**M1 IM/MPA - EDP et Différences Finies.**

**Equation de transport**

On s'intéresse à l'équation de transport à vitesse $c$ en dimension 1 :

$$\begin{cases} \partial_t u(t,x) + c(t,x)\partial_x u(t,x) = 0, & t, x \in \mathbb{R}, \\ u(0,x) = u_0(x), & x \in \mathbb{R}, \end{cases} \tag{1}$$

où $u_0 \in \mathcal{C}^1(\mathbb{R})$ et $c$ est supposée globalement lipschitzienne en $x$.

**Rappels sur les caractéristiques :** Pour tout $x \in \mathbb{R}$, on définit $X_x$ la solution (sous conditions d'existence et unicité) du problème de Cauchy

$$\begin{cases} \dfrac{\mathrm{d}X_x}{\mathrm{d}t}(t) = c(t, X_x(t)) & t \in \mathbb{R}, \\ X_x(0) = x. \end{cases} \tag{2}$$

Alors quelque soit $x \in \mathbb{R}$, $u$ est constante le long de la *caractéristique* $t \mapsto (t, X_x(t))$, c'est à dire $u(t, X_x(t)) = u(0, X_x(0))$.

On en déduit que pour tout $t, x \in \mathbb{R}$, $u(t,x) = u_0(X_y(0))$ avec $X_y(t) = x$.

En particulier, si $c(t,x) = c \in \mathbb{R}$, alors $u(t,x) = u_0(x - ct)$.

# I Résolution numérique - conditions de Dirichlet

## I.1 Notations

On souhaite déterminer une solution approchée par la méthode des différences finies de l'équation (1) à vitesse $c(t,x) = c$ constante.

En pratique, nous sommes contraints à ne décrire cette solution que sur un domaine d'espace-temps borné, on considérera $t \in [0, t_{\max}]$, $x \in [0, L]$.

On note $T$ et $X$ les discrétisations de $[0, t_{\max}]$, et $[0, L]$ en respectivement $N+1$ et $M+1$ points, de pas $\Delta_t$ et $\Delta_x$. On note $U \in M_{M+1,N+1}$ la matrice des $u_i^n$ où $u_i^n$ désigne l'approximation de $u(n\Delta_t, i\Delta_x)$, $i = 0, \dots, M$ ; $n = 0, \dots N$.

On utilisera le schéma explicite en temps et décentré amont (resp. aval) en espace si $c > 0$ (resp.$c < 0$), aussi appelé schéma *Upwind* :

$$\boxed{\begin{array}{c}\textit{Trouver } (u_i^n)_{i,n} \textit{solution de} \\ u_i^{n+1} = u_i^n - r(u_i^n - u_{i-1}^n), \text{ si } c > 0 \\ u_i^{n+1} = u_i^n - r(u_{i+1}^n - u_i^n), \text{ si } c < 0 \\ \text{(+cond. initiale et de bord), où } r := c\dfrac{\Delta t}{\Delta_x}\end{array}}$$

Ce schéma est stable sous la condition (CFL) :

$$|r| = |c|\Delta_t/\Delta_x \leq 1$$

Notons que nous n'avons besoin de définir de condition de bord que sur le bord *entrant* : le bord gauche si $c > 0$, droit si $c < 0$.

## I.2 Résolution sur l'intervalle $[0, L]$

Considérons la condition initiale

$$u_0(x) = \begin{cases} 0 \text{ si } x < 2, \\ (x-2)^6 \text{ si } 2 \leq x \leq 3, \\ 2-(x-4)^6 \text{ si } 3 \leq x \leq 4, \\ 2 \text{ si } x > 4. \end{cases}$$

**1)** Ecrire la fonction python `u0sigmoid` correspondante. [1]

**2)** Écrire une fonction python *SchemaUpwind* qui prend en paramètre $c$, $t_{\max}$, $L$, $u0$, $M$ et $N$. Cette fonction teste le signe de $c$ puis résout le schéma *Upwind* appliqué à l'équation de transport à vitesse $c$ sur $[0, t_{\max}] \times [0, L]$ à l'aide du schéma. La fonction renverra $T$, $X$, et la matrice $U$.

