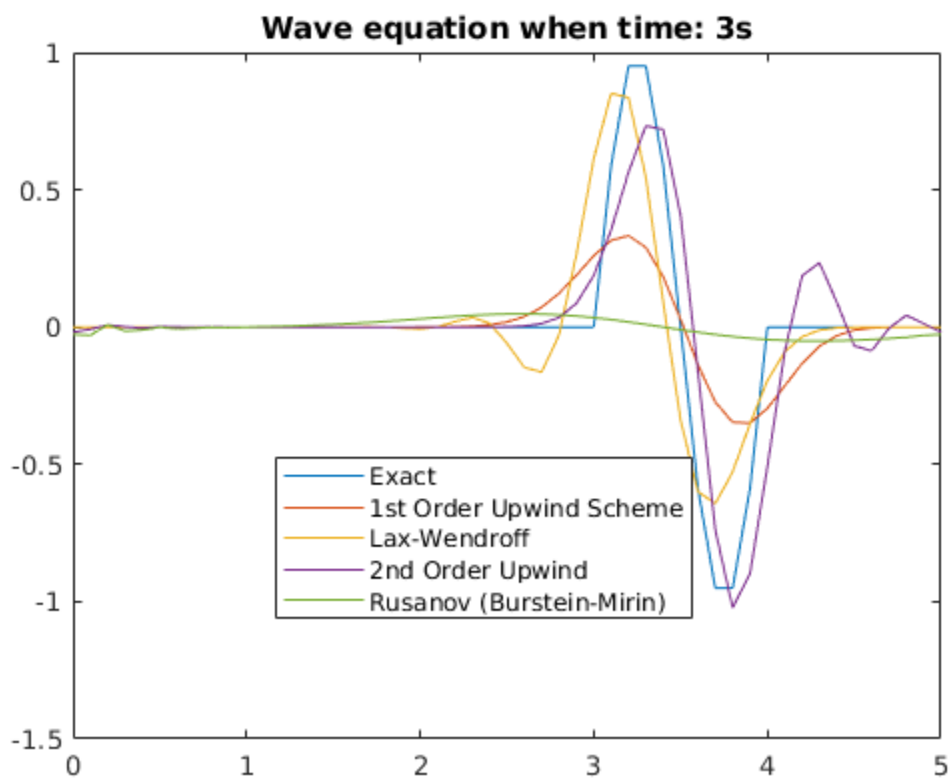


Wave equation when time: 3s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5



Wave equation when time: 1s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5


Wave equation when time: 2s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5


Wave equation when time: 3s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5

Published with MATLAB® R2018a

```
% The following code is for the wave equation
% using 1st order upwind scheme

function L_A = wave_methods_tabulation(dx,Y,time)

dt = Y*dx;

% rows in the matrix
t = 0:dt:time;
% columns in the matrix
L = 0:dx:5;

%initial condition in wavelenght
IC1 = sin(2*pi*(0:dx:1));
%secondary initial condition in wavelength
IC2 = 0*((1+dx):dx:5);

u1 = zeros(length(t),length(L));
u1(size(u1,1),:) = [IC1,IC2];

% This is the 1st order upwind scheme
for w = length(t)-1:-1:1

    for i=2:1:length(L)
       u1(w,i) = (Y)*(-u1(w+1,i)+u1(w+1,i-1))+u1(w+1,i);
    end

    u1(w,length(L)) = u1(w+1,length(L)-1);
    u1(w,1) = u1(w+1,length(L)-1);

end

%
 -------------------------------------------------------------------------
% The following code is for the wave equation
% using lax-wendroff method

u2 = zeros(length(t),length(L));
u2(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1
        for i=2:1:length(L)-1
            u2(w,i) = u2(w+1,i)-(1/2)*(Y)*(u2(w+1,i+1)-u2(w
+1,i-1))+...
                (1/2)*(Y)^2*(u2(w+1,i+1)-2*u2(w+1,i)+u2(w+1,i-1));
        end

    u2(w,length(L)) = u2(w+1,length(L)-1);
    u2(w,1) = u2(w+1,length(L)-1);
end
```

```
%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using second order upwind
% a predictor is used and then the other values are iterated
% using a 1 step algorithm

u3 = zeros(length(t),length(L));
u3(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    u3(w,2) = u3(w+1,2)-(Y)*(u3(w+1,2)-u3(w+1,1));
        for i=3:1:length(L)
            u3(w,i) = u3(w+1,i)-(dt/dx)*(u3(w+1,i)-u3(w+1,i-1))+...
                (1/2)*(dt/dx)*((dt/dx)-1)*(u3(w+1,i)-...
                2*u3(w+1,i-1)+u3(w+1,i-2));
        end

    u3(w,length(L)) = u3(w+1,length(L)-1);
    u3(w,1) = u3(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using Rusanov method

u4 = zeros(length(t),length(L));
u4(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    %for the 1st step before n+1
    for i=1:1:length(L)-1
        u4_1(1,i) = (1/2)*(u4(w+1,i+1)+u4(w+1,i))-...
            (1/3)*Y*(u4(w+1,i+1)-u4(w+1,i));
    end

    %for the 2nd step before n+1
    for i=2:1:length(L)-1
        u4_2(1,i) = u4(w+1,i)-(2/3)*Y*(u4_1(1,i)-u4_1(1,i-1));
    end

    for i=3:1:length(L)-2
        u4(w,i)=u4(w+1,i)-(1/24)*Y*(-2*u4(w+1,i+2)+7*u4(w+1,i+1)-...
            7*u4(w+1,i-1)+2*u4(w+1,i-2))-...
            (6/8)*Y*(u4_2(1,i-1)-u4_2(1,i-2))-...
            ((4*Y^2-Y^4)/24)*(u4(w+1,i+2)-4*u4(w+1,i+1)+6*u4(w+1,i)...
            -4*u4(w+1,i-1)+u4(w+1,i-2));
    end

    u4(w,length(L)-1)=u4(w+1,length(L)-1)-(1/24)*Y*(-2*u4(w
+1,2)+7*u4(w+1,length(L))-...
```

```
    7*u4(w+1,length(L)-2)+2*u4(w+1,length(L)-3))-...
    (3/8)*Y*(u4_2(1,length(L)-2)-u4_2(1,length(L)-3))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2)-4*u4(w+1,length(L))+6*u4(w
+1,length(L)-1)...
    -4*u4(w+1,length(L)-2)+u4(w+1,length(L)-3));

    u4(w,2)=u4(w+1,2)-(1/24)*Y*(-2*u4(w+1,2+2)+7*u4(w+1,2+1)-...
    7*u4(w+1,2-1)+2*u4(w+1,length(L)-1))-...
    (3/8)*Y*(u4_2(1,2-1)-u4_2(1,length(u4_2)))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2+2)-4*u4(w+1,2+1)+6*u4(w+1,2)...
    -4*u4(w+1,2-1)+u4(w+1,length(L)-1));

    u4(w,length(L)) = u4(w+1,length(L)-1);
    u4(w,1) = u4(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using the exact solution
% so u(x) = sin(2*pi*x) from 0,x,1
% u(x) = 0 for 1,x,5
% hence, for the exact solution, u(x,t) = F(x-ct)

u_exact = zeros(length(t),length(L));

for w = length(t):-1:1
    for i=1:1:length(L)
        if ((L(i)-t(w))>=0) && (((L(i)-t(w)))<=1)
            u_exact(w,i) = sin(2*pi*(L(i)-t(w)));
        else
            u_exact(w,i) = 0;
        end
    end
end

%
 -----------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% for the 1st order upwind scheme

L1 = zeros(3,length(L));

for i=1:1:length(L)
    L1(1,i) = abs(u1(1,i)-u_exact(length(t),i));
end
L_A(1,1) = max(L1(1,:));

for i=1:1:length(L)
    L1(2,i) = abs(u1(1,i)-u_exact(length(t),i));
end
L_A(1,2) = sum(L1(2,:))/length(L);

for i=1:1:length(L)
```

```
    L1(3,i) = abs(u1(1,i)-u_exact(length(t),i))^2;
end
L_A(1,3) = (sum(L1(3,:))/length(L))^(1/2);

%
 -----------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using lax-wendroff method

for i=1:1:length(L)
    L1(1,i) = abs(u2(1,i)-u_exact(length(t),i));
end
L_A(2,1) = max(L1(1,:));

L1(2,1) = abs(u2(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u2(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(2,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u2(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u2(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(2,3) = (L1(3,length(L))/length(L))^(1/2);

%
 -----------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using second order upwind

for i=1:1:length(L)
    L1(1,i) = abs(u3(1,i)-u_exact(length(t),i));
end
L_A(3,1) = max(L1(1,:));

L1(3,1) = abs(u3(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u3(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(3,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u3(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u3(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(3,3) = (L1(3,length(L))/length(L))^(1/2);

%
 -----------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using Rusanov method

for i=1:1:length(L)
```

```
    L1(1,i) = abs(u4(1,i)-u_exact(length(t),i));
end
L_A(4,1) = max(L1(1,:));

L1(3,1) = abs(u4(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u4(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(4,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u4(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u4(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(4,3) = (L1(3,length(L))/length(L))^(1/2);

end
```

```
Not enough input arguments.

Error in wave_methods_tabulation (line 6)
dt = Y*dx;
```

