# Unsteady Convection Diffusion

Samadrita Karmakar

**Goal:**

$$u_t + a u_x = 0 \quad x \in (0,1),\ t \in (0, 0.6]$$
$$u(x,0) = u_0(x)\, x \in (0,1)$$
$$u(0,t)\, 1 \in t \in (0, 0.6]$$
$$u_o(x) = 1\ if\ x \leq 0.2$$
$$u_o(x) = 0\ otherwise$$

**a). Compute the Courant number.**
**b). Solve the problem using the Crank-Nicholson scheme in time and linear finite element for the Galerkin scheme in space. Verify solution accuracy.**
**c). Implement the Crank-Nicholson scheme in time and the least-squares formulation in space. Comment the results.**
**d). Solve the problem using the second-order Lax-Wendroff method. Verify solution accuracy. Metion necessary changes for accuracy. Comment the results.**
**e). Implement the second-order two-step Lax-Wendroff method. Comment the results.**

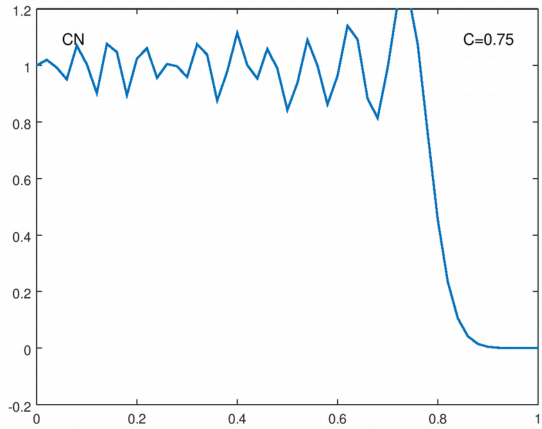
**Solution**

**a). Courant Number is calculated as** $C = \frac{|a| \Delta t}{h} = \frac{1 \times 1.5 \times 10^{-2}}{2 \times 10^{-2}} = 0.75$

**b). Implementation of Crank-Nicholson Scheme in time and linear finite element for the Galerkin scheme in space.**

CODE

```
% MATRICES COMPUTATION
% Loop on elements
for i=1:numel
    unos = ones (ngaus,1);
    h = xnode(i+1)-xnode(i);
    xm = (xnode(i)+xnode(i+1))/2;
    weight = wpg*h/2;
    isp = [i i+1];
    % Loop on Gauss points (numerical quadrature)
    for ig=1:ngaus
        N = N_mef(ig,:);
        Nx = Nxi_mef(ig,:)*2/h;
        w_ig = weight(ig);
        x = xm + h/2*xipg(ig); % x-coordinate of the current Gauss point
        % Matrices assembly
        A(isp,isp) = A(isp,isp) + w_ig*(N'*N - dt_2*(a*Nx)'*N);
        B(isp,isp) = B(isp,isp) + w_ig*dt*(a*Nx)'*N;
        f(isp) = f(isp) + w_ig*(N')*SourceTerm(x);
    end
end
```



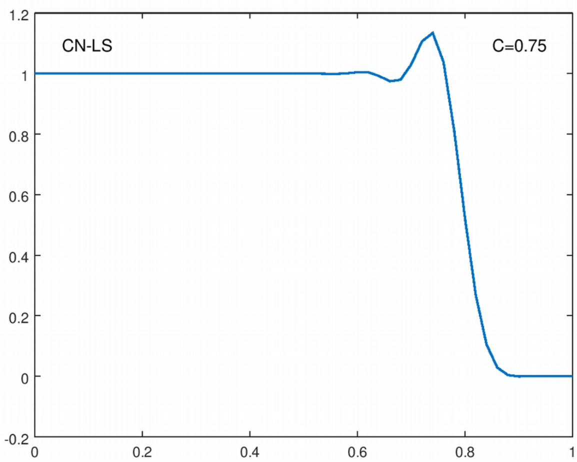

*Crank Nicholson and Galerkin Scheme*

**Comment:** Solution is stable in time discretization since Crank-Nicholson Scheme is unconditionally stable but is unstable in Galerkin scheme in space discretization. This results in oscillations as shown by the graph above.

