UNIVERSITÉ DE POITIERS **2008-2009**
Agrégation de Mathématiques

Corrigé TP advection

1. L'équation de transport (ou d'advection, ou de convection) $u_t + au_x = 0$ est le prototype des équations hyperboliques. En général, a a la dimension d'une vitesse.
2. Il est facile de vérifier que $u(t,x) = u_0(x-at)$ est solution, si $u_0 \in C^1(\mathbb{R})$, u_0 L-périodique. Cela donne l'existence. On va montrer l'unicité pour le problème plus simple suivant :

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + a\dfrac{\partial u}{\partial x}(t,x) = 0 & t \geq 0,\ x \in \mathbb{R} \\ u(t,\cdot) \text{ est L-périodique,} & \forall t \geq 0 \\ u(0,x) = u_0(x) & x \in \mathbb{R}. \end{cases} \tag{1}$$

(il est clair que toute solution $C^1(\mathbb{R}_+ \times \mathbb{R})$ de (1) est solution de notre problème - l'implication inverse est plus délicate, d'où l'intérêt de considérer (1)). Soit donc $u \in C^1([0,+\infty) \times \mathbb{R})$ une solution de (1). Posons, pour $y \in \mathbb{R}$ fixé, $v(t) = u(t, y+at)$. Alors $v \in C^1([0,+\infty))$ et $v'(t) = u_t(t,y+at) + au_x(t,y+at) = 0$, donc $v(t) = v(0) = u(0,x) = u_0(x)$, i.e. $u(t,y+at) = u_0(y)$. Ceci est vrai pour tout $t \geq 0$ et tout $y \in \mathbb{R}$, donc en posant $x = y + at$, on obtient que $u(t,x) = u_0(x-at)$. Réciproquement, on voit que $u(t,x) = u_0(x-at)$ est solution de (1).

Remarque : même si u_0 n'est pas de classe C^1, la fonction $u(t,x) = u_0(x-at)$ donne un candidat pour la solution de l'équation de transport, où les dérivées sont considérées dans un sens plus faible (dérivée au sens des distributions). C'est le sens des exemples numériques proposés ci-dessous.
3. Les 5 schémas proposés sont explicites. Ils sont également tous à un pas, sauf le Leapfrog (saute-mouton) qui est à deux pas (il nécessite un schéma auxilaire pour définir u_j^1). Les schémas explicites sont très faciles à programmer. Les schémas proposés, à l'exception de Leapfrog, s'écrivent sous la forme $v_j^{n+1} = Av_{j-1}^n + Bv_j^n + Cv_{j+1}^n$.

On rassemble ci-dessous les particularités de ces schémas. On a noté $\lambda = k/h$.

Schéma	FTFS	FTBS	FTCS	Lax-Friedrichs
A	0	$a\lambda$	$a\lambda/2$	$(1+a\lambda)/2$
B	$1+a\lambda$	$1-a\lambda$	1	0
C	$-a\lambda$	0	$-a\lambda/2$	$(1-a\lambda)/2$
ordre	$O(k)+O(h)$	$O(k)+O(h)$	$O(k)+O(h^2)$	$O(k)+O(h^2/k)+O(h^2)$
stable	$-1 \leq a\lambda \leq 0$	$0 \leq a\lambda \leq 1$	instable	$\lvert a\rvert\lambda \leq 1$
dissipatif	$-1 < a\lambda < 0$	$0 < a\lambda < 1$	/	$\lvert a\rvert\lambda < 1$

Remarquer que pour $a\lambda = 1$, les deux schémas stables FTCS et Lax-Friedrichs sont égaux : ils correspondent à un décalage $v_j^{n+1} = v_{j-1}^n$

Dans les figures ci-dessus, on a utilisé la fonction chapeau $u_0(x) = \begin{cases} 1-|x| & |x| \leq 1, \\ 0 & \text{sinon} \end{cases}$, avec assez de marge sur les bords pour ne pas avoir de problème avec les conditions aux limites. On a utilisé $\lambda = 0.7$, $h = 0.1$, $k = \lambda h = 0.07$ et la solution est représentée à $t_n = 1.05 = 15k$. La solution exacte est également représentée.

