

### T.P. 3 : Equations de transport *Correction*

```
function [res] = divers_schema(xin)

global choixCI;
global L;

a      = 0.1; %% vitesse d'advection
Tmax   = 10; %% temps maximum de simulation
L      = 5;   %% longueur du domaine

N      = 201; %% nb de points de discretisation en espace
dx     = L/(N-1); %% pas d'espace
nu     = 0.9; %% nombre de Courant-Friedrichs-Levy
dt     = nu*dx/a; %% pas de temps

choixCI=1
schemas={'FTCS' , 'Lax' , 'upwind' , 'Lax - Wendroff'};
choixSCH=3

%% initialisation du maillage et de la donnee initiale :
x = linspace(0,L,N)';
u = condinit(x);

%% boucle en temps : a chaque temps discret, on effectue le calcul
%% de facon vectorielle en tenant compte de la condition de periodicite
nframes=Tmax/dt;
F = moviein(nframes);

frame=1;

for t =dt:dt:Tmax

    uold=u;

    switch choixSCH
    case 1 %% FTCS
        u(2:(N-1)) = uold(2:N-1) - nu/2*(uold(3:N)-uold(1:N-2));
        u(N) = uold(N) - nu/2*(uold(2)-uold(N-1));
        u(1)=u(N);
    case 2 %% Lax
        u(2:N-1) = (1 - nu)/2*uold(3:N) + (1+nu)/2*uold(1:N-2);
        u(N) = (1 - nu)/2*uold(2)+ (1+nu)/2*uold(N-1);
        u(1)=u(N);
    end
end
```

```

case 3 %% upwind
    u(2:N) = uold(2:N) - nu*(uold(2:N)-uold(1:N-1));
    u(1) = u(N);
case 4 %% Lax - Wendroff
    u(2:N-1) = uold(2:N-1) - nu/2*(uold(3:N)-uold(1:N-2)) + ...
        nu^2/2*(uold(3:N)- 2*uold(2:N-1) + uold(1:N-2));
    u(N) = uold(N) - nu/2*(uold(2)-uold(N-1)) + ...
        nu^2/2 * (uold(2)- 2*uold(N) + uold(N-1));
    u(1)=u(N);
end

xnew = x-a*t;
xnew_per = xnew - L*floor(xnew / L);
uexact = condinit(xnew_per);

plot(x,u,x,uexact);

legend('sol numerique','sol exacte');
title(['Schema de type ', schemas{choixSCH}]);

F(frame) = getframe;frame=frame+1;

pause(0.01);

end

movie(F,2);
movie2avi(F,[schemas{choixSCH}, '.avi']);
end

function [v] = condinit(x)
    global choixCI;
    global L;
    switch choixCI
        case 1
            v = 0 + ((x>1.) & (x<1.5));
        case 2
            v = sin(8*pi*x/L);
    end
end
end

```