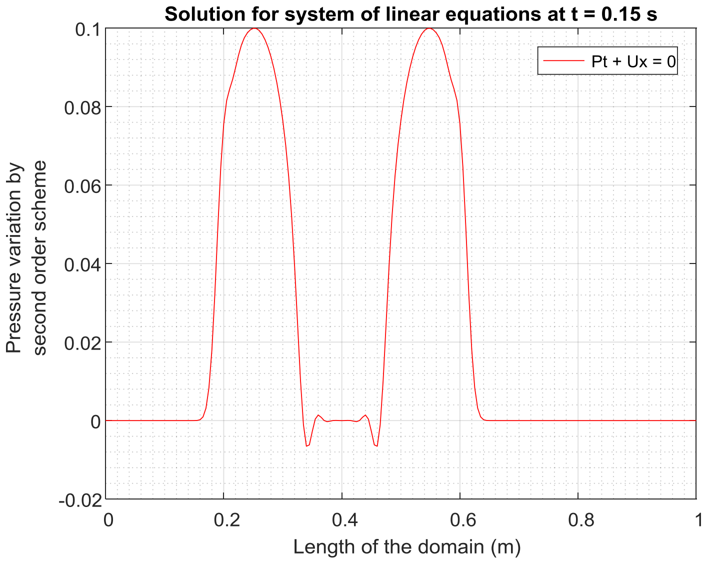
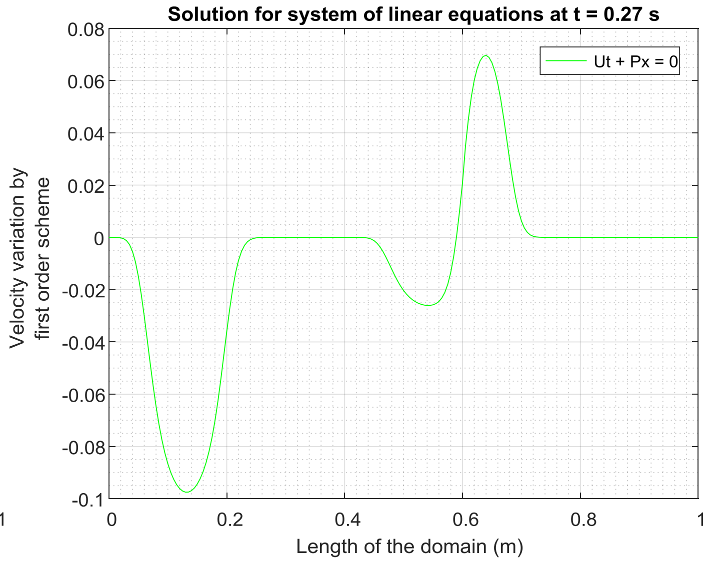
6/9/2018