On a ensuite utilisé un créneau.

Voici le programme Scilab ayant permis de faire ces dessins : **Programme Scilab**

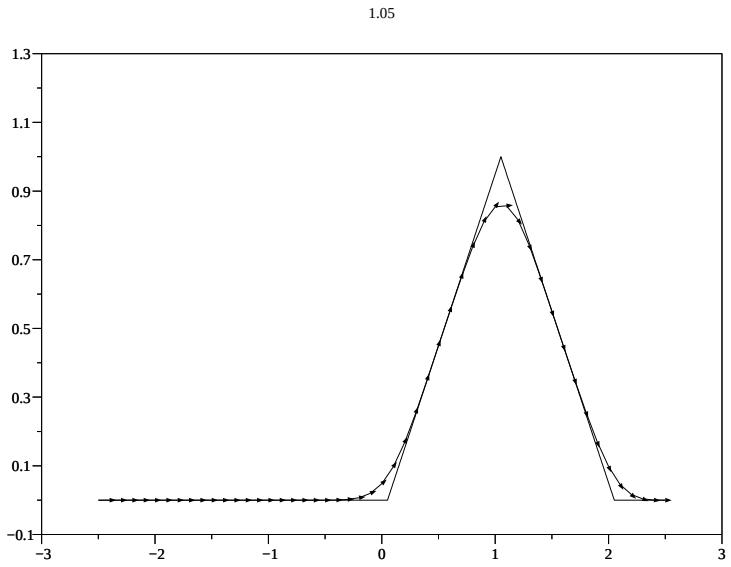

Figure 1: FTBS

```
//Differences finies pour l'equation de transport u_t+u_x=0
//schemas explicites a 1 pas du type v_j^{n+1}=Av_{j-1}^n+Bv_j^n+Cv_{j+1}^n


clear
 listu0=['creneau','casu0=1';
  'chapeau','casu0=2';
  'reguliere','casu0=3'];


listschemas=['FTFS','schema=1';
   'FTBS','schema=2';
   'FTCS','schema=3';
   'Lax-Friedrichs','schema=4';
   'Lax-Wendroff','schema=5'];


 listlambda=['1','lambda=1';
  '1.2','lambda=1.2';
  '0.7','lambda=0.7';
  '0.1','lambda=0.1'];

while %t
 num=x_choose(listu0(:,1),'Choisir un u0')
   if num==0 then break
  else
execstr(listu0(num,2));
 select casu0
    case 1
```

