```matlab
clear variables; close all; clc;

% approximate order of Upwind Scheme and Lax-Wendroff Method
% for linear advection PDE with
% advection speed a = 1

% u_t + u_x = 0
% u0(x) = {1, x<= 0.5; 0, x>0.5}
% tf = 1, 201 gridpoints, 500 timesteps


%% UPWIND SCHEME

% Temporal convergence

Nx = 100; % N intervals, N+1 points
tf = 1;
x0 = 0; xf = 1;
xvals = linspace(x0, xf, Nx + 1)';
dx = xvals(2) - xvals(1);
Dx = gallery('tridiag', Nx + 1, -1, 1, 0);
Dx(1, end) = -1; % period BC


u0 = @(x) sin(2*pi*x); % Heaviside step function about x=0.5

exact = u0(xvals - tf);
lambdavals = (0.01:0.02:1)';
errors = zeros(numel(lambdavals), 1);

for k = 1:numel(lambdavals)
    dt = lambdavals(k)*dx;
    tvals = (0:dt:tf)';
    if tvals(end) ~= tf
        tvals = [tvals; tf];
    end

    u = u0(xvals); % initial condition

    for n = 2:numel(tvals)
        u = u - (dt/dx)*(Dx*u);
    end
    errors(k) = (dx*sum(abs(exact - u)));
end

figure(1); clf;
loglog(lambdavals, errors, 'black--', 'LineWidth', 1); hold on;
loglog(lambdavals, lambdavals, 'b-', 'LineWidth', 1.5);
title('Upwind scheme'); xlabel('\lambda'); ylabel('error');
legend('Upwind', 'Order 1');


% Spatial Convergence
Nxvals = [20, 40, 80, 160, 320, 640];
errors = zeros(numel(Nxvals), 1);
```

```
for k = 1:numel(errors)
    Nx = Nxvals(k);
    xvals = linspace(x0, xf, Nx + 1)';
    dx = xvals(2) - xvals(1);
    Dx = gallery('tridiag', Nx + 1, -1, 1, 0);
    Dx(1, end) = -1; % period BC
    dt = 0.5*dx;
    tvals = (0:dt:tf)';
    if tvals(end) ~= tf
        tvals = [tvals; tf];
    end
    exact = u0(xvals - tf);

    u = u0(xvals); % initial condition

    for n = 2:numel(tvals)
        u = u - (dt/dx)*(Dx*u);
    end
    errors(k) = (dx*sum(abs(exact - u)));
end

disp('Upwind Spatial Accuracy Order = ');
disp(log2(errors(1:end-1)./errors(2:end)));


%% LAX-WENDROFF
clear variables;

% Temporal convergence

Nx = 200; % N intervals, N+1 points
tf = 1;
x0 = 0; xf = 1;
xvals = linspace(x0, xf, Nx + 1)';
xvals = xvals(1:end-1);
dx = xvals(2) - xvals(1);

Dx = gallery('tridiag', Nx, -1, 0, 1);
Dx(1, end) = -1; Dx(end, 1) = 1; % periodic BCs

Dxx = gallery('tridiag', Nx, 1, -2, 1);
Dxx(1, end) = 1; Dxx(end, 1) = 1; % periodic BCs

u0 = @(x) sin(2*pi*x);
exact = u0(xvals - tf);
lambdavals = (0.01:0.02:1)';
errors = zeros(numel(lambdavals), 1);

for k = 1:numel(lambdavals)
    dt = lambdavals(k)*dx;
    tvals = (0:dt:tf)';
    if tvals(end) ~= tf
        tvals = [tvals; tf];
    end
```

```
    u = u0(xvals); % initial condition

    for n = 2:numel(tvals)
        u = u - (dt/(2*dx))*(Dx*u) + (dt^2/(2*dx^2))*(Dxx*u); % Lax-Wendroff
    end
    errors(k) = (dx*sum(abs(exact - u)));
end

figure(2); clf;
loglog(lambdavals, errors, 'black--', 'LineWidth', 1); hold on;
loglog(lambdavals, lambdavals.^2, 'b-', 'LineWidth', 1.5);
title('Lax-Wendroff scheme'); xlabel('\lambda'); ylabel('error');
legend('Upwind', 'Order 2');


% Spatial Convergence
clear variables;
Nxvals = [20, 40, 80, 160, 320, 640];
tf = 1;
x0 = 0; xf = 1;
u0 = @(x) sin(2*pi*x);
errors = zeros(numel(Nxvals), 1);

for k = 1:numel(errors)
    Nx = Nxvals(k);
    xvals = linspace(x0, xf, Nx + 1)';
    xvals = xvals(1:end-1);
    dx = xvals(2) - xvals(1);

    Dx = gallery('tridiag', Nx, -1, 0, 1);
    Dx(1, end) = -1; Dx(end, 1) = 1; % periodic BCs

    Dxx = gallery('tridiag', Nx, 1, -2, 1);
    Dxx(1, end) = 1; Dxx(end, 1) = 1; % periodic BCs

    dt = 0.5*dx;
    tvals = (0:dt:tf)';
    if tvals(end) ~= tf
        tvals = [tvals; tf];
    end
    exact = u0(xvals - tf);

    u = u0(xvals); % initial condition

    for n = 2:numel(tvals)
        u = u - (dt/(2*dx))*(Dx*u) + (dt^2/(2*dx^2))*(Dxx*u); % Lax-Wendroff
    end
    errors(k) = (dx*sum(abs(exact - u)));
end

disp('Lax-Wendroff Spatial Accuracy Order = ');
disp(log2(errors(1:end-1)./errors(2:end)));
```