Assignment 2

Numerical Methods For Conservation Laws



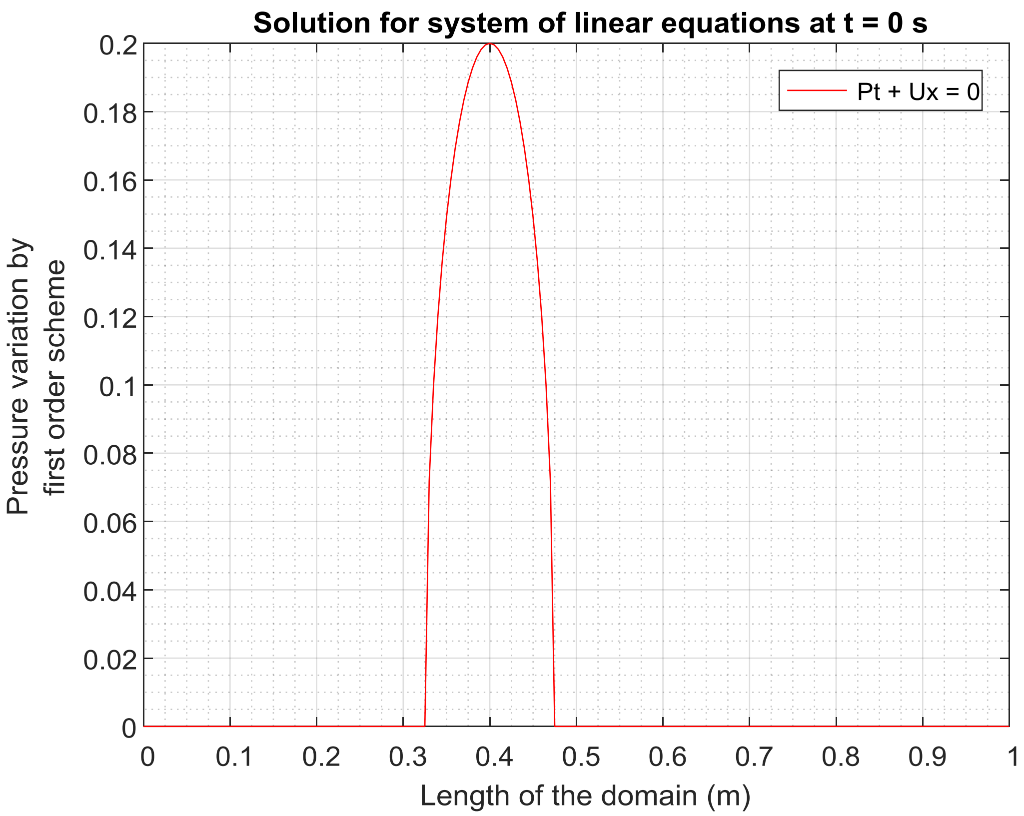


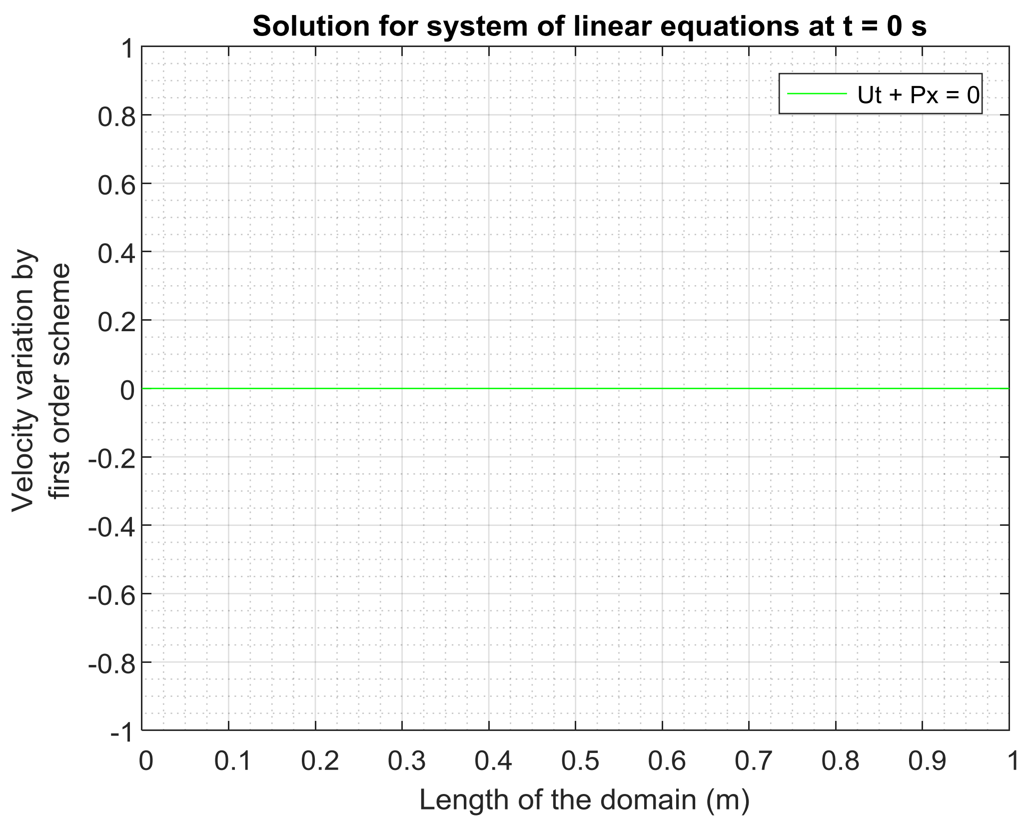
173109003 | Sanit Prashant Bhatkar

Results and Discussion

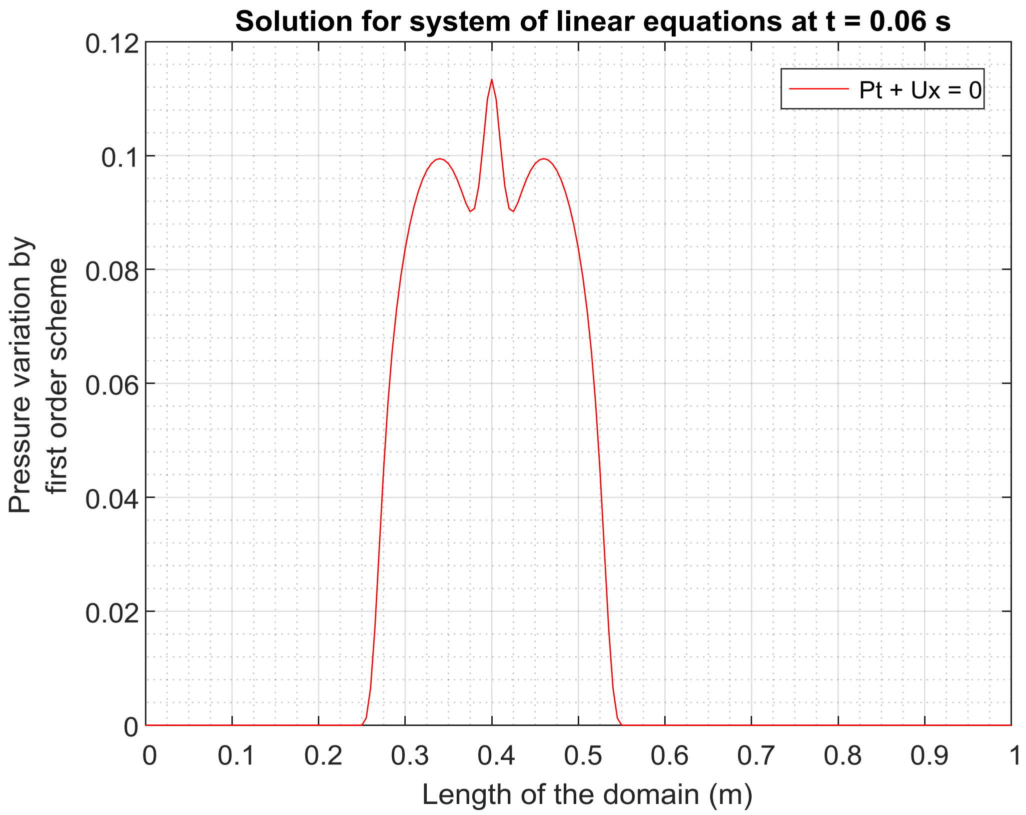
Problem 1 by upwind scheme

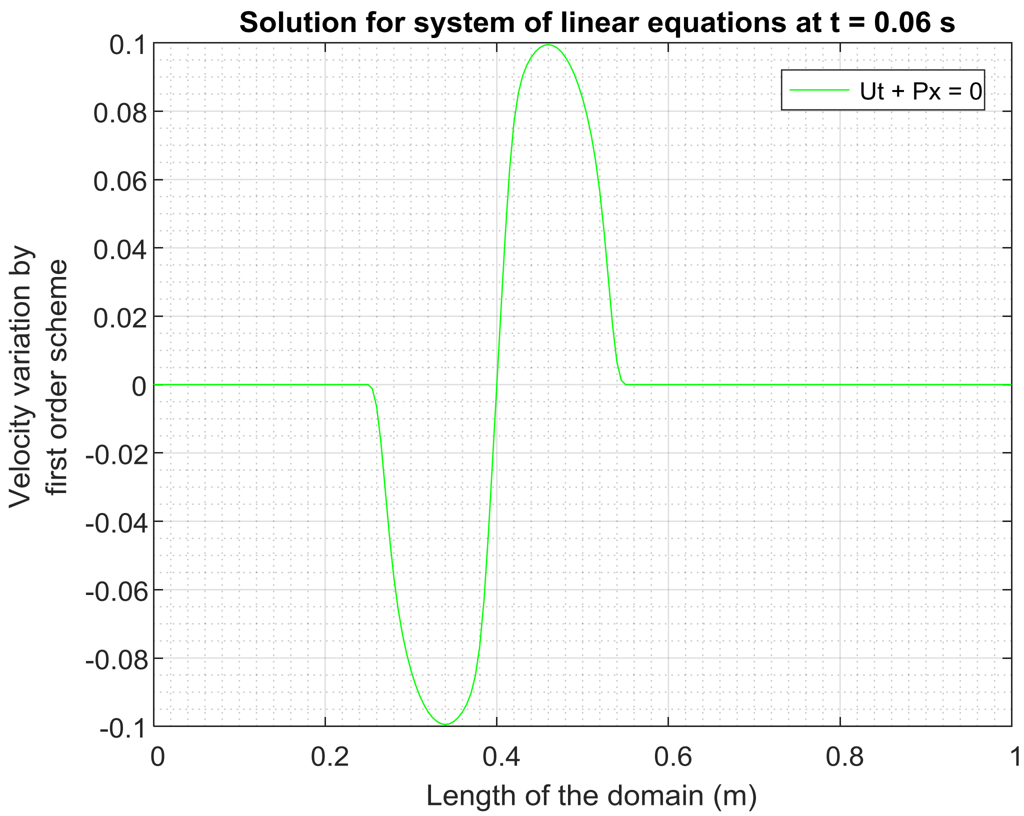
A. Solution of at t = 0



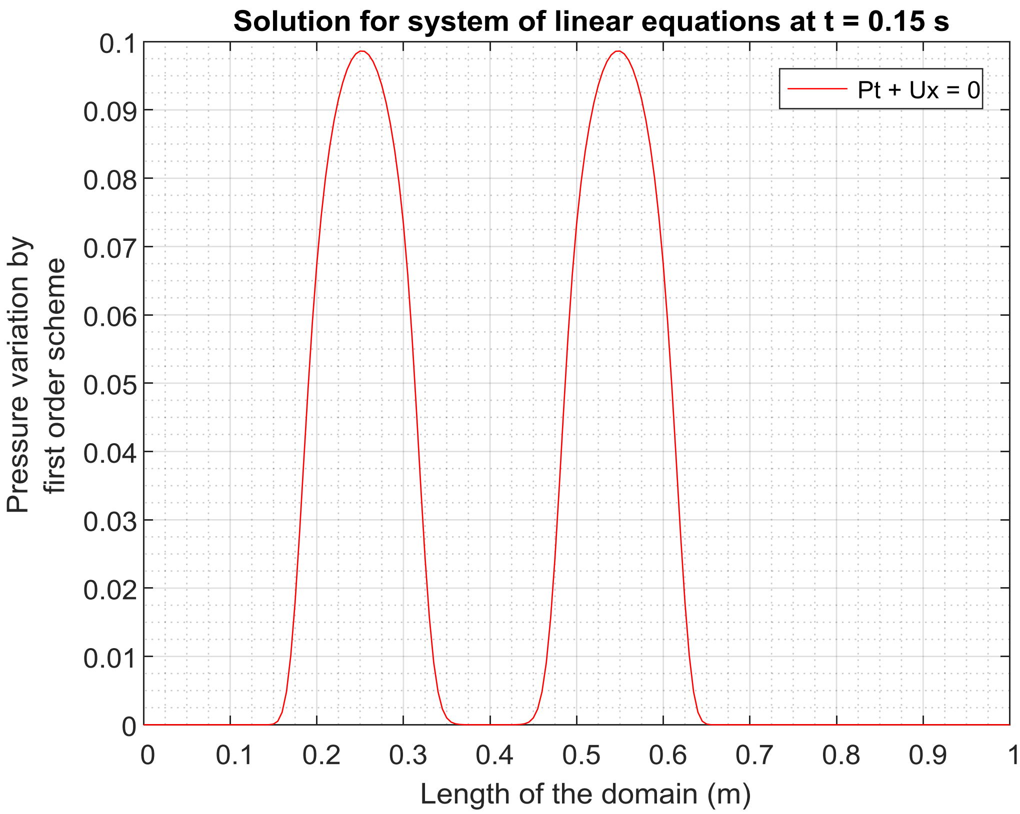


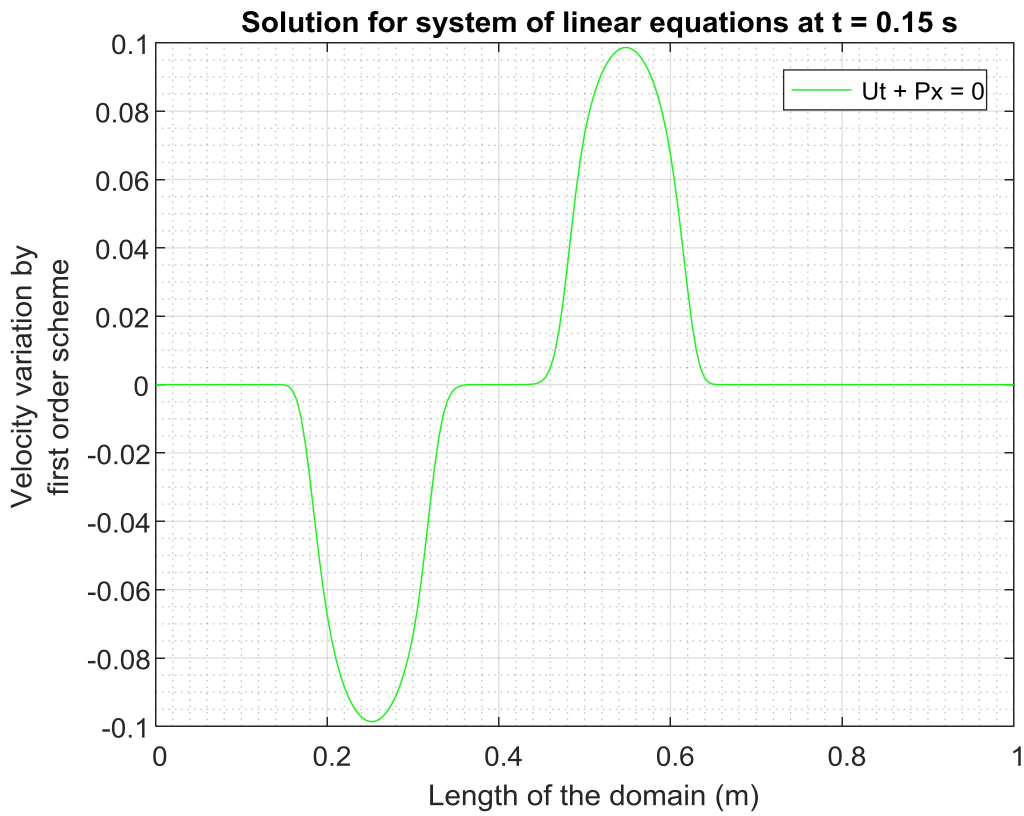
B. Solution of at t = 0.06



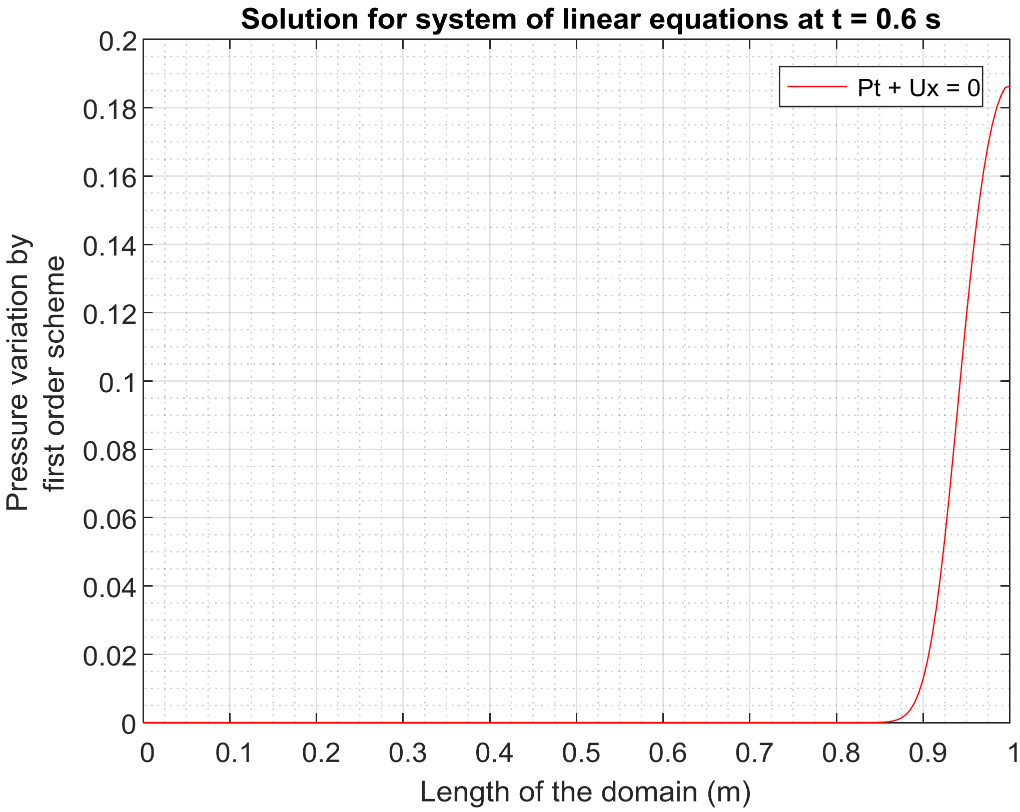


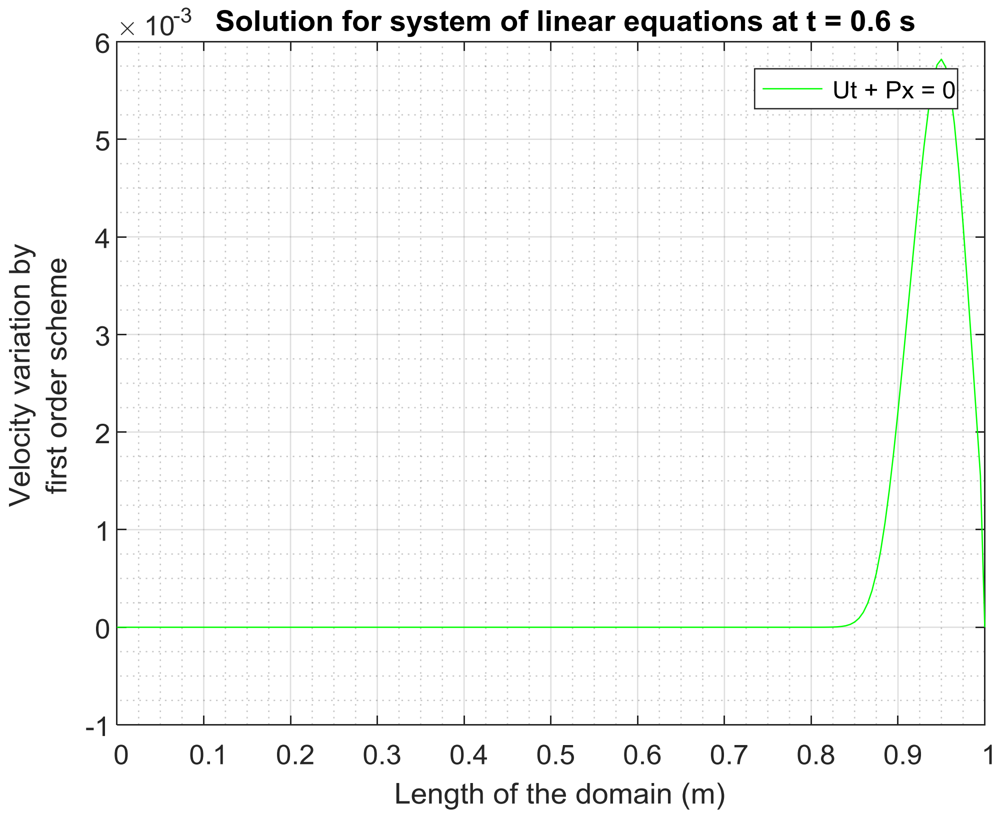
C. Solution of at t = 0.15





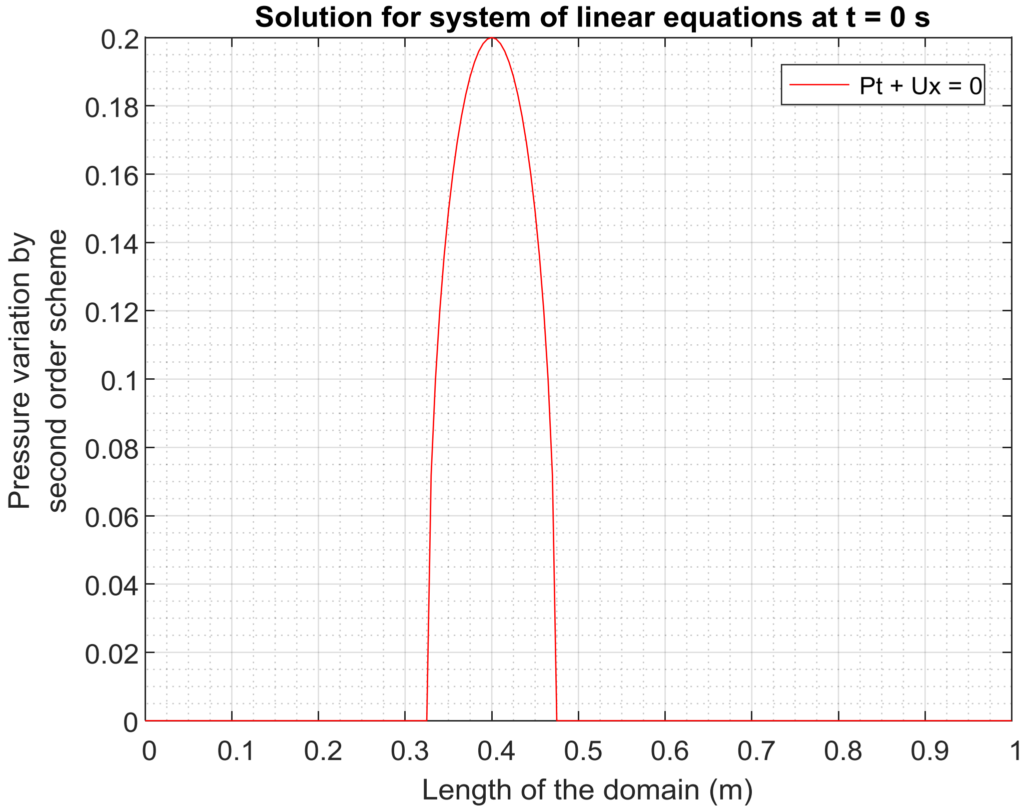
D. Solution of at t = 0.6

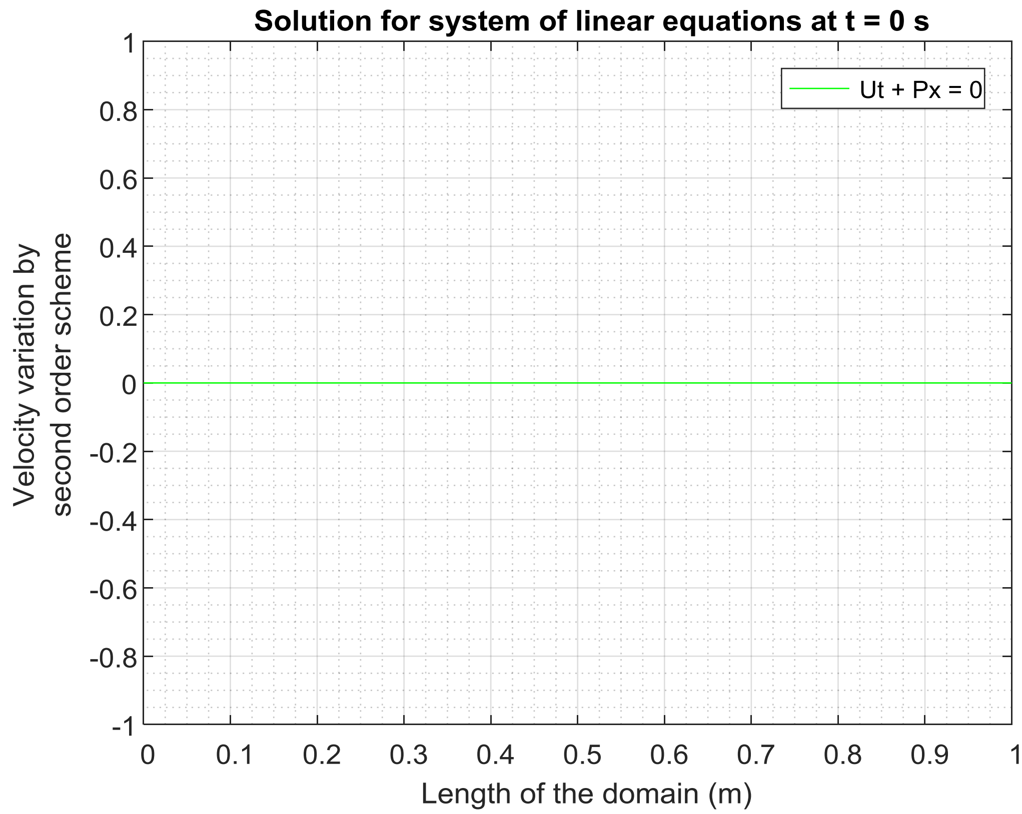




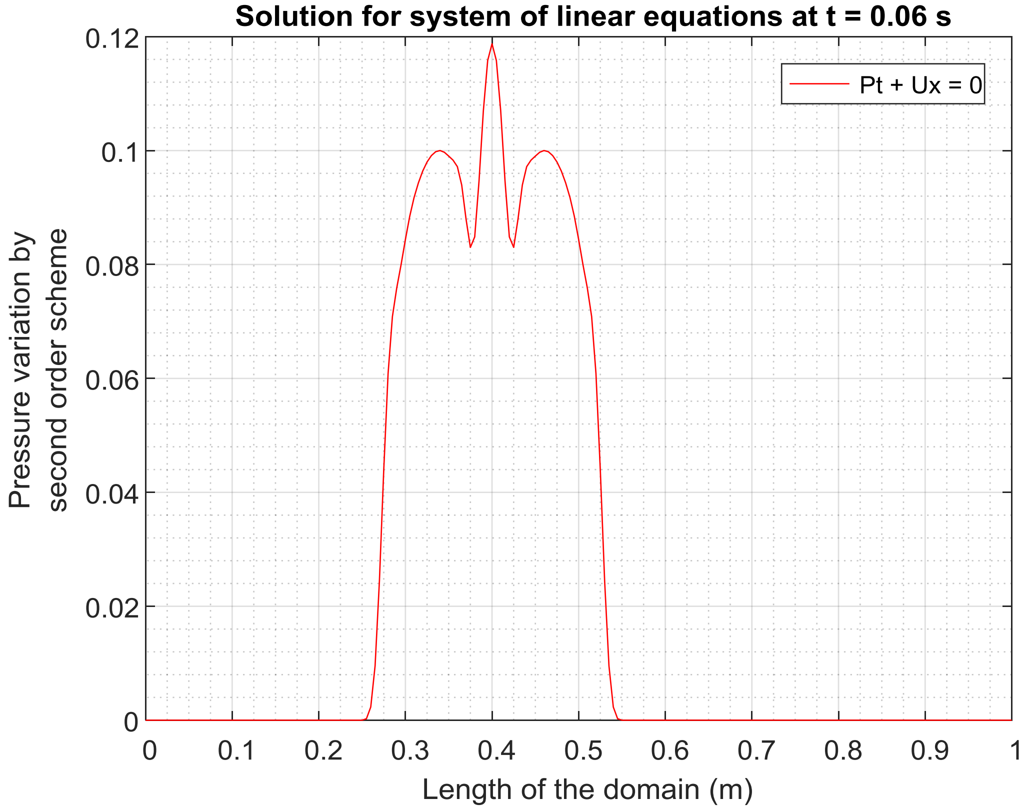
Problem 1 by central scheme

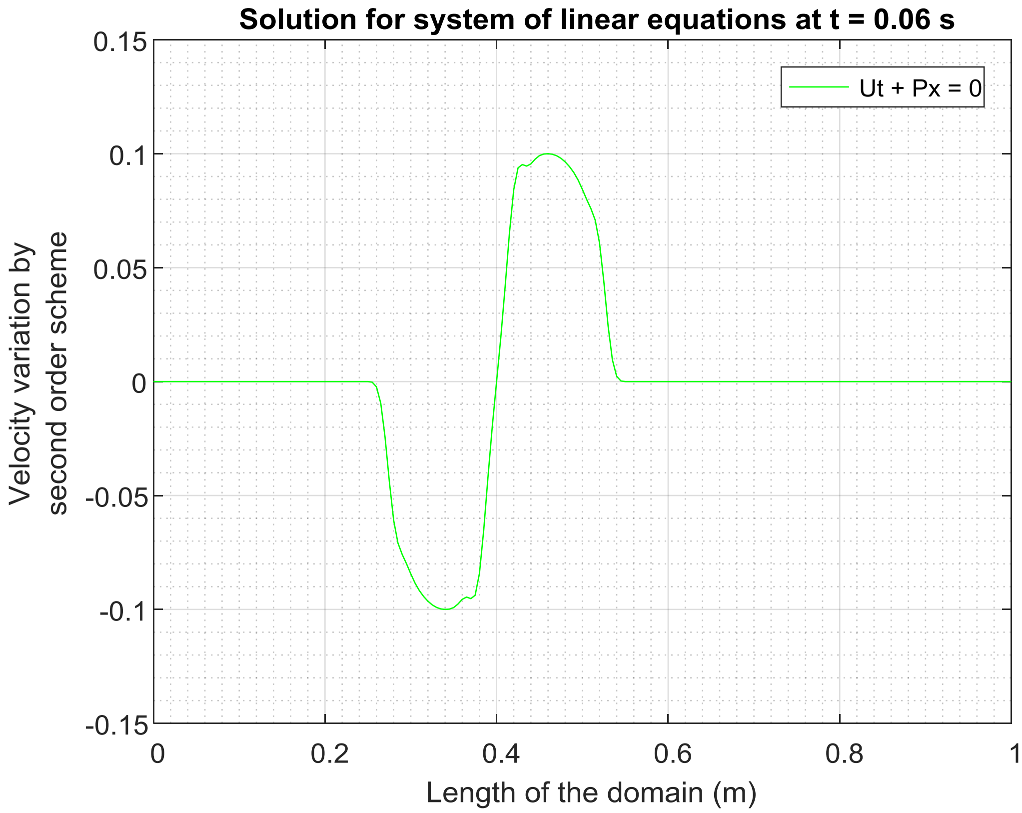
A. Solution at t = 0



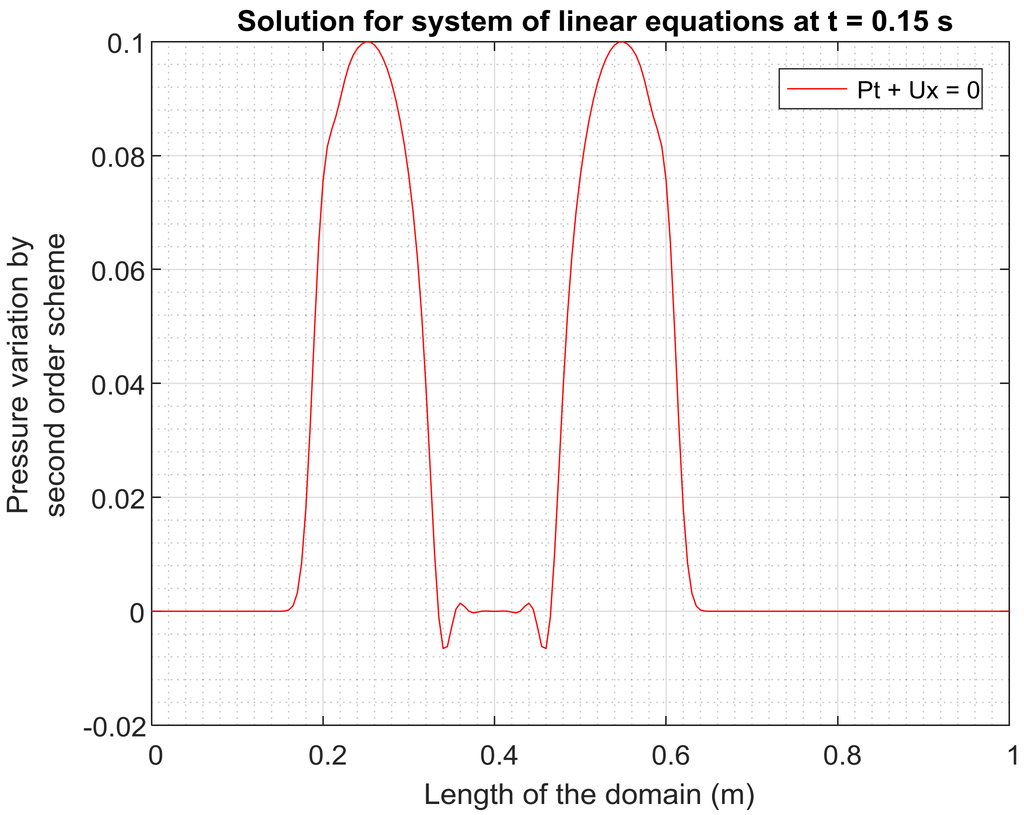


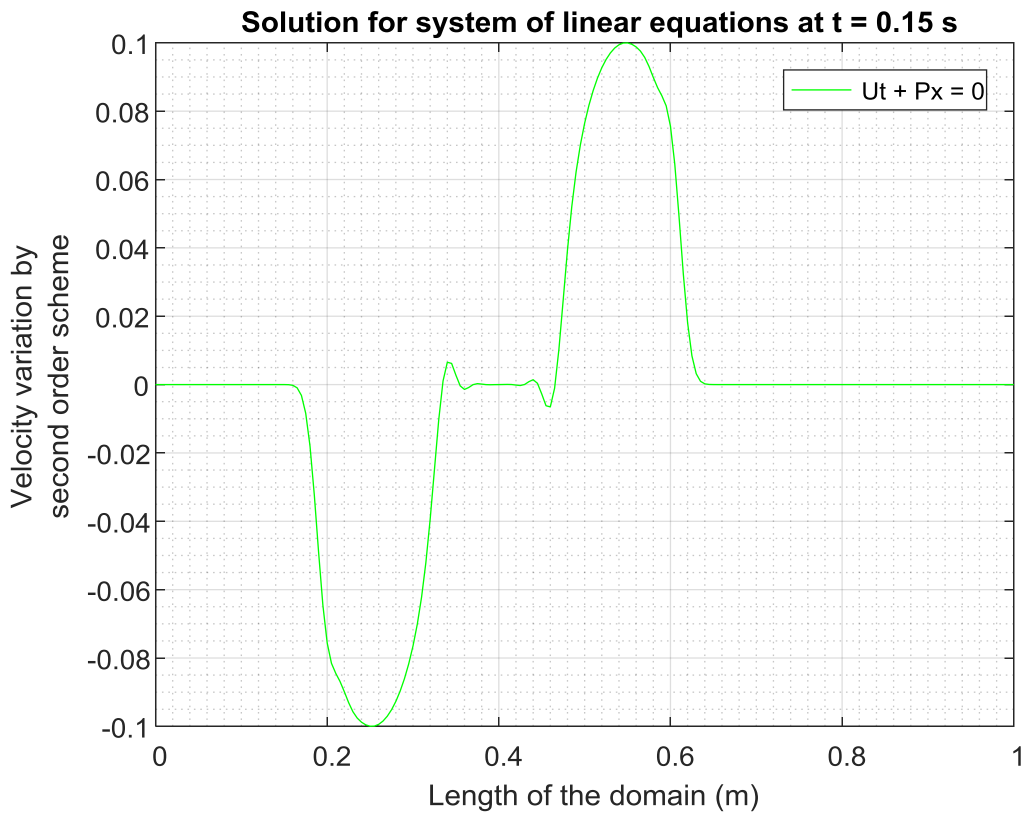
B. Solution at t = 0.06



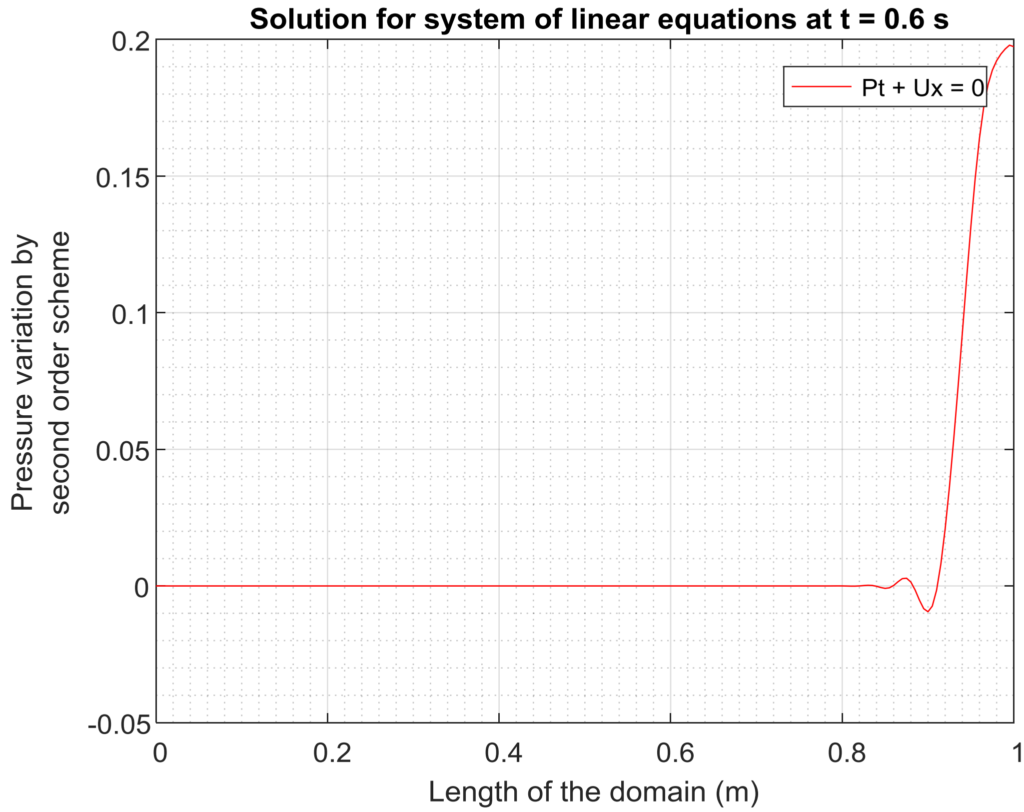


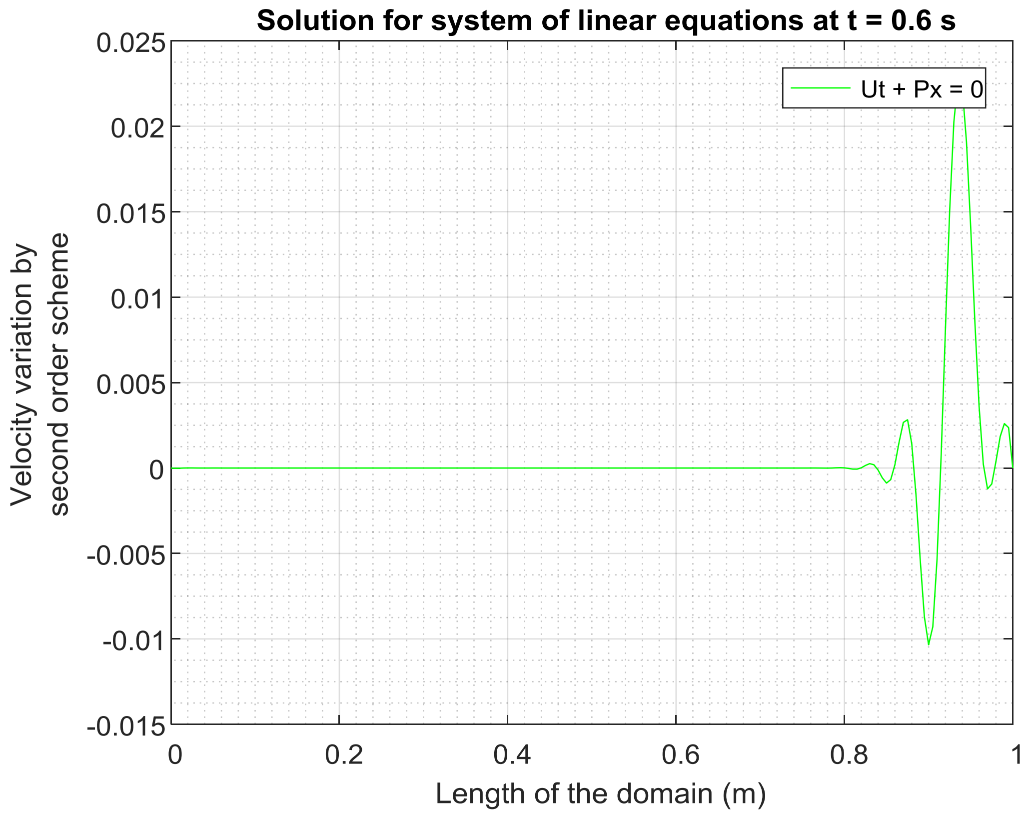
C. Solution at t = 0.15





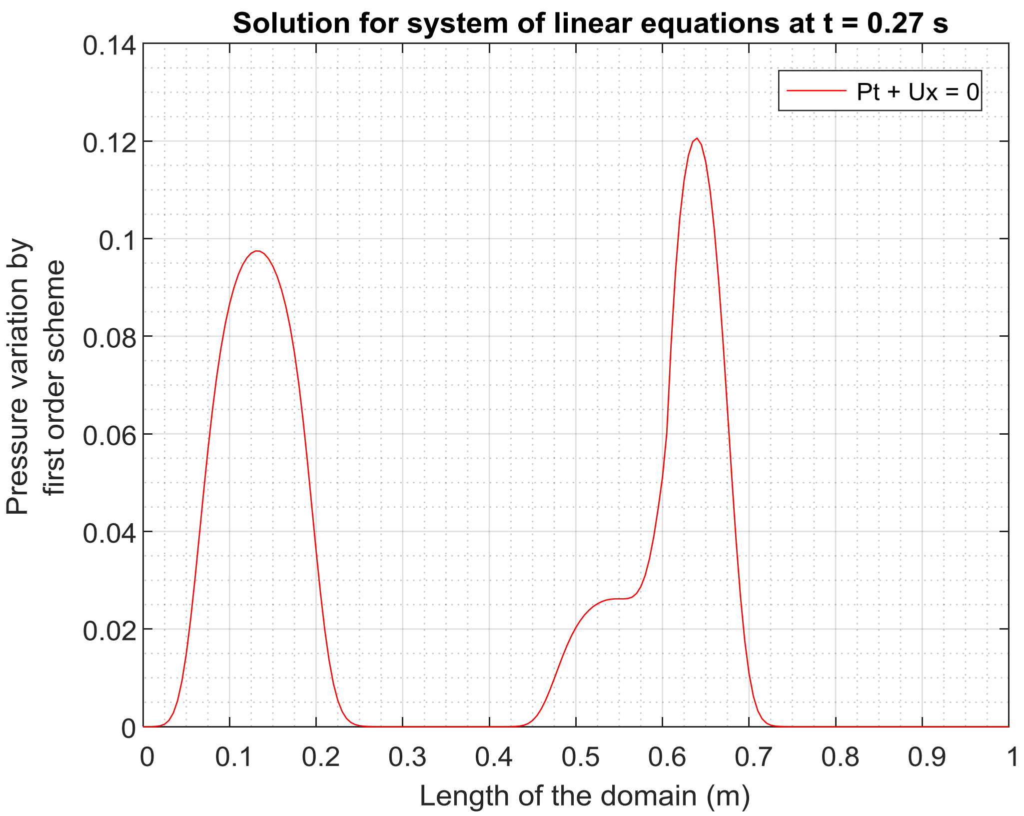
