

```

// Script 3: Animation graphique
clear, global("condI","scheme","L");
// Parametres physiques
a = 0.1;    // vitesse d'advection
Tmax=20;    // temps maximum
L = 5 ;     // longueur du domaine
condI=1;    // flag pour le choix de la condition initiale
            // =1 le creneau, =2 la sinusoide
scheme=1;   // flag pour le choix du schema
function [v] = condinit(x)    // condition initiale
    select condI
    case 1
        v=bool2s((x>1.)&(x<1.5));
    case 2
        v=sin(8*%pi*x/L);
    end
endfunction
// Parametres numeriques
N = 201;    // Nd points discretisation espace
dx = L/(N-1); // pas d'espace
s = 0.8;    // nombre de Courant
dt = s*dx/a; // pas de temps
// Initialisation maillage et donnee initiale
x=linspace(0,L,N); // partition equidistante avec N points
u=condinit(x);     // solution initiale
// On demarre la boucle en temps
tps = dt:dt:Tmax;
for j=1:length(tps)
    uold = u; // on stocke la solution precedente
    uexact=condinit(x-a*tps(j)); // solution exacte au temps t
    select scheme
    case 1
        u(2:N)=uold(2:N)-s*(uold(2:N)-uold(1:N-1));
    case 2
        u(N)=uexact(N);
        u(2:N-1)=uold(2:N-1)-s/2*(uold(3:N)-uold(1:N-2))+...
            s^2/2*(uold(3:N)-2*uold(2:N-1)+uold(1:N-2));
    end
    u(1)=u(N); // on utilise la periodicite
    if floor(j/5)*5 ==j then // on trace la solution
        clf(), plot(x,u,'rx-',x,uexact,'bo-');
        legend('Solution approchee', 'Solution exacte');
        xtitle('Comparaison entre la solution exacte et
approchee','x','Solution');
        halt('Press a key')
    end
end
end

```

```

// Script 4: Lax Wendroff implicite
clear, global("condI","scheme","L");
// Parametres physiques
a = 0.1; // vitesse d'advection
Tmax=20; // temps maximum
L = 5 ; // longueur du domaine
condI=1; // flag pour le choix de la condition initiale
function [v] = condinit(x) // condition initiale
    select condI
    case 1
        v=bool2s((x>1.)&(x<1.5));
    case 2
        v=sin(8*pi*x/L);
    end
endfunction
// Parametres numeriques
N = 201; // Nd points discretisation espace
dx = L/(N-1); // pas d'espace
s = 0.8; // nombre de Courant
dt = s*dx/a; // pas de temps
// Initialisation maillage et donnee initiale
x=linspace(0,L,N); // partition equidisante avec N points
u=condinit(x); // solution initiale
// Construction de la matrice pour le schema implicite
A=diag((1+s^2)*ones(N,1))-diag(s^2/2*ones(N-1,1),-1)-
diag(s^2/2*ones(N-1,1),1);
A(1,N)=-s^2/2;A(N,1)=-s^2/2;
// On demarre la boucle en temps
tps = dt:dt:Tmax;
for j=1:length(tps)
    uold =matrix(u,length(u),1); // on stocke la solution precedente
    uexact=condinit(x-a*tps(j)); // solution exacte au temps t
    Wn(N)=uexact(N);
    Wn(2:N-1)=uold(2:N-1)-s/2*(uold(3:N)-uold(1:N-2));
    Wn(1)=Wn(N);
    u=A\Wn; // calcul de la solution
    if floor(j/5)*5 ==j then // on trace la solution
        clf(), plot(x,u,'rx-',x,uexact,'bo-');
        legend('Solution approchee', 'Solution exacte');
        xtitle('Comparaison entre la solution exacte et
approchee','x','Solution');
        halt('Press a key')
    end
end
end

```