MNE-TP2 : Schémas diffusifs et dispersifs

ZHONG Ming

28 février 2019

Table des matières

1 Rappel du problème

Dans ce TP, on étudie des schémas numériques pour l'équation de transport :

$$(ET)\begin{cases} \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 & \text{pour } (x,y) \in [0,L] \times R^+ \\ u(x,0) = u^0(x) \end{cases}$$

Le but est de mettre en évidence des propriétés (généralement non désignées) de diffusion et de dispersion numériques.

2 Méthode numérique

2.1 Les schémas

On considère les 3 schémas suivants :

- **décentré** :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} \;+\; c\frac{u_j^n - u_{j-1}^n}{\Delta x} \;=\; 0$$

- **Lax-Friedrichs** :

$$\frac{1}{\Delta t}\left\{u_j^{n+1} - \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n\right)\right\} \;+\; c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \;=\; 0$$

- **Lax-Wendroff** :

$$\frac{u_j^{n+1} + u_j^n}{\Delta t} \;+\; c\frac{u_{j+1}^n + u_{j-1}^n}{2\Delta x} \;-\; \frac{c^2}{2}\Delta t\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \;=\; 0$$

2.2 Discrétisation

- **spatiale** : $x_j = (j-1)\,\Delta x,\ j = 1, ..., Nx$
- **temporelle** : $t^n = (n-1)\,\Delta t,\ n = 1, ..., Nt$
- **de la solution** : $u_j^n \;=\; u\left(x_j, t^n\right)$
- **de l'équation** :
  - décentré : $u_j^{n+1} = \left(1 - c\frac{\Delta t}{\Delta x}\right) u_j^n + c\frac{\Delta t}{\Delta x} u_{j-1}^n$
  - Lax-Friedrichs : $u_j^{n+1} = \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n\right) - \frac{c\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right)$
  - Lax-Wendroff : $u_j^{n+1} = u_j^n - \frac{c^2\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{c^2\Delta t^2}{2\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$

2.3 Condition de stabilité

Ces trois schémas sont conditionnellement stables sous la condition CFL :

$$c\frac{\Delta t}{\Delta x} \;<\; 1$$

Afin d'assurer qu'ils sont toujours stables, on donne tout d'abord λ ($\lambda = c\frac{\Delta t}{\Delta x}$) une valeur constante. Quand le pas d'espace Δx est fixé, on a le pas de temps $\Delta t = \frac{\lambda \Delta x}{c}$.

3 Algorithme

·étap 1 :

- définition la condition initiale : function y=Uinit(x)

- définition la fonction exacte : function y=Uexact(x,t) $(u\left(x,t\right)=u_0\left(x-ct\right))$
- initiation des paramètres

·étap 2 :
for n = 1 :Nt

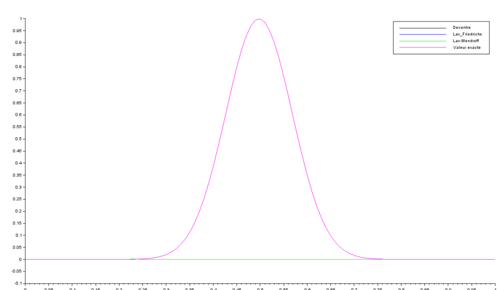
- initialiser les valeurs aux bords = Uexact(0,t(n)) et Uexact(1,t(n))
- for j=2 :Nx
  - décentré : $u_j^{n+1}=\left(1-c\frac{\Delta t}{\Delta x}\right)u_j^n+c\frac{\Delta t}{\Delta x}u_{j-1}^n$
  - Lax-Friedrichs : $u_j^{n+1}=\frac{1}{2}\left(u_{j-1}^n+u_{j+1}^n\right)-\frac{c\Delta t}{2\Delta x}\left(u_{j+1}^n-u_{j-1}^n\right)$
  - Lax-Wendroff : $u_j^{n+1}=u_j^n-\frac{c^2\Delta t}{2\Delta x}\left(u_{j+1}^n-u_{j-1}^n\right)+\frac{c^2\Delta t^2}{2\Delta x^2}\left(u_{j+1}^n-2u_j^n+u_{j-1}^n\right)$
- $u_j^n\leftarrow u_j^{n+1}$

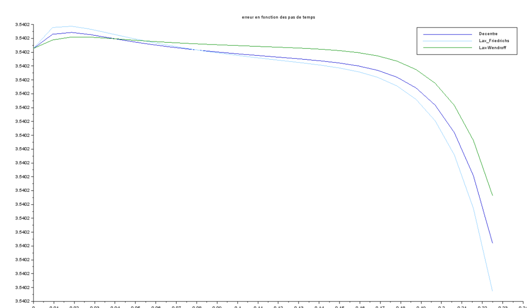
·étap 3 :
Calculer l'erreur en fonction des pas de temps :
for j=1 :Nt
$err(j)=norm(Vexacte(:,j)-V(:,j),2)$