D. Solution at t = 0.6

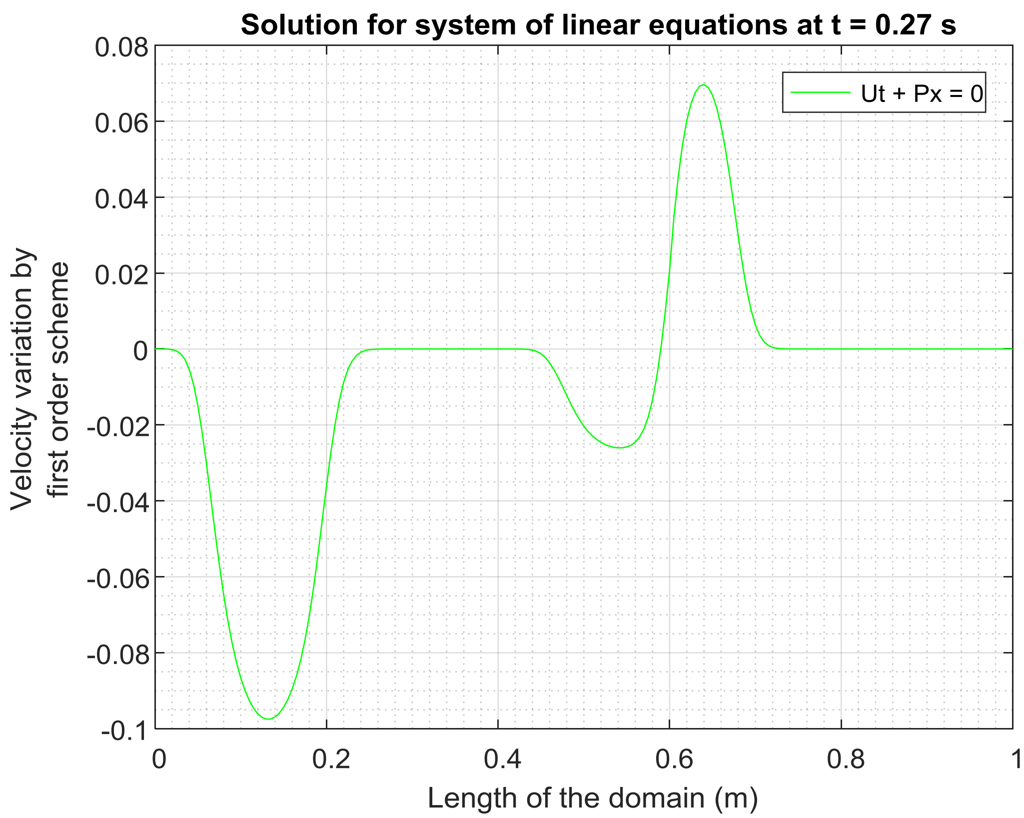




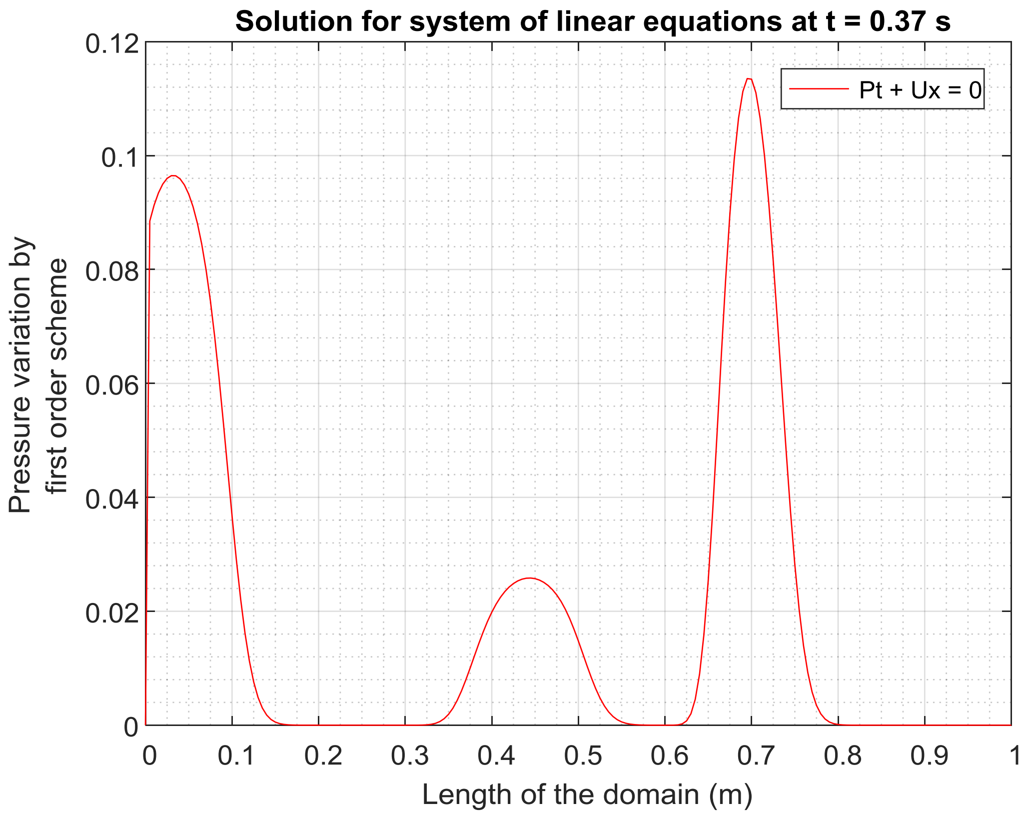
Problem 2 by upwind scheme

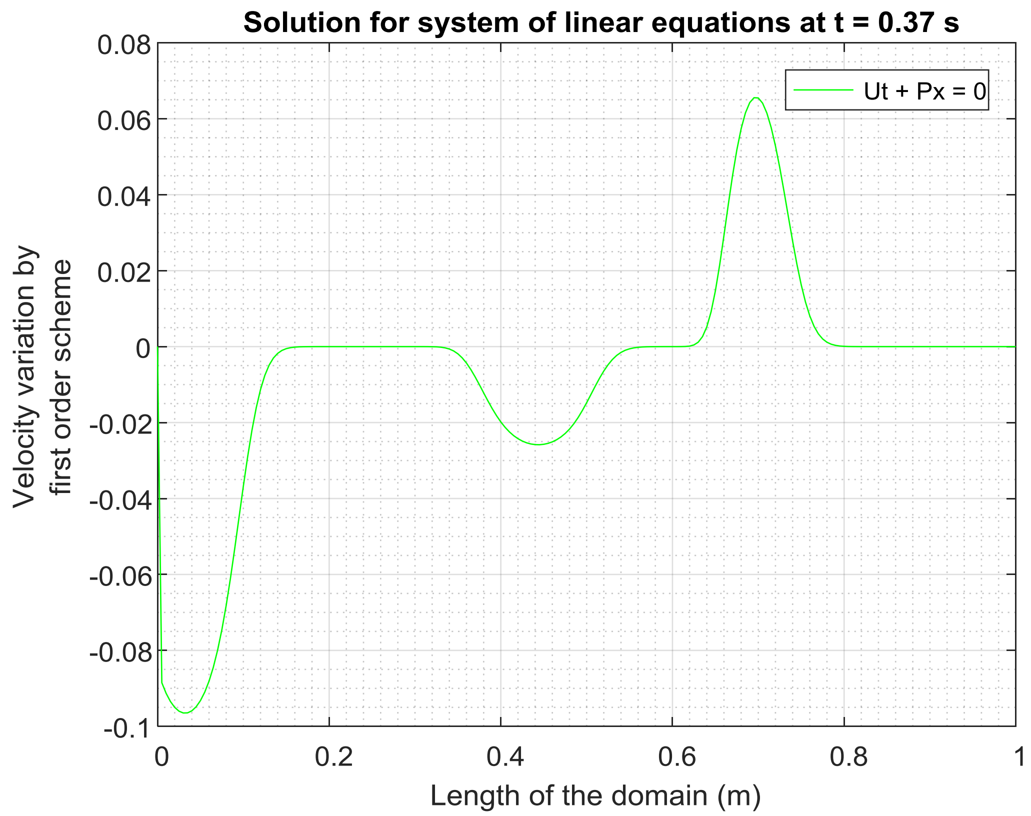
A. Solution at t = 0.27



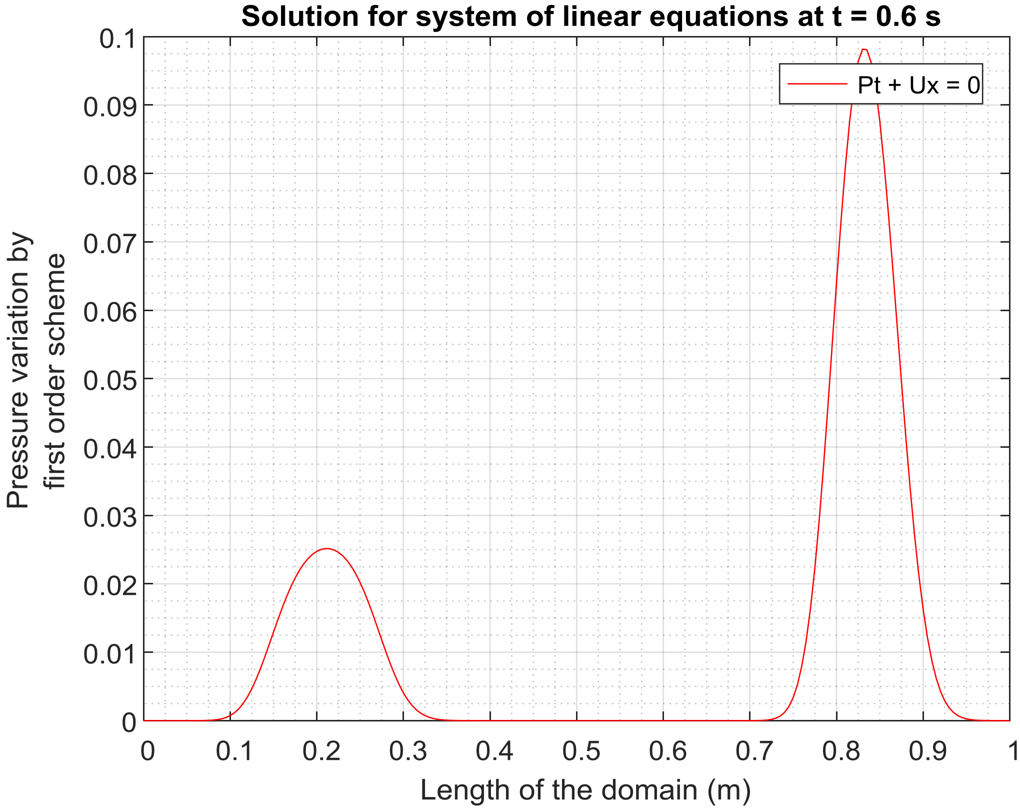


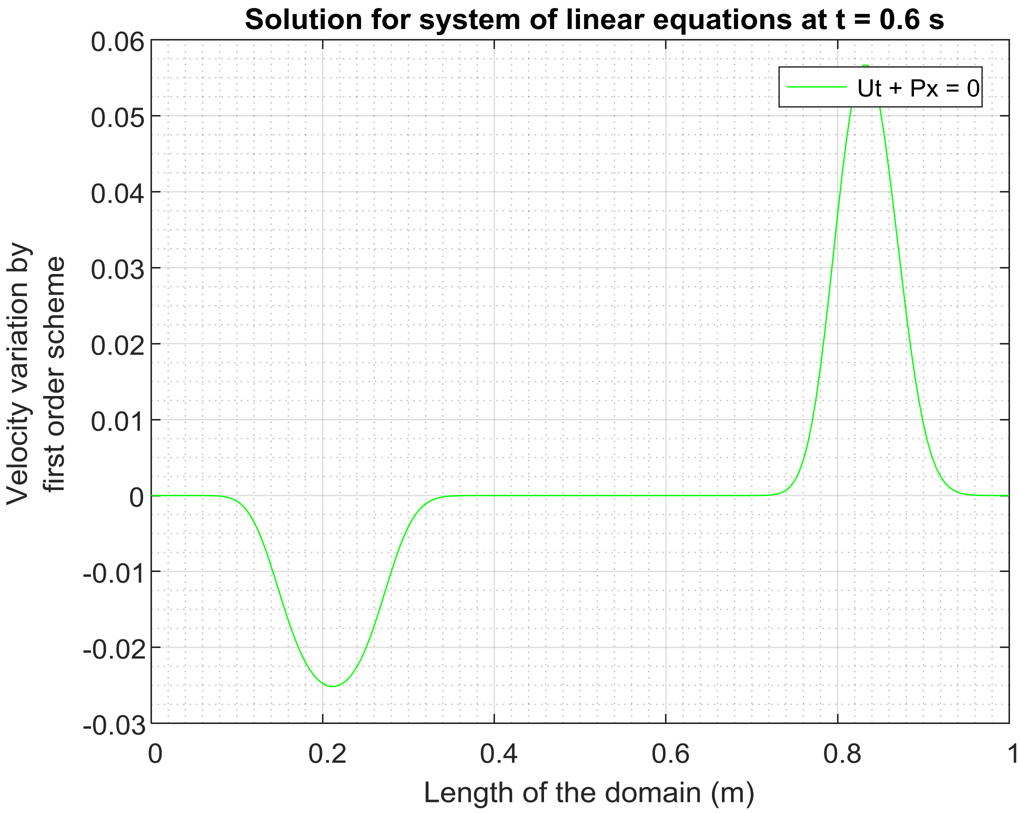
B. Solution at t = 0.37





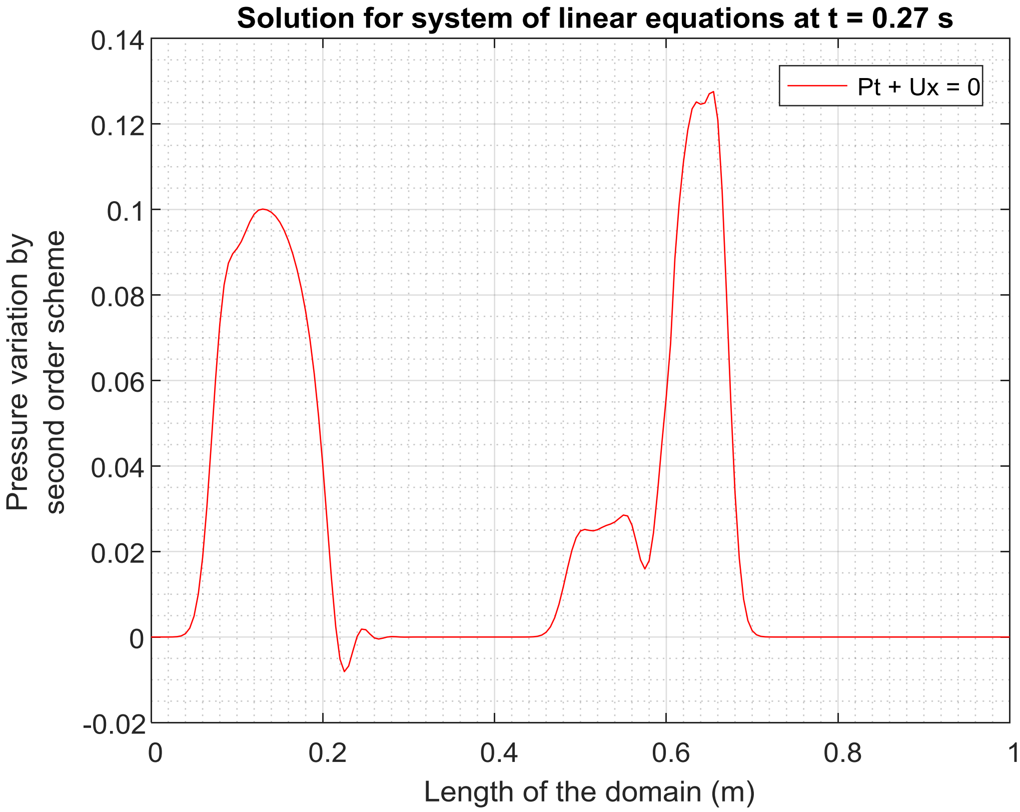
C. Solution at t = 0.6

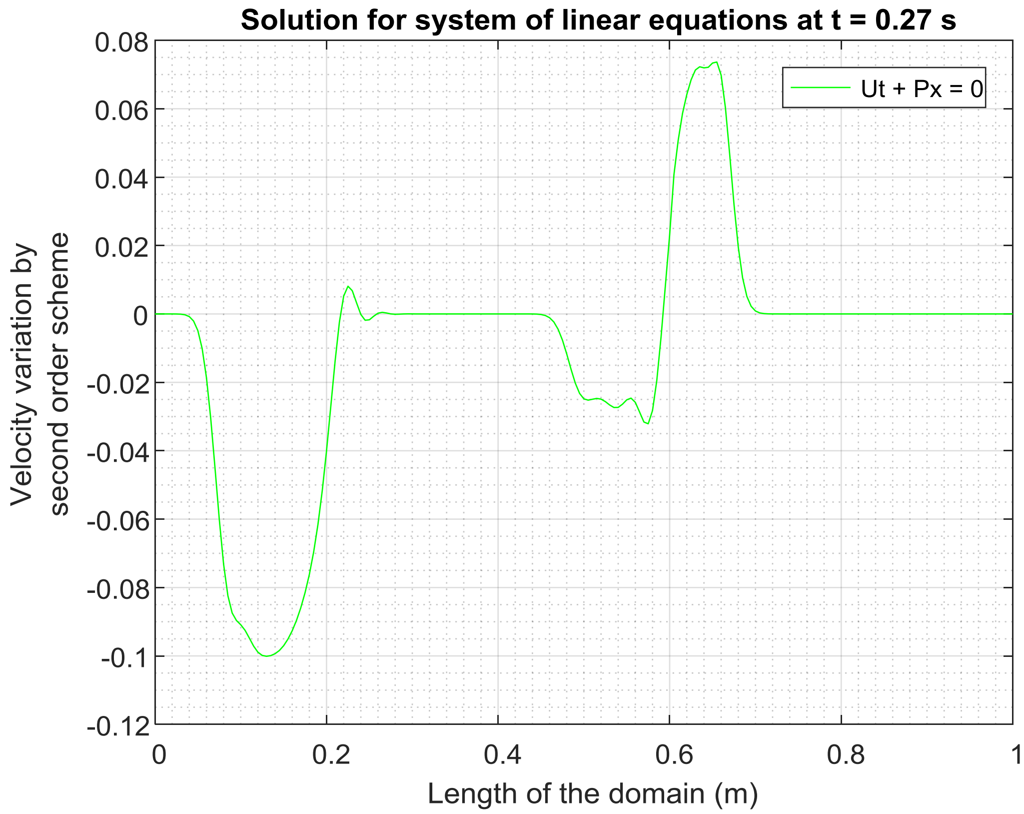




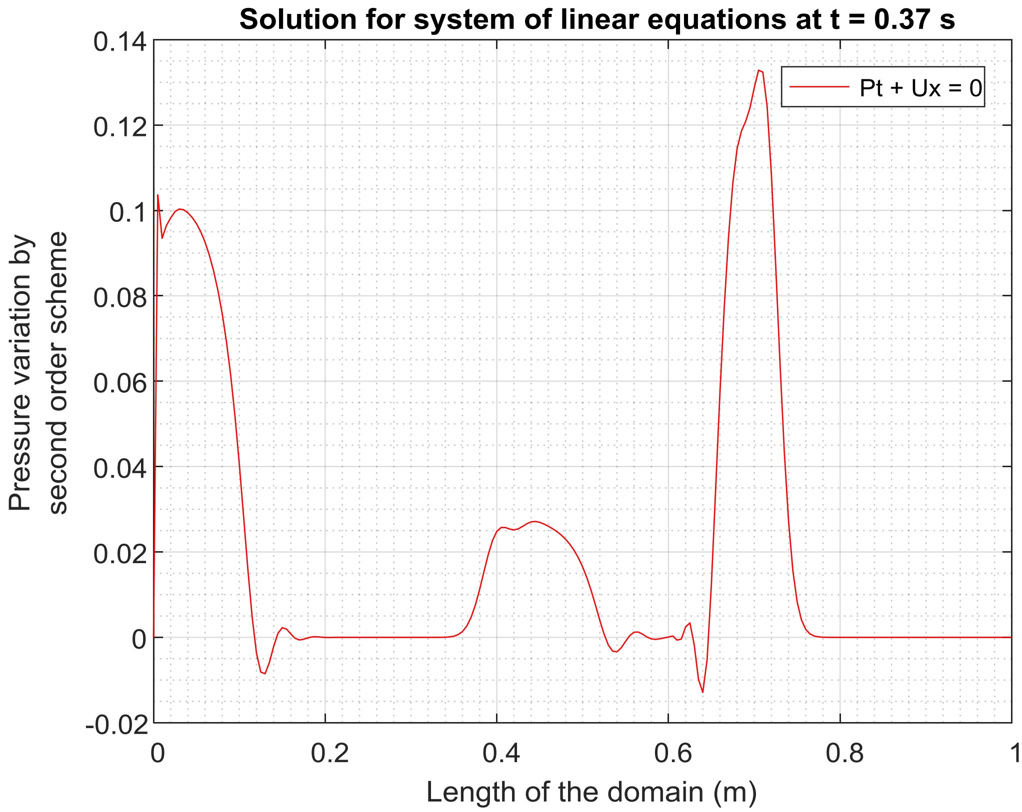
Problem 2 by central difference scheme

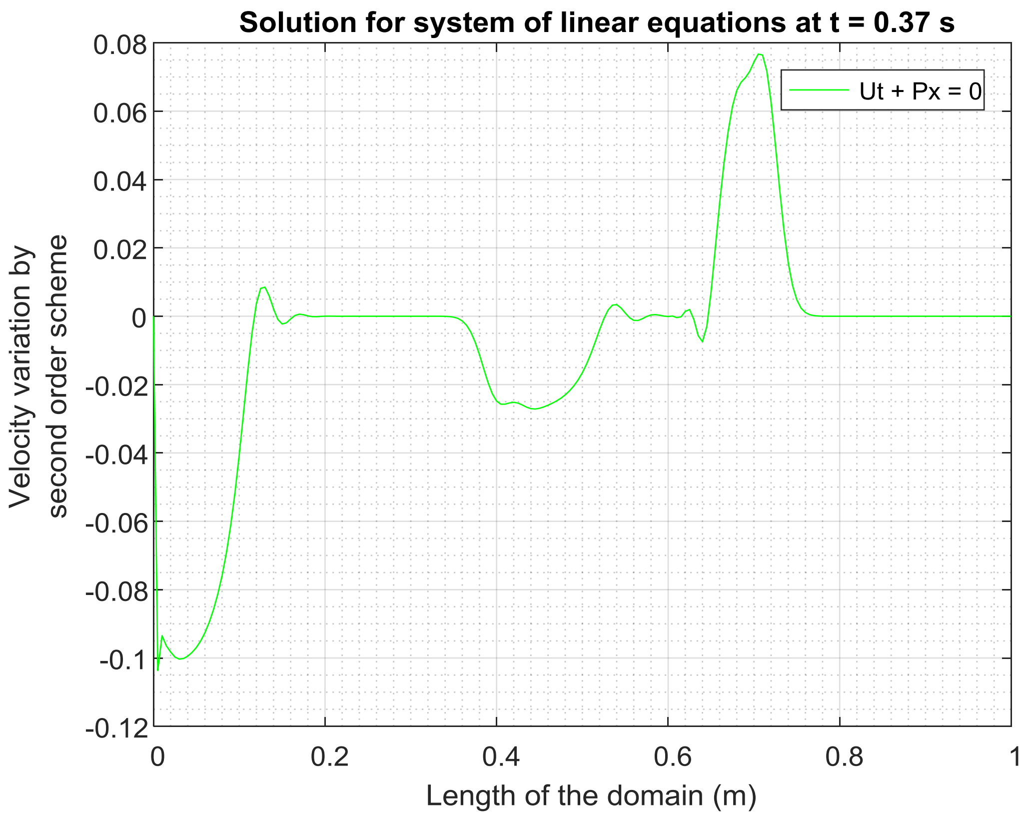
A. Solution at t = 0.27



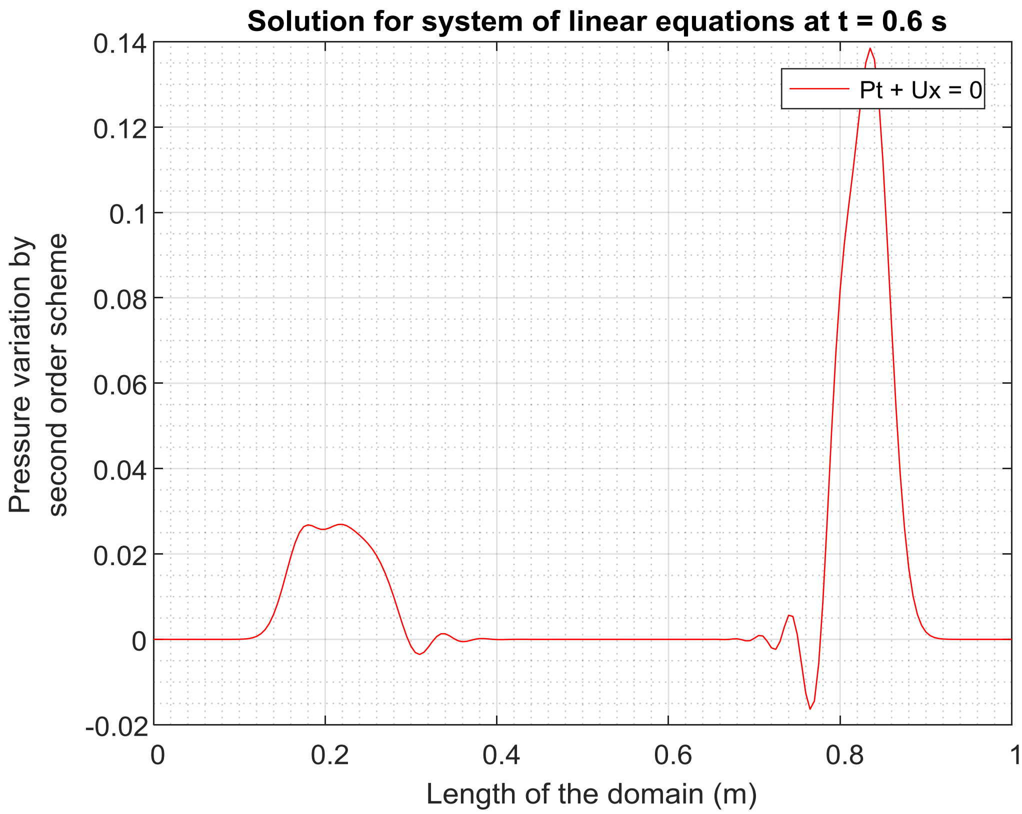


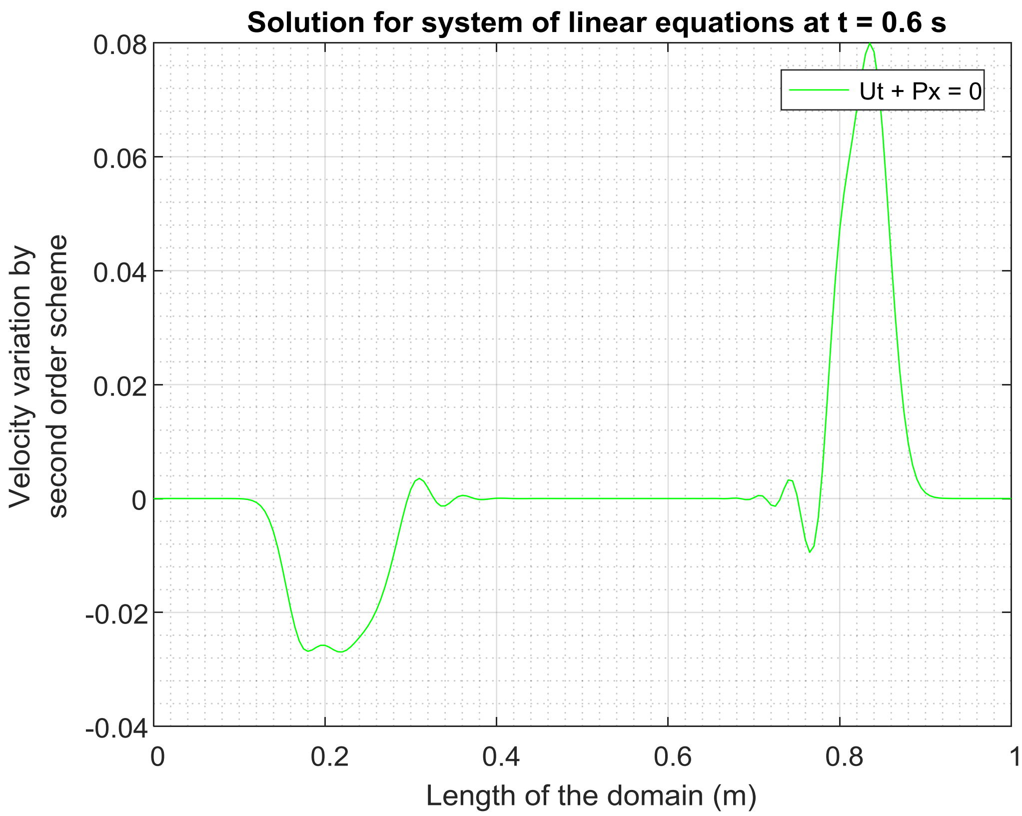
B. Solution at t = 0.37





C. Solution at t = 0.6





Code of the problem

```
%==============================================================%
% Assignment 2 Numerical Methods for Conservation Laws AE 617
%
% Assignment Number 2. System of Linear Equations
%
% AUTHOR:
% Sanit P. Bhatkar (173109003@iitb.ac.in)
% Roll No: 173109003
% Place: IIT BOMBAY.
%==============================================================%

