Méthodes numériques, travaux pratiques

**le rapport devra être envoyé par mail (guillaume.fuhr@univ-amu.fr)
avant le 06 mars 2016 au format pdf**

Contents

Introduction

Le but du sujet proposé est d'étudier une équation d'advection-diffusion du point de vue des méthodes numériques de résolution. L'équation de base étudiée est la suivante :

$$\frac{\partial u(x,t)}{\partial t} + V\nabla u(x,t) = C\nabla^2 u(x,t) \tag{1}$$

$$\nabla := \begin{cases} \partial_x \\ \partial_y \\ \partial_z \end{cases}$$

Le domaine radial sera de taille $[L_x \times L_y] = [2\pi \times 2\pi]$, pour les diverses méthodes étudiées, il faudra choisir des valeurs pour Δt et Δx permettant d'atteindre un temps dans la simulation de l'ordre de $\simeq \mathbf{10}$ fois le temps caractéristique de la dynamique considérée.
Diffusion : $\tau_D = L^2/C$
Advection : $\tau_A = L/V$

Les coefficients et paramètres de grille ne sont pas imposés mais peuvent-être choisi librement. Un point de départ pour ces diverses valeurs vous pouvez utiliser une grille de taille $[8\times 8, 16\times 16, 32\times 32, \cdots, 256\times 256]$ et des valeurs pour C, V de l'ordre de $\left[10^{-2}, 10\right]$

1 1D equation

1.1 Diffusion

Dans cette partie, seul le terme de diffusion sera considéré :

$$\frac{\partial u(x,t)}{\partial t} = C\frac{\partial^2 u(x,t)}{\partial x^2} \tag{2}$$

1. Utilisez le programme advdiff pour résoudre l'équation utilisant la méthode d'Euler explicite. Déduisez du graphe de la solution aux divers temps les conditions de bords utilisées. Comment cela est-il implémenté dans le code?

2. Comparez les solutions obtenues quand $C\Delta t/\Delta x^2$ est $< 1/2, = 1/2 - \epsilon, = 1/2, = 1/2 + \epsilon, > 1/2$

3. Calculez le taux de croissance obtenu numérique et comparez le au taux analytique.

4. Pour Δx donné, observez vous une limite de stabilité pour le schéma RK4 en fonction du pas Δt? Estimez cette valeur à ± 0.01 prêt et comparez là au Δt correspondant à la méthode d'Euler explicite.

5. Théoriquement, il n'y a pas de limitation sur le pas de temps lors de l'utilisation d'un schéma implicite, est-ce vérifié numériquement? Commentez.

1.2 Advection

On s'interesse maintenant uniquement à l'équation d'advection :

$$\frac{\partial u(x,t)}{\partial t} + V\frac{\partial u(x,t)}{\partial x} = 0$$

1. Comment sont codées les dérivées spatiales dans le code? Est-ce un schéma avant, arrière ou centré?

2. Au sens physique, que fait une advection? en particulier, est-ce que la structure radiale de la solution est modifiée?

3. Qu'en est il dans les résultats obtenus avec les simulations numériques dans les deux cas suivant :

   - Quand on prend une fonction sin pour $u_0(x)$
   - Quand on prend une fonction de heavyside pour $u_0(x)$

4. Vérifiez si l'utilisation d'une méthode de RK4 permet d'utiliser un terme de dérivé centré en espace.

5. Comment évolue le système quand $V\Delta t/\Delta x$ est $< 1, \simeq 1, = 1, > 1$ pour les méthodes d'Euler et RK4.

6. Estimez la valeur maximale pour $V\Delta t/\Delta x$ si vous considérez un schéma RK4

1.3 Advection-Diffusion

Regardons maintenant l'équation complète :

$$\frac{\partial u(x,t)}{\partial t} + V\frac{\partial u(x,t)}{\partial x} = C\frac{\partial^2 u(x,t)}{\partial x^2}$$

1. Quel type de dérivée est utilisée pour le terme d'advection dans le schéma d'Euler implicite implémenté?

2. Comme dans le cas avec uniquement la diffusion, calculez le coefficient de diffusion à partir des données obtenues, ce résultat correspond-il à la valeur attendue?

3. Calculez à partir de vos résultats, le taux de croissance suivant qu'une dérivée centrée ou arrière est utilisée en espace pour le terme d'advection? Ce résultat dépend-il de la valeur du coefficient d'advection?

4. Dans les parties précédentes, deux critères de stabilités ont pu être mis en évidence, que se passe-t-il si l'on choisit des paramètres ne vérifiant que l'un des 2?

2 Appendices

2.1 Résultats analytiques

2.1.1 Solution générale de l'équation de la chaleur dans un domaine de taille finie

cas 1D

$$\begin{aligned}
\partial_t u(x,t) &= C\partial_x^2 u(x,t) + Du(x,t) + S(x) && (3)\\
\text{Domaine :} & \quad 0 \le x \le Lx && (4)\\
u(x,0) &= u_0(x) && (5)\\
u(0,t) &= 0 && (6)\\
u(0,Lx) &= 0 && (7)\\
u(x,t) &= \int_0^{Lx} u_0(x)G(x,\xi,t)\mathrm{d}\xi + \int_0^t \int_0^{Lx} S(\xi)G(x,\xi,t-\tau)\mathrm{d}\xi\mathrm{d}\tau && (8)\\
G(x,\xi,t) &= \frac{2}{Lx}e^{(Dt)}\sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{Lx}\right)\sin\left(\frac{n\pi\xi}{Lx}\right)\exp\left(-\frac{Cn^2\pi^2 t}{Lx^2}\right) && (9)\\
& && (10)
\end{aligned}$$