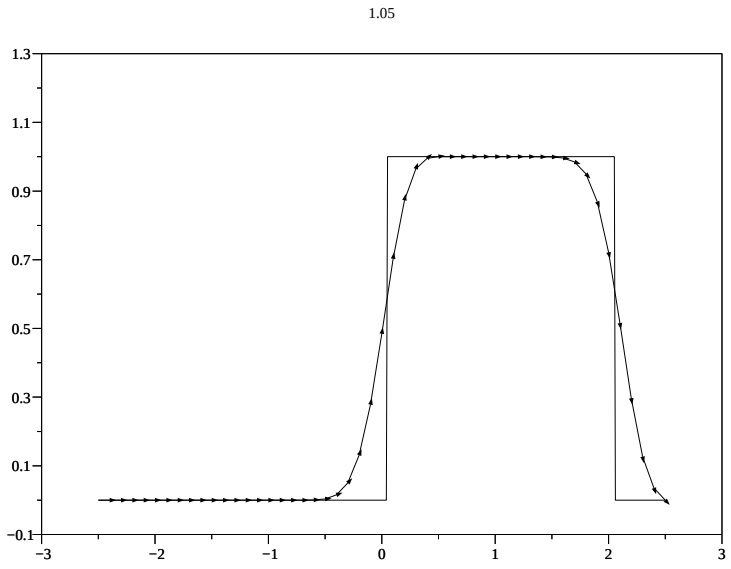

Figure 2: FTBS

```
        deff('y=u0(x)','y=bool2s((abs(x)<=1))');
    case 2
 deff('y=u0(x)','y=bool2s((abs(x)<=1)).*(ones(size(x,1),size(x,2))-abs(x))');
    case 3
deff('y=u0(x)','y=0.5*bool2s((abs(x)<=1)).*(ones(size(x,1),size(x,2))+cos(%pi*x))');
    end
   end


  num=x_choose(listlambda(:,1),'Choisir un lambda')
  execstr(listlambda(num,2));

  M=10;  h=1/M;//pour le pas d'espace

  k=lambda*h;
  N=15;

  num=x_choose(listschemas(:,1),'Choisir un schema')
  execstr(listschemas(num,2));
  select schema
  case 1 //FTFS
   A=0;B=1+lambda;C=-lambda;
  case 2 //FTBS
       A=lambda;B=1-lambda;C=0;
  case 3  //FTCS
     A=lambda/2;B=1;C=-lambda/2;
  case 4  //Lax-Friedrichs
     A=(1+lambda)/2;B=0;C=(1-lambda)/2;
  case 5   //Lax-Wendroff
```

1.05

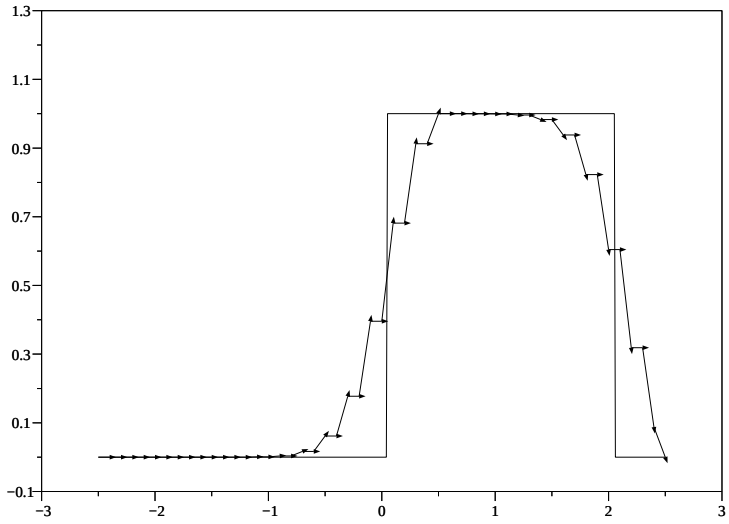

Figure 3: Lax-Friedrichs

```
  A=lambda/2+lambda^2/2;B=1-lambda^2;C=-lambda/2+lambda^2/2;

end


xmin=-1-h*N;
xmax=1+h*N;
x=[xmin:h:xmax];
xbis=[xmin:0.01:xmax];
ymin=-0.1;
ymax=1.2;

y=u0(xbis);
v=u0(x);LV=length(v);
xbasc();
plot2d(xbis',y',rect=[xmin,ymin,xmax,ymax]);
plot2d4(x',v',rect=[xmin,ymin,xmax,ymax]);
halt();
for n=1:N
t=n*k;
xprime=xbis-t*ones(size(xbis,1),size(xbis,2));
y=u0(xprime);
v(2:LV-1)=A*v(1:LV-2)+B*v(2:LV-1)+C*v(3:LV);
xbasc();
plot2d(xbis',y',rect=[xmin,ymin,xmax,ymax]);
plot2d4(x',v',rect=[xmin,ymin,xmax,ymax]);
xtitle(string(t))
halt();
end;
```

```
  end
```