On imposera la condition de bord $\forall n$, $u_0^n = u_0(0 - ct_n) = u_0(-cn\Delta_t)$ si $c > 0$, et $\forall n$, $u_M^n = u_0(L - ct_n)$ si $c < 0$.

**3)** Résoudre numériquement l'équation avec $c = 1$, $t_{\max} = L = 10$ et les pas $\Delta_x = 0.05$ et $\Delta_t = 0.025$. Tracer la solution numérique obtenue comme une surface 3D avec la fonction `plot_surface` de python.

1. **Attention :** En python, un test tel que `if X<2` n'est pas valide si $X$ est un *array* !
`The truth value of an array with more than one element is ambiguous.`
On ne pourra donc pas exécuter cette fonction directement sur un *array*. On peut choisir de passer par une boucle. Une solution plus efficace est de vectoriser votre fonction `u0sigmoid`. Pour cela, on ajoutera simplement (après la définition de `u0sigmoid`) la ligne suivante :

```
u0sigmoid  = np.vectorize(u0sigmoid, otypes=[np.float64])
```

La commande `u0sigmoid(X)` sera alors valide même si $X$ est un array.

**4)i)** Comparez (graphiquement, en 2D) la solution numérique et la solution exacte aux temps $t = 0,\ 2\ 5,\ 6$ et 10. Que constate-t-on ?

**ii)** Mêmes questions avec $c = -1$.
**iii)** Tester votre code avec la condition initiale

$$u_0(x) = \arctan(x)\ \forall x \in \mathbb{R}.$$

## II Suivi d'un front - conditions périodiques

Considérons la condition initiale en forme de "cloche"

$$u_0(x) = \begin{cases} \exp(\frac{-1}{1-(x-2)^2}) \text{ si } 1 < x < 3, \\ 0 \text{ sinon.} \end{cases}$$

On sait qu'à cause du transport, la "cloche" sort du domaine $[0, L]$ assez rapidement. Afin de pouvoir suivre son transport sur un temps plus long sans avoir à augmenter la valeur de $L$, on peut résoudre l'équation uniquement sur $[0, L]$ et imposer les conditions de bord périodiques

$$\forall t,\ u(t, 0) = u(t, L)$$

(on replie l'intervalle $[0, L]$ sur lui même, le point $x_0$ coïncide avec $x_M$).

La solution exacte de cette équation est donnée pour tout $t \in \mathbb{R},\ x \in [0, L]$ par $u(t, x) = u_0(x - ct - nL)$ si $x - ct \in [nL, (n+1)L[$ avec $n \in \mathbb{Z}$. En particulier : cette solution est périodique en $t$, de période $L/c$.

Numériquement, cette condition s'écrit pour tout $0 \leq n \leq N$ :

$$u_0^{n+1} = u_0^n - r(u_0^n - u_{M-1}^n) \text{ si } c > 0 \text{ et } u_{M-1}^{n+1} = u_{M-1}^n - r(u_0^n - u_{M-1}^n) \text{ pour } c < 0.$$

**5)** Écrire une fonction python *SchemaUpwindPeriod* qui prend en paramètre $c$, $t_{\max}$, $L$, $u0$, $M$ et $N$. Cette fonction résoudra l'équation de transport à vitesse constante $c$ et à conditions de bord périodiques à l'aide de l'algorithme *Upwind*. La fonction renverra $T$, $X$, et la matrice $U$. Attention : cette fois l'espace est constitué de $M$ (et non $M + 1$ points) !

**6)** Résoudre numériquement l'équation à conditions de bords périodiques avec $c = 1$, $t_{\max} = 15$, $L = 5$, $\Delta_x = 0.05$, $\Delta_t = 0.025$ et la condition initiale *en cloche*. Tracer la solution numérique obtenue comme une surface 3D avec la fonction `plot_surface` de python.

**7)** Comparez la solution numérique et la solution exacte aux temps $t = 0,\ 5,\ 10$ et 15 et 20. Qu'observe-t-on ?

Remarque : la solution exacte étant de période 5, il suffit de la tracer au temps 0.

