
Équation d'advection, Équation d'onde

NOM :

Prénom :

Le but de ce TP numérique est d'étudier différents schémas numériques pour la résolution des **équation d'advection et équation d'onde**. Ce sujet, les notes de cours associées ainsi que les ébauches de programmes du TP précédent sont disponibles à l'adresse suivante : <http://userpages.irap.omp.eu/~rbelmont/>.

1 Équation d'advection

On se propose de résoudre numériquement, sur le domaine $x \in [0, L]$, l'équation d'advection 1D pour un champ $u(t, x)$, une vitesse d'advection constante v avec un jeu de conditions initiales et de conditions aux bords **périodiques** :

$$\begin{aligned}\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} &= 0 \\ u(0, x) &= \sin\left(\frac{2\pi x}{L}\right) \\ u(t, 0) &= u(t, L)\end{aligned}$$

On rappelle que le paramètre CFL pour une équation d'advection est :

$$\lambda = \frac{v\delta t}{\delta x}$$

Les solutions numériques pourront être comparées à la solution analytique de cette équation :

$$T(t, x) = \sin\left(\frac{2\pi(x - vt)}{L}\right)$$

Sans perte de généralité, on pourra prendre $L = 1\text{m}$ et $v = 1 \text{ms}^{-1}$ dans la suite. Pour démarrer on pourra prendre $n_x = 128$ points en x , une condition CFL conservative $\lambda = 0.25$ et intégrer jusque $t_{\max} = 0.1\text{s}$. Ces paramètres seront ensuite à faire varier.

Pour tester l'évolution à grand temps des différents schémas, il pourra être astucieux de choisir des temps multiples du temps de traversée de la boîte.

1.1 Schémas explicites

Une ébauche de programme python vous est proposée dans le fichier `adv_exp1.py`. Attention : le code fourni n'utilise a priori pas de cellule fantôme pour les conditions aux limites et est plutôt basé sur une écriture spécifique du schéma numérique aux bords.

Q01– S'inspirer de ce programme et implémenter le schéma **FTCS** pour l'équation d'advection.

Q02– Étudier l'effet du paramètre CFL et tester la stabilité du schéma FTCS.

Q03– Chercher sur internet la définition et les propriétés du schéma explicite de type **Lax-Friedrich**. L'implémenter.

Q04– Étudier l'effet du paramètre CFL et tester la stabilité du schéma de Lax-Friedrich.

Q05– Quelle différence avec la solution théorique observe-t-on lorsque le temps devient suffisamment grand ? Comment s'appelle ce phénomène ? Interpréter ce résultat en terme d'erreur de consistance.

Q06– Pour un temps suffisamment long (par exemple $t_{\max} = 10\text{s}$) et $\lambda = 0.99$, mesurer, avec une syntaxe inspirée de `print("Erreur au max: ", abs(1-max(u)/max(uexacte(t,x))))` l'erreur au maximum de la Gaussienne pour différents nombres de points :

| n_x | 128 | 256 | 512 | 1014 |
|------------|-----|-----|-----|------|
| ϵ | | | | |

Ces résultats sont-ils compatibles avec l'ordre du schéma ?

Q07– Implémenter le schéma explicite de type **upwind**.

Q08– Étudier l'effet du paramètre CFL et tester la stabilité du schéma upwind.

Q09– Quels effets numériques sont observés à grand temps ? Interpréter ces effet dans le cadre de l'analyse d'erreur.

Q10– Changer le signe de la vitesse. Que faut-il faire pour conserver la nature upwind du schéma ?

1.2 Schéma implicite de Crank-Nicolson

Une ébauche de programme python vous est proposée dans le fichier `adv_impl.py`. Il suppose que le schéma implicite peut se mettre sous forme matricielle avec une matrice cyclique. Dans ce cas, l'inversion de matrice peut se faire avec des algorithmes beaucoup plus rapides qu'une inversion complète et générale. Ici, en python, cette inversion se fait avec la fonction `cyclic` à qui il faut fournir les 3 diagonales, les deux coins en bas à gauche et en haut à droite ainsi que le membre de droite.

$$\begin{pmatrix} b_0 & c_0 & 0 & \dots & 0 & 0 & \beta \\ a_0 & b_1 & c_1 & \dots & 0 & 0 & 0 \\ 0 & a_1 & b_2 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & b_{n-3} & c_{n-3} & 0 \\ 0 & 0 & 0 & \dots & a_{n-3} & b_{n-2} & c_{n-2} \\ \alpha & 0 & 0 & \dots & 0 & a_{n-2} & b_{n-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \dots \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ \dots \\ s_{n-3} \\ s_{n-2} \\ s_{n-1} \end{pmatrix}$$

D'autre part, on rappelle que l'erreur du schéma de Crank-Nicolson est en

$$\epsilon \sim c(dx^2/6)(1 - \lambda^2/2)d^3u/dx^3$$

Q11– En s'inspirant du schéma implicite pour l'équation de diffusion, écrire le schéma de Crank-Nicolson (centré en espace et en temps) de l'équation d'advection pour tous les points strictement intérieurs au domaine.

Q12– Écrire le schéma aux deux bords, pour des conditions périodiques.