Published with MATLAB® R2018a

```
ans =

    0.6780    0.1064    0.2180
    0.6140    0.0842    0.1709
    0.5084    0.0802    0.1525
    0.8801    0.1385    0.2818


ans =

    0.4828    0.0744    0.1563
    0.2909    0.0331    0.0689
    0.2372    0.0288    0.0587
    0.7781    0.1207    0.2455


ans =

    0.1974    0.0225    0.0502
    0.0696    0.0030    0.0098
    0.0607    0.0027    0.0088
    0.3443    0.0534    0.1136

>>
```

Using these values in order to get the slope of the L’s

Upwind – Linf

M	Linf	L1	L2
51	34.577966429	5.4279125162	11.119936455
101	48.759412042	7.5125015888	15.788160254
401	79.138082958	9.0353392455	20.136115703

Lax

M	Linf	L1	L2
51	31.313618464	4.2939984685	8.7135391091
101	29.382984056	3.3388532795	6.9607505166
401	27.893939147	1.2032953899	3.9097815544

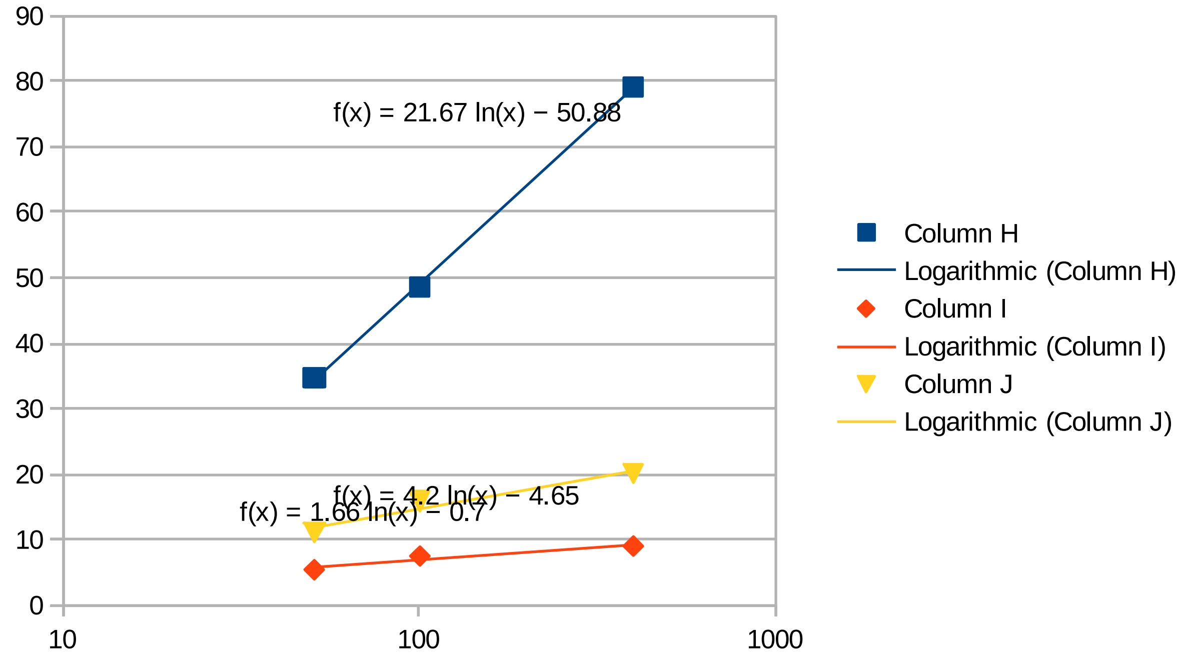
2nd order

M	Linf	L1	L2
51	25.928333112	4.0907952592	7.77669035
101	23.95786242	2.9061410655	5.9246052886
401	24.338135305	1.0997843945	3.5245526333

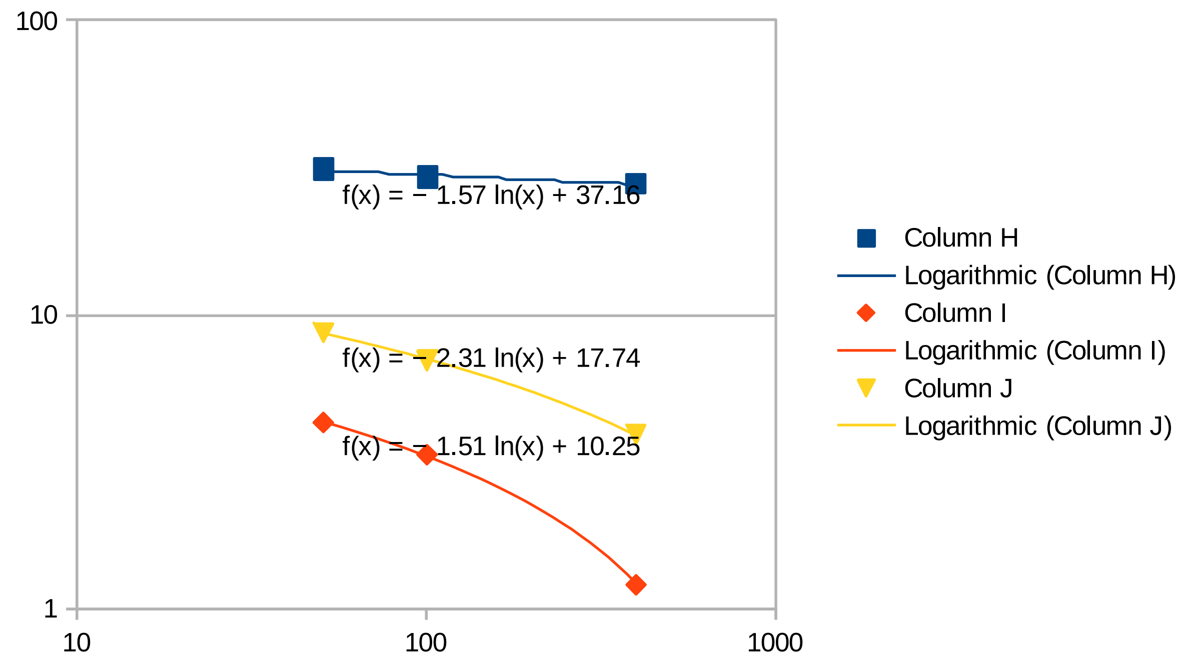
Rusanov

M	Linf	L1	L2
51	44.883125621	7.0628945728	14.372560867
101	78.583731008	12.193095747	24.798719762
401	138.07996363	21.412916523	45.570274082