clc
clearvars

%% Input variables

xo=0.4;
xm=0.075;
pm=0.2;
dx=0.005;
dt=0.004;
prbno=input('\nInput problem number: ');
fprintf('Scheme\n1.FO Upwind  2.SO Central Difference\n');
scno=input('\nScheme Number: ');
tf=input('\nInput final time: ');




%% Grid formation

x=0:dx:1;
t=0:dt:tf;

%% Initial condition

[sx sx]=size(x);
[st st]=size(t);

% Initial condition for rho

rho=ones(1,sx);

if prbno==2

dum=find(x > 0.6);
i=dum(1);

while i <= dum(end)

rho(1,i)=3;
i=i+1;

end

end

k=1;
```

```
c=(k./rho).^0.5;
z=rho.*c;

u=zeros(1,sx);
p=zeros(1,sx);


dum=find(abs(x-xo)<(xm));

i=dum(1);

while i <= dum(end)

p_ini=pm*(1-((x(i)-xo)/xm)^2)^0.5;
p(1,i)=p_ini;
i=i+1;

end



%% Definition of the system of equations

% Ux+Fx=0 is type of equations

U=zeros(2,sx);
dU=zeros(2,sx-1);
U(1,1:end)=p;
U(2,1:end)=u;

Uo=zeros(2,sx);
Uo(1,1:end)=p;
Uo(2,1:end)=u;

%% Initial condition plot

figure
plot(x,U(1,1:end),'-r');
legend('Pt + Ux = 0');
xlabel('Length of the domain (m)');
ylabel({'Pressure variation'});
title(['Solution for system of linear equations at t = ',num2str(0),' s']);