Q13– Montrer qu'il peut se mettre sous la forme matricielle $M_{ij}u_j = s_i$ où M_{ij} est cyclique. Expliciter les termes a_i de la diagonale inférieure, b_i de la grande diagonale, c_i de la diagonale supérieure, α du coin inférieur gauche, et β du coin supérieur droit. Calculer finalement les composantes s_i .

Q14– S'inspirer du programme fourni et implémenter ce schéma avec une condition initiale sinusoïdale de fréquence k : $u(0, x) = \sin(2\pi kx/L)$.

Q15– Tester la stabilité de ce schéma pour différentes valeurs du coefficient CFL (il est préférable pour cela de prendre une basse résolution : $n_x = 64$ par exemple).

Q16– Quelle différence avec la solution théorique observe-t-on lorsque le temps devient suffisamment grand ? Comment s'appelle ce phénomène ? Interpréter ce résultat en terme d'erreur de consistance. La solution numérique est-elle en avance ou en retard sur la solution théorique ?

Q17– On s'intéresse maintenant au décalage Δx observé entre la solution numérique et la solution analytique à $t_{\max} = 10$ s, pour un paramètre CFL $\lambda = 1$. Celui-ci pourra être mesuré avec des commandes inspirées de :

```
nn = 20000; xx = linspace(0,L-dx,nn)
uuexa = interp(xx,x,uexacte(t,x))
uunum = interp(xx,x,u)
print "Dx = ",(argmax(correlate(uunum,uuexa,'full'))-nn+1)*1.*L/nn
```

Mesurer ce décalage pour différents nombres de points et $k = 1$, puis pour différentes fréquences de signal et $n_x = 256$:

| | | | | | |
|------------------------|----|----|-----|-----|-----|
| $k = 1, n_x = \dots$ | 32 | 64 | 128 | 256 | 512 |
| ϵ | | | | | |
| $n_x = 256, k = \dots$ | 1 | 2 | 4 | 6 | |
| ϵ | | | | | |

Est-ce compatible avec l'erreur de phase pour un schéma spatial centré d'ordre 2 (en $\Delta x \propto (k\Delta x)^2$) vue en cours ?

Q18– Étudier finalement, pour $0 < t < 10$ s, l'évolution temporelle d'une gaussienne ($n_x = 128$) puis d'un créneau¹ de largeur $l = 0.2$ ($n_x = 256$). Interpréter les effets observés en terme d'ondes.

1.3 Équation d'advection non-linéaire

L'équation linéaire d'advection est un bon test pour les schémas numériques car on peut comparer leur résultats à des solutions analytiques. Cependant, le réel intérêt des méthodes numériques apparait dans la résolution d'équations pour lesquelles on ne sait pas calculer de solution analytique simple. L'équation de Burgers est un exemple classique d'équation non-linéaire :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0$$

De tels termes apparaissent naturellement dans la dérivée Lagrangienne $d/dx = \partial/\partial t + \vec{u} \cdot \vec{\nabla}$ des équations hydrodynamiques.

Q19– Adapter le schéma de **Lax-Friedrich** pour résoudre cette équation, avec une condition initiale sinusoïdale. On admettra qu'avec une condition initiale telle que $|u(0, x)| < 1$, le signal reste toujours inférieur à 1 : $\forall t > 0, |u(t, x)| < 1$. Comment définir le paramètre CFL pour que le schéma soit stable ?

Q20– Qu'observez vous ? Interpréter le raidissement en terme de vitesse d'advection.

1. $u(0, x) = 1$ pour $L/2 - l/2 < x < L/2 + l/2$, et 0 ailleurs

2 Équation d'onde

On se propose maintenant de résoudre numériquement sur le domaine $x \in [0, L]$, l'équation d'onde pour un champ $u(t, x)$. Celle-ci s'écrit :

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$$

où c est la vitesse de propagation de l'onde. Une fois discrétisée en temps et en espace avec des pas δt et δx respectivement, le paramètre CFL de cette équation s'écrit : $\lambda = c\delta t/\delta x$.

Il s'agit d'une équation d'ordre 2 en temps, qui nécessite donc la connaissance de deux conditions initiales : par exemple une sur le champ $u(0, x)$ et une sur sa dérivée temporelle $u'(0, x)$.

Q21– Écrire un schéma **centré en espace et en temps** pour cette équation. Appliquer pour cela à la dérivée temporelle la même méthode de discrétisation que celle utilisée pour la dérivée spatiale dans l'équation de diffusion.

Numériquement, il s'agit d'une méthode à deux niveaux, c'est à dire d'une double récurrence. Il faudra donc utiliser 2 tableaux (u_a et u_b par exemple) qui représenteront à chaque pas de temps la valeur du champ aux deux pas de temps précédents. Il faudra également initialiser la récurrence avec les valeurs $u(x)$ aux deux premiers pas de temps. On admettra que le premier pas de temps u^1 peut se calculer de la manière suivante en fonction des conditions initiales $(u')^0$:

$$\forall i, \quad u_i^1 = u_i^0 + dt(u')_i^0 + \lambda (u_{i+1}^0 - 2u_i^0 + u_{i-1}^0)$$

Q22– Implémenter ce schéma numérique avec des conditions aux limites périodiques et une condition initiale statique Gaussienne.

Q23– Quel effet numérique observe-t-on à grand temps ?

Q24– Implémenter et tester d'autres conditions aux limites. Par exemple des conditions reflectives correspondent à $du/dx = 0$ aux bords.