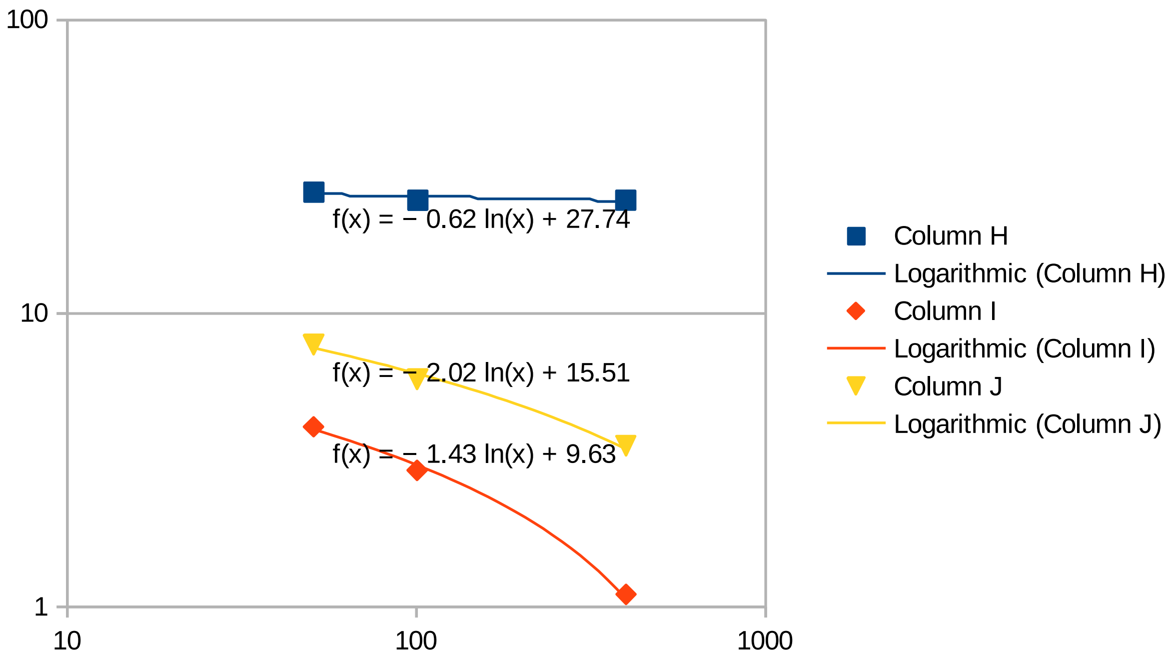
x values and the mesh points and y value is the L*M



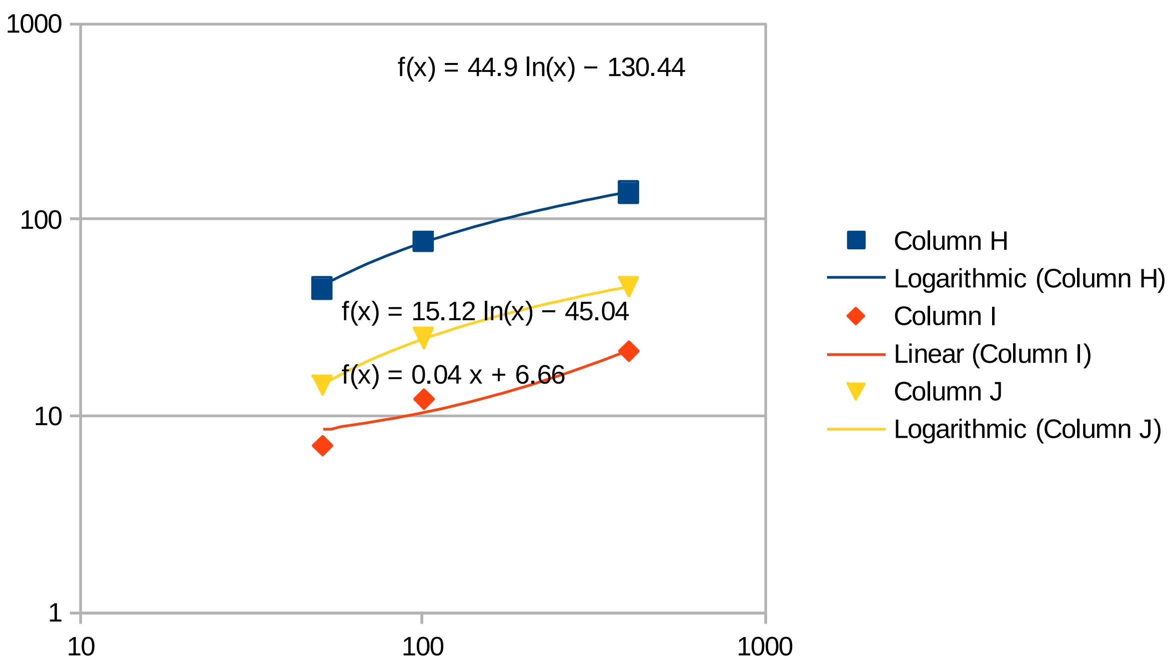
1st order upwind (slopes are variable)



Lax method (slopes are variable)



2nd order upwind (slopes are variable)



Rusanov

```
% The following code is for the wave equation
% using 1st order upwind scheme

function wave_methods_linear(dx,Y,time,n)

dt = Y*dx;

% rows in the matrix
t = 0:dt:time;
% columns in the matrix
L = 0:dx:5;

%initial condition in wavelenght
%IC1 = sin(2*pi*(0:dx:1));
IC1 = ones(1,length(0:dx:1));
%secondary initial condition in wavelength
IC2 = 0*((1+dx):dx:5);

u1 = zeros(length(t),length(L));
u1(size(u1,1),:) = [IC1,IC2];

% This is the 1st order upwind scheme
for w = length(t)-1:-1:1

    for i=2:1:length(L)
       u1(w,i) = (Y)*(-u1(w+1,i)+u1(w+1,i-1))+u1(w+1,i);
    end

    u1(w,length(L)) = u1(w+1,length(L)-1);
    u1(w,1) = u1(w+1,length(L)-1);

end

%
 ------------------------------------------------------------------------
% The following code is for the wave equation
% using lax-wendroff method

u2 = zeros(length(t),length(L));
u2(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1
        for i=2:1:length(L)-1
            u2(w,i) = u2(w+1,i)-(1/2)*(Y)*(u2(w+1,i+1)-u2(w
+1,i-1))+...
                (1/2)*(Y)^2*(u2(w+1,i+1)-2*u2(w+1,i)+u2(w+1,i-1));
        end

    u2(w,length(L)) = u2(w+1,length(L)-1);
    u2(w,1) = u2(w+1,length(L)-1);
end
```

```
%
 ------------------------------------------------------------------------
% The following code is for the wave equation
% using second order upwind
% a predictor is used and then the other values are iterated
% using a 1 step algorithm

u3 = zeros(length(t),length(L));
u3(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    u3(w,2) = u3(w+1,2)-(Y)*(u3(w+1,2)-u3(w+1,1));
        for i=3:1:length(L)
            u3(w,i) = u3(w+1,i)-(dt/dx)*(u3(w+1,i)-u3(w+1,i-1))+...
                (1/2)*(dt/dx)*((dt/dx)-1)*(u3(w+1,i)-...
                2*u3(w+1,i-1)+u3(w+1,i-2));
        end

    u3(w,length(L)) = u3(w+1,length(L)-1);
    u3(w,1) = u3(w+1,length(L)-1);
end

%
 ------------------------------------------------------------------------
% The following code is for the wave equation
% using Rusanov method

u4 = zeros(length(t),length(L));
u4(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    %for the 1st step before n+1
    for i=1:1:length(L)-1
        u4_1(1,i) = (1/2)*(u4(w+1,i+1)+u4(w+1,i))-...
            (1/3)*Y*(u4(w+1,i+1)-u4(w+1,i));
    end

    %for the 2nd step before n+1
    for i=2:1:length(L)-1
        u4_2(1,i) = u4(w+1,i)-(2/3)*Y*(u4_1(1,i)-u4_1(1,i-1));
    end

    for i=3:1:length(L)-2
        u4(w,i)=u4(w+1,i)-(1/24)*Y*(-2*u4(w+1,i+2)+7*u4(w+1,i+1)-...
            7*u4(w+1,i-1)+2*u4(w+1,i-2))-...
            (6/8)*Y*(u4_2(1,i-1)-u4_2(1,i-2))-...
            ((4*Y^2-Y^4)/24)*(u4(w+1,i+2)-4*u4(w+1,i+1)+6*u4(w+1,i)...
            -4*u4(w+1,i-1)+u4(w+1,i-2));
    end

    u4(w,length(L)-1)=u4(w+1,length(L)-1)-(1/24)*Y*(-2*u4(w
+1,2)+7*u4(w+1,length(L))-...
```