**8)** Reprendre la question 7 avec en choisissant vos paramètres tels que $|r| := |c|\Delta_t/\Delta_x = 1$. Commenter.

### II.1 Schéma corrigé et schéma de Lax-Wendroff

Les sections précédentes illustrent que le schéma *Upwind* est convergent mais très diffusif. Dans le cas particulier où $r = 1$, la solution numérique est exacte, mais cette condition ne peut pas toujours être garantie (Par exemple si l'équation de transport est couplée à une autre équation, qui vient ajouter ses propres conditions de stabilité, ou si la vitesse $c$ n'est pas constante).

**9)** On se propose d'ajouter un terme au schéma précédent afin de compenser cette diffusion numérique. En repartant du code de la fonction *SchemaUpwindPeriod*, créer une fonction *SchemaUpwindPeriodCorrige* qui, au lieu du schéma *Upwind*, implémente le schéma

$$\begin{array}{c}\textit{Trouver } (u_i^n)_{i,n} \textit{solution de} \\ u_i^{n+1} = u_i^n - r(u_i^n - u_{i-1}^n) + \dfrac{r(r-1)}{2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n), \text{ si } c > 0 \\ u_i^{n+1} = u_i^n - r(u_{i+1}^n - u_i^n) + \dfrac{r(r+1)}{2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n), \text{ si } c < 0 \\ \text{(+cond. de bord), où } r := c\frac{\Delta_t}{\Delta_x}\end{array}$$

**10)** Tester votre fonction en reprenant les questions 6) et 7).

### II.2 Bonus : Vitesse variable

**12)** Adapter la fonction *SchemaUpwindPeriod* au cas où la vitesse est une fonction ($c = c(t,x)$) et tester votre code. Pour simplifier, on pourra ne considérer que le cas $\forall t, x,\ c(t,x) > 0$. Tester votre code avec la vitesse $c : [0, t_{\max}] \times [0, L] \to \mathbb{R}_*^+$ de votre choix, on veillera simplement à respecter la condition $\max\limits_{t,x}(c(t,x))\Delta_t/\Delta_x \leq 1$.

# A Construction du schéma

## A.1 Pourquoi le schéma Upwind est diffusif ?

Pour éviter de discuter les signes, on se place dans le cadre où $c$ est positif.

Par des développements de Taylor à l'ordre 3 (en supposant $u$ suffisamment régulière) on montre que [2] :

$$\begin{aligned}&\tfrac{u(t+\Delta_t,x)-u(t,x)}{\Delta_t} + c\tfrac{u(t,x)-u(t,x-\Delta_x)}{\Delta_x} \\ &= \underbrace{\partial_t u + c\partial_x u(t,x)}_{=0} + \underbrace{\frac{c\Delta_x}{2}(r-1)\partial_x^2 u(t,x) + \frac{c\Delta_x^2}{6}(1-r^2)\partial_x^3 u(t,x) + \mathcal{O}(\Delta_x^3)}_{\text{erreur de consistance}}\end{aligned}$$

Le terme dominant de l'erreur de troncature est donc proportionnel à $\partial_x^2 u \times \Delta_x$.

On en déduit que le schéma Upwind est équivalent à $\partial_t u + c\partial_x u = 0$ à l'ordre 1 mais aussi, plus précisément, à $\partial_t u + c\partial_x u + = -\frac{c\Delta_x}{2}(r-1)\partial_x^2 u(t,x)$ à l'ordre 2.

Cette dernière équation est une équation de convection (via $\partial_x$) et de diffusion (via $\partial_x^2$), avec coefficient de diffusion $\frac{-c\Delta_x}{2}(r-1) = \frac{c\Delta_x}{2}(1-r)$.

On observe alors que

- si $r = 1$, le terme de diffusion s'annule.
- $r > 1$, on a un coefficient de diffusion négatif (anti-diffusion), qui fait croitre la solution au lieu de la diffuser, d'où l'instabilité du schéma.
- Si $r < 1$, le coefficient est positif : le schéma est diffusif.