set(gca,'XMinorGrid','on');
set(gca,'YMinorGrid','on');
grid on

figure
plot(x,U(2,1:end),'-g');
legend('Ut + Px = 0');
xlabel('Length of the domain (m)');
ylabel({'Velocity variation'});
title(['Solution for system of linear equations at t = ',num2str(0),' s']);

set(gca,'XMinorGrid','on');
set(gca,'YMinorGrid','on');
grid on
```

```
% dU_(i+1/2) calculations

for i=1:sx-1

    dU(1,i)=U(1,i+1)-U(1,i);
    dU(2,i)=U(2,i+1)-U(2,i);

end


%% Definition of eigen vector

for i=1:sx-1

% a1r1+a2r2
% r1=[-rho*c 1]
% r2=[rho*c 1]

R=[-z(i),z(i);1,1];
A=zeros(2,sx-1);

%Calculation for a1 and a2

    if rank(dU(1:end,i))~=0
    A(1:end,i)=(R)\dU(1:end,i);
    end

end



%% Calculation for lambda

for i=1:sx

lamb(1:2,i)=[-c(i);c(i)];
sg(1:2,i)=lamb(1:2,i)./c(i);

end

if scno==1
%% First order Upwind scheme

for d=1:st

%% Inner node formulation

for i=2:sx-1

R=[-z(i),z(i);1,1];

% U(1:end,i)=Uo(1:end,i)-(dt/dx)*(max(sg(1),0)*lamb(1)*A(1,i-
1)*R(1:end,1)+max(sg(2),0)*lamb(2)*A(2,i-
1)*R(1:end,2)+min(sg(1),0)*lamb(1)*A(1,i)*R(1:end,1)+min(sg(2),0)*lamb(2)*A(2