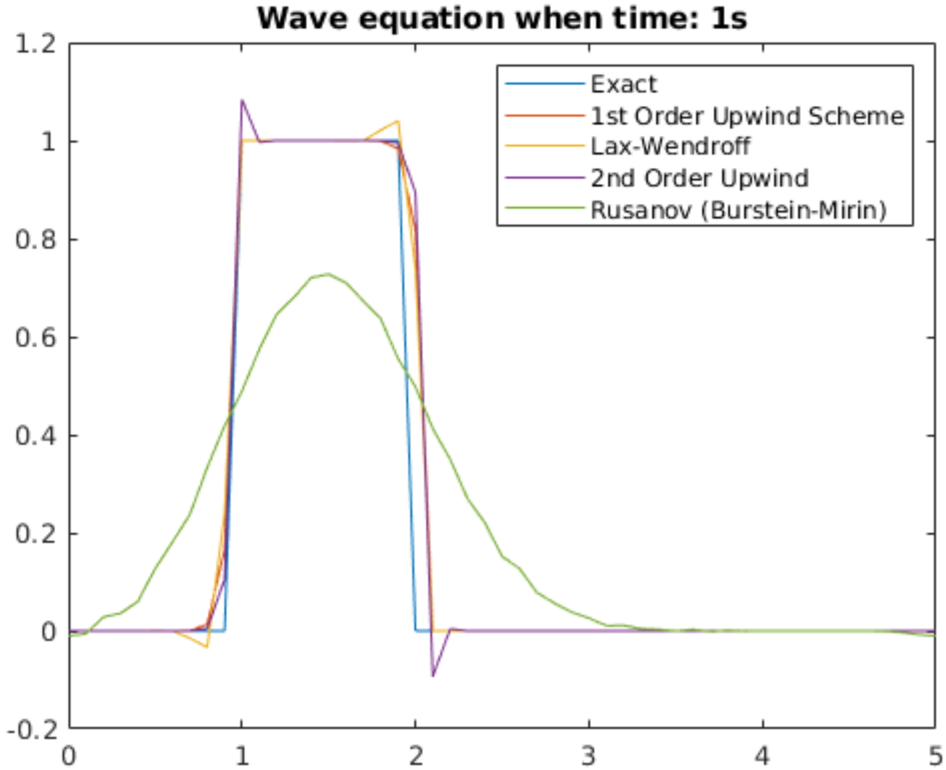
```
    7*u4(w+1,length(L)-2)+2*u4(w+1,length(L)-3))-...
    (3/8)*Y*(u4_2(1,length(L)-2)-u4_2(1,length(L)-3))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2)-4*u4(w+1,length(L))+6*u4(w
+1,length(L)-1)...
    -4*u4(w+1,length(L)-2)+u4(w+1,length(L)-3));

    u4(w,2)=u4(w+1,2)-(1/24)*Y*(-2*u4(w+1,2+2)+7*u4(w+1,2+1)-...
    7*u4(w+1,2-1)+2*u4(w+1,length(L)-1))-...
    (3/8)*Y*(u4_2(1,2-1)-u4_2(1,length(u4_2)))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2+2)-4*u4(w+1,2+1)+6*u4(w+1,2)...
    -4*u4(w+1,2-1)+u4(w+1,length(L)-1));

    u4(w,length(L)) = u4(w+1,length(L)-1);
    u4(w,1) = u4(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using the exact solution
% so u(x) = sin(2*pi*x) from 0,x,1
% u(x) = 0 for 1,x,5
% hence, for the exact solution, u(x,t) = F(x-ct)

u_exact = zeros(length(t),length(L));

for w = length(t):-1:1
    for i=1:1:length(L)
        if ((L(i)-t(w))>=0) && (((L(i)-t(w)))<=1)
            u_exact(w,i) = 1;
        else
            u_exact(w,i) = 0;
        end
    end
end

figure(n)
plot(L,u_exact(length(t),:),L,u1(1,:),L,u2(1,:),L,u3(1,:),L,u4(1,:));
title(['Wave equation when time: ', num2str(time),'s'])
legend('Exact','1st Order Upwind Scheme','Lax-Wendroff','2nd Order
 Upwind',...
    'Rusanov (Burstein-Mirin)')


end
```

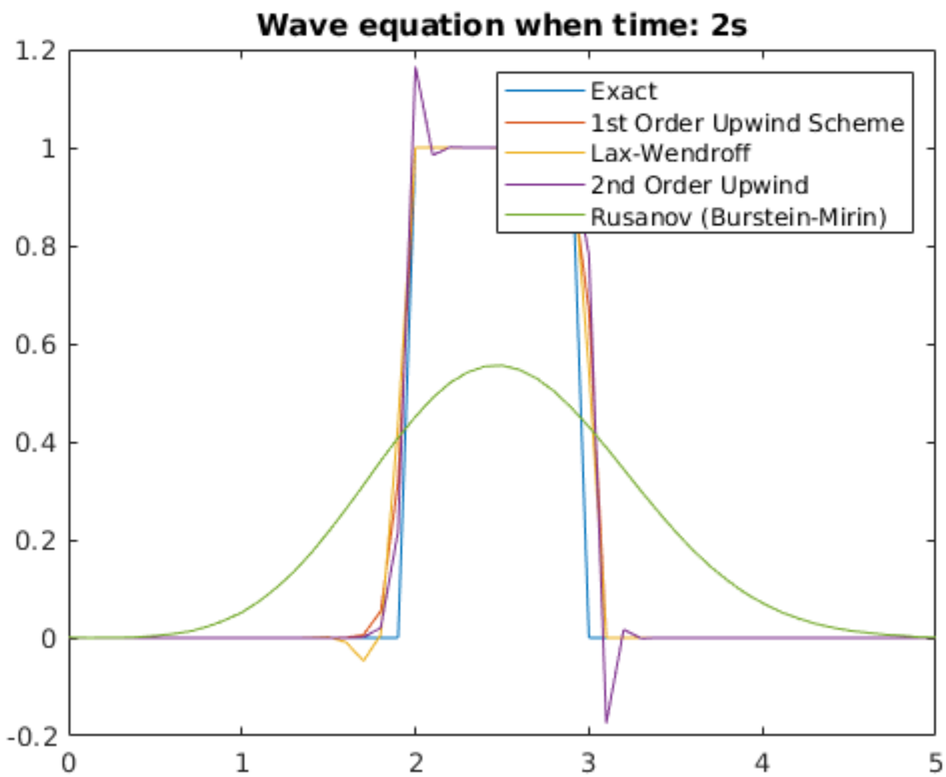
Not enough input arguments.