### A.1.1 Comment corriger cette diffusivité ?

Pour limiter ce phénomène de diffusion numérique, dominant dans l'erreur du schéma *Upwind*, on ajoute au schéma un terme dont le but est de compenser le terme de de diffusion numérique $\frac{c\Delta_x}{2}(1-r)$

Approchons ce terme de diffusion grâce à l'approximation de la dérivée seconde

$$\frac{u(t,x+\Delta_x) - 2u(t,x) + u(t,x-\Delta_x)}{\Delta_x^2} = \partial_x^2 u(t,x) + \mathcal{O}(\Delta_x^2)$$

2. Remarquons que pour obtenir une équation où les dérivées les dérivées d'ordre $> 1$ sont uniquement par rapport à $x$ on utilise simplement le fait que

$$\partial_t u = -c\partial_x u \Longrightarrow \partial_t^2 u = -c\partial_t\partial_x u = -c\partial_x\partial_t u = c^2\partial_x^2 u$$

et que, de même, $\partial_t^3 u = -c^3\partial_x^3 u$. De plus $\Delta_t$ et $\Delta_x$ sont liés par la relation $r = c\Delta_t/\Delta_x$, donc les classes $\mathcal{O}(\Delta_x^k)$ et $\mathcal{O}(\Delta_t^k)$ sont identiques.

On a donc :

$$
\begin{aligned}
& \tfrac{c\Delta_x}{2}(r-1)\partial_x^2 u(t,x) \\
= \quad & \tfrac{c\Delta_x}{2}(r-1)\tfrac{u(t,x+\Delta_x)-2u(t,x)+u(t,x-\Delta_x)}{\Delta_x^2} + \tfrac{c\Delta_x}{2}(r-1)\mathcal{O}(\Delta_x^2) \\
= \quad & \tfrac{c\Delta_x}{2}(r-1)\tfrac{u(t,x+\Delta_x)-2u(t,x)+u(t,x-\Delta_x)}{\Delta_x^2} + \mathcal{O}(\Delta_x^3)
\end{aligned}
\tag{3}
$$

En retranchant ce terme au schéma Upwind, on obtient le schéma

$$\frac{u_i^{n+1}-u_i^n}{\Delta_t} + c\frac{u_i^n - u_{i-1}^n}{\Delta_x} - \frac{c\Delta_x}{2}(r-1)\frac{u_{i+1}^n - 2u_i^n + u_{i+1}^n}{\Delta_x^2} = 0$$

Dont la formule de récurrence est donc bien

$$u_i^{n+1} = u_i^n - r(u_i^n - u_{i-1}^n) + \frac{r(r-1)}{2}(u_{i+1}^n - 2u_i^n + u_{i+1}^n). \tag{4}$$

Calculons l'erreur de troncature du schéma (4) :

$$
\begin{aligned}
& \tfrac{u(t+\Delta_t,x)-u(t,x)}{\Delta_t} + c\tfrac{u(t,x)-u(t,x-\Delta_x)}{\Delta_x} - \tfrac{c\Delta_x}{2}(r-1)\tfrac{u(t,x+\Delta_x)-2u(t,x)+u(t,x-\Delta_x)}{\Delta_x^2} \\
&= \underbrace{\partial_t u + c\partial_x u(t,x)}_{=0} + \underbrace{\mathcal{O}(\Delta_x^3) + \frac{c\Delta_x^2}{6}(1-r^2)\partial_x^3 u(t,x) + \mathcal{O}(\Delta_x^3)}_{\text{erreur de consistance}} \\
&= \tfrac{c\Delta_x^2}{6}(1-r^2)\partial_x^3 u(t,x) + \mathcal{O}(\Delta_x^3)
\end{aligned}
$$

Cette fois, le terme dominant de l'erreur de consistance est d'ordre $\Delta_x^2$. On est passé d'un schéma d'ordre 1 (en temps et espace) à un schéma d'ordre 2. De plus, le terme dominante de l'erreur est proportionnel à une dérivée d'ordre 3. On a alors un schéma considérablement moins diffusif (le prochain terme de diffusion ($\partial_x^4$) est en $\Delta_x^3$. Le principale défaut de ce schéma est son caractère dispersif (mauvaise vitesse de propagation des ondes), qui le rend peu souhaitable pour des conditions initiales très raides ou discontinues.

Ce schéma peut être vu comme une variante (très proche) du schéma de Lax-Wendroff, qui s'écrit traditionnellement avec une approximation centrée de la dérivée spatial d'ordre 1 et une correction de la diffusion numérique selon la méthode présentée ici.