,i)*R(1:end,2));

% when lamb > 0
U(1:end,i)=Uo(1:end,i)-(dt/dx)*(max(sg(1,i),0)*lamb(1,i)*A(1,i-
1)*R(1:end,1));
```

```
U(1:end,i)=U(1:end,i)-(dt/dx)*(max(sg(2,i),0)*lamb(2,i)*A(2,i-1)*R(1:end,2));

% when lamb < 0
% Important note:a
% As the min(sg(2),0) adds extra negative sign to the formula
% Instead of negative sign, positive sign is used
% Formula is still unchanged and it can be shown on paper

U(1:end,i)=U(1:end,i)+(dt/dx)*(min(sg(1,i),0)*lamb(1,i)*A(1,i)*R(1:end,1));
U(1:end,i)=U(1:end,i)+(dt/dx)*(min(sg(2,i),0)*lamb(2,i)*A(2,i)*R(1:end,2));


end

%% Boundary node formulation

% Pressure boundary
% when lamb > 0
U(1,end)=Uo(1,end)-(dt/dx)*(max(sg(1,end),0)*lamb(1,end)*A(1,end)*R(1,1));
U(1,end)=U(1,end)-(dt/dx)*(max(sg(2,end),0)*lamb(2,end)*A(2,end)*R(1,2));

% Velocity boundary
U(2,end)=0;

%% Value update

% dU_(i+1/2) calculations

for i=1:sx-1

    dU(1,i)=U(1,i+1)-U(1,i);
    dU(2,i)=U(2,i+1)-U(2,i);

end

%Calculation for a1 and a2
for i=1:sx-1

    R=[-z(i),z(i);1,1];

    if rank(dU(1:end,i))~=0