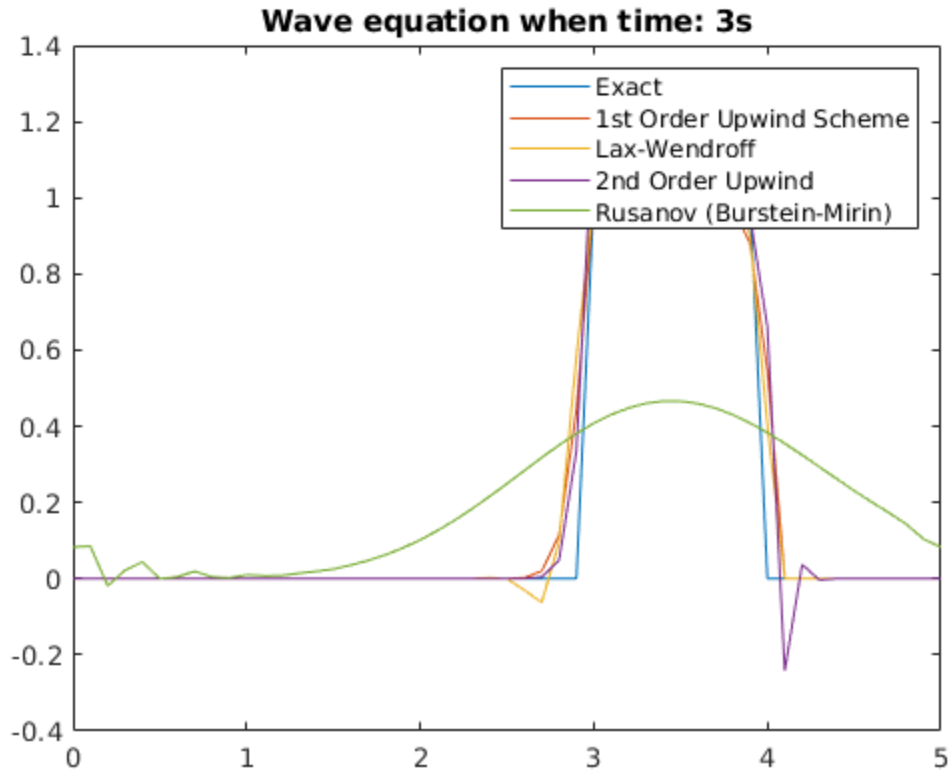
Error in wave_methods_linear (line 6)
*dt = Y*dx;*

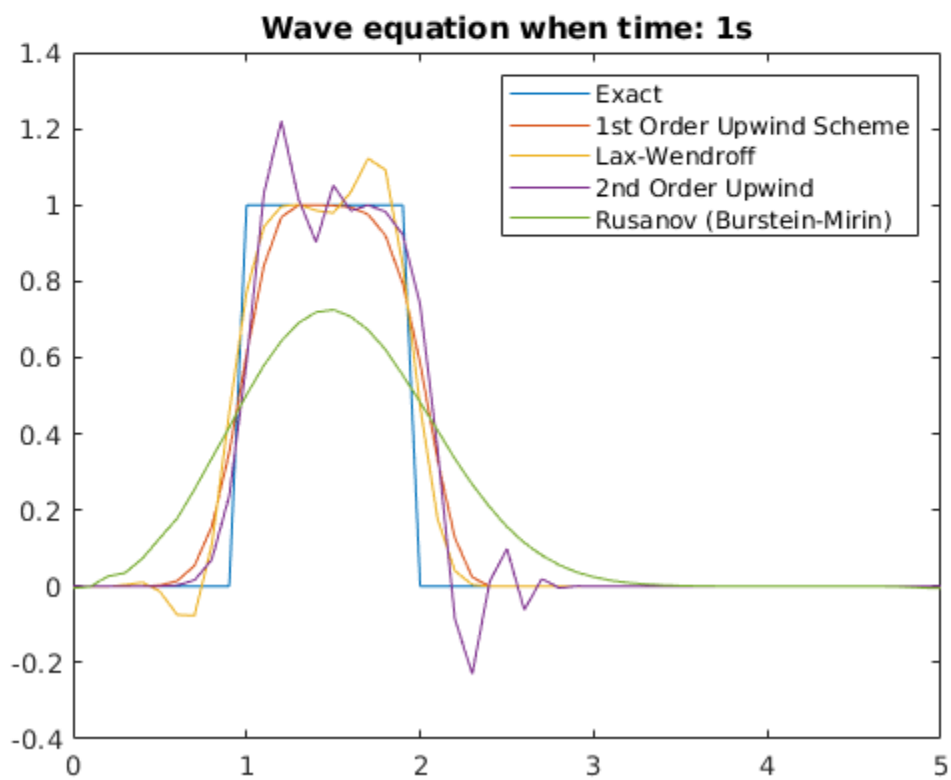
Published with MATLAB® R2018a

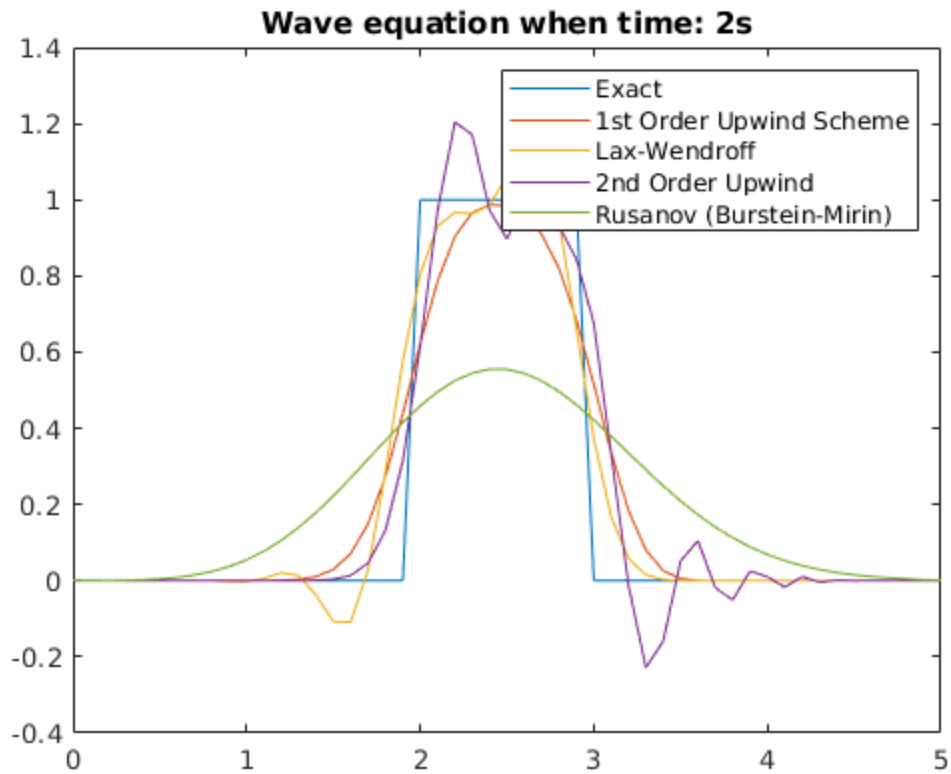
```
clc
clear all

% for Y = 1
% 0.98 was needed in order to be able to distinguish, still 0.98->1
% wave_methods(0.1,0.98,1,1);
% wave_methods(0.1,0.98,2,2);
% wave_methods(0.1,0.98,3,3);

% for Y = 0.75
% wave_methods(0.1,0.75,1,4);
% wave_methods(0.1,0.75,2,5);
% wave_methods(0.1,0.75,3,6);

% tabulate the tables for L's of the numerical solution
% using v = 0.75, dx = 0.1, dx = 0.05, dx = 0.01 at t = 3. (wave)
% wave_methods_tabulation(0.1,0.75,3)
% wave_methods_tabulation(0.05,0.75,3)
% wave_methods_tabulation(0.01,0.75,3)

% % for Y = 1
% 0.98 was needed in order to be able to distinguish, still 0.98->1
wave_methods_linear(0.1,0.98,1,7);
wave_methods_linear(0.1,0.98,2,8);
wave_methods_linear(0.1,0.98,3,9);
%
% % for Y = 0.75
wave_methods_linear(0.1,0.75,1,10);
wave_methods_linear(0.1,0.75,2,11);
wave_methods_linear(0.1,0.75,3,12);

% tabulate the tables for L's of the numerical solution
% using v = 0.75, dx = 0.1, dx = 0.05, dx = 0.01 at t = 3.
% wave_methods_linear_tabulation(0.1,0.75,3)
% wave_methods_linear_tabulation(0.05,0.75,3)
% wave_methods_linear_tabulation(0.01,0.75,3)
```


