

## TP 5

### Schémas aux différences finies pour l'équation de transport.

Ecrire une fonction `transport(c, T, xmin, xmax, dx, dt)` qui calcule la solution  $u(x, t)$  de l'EDP  $u_t + cu_x = 0$  avec la donnée initiale  $u(x, 0) = f(x)$  sur le segment  $x_{min} \leq x \leq x_{max}$  pour  $0 \leq t \leq T$ . Le réel  $c$  est une constante.

Vous utiliserez la méthode des différences finies avec divers schémas explicites : schéma amont, schéma centré, schéma de Lax-Friedrichs, schéma de Lax-Wendroff avec le pas d'espace  $dx$  et le pas de temps  $dt$  (voir les schémas plus loin).

La fonction `transport` calcule une matrice  $u = u(j, n)$  où  $u(j, n)$  est censée approcher  $u(x_j, t_n)$ . La grille de points  $(x_j, t_n)$  est obtenue en discrétisant le rectangle  $(x_{min}, x_{max}) \times (0, T)$  avec un pas d'espace  $\delta x$  et un pas de temps  $\delta t$ . Coder les schémas explicites suivant (où on a noté  $r = c\delta t/\delta x$ .)

$$\text{amont : } u_j^{n+1} = r u_{j-1}^n + (1-r) u_j^n.$$

$$\text{centré : } u_j^{n+1} = \frac{r}{2} u_{j-1}^n + u_j^n - \frac{r}{2} u_{j+1}^n.$$

$$\text{Lax - Friedrichs : } u_j^{n+1} = \left(\frac{1}{2} + \frac{r}{2}\right) u_{j-1}^n + \left(\frac{1}{2} - \frac{r}{2}\right) u_{j+1}^n.$$

$$\text{Lax - Wendroff : } u_j^{n+1} = \frac{1}{2}(r^2 + r) u_{j-1}^n + (1-r^2) u_j^n + \frac{1}{2}(r^2 - r) u_{j+1}^n.$$

La fonction devra tracer sur une même figure les 3 courbes  $x \mapsto u(x, 0)$ ,  $x \mapsto u(x, T)$  et la solution exacte  $x \mapsto f(x - cT)$  pour chacun des schémas. La condition initiale  $f(x)$  pourra être de trois types différents : indicatrice  $1_{(x < 0)}$  puis une cloche de Gauss  $\exp(-25(x - (x_{min} + x_{max})/2)^2)$ , ou encore une fonction indicatrice lissée :  $x \mapsto 4x^2(3 - 4x) \cdot 1_{(0 < x < 0.5)} + 1_{(x \geq 0.5)}$ .

```
Python
# donnée initiale
def f(x):
    # commenter la fonction choisie.
    # fonction 1_(x<0)
    y = np.zeros_like(x)
    y[x < 0] = 1
    # cloche de Gauss.
    # y = np.exp(-25 * (x - (xmin + xmax) / 2) ** 2)
    return y
```

Pour simplifier, vous prendrez des *conditions limites constantes* au cours des itérations :

```
Python
# conditions au bord constantes
u[0, n + 1] = u[0, n]
u[-1, n + 1] = u[-1, n]
```

Comparer les résultats des différents schémas pour les paramètres suivants  $c = 0.5$ ,  $T = 0.75$ ,  $\delta t = 0.01$ ,  $\delta x = 0.01$ ,  $x_{min} = -1$ ,  $x_{max} = 1$ . Prenez ensuite  $\delta t = 0.02$  puis  $\delta t = 0.021$ . Que se passe-t-il ?  
Changer enfin  $c$  en  $c = -0.5$ . Que constatez-vous ?