    A(1:end,i)=(R)\dU(1:end,i);
    end

end

Uo=U;

end



figure
plot(x,U(1,1:end),'-r');
legend('Pt + Ux = 0');
xlabel('Length of the domain (m)');
ylabel({'Pressure variation by';'first order scheme'});
title(['Solution for system of linear equations at t = ',num2str(tf),' s']);

set(gca,'XMinorGrid','on');
```

```
set(gca,'YMinorGrid','on');
grid on

figure
plot(x,U(2,1:end),'-g');
legend('Ut + Px = 0');
xlabel('Length of the domain (m)');
ylabel({'Velocity variation by';'first order scheme'});
title(['Solution for system of linear equations at t = ',num2str(tf),' s']);

set(gca,'XMinorGrid','on');
set(gca,'YMinorGrid','on');
grid on


end

if scno==2

%% Second order central differnce scheme

% Courant number definition

nu=lamb*dt/dx;

for d=1:st
%% Inner node formulation

for i=2:sx-1

R=[-z(i),z(i);1,1];

U(1:end,i)=Uo(1:end,i)-((0.5*dt)/dx)*((1+nu(1,i))*lamb(1,i)*A(1,i-
1)*R(1:end,1)+ (1+nu(2,i))*lamb(2,i)*A(2,i-1)*R(1:end,2));
U(1:end,i)=U(1:end,i)-((0.5*dt)/dx)*((1-nu(1,i))*lamb(1,i)*A(1,i)*R(1:end,1)+
(1-nu(2,i))*lamb(2,i)*A(2,i)*R(1:end,2));

end