Wave equation when time: 1s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5

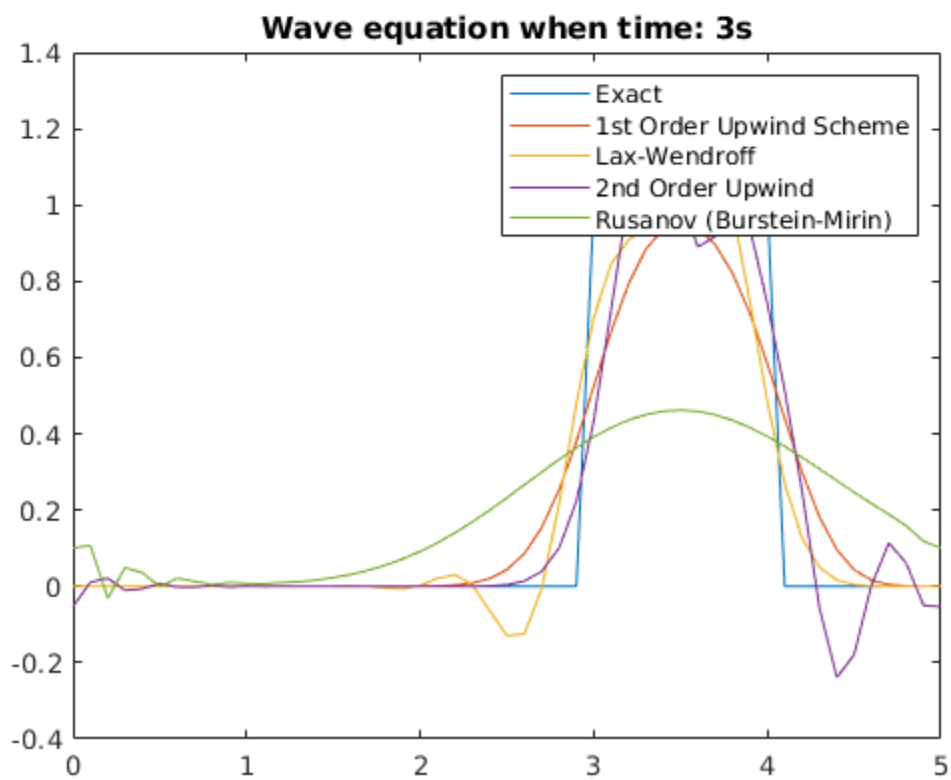

Wave equation when time: 2s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5


Wave equation when time: 3s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1
2
3
4
5


Wave equation when time: 1s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1
2
3
4
5



Wave equation when time: 2s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1
2
3
4
5



Wave equation when time: 3s
Exact
1st Order Upwind Scheme
Lax-Wendroff
2nd Order Upwind
Rusanov (Burstein-Mirin)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1
2
3
4
5

