clear all;

clc;

close all;

%this code solves, and simulates a scalar conservation law equation of

% df(x,t)/dt + a df(x,t)/dx = 0

hold on

%numerical grid

xlength=2; %grid length, upper limit of domain set to 2

n=1000; %number of grid points

h=xlength/(n-1); %gap between grid points

% set numerical and physical parameters

D=1;

U=1;

dt = 0.000001;

x = zeros(1,n);

f = zeros(1,n);

fn = zeros(1,n);

freal= zeros(1,n);

x(1) = 0D0;

% assigning values to array x(i)

for i=2:n

x(i)=x(i-1)+h;

end

% initialising function

for i = 1:n

f(i) = 0.75\*exp(-((x(i)-0.5)/0.1).^2);

end

for time= [0.5:0.5:1] % creates time matrix values with increments of 0.5

nt = 0.5/dt; % find out %number of iterations (nt)

for k = 1:nt

for i = 2:n-1

% Determining difference equation for diffusion-only situation

% flux= D \* (f(i+1)-(2\*f(i))+f(i-1));

fn(i) = f(i)-((U\*(dt/(2\*h)))\*(f(i+1)-f(i-1)))+((D\*(dt/(h^2)))\*(f(i+1)-(2\*f(i))+f(i-1)));

end

% boundary condition

fn(1) = fn(n);

fn(n) = fn(1);

f = fn;

end

warning off

% Plots Distance vs Concentration for diffusion-only situation

plot(x,f)

grid on

xlabel('Distance (m)');

ylabel('Concentration (c)')

legend('t=0.5','t=1')

% shg

% pause(0.1)

end

hold off;