Un exemple de programmation **MAPLE** :

```
> ?Matrix
>
> u0:=x->sin(2*Pi*x);

                        u0 := x -> sin(2 Pi x)

> ?animate
> with(plots):
>
Warning, the name changecoords has been redefined

> animate( plot, [u0(x-t), x=0..1], t=0..5, frames=50 );

> transport:=proc(L,T,a,M,N)
> # construction de la matrice v
> local j,n,v,h,k,LL;
> global u0;
> h:=L/M;
> k:=T/N;
> v:=Matrix(M+1,N+1);
> for j from 1 to M+1 do v[j,1]:=evalf(u0((j-1)*h)); od;
> for n from 1 to N do
>     for j from 2 to M+1 do
>         v[j,n+1]:=evalf( v[j,n]-a*(k/h)*(v[j,n]-v[j-1,n]) );
>         v[1,n+1]:=evalf( v[M+1,n+1] );
>     od;
> od;
> # affichage aux differents temps
> LL:=Matrix(M+1,N+1);
> for n from 1 to N+1 do
>     for j from 1 to M+1 do
>         LL[j,n]:=[evalf((j-1)*h),v[j,n]];
>     od;
> od;
> return(LL);
> end proc:
> L:=1;T:=2;M:=10;N:=20;LL:=transport(L,T,1,M,N):evalm(LL):

                                L := 1


                                T := 2


                               M := 10
```

```
                              N := 20
```

```
> Nbis:=10;for n from 8 to Nbis do
> LLL:=seq(LL[j,n],j=1..M+1):plot({[LLL],u0(x-(n-1)*T/N)},x=0..1); od;
```

```
                             Nbis := 10
```

```
  LLL := [0., 0.9510565165], [0.1000000000, 0.5877852524],

         [0.2000000000, 0.], [0.3000000000, -0.5877852524],

         [0.4000000000, -0.9510565165], [0.5000000000, -0.9510565165],

         [0.6000000000, -0.5877852524], [0.7000000000, 0.],

         [0.8000000000, 0.5877852524], [0.9000000000, 0.9510565165],

         [1., 0.9510565165]


  LLL := [0., 0.9510565165], [0.1000000000, 0.9510565165],

         [0.2000000000, 0.5877852524], [0.3000000000, 0.],

         [0.4000000000, -0.5877852524], [0.5000000000, -0.9510565165],

         [0.6000000000, -0.9510565165], [0.7000000000, -0.5877852524],

         [0.8000000000, 0.], [0.9000000000, 0.5877852524],

         [1., 0.9510565165]


  LLL := [0., 0.5877852524], [0.1000000000, 0.9510565165],

         [0.2000000000, 0.9510565165], [0.3000000000, 0.5877852524],

         [0.4000000000, 0.], [0.5000000000, -0.5877852524],

         [0.6000000000, -0.9510565165], [0.7000000000, -0.9510565165],

         [0.8000000000, -0.5877852524], [0.9000000000, 0.],

         [1., 0.5877852524]
```