Published with MATLAB® R2018a

```
% The following code is for the wave equation
% using 1st order upwind scheme

function L_A = wave_methods_linear_tabulation(dx,Y,time)

dt = Y*dx;

% rows in the matrix
t = 0:dt:time;
% columns in the matrix
L = 0:dx:5;

%initial condition in wavelenght
%IC1 = sin(2*pi*(0:dx:1));
IC1 = ones(1,length(0:dx:1));
%secondary initial condition in wavelength
IC2 = 0*((1+dx):dx:5);

u1 = zeros(length(t),length(L));
u1(size(u1,1),:) = [IC1,IC2];

% This is the 1st order upwind scheme
for w = length(t)-1:-1:1

    for i=2:1:length(L)
       u1(w,i) = (Y)*(-u1(w+1,i)+u1(w+1,i-1))+u1(w+1,i);
    end

    u1(w,length(L)) = u1(w+1,length(L)-1);
    u1(w,1) = u1(w+1,length(L)-1);

end

%
 -------------------------------------------------------------------------
% The following code is for the wave equation
% using lax-wendroff method

u2 = zeros(length(t),length(L));
u2(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1
        for i=2:1:length(L)-1
            u2(w,i) = u2(w+1,i)-(1/2)*(Y)*(u2(w+1,i+1)-u2(w
+1,i-1))+...
                (1/2)*(Y)^2*(u2(w+1,i+1)-2*u2(w+1,i)+u2(w+1,i-1));
        end

    u2(w,length(L)) = u2(w+1,length(L)-1);
    u2(w,1) = u2(w+1,length(L)-1);
end
```

```
%
 --------------------------------------------------------------------------
% The following code is for the wave equation
% using second order upwind
% a predictor is used and then the other values are iterated
% using a 1 step algorithm

u3 = zeros(length(t),length(L));
u3(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    u3(w,2) = u3(w+1,2)-(Y)*(u3(w+1,2)-u3(w+1,1));
        for i=3:1:length(L)
            u3(w,i) = u3(w+1,i)-(dt/dx)*(u3(w+1,i)-u3(w+1,i-1))+...
                (1/2)*(dt/dx)*((dt/dx)-1)*(u3(w+1,i)-...
                2*u3(w+1,i-1)+u3(w+1,i-2));
        end

    u3(w,length(L)) = u3(w+1,length(L)-1);
    u3(w,1) = u3(w+1,length(L)-1);
end

%
 --------------------------------------------------------------------------
% The following code is for the wave equation
% using Rusanov method

u4 = zeros(length(t),length(L));
u4(size(u1,1),:) = [IC1,IC2];

for w = length(t)-1:-1:1

    %for the 1st step before n+1
    for i=1:1:length(L)-1
        u4_1(1,i) = (1/2)*(u4(w+1,i+1)+u4(w+1,i))-...
            (1/3)*Y*(u4(w+1,i+1)-u4(w+1,i));
    end

    %for the 2nd step before n+1
    for i=2:1:length(L)-1
        u4_2(1,i) = u4(w+1,i)-(2/3)*Y*(u4_1(1,i)-u4_1(1,i-1));
    end

    for i=3:1:length(L)-2
        u4(w,i)=u4(w+1,i)-(1/24)*Y*(-2*u4(w+1,i+2)+7*u4(w+1,i+1)-...
            7*u4(w+1,i-1)+2*u4(w+1,i-2))-...
            (6/8)*Y*(u4_2(1,i-1)-u4_2(1,i-2))-...
            ((4*Y^2-Y^4)/24)*(u4(w+1,i+2)-4*u4(w+1,i+1)+6*u4(w+1,i)...
            -4*u4(w+1,i-1)+u4(w+1,i-2));
    end

    u4(w,length(L)-1)=u4(w+1,length(L)-1)-(1/24)*Y*(-2*u4(w
+1,2)+7*u4(w+1,length(L))-...
```

```
    7*u4(w+1,length(L)-2)+2*u4(w+1,length(L)-3))-...
    (3/8)*Y*(u4_2(1,length(L)-2)-u4_2(1,length(L)-3))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2)-4*u4(w+1,length(L))+6*u4(w
+1,length(L)-1)...
    -4*u4(w+1,length(L)-2)+u4(w+1,length(L)-3));

    u4(w,2)=u4(w+1,2)-(1/24)*Y*(-2*u4(w+1,2+2)+7*u4(w+1,2+1)-...
    7*u4(w+1,2-1)+2*u4(w+1,length(L)-1))-...
    (3/8)*Y*(u4_2(1,2-1)-u4_2(1,length(u4_2)))-...
    ((4*Y^2-Y^4)/24)*(u4(w+1,2+2)-4*u4(w+1,2+1)+6*u4(w+1,2)...
    -4*u4(w+1,2-1)+u4(w+1,length(L)-1));

    u4(w,length(L)) = u4(w+1,length(L)-1);
    u4(w,1) = u4(w+1,length(L)-1);
end

%
 -----------------------------------------------------------------------
% The following code is for the wave equation
% using the exact solution
% so u(x) = sin(2*pi*x) from 0,x,1
% u(x) = 0 for 1,x,5
% hence, for the exact solution, u(x,t) = F(x-ct)

u_exact = zeros(length(t),length(L));

for w = length(t):-1:1
    for i=1:1:length(L)
        if ((L(i)-t(w))>=0) && (((L(i)-t(w)))<=1)
            u_exact(w,i) = sin(2*pi*(L(i)-t(w)));
        else
            u_exact(w,i) = 0;
        end
    end
end

%
 -----------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% for the 1st order upwind scheme

L1 = zeros(3,length(L));

for i=1:1:length(L)
    L1(1,i) = abs(u1(1,i)-u_exact(length(t),i));
end
L_A(1,1) = max(L1(1,:));

for i=1:1:length(L)
    L1(2,i) = abs(u1(1,i)-u_exact(length(t),i));
end
L_A(1,2) = sum(L1(2,:))/length(L);

for i=1:1:length(L)
```

```
    L1(3,i) = abs(u1(1,i)-u_exact(length(t),i))^2;
end
L_A(1,3) = (sum(L1(3,:))/length(L))^(1/2);

%
 ------------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using lax-wendroff method

for i=1:1:length(L)
    L1(1,i) = abs(u2(1,i)-u_exact(length(t),i));
end
L_A(2,1) = max(L1(1,:));

L1(2,1) = abs(u2(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u2(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(2,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u2(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u2(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(2,3) = (L1(3,length(L))/length(L))^(1/2);

%
 ------------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using second order upwind

for i=1:1:length(L)
    L1(1,i) = abs(u3(1,i)-u_exact(length(t),i));
end
L_A(3,1) = max(L1(1,:));

L1(3,1) = abs(u3(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u3(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(3,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u3(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u3(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(3,3) = (L1(3,length(L))/length(L))^(1/2);

%
 ------------------------------------------------------------------------
% Now lets tabulate the table for L_inf, L_1, L_2
% using Rusanov method

for i=1:1:length(L)
```

```
    L1(1,i) = abs(u4(1,i)-u_exact(length(t),i));
end
L_A(4,1) = max(L1(1,:));

L1(3,1) = abs(u4(1,1)-u_exact(length(t),1));
for i=2:1:length(L)
    L1(2,i) = abs(u4(1,i)-u_exact(length(t),i))+L1(2,i-1);
end
L_A(4,2) = L1(2,length(L))/length(L);

L1(3,1) = (abs(u4(1,1)-u_exact(length(t),1))^2)/length(L);
for i=2:1:length(L)
    L1(3,i) = abs(u4(1,i)-u_exact(length(t),i))^2+L1(3,i-1);
end
L_A(4,3) = (L1(3,length(L))/length(L))^(1/2);

end
```

```
Not enough input arguments.

Error in wave_methods_linear_tabulation (line 6)
dt = Y*dx;
```

Published with MATLAB® R2018a

```
ans =

    1.8465    0.2196    0.5016
    2.0838    0.2284    0.5506
    1.9465    0.2364    0.5338
    1.6676    0.2334    0.4626


ans =

    1.9195    0.2090    0.5122
    2.1756    0.2173    0.5506
    1.9970    0.2228    0.5387
    1.7867    0.2197    0.4824


ans =

    1.9980    0.2011    0.5306
    2.0235    0.2060    0.5468
    2.0000    0.2083    0.5456
    1.9580    0.2026    0.5169
```

Using these values in order to get the slope of the L's

Upwind – Linf

M	Linf	L1	L2
51	94.1715	11.1996	25.5816
101	193.8695	21.109	51.7322
401	801.198	80.6411	212.7706

Lax

M	Linf	L1	L2
51	106.2738	11.6484	28.0806
101	219.7356	21.9473	55.6106
401	811.4235	82.606	219.2668

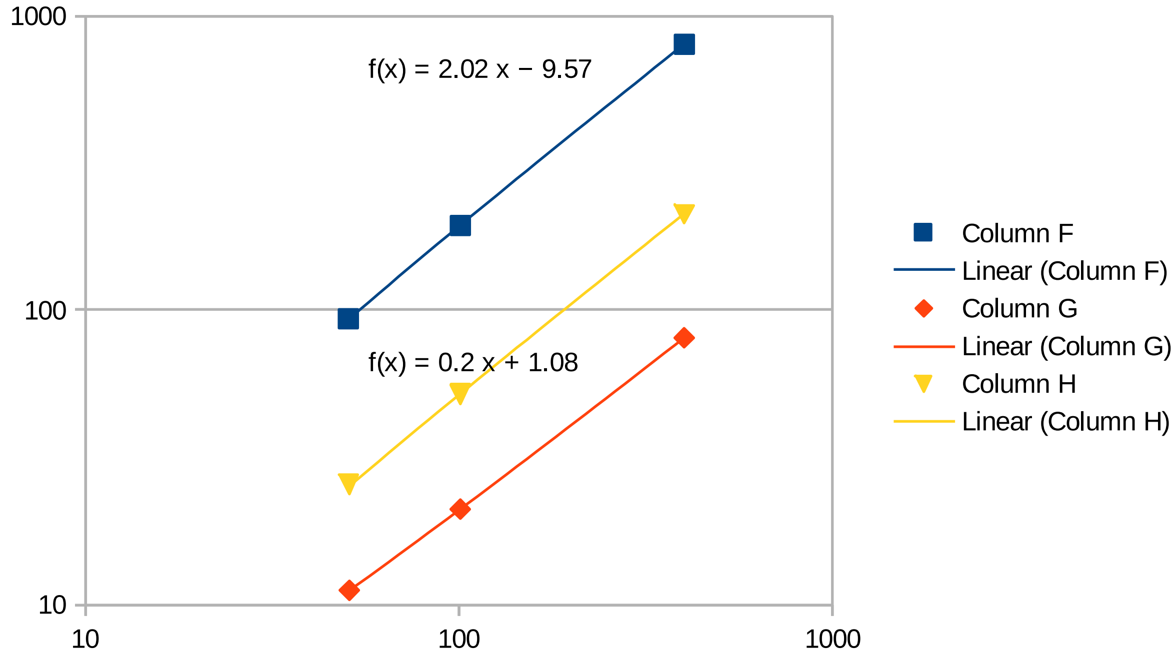
2^{nd} order

M	Linf	L1	L2
51	97.325	12.0564	27.2238
101	201.697	22.5028	54.4087
401	802	83.5283	218.7856