4 Résultats

4.1 1ème cas-test

- la condition initiale : u^0(x)=exp(-$\frac{(x-x_0)^2}{\sigma^2}$)
- la condition CFL : $\lambda=0.9$
- nombre de points d'espace : Nx = 100
- nombre de points de temps : Nt = 25

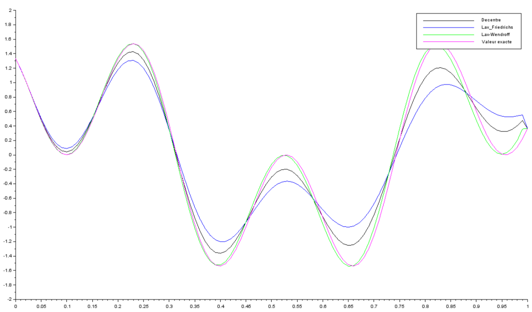


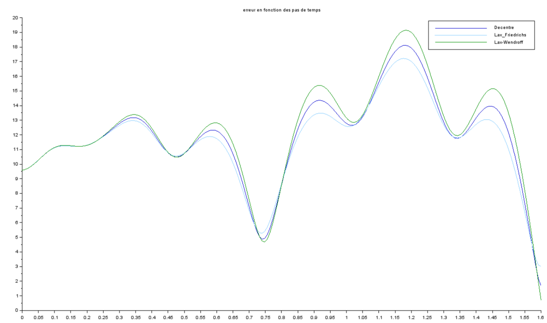


Nous avons donc à gauche les courbes calculées et exactes, et à droite l'erreur en fonction des pas de temps.

4.2 2ème cas-test

- la condition initiale : u^0(x)=sin($\frac{m\pi x}{L}$)+sin($\frac{m\pi x}{3L}$)
- la condition CFL : $\lambda = 0.8$
- nombre de points d'espace : Nx = 100
- nombre de points de temps : Nt = 200



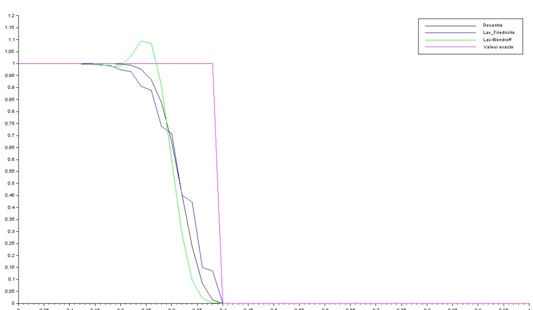


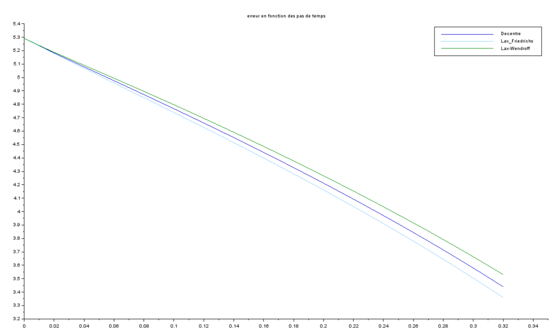
A gauche les courbes calculées et exactes, et à droite l'erreur en fonction des pas de temps.

On remarque que la solution qui se rapproche au plus de la solution exacte est la la solution correspondant au schéma de Lax-Wendroff, qui est un schéma dispersifs. La solution la moins précise est la courbe du schéma de Lax-Friedreichs, qui est un schéma diffusif.

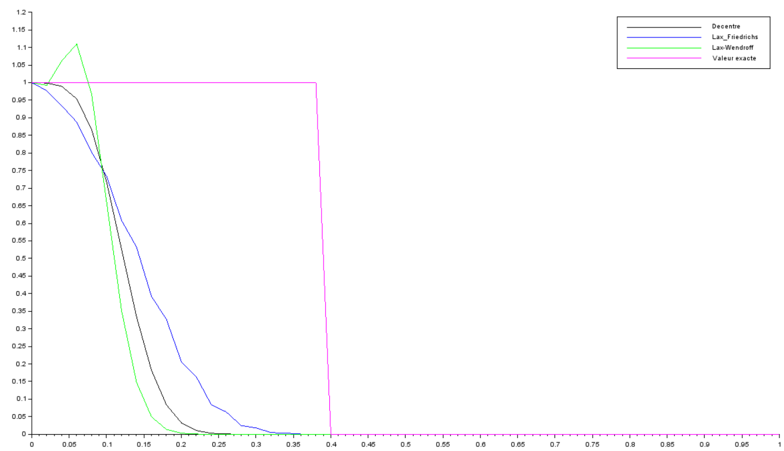
4.3 3ème cas-test

- la condition initiale : u^0(x)= 1 si $x < x_0$ et 0 sinon
- la condition CFL : $\lambda = 0.8$
- nombre de points d'espace : Nx = 50
- nombre de points de temps : Nt = 20





A gauche les courbes calculées et exactes, et à droite l'erreur en fonction des pas de temps.



la condition CFL : $\lambda = 0.3$

On constate que plus le coefficient lambda est proche de 1, plus la solution approximative sera proche de la solution exacte.