%% Boundary node formulation

% Pressure boundary
U(1,end)=Uo(1,end)-((0.5*dt)/dx)*((1+nu(1,end))*lamb(1,end)*A(1,end)*R(1,1)+
(1+nu(2,end))*lamb(2,end)*A(2,end)*R(1,2));

% Pressure boundary
U(2,end)=0;

%% Value update

% dU_(i+1/2) calculations

for i=1:sx-1

    dU(1,i)=U(1,i+1)-U(1,i);
    dU(2,i)=U(2,i+1)-U(2,i);

end

%Calculation for a1 and a2
```

```
for i=1:sx-1

   R=[-z(i),z(i);1,1];

    if rank(dU(1:end,i))~=0
    A(1:end,i)=(R)\dU(1:end,i);
    end

end

Uo=U;

end


figure
plot(x,U(1,1:end),'-r');
legend('Pt + Ux = 0');
xlabel('Length of the domain (m)');
ylabel({'Pressure variation by';'second order scheme'});
title(['Solution for system of linear equations at t = ',num2str(tf),' s']);

set(gca,'XMinorGrid','on');
set(gca,'YMinorGrid','on');
grid on

figure
plot(x,U(2,1:end),'-g');
legend('Ut + Px = 0');
xlabel('Length of the domain (m)');
ylabel({'Velocity variation by';'second order scheme'});
title(['Solution for system of linear equations at t = ',num2str(tf),' s']);

set(gca,'XMinorGrid','on');
set(gca,'YMinorGrid','on');
grid on



end
```