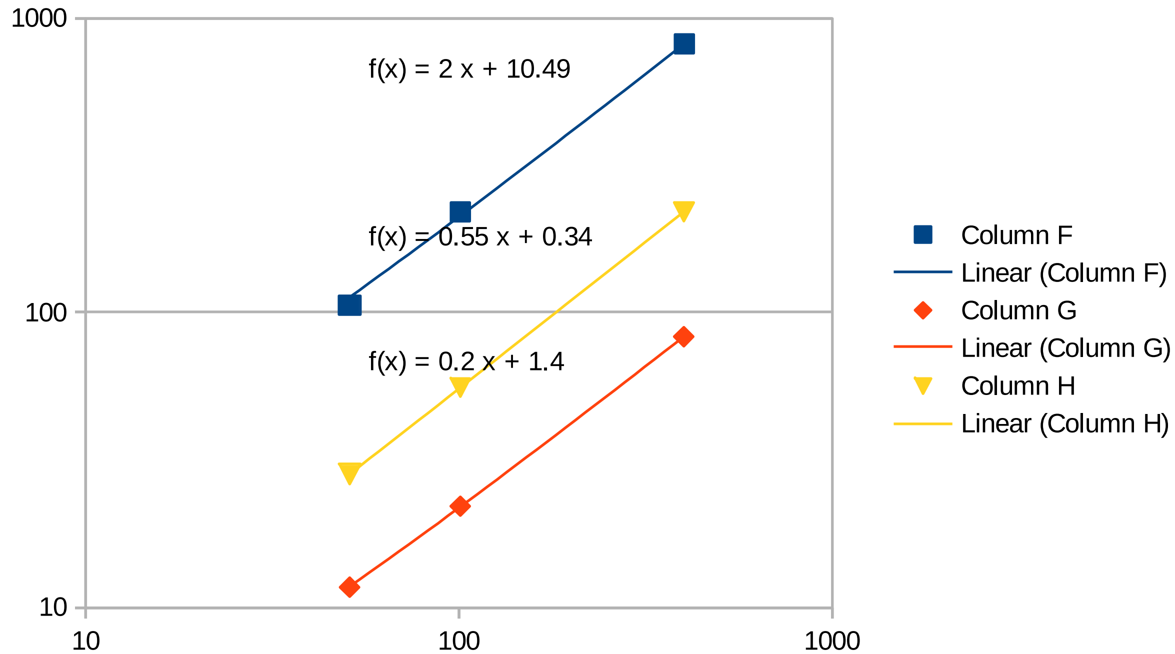
Rusanov

M	Linf	L1	L2
51	85.0476	11.9034	23.5926
101	180.4567	22.1897	48.7224
401	785.158	81.2426	207.2769

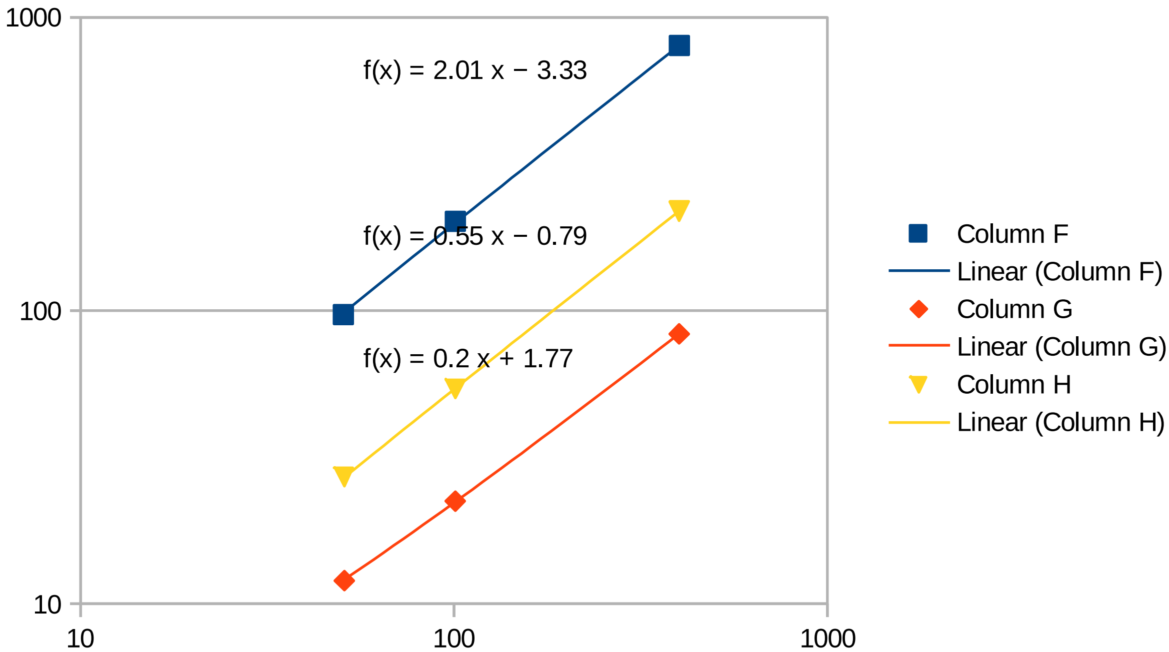
x values and the mesh points and y value is the L*M



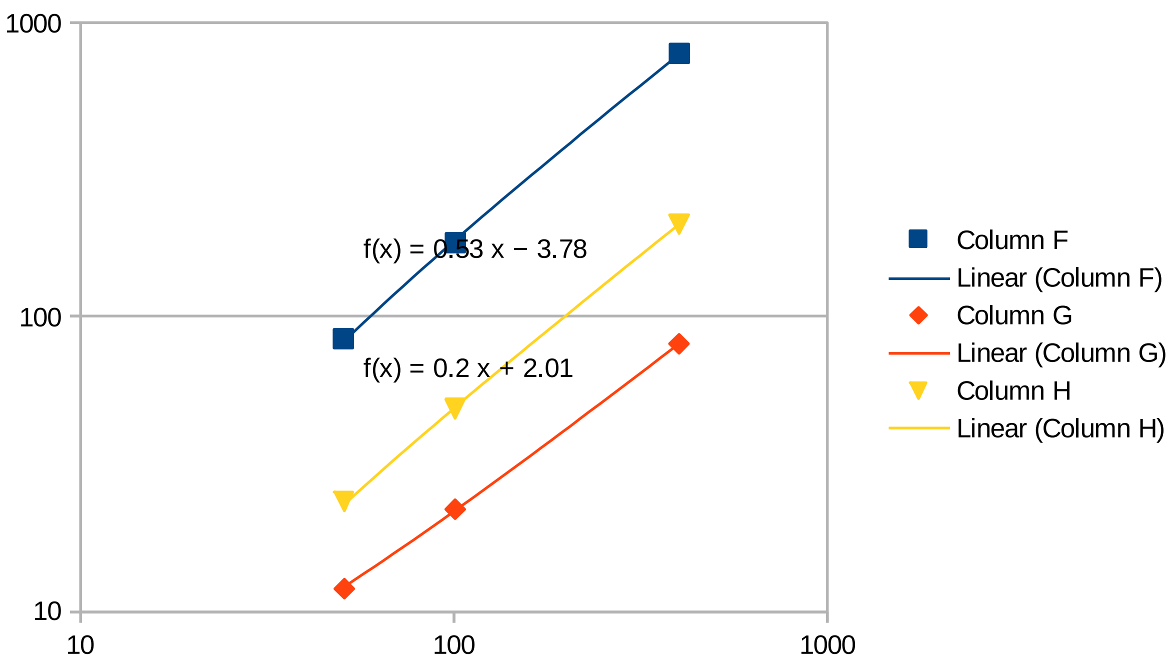
1st order upwind (slopes are variable)



Lax method (slopes are variable)



2nd order upwind (slopes are variable)



Rusanov