5 Conclusion

- plus le coefficient est proche de 1, mieux on aura une bonne approximation qui est proche de la solution exacte.
- grâce à l'étude de la comparaison des trois schémas : le schéma Décentré, celui de Lax-Friedrichs et celui de Lax-Wendroff nous avons remarqué que les schémas Decentred et Lax-Wendroff étaient des schémas dispersifs et que le schéma Lax-Friedrich était un schéma diffusif. Nous avons également remarqué que c'est la solution approximative du schéma de Lax-Wendroff la plus proche de la solution exacte.

6 Code Scilab

```
// author : ZHONG Ming
// equation de transport: schema Euler Decentre, Lax_Friedrichs, Lax-Wendroff
clear()

sigma = 0.1
x0 = 0.25
m = 7
c = 1

function y=Uinit(x)
//     y = exp(-(x-x0)**2/sigma**2)
```

```
//     y = sin((m*%pi*x)) + sin((m*%pi*x)/3)

            if x < x0 then
                y = 1;
            else
                y = 0;
            end
endfunction

function y=Uexact(x,t)
    y = Uinit(x-c*t)
endfunction

function y=a(t)
    y= Uexact(0,t)
endfunction

function y=b(t)
    y= Uexact(1,t)
endfunction


//la condition CFL : lambda = c * dt / dx
lambda = 0.3

//la discr tisation du probl me en espace :
//Espace total :
L = 1
//Nombre de points d'espace :
Nx = 50
//Pas d ' espace :
dx = L / Nx


//la discr tisation du probl me en temps :
//Temps initial :
T0 = 0
//Nombre de points de temps :
Nt = 20
//Pas de temps :
dt = lambda * dx / c
//Temps final :
Tfin = T0 + dt * Nt

//valeur
V_dec=zeros(Nx+1,Nt)
V_laxf=zeros(Nx+1,Nt)
V_laxw=zeros(Nx+1,Nt)
x = linspace(0,L,Nx+1)
t = linspace(T0,Tfin,Nt)
```

```
//quand t = n, x = 0,1,2,...,Nx
U_dec_n=zeros(Nx+1,1)
U_laxf_n=zeros(Nx+1,1)
U_laxw_n=zeros(Nx+1,1)
//quand t = n+1, x = 0,1,2,...,Nx
U_dec_np1=zeros(Nx+1,1)
U_laxf_n=zeros(Nx+1,1)
U_laxw_n=zeros(Nx+1,1)

for n = 1:Nt

    U_dec_np1(1)=a(t(n))
    U_dec_np1(Nx+1)=b(t(n))
    U_laxf_np1(1)=a(t(n))
    U_laxf_np1(Nx+1)=b(t(n))
    U_laxw_np1(1)=a(t(n))
    U_laxw_np1(Nx+1)=b(t(n))

    for j=2:Nx
        U_dec_np1(j) = c*dt/dx*(U_dec_n(j-1)-U_dec_np1(j)) + U_dec_n(j)
        U_laxf_np1(j) = 0.5*(U_laxf_n(j-1)+U_laxf_n(j+1)) - c*dt/2/dx*(
            U_laxf_n(j+1)-U_laxf_n(j-1))
        U_laxw_np1(j) = U_laxw_n(j) - c*dt/2/dx*(U_laxw_n(j+1)-U_laxw_n(j-1))
            + c^2*dt^2/2/dx^2*(U_laxw_n(j+1)-2*U_laxw_n(j)+U_laxw_n(j-1))
    end

//    if norm(U_dec_np1-U_dec_n) < 0.005
//        then break
//    end

    U_dec_n = U_dec_np1
    U_laxf_n = U_laxf_np1
    U_laxw_n = U_laxw_np1
    V_dec(:,n) = U_dec_n
    V_laxf(:,n) = U_laxf_n
    V_laxw(:,n) = U_laxw_n
end

//valeur exacte
VE=zeros(Nx+1,Nt)

for i=1:Nt
    for j=1:Nx+1
        VE(j,i)=Uexact(x(j),t(n))
    end
end

y = linspace(0,L,Nx+1)
for i=1:Nx+1
    y(i) = Uexact(x(i),0.15)
```

```
end
scf(1)
plot2d(x,U_dec_np1,style=1)
plot2d(x,U_laxf_np1,style=2)
plot2d(x,U_laxw_np1,style=3)
plot2d(x,y,style=6)
legend("Decentre","Lax_Friedrichs","Lax-Wendroff","Valeur exacte")

//l'erreur en fonction des pas de temps
for j=1:Nt
    err_dec_t(j)=norm(VE(:,j)-V_dec(:,j),2)
    err_laxf_t(j)=norm(VE(:,j)-V_laxf(:,j),2)
    err_laxw_t(j)=norm(VE(:,j)-V_laxw(:,j),2)
end
scf(3)
xtitle("erreur en fonction des pas de temps")
plot2d(t,err_dec_t,style=11)
plot2d(t,err_laxf_t,style=12)
plot2d(t,err_laxw_t,style=13)
legend("Decentre","Lax_Friedrichs","Lax-Wendroff")
```