### c). Implementation of the Crank-Nicholson scheme in time and the least-squares formulation in space.

CODE

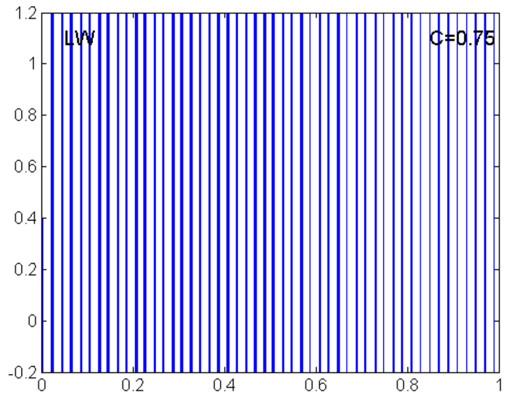
```
% MATRICES COMPUTATION
% Loop on elements
for i=1:numel
    unos = ones (ngaus,1);
    h = xnode(i+1)-xnode(i);
    xm = (xnode(i)+xnode(i+1))/2;
    weight = wpg*h/2;
    isp = [i i+1];
    % Loop on Gauss points (numerical quadrature)
    for ig=1:ngaus
        N = N_mef(ig,:);
        Nx = Nxi_mef(ig,:)*2/h;
        w_ig = weight(ig);
        x = xm + h/2*xipg(ig); % x-coordinate of the current Gauss point
        % Matrices assembly
        A(isp,isp) = A(isp,isp) + w_ig*(N'*N + dt_2*a*(N'*Nx + Nx'*N + dt_2*a*Nx'*Nx));
        B(isp,isp) = B(isp,isp) -  w_ig*dt*a*(N'*Nx + dt_2*a*Nx'*Nx);
      % f(isp) = f(isp) + w_ig*(N')*SourceTerm(x);
    end
end
```




*Crank Nicholson and Least Square Scheme*

**Comment:** Solution is stable since Crank-Nicholson Scheme is unconditionally stable and Galerkin scheme is stabilized in space discretization by using Least Square Method.

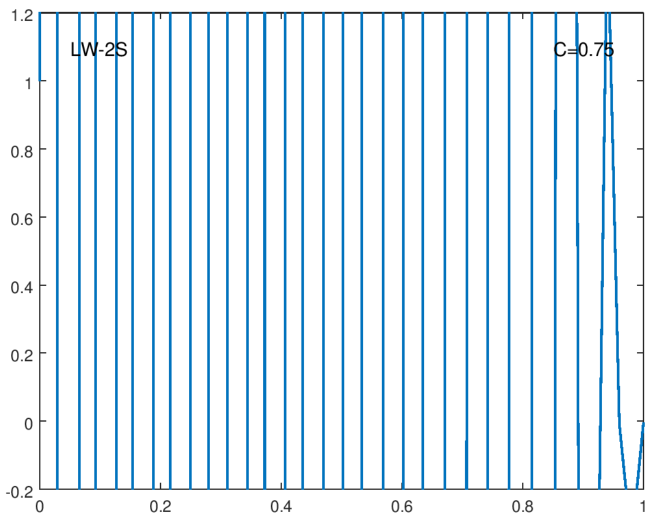
**d). Solving the problem using the second-order Lax-Wendroff method.**




*Second-Order Lax-Wendroff*

**Comment:** As is observed from the above graph, the method is unstable for C=0.75. The Lax-Wendroff scheme is unstable for $C^2 > 1/3$. It will generate more stable results for $C^2<1/3$.

**e). Implemention of the second-order two-step Lax-Wendroff method.**




*Second-Order Two Step Lax-Wendroff*

CODE

```
% Allocate storage
A1 = zeros(numnp,numnp);
B1 = zeros(numnp,numnp);
f1 = zeros(numnp,1);
A2 = zeros(numnp,numnp);
B2 = zeros(numnp,numnp);
f2 = zeros(numnp,1);
C2 = zeros(numnp,numnp);


% MATRICES COMPUTATION
% Loop on elements
for i=1:numel
    unos = ones (ngaus,1);
    h = xnode(i+1)-xnode(i);
    xm = (xnode(i)+xnode(i+1))/2;
    weight = wpg*h/2;
    isp = [i i+1];
    % Loop on Gauss points (numerical quadrature)
    for ig = 1:ngaus
        N = N_mef(ig,:);
        Nx = Nxi_mef(ig,:)*2/h;
        w_ig = weight(ig);
        x = xm + h/2*xipg(ig); % x-coordinate of the current Gauss point
        % Matrices assembly
        A1(isp,isp) = A1(isp,isp) + w_ig*(N'*N);
        B1(isp,isp) = B1(isp,isp) - w_ig*((dt/2*N'*(a*Nx)));
        f1(isp) = f1(isp) + w_ig*(N')*SourceTerm(x);
        A2(isp,isp) = A2(isp,isp) + w_ig*(N'*N);
        B2(isp,isp) = B2(isp,isp);
        f2(isp) = f2(isp) + w_ig*(N')*SourceTerm(x);
        C2(isp,isp) = C2(isp,isp) - w_ig*(dt*N'*(a*Nx));

    end
end
```

**Comment:** The scheme is unstable for C=0.75. The solutions may be improved using the Discontinuous Galerkin in space and the second-order two-step Lax-Wendroff method in time.