>
>

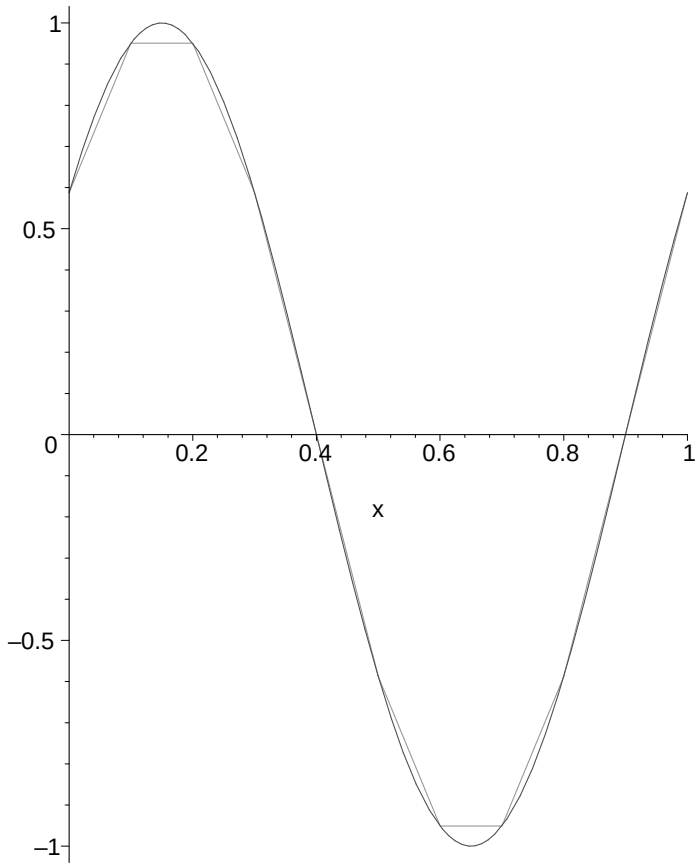

Figure 4: La dernière figure affichée par “transport.mws”

Analyse de stabilité : on fait l'exemple de FTBS. En discrétisant d'abord en espace décentré aval, l'edp devient

$$\frac{du_j(t)}{dt} + a\frac{u_j(t) - u_{j-1}(t)}{dt} = 0, \quad 1 \leq i \leq M,$$

i.e. $U'(t) + AU(t) = 0$, avec

$$U(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_M(t) \end{pmatrix}, \quad A = \frac{a}{h}\begin{pmatrix} 1 & 0 & \cdots & -1 \\ -1 & 1 & 0 & \cdots \\ 0 & \ddots & \ddots & \\ \cdots & & -1 & 1 \end{pmatrix}.$$

Remarquer que u_0 n'est pas une inconnue, car $u_0 = u_M$ avec les conditions périodiques. On peut calculer explicitement les valeurs propres de A (suite récurrente). On trouve, en posant $A = (a/h)\tilde{A}$, que $(v_1, \ldots, v_M)$ est un vecteur propre de $\tilde{A}$ pour la v.p. λ ssi $v_i = (1-\lambda)^{i-1}v_1$, avec $v_{M+1} = v_1$, i.e. (comme $v_1 \neq 0$), $(1-\lambda)^{-N} = 1$. Les v.p. sont

$$1 - \lambda_j = e^{2ij\pi/M} \iff \lambda_j = 1 - \cos(\frac{2j\pi}{M}) - i\sin(\frac{2j\pi}{M}), \quad 0 \leq j \leq M-1.$$

En particulier, $Re(-\lambda_j) \leq 0$, donc bien conditionné.

Si on discrétise $U'(t) = AU(t)$ par Euler explicite, alors par diagonalisation on est ramené à étudier la condition de stabilité pour

$$z'(t) = -\beta_j z(t),$$

avec

$$\beta_j = \frac{a}{h}\lambda_j, \text{ et donc } Re(\beta_j) \geq 0.$$

Euler explicite s'écrit

$$z_{n+1} = z_n - k\beta_j z_n, \quad n \geq 0.$$

La condition de stabilité est : $|1 - k\beta_j| \leq 1$, pour tout j. En calculant :

$$\rho_j^2 = |1 - k\beta_j|^2 = [1 - \frac{ka}{h} + \frac{ka}{h}\cos(\frac{2j\pi}{M})]^2 + [\frac{ka}{h}\sin(\frac{2j\pi}{M})]^2.$$

En développant puis simplifiant,

$$\begin{aligned} \rho_j^2 &= (1 - \frac{ka}{h})^2 + 2(1 - \frac{ka}{h})\frac{ka}{h}\cos(\frac{2j\pi}{M}) + (\frac{ka}{h})^2, \\ &= [(1 - \frac{ka}{h}) + \frac{ka}{h}]^2 + 2(1 - \frac{ka}{h})\frac{ka}{h}(\cos(\frac{2j\pi}{M}) - 1). \end{aligned}$$

On a toujours $\cos(\frac{2j\pi}{M}) - 1 \leq 0$, donc la condition de stabilité est $1 - \frac{ka}{h} \geq 0$, d'où le résultat annoncé dans le tableau.