cas 2D

$$\begin{aligned}
\partial_t u(x,y,t) &= C\partial_x^2 u(x,t) + C\partial_y^2 u(x,y,t) + S(x,y) && (11)\\
\text{Domaine :} & \quad 0 \le x \le Lx && (12)\\
& \quad -\infty \le y \le \infty && (13)\\
u(x,y,0) &= u_0(x,y) && (14)\\
u(0,y,t) &= 0 && (15)\\
u(Lx,y,t) &= 0 && (16)\\
u(x,t) &= \int_{-\infty}^{\infty}\int_0^{Lx} u_0(\xi,\eta)G(x,y,\xi,\eta,t)\mathrm{d}\xi\mathrm{d}\eta && (17)\\
& + \int_0^t \int_0^{Lx}\int_{-\infty}^{\infty} S(\xi,\eta)G(x,y,\xi,\eta,t-\tau)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\tau && (18)\\
G(x,\xi,t) &= G_1(x,\xi,t)G_2(y,\eta,t) && (19)\\
G_1(x,\xi,t) &= \frac{2}{Lx}\sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{Lx}\right)\sin\left(\frac{n\pi\xi}{Lx}\right)\exp\left(-\frac{an^2\pi^2 t}{Lx^2}\right) && (20)\\
G_2(y,\eta,t) &= \frac{1}{2\sqrt{\pi C t}}\left[\exp\left(-\frac{(y-\eta)^2}{4Ct}\right) - \exp\left(-\frac{(y+\eta)^2}{4Ct}\right)\right] && (21)
\end{aligned}$$

2.2 Méthodes numériques

La convention suivante est utilisée dans la suite :

$$f(x,t) \to f(x_i,t_j) \to f(x+i\Delta x, t+j\Delta t) \to f_{t+i}^{x+j}$$

2.2.1 Dérivés

Partant d'une équation type,

$$\frac{\partial f(x,t)}{\partial t} = L(t, f(x,t)) \tag{22}$$

$$\begin{array}{lrcl} \text{dérivé avant :} & \partial_t f(t) & = & \frac{f^{t+1}-f^t}{\Delta t} \\ \text{dérivé arrière :} & \partial_t f(t) & = & \frac{f^t - f^{t-1}}{\Delta t} \\ \text{dérivé centrée :} & \partial_t f(t) & = & \frac{f^{t+1}-f^{t-1}}{2\Delta t} \end{array} \tag{23}$$

2.2.2 Euler explicite

$$\frac{f^{t+1}-f^t}{\Delta t} = L(t, f^t) \tag{24}$$

$$f^{t+1} = f^t + \Delta t L(t, f^t) \tag{25}$$

2.2.3 Euler implicite

$$\frac{f^{t+1}-f^t}{\Delta t} = L(t+\Delta t, f^{t+1}) \tag{26}$$

$$f^{t+1} = f^t + \Delta t L(t+\Delta t, f^{t+1}) \tag{27}$$

2.2.4 Cranck-Nicholson

$$\frac{f^{t+1}-f^t}{\Delta t} = \frac{1}{2}L(t, f^t) + \frac{1}{2}L(t+\Delta t, f^{t+1}) \tag{28}$$

$$f^{t+1} = f^t + \Delta t\left(\frac{1}{2}L(t, f^t) + \frac{1}{2}L(t+\Delta t, f^{t+1})\right) \tag{29}$$

2.2.5 Runge-Kutta 4

$$f^{t+1} = f^t + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{30}$$

$$\text{with} \tag{31}$$

$$k_1 = \Delta t L(t, f^t) \tag{32}$$

$$k_2 = \Delta t L(t + \frac{\Delta t}{2}, f^t + \frac{1}{2}k_1) \tag{33}$$

$$k_3 = \Delta t L(t + \frac{\Delta t}{2}, f^t + \frac{1}{2}k_2) \tag{34}$$

$$k_4 = \Delta t L(t + \Delta t, f^t + k_3) \tag{35}$$

$$\tag{36}$$

2.3 Code AdvDiff1D

2.3.1 téléchargement et compilation

adresse de téléchargement : **https://github.com/GFuhr/MF_FCM6/zipball/master**
Code source se trouve dans le dossier TP1.

Comment compiler

- Sous linux (Mac?) , la compilation se fait via la commande **make**.
- Pour éxécuter le programme généré sous linux, tapez **./advdiff.exe**
- Sous Windows, dans un éditeur type VisualStudio ou Code::Blocks, chargez les fichiers *advdiff.c* et *advdiff.h*.

2.3.2 Conditions initiales

temps

	$advection$	$diffusion$
$t = 0$	$u^0(x)$	$u^0(x)$

espace

	$advection$	$diffusion$
$x = 0$	$u(t, 0)$	$u(t, 0)$
$x = L_x$		$u(t, L_x)$

Champs initiaux

$$\begin{array}{c|c|}
u^0(x) & A\sin(\sigma \frac{\pi}{L_x}(x - x_0)) \\
u^0(x) & A\exp(-(x - x_0)^2/\sigma^2) \\
u^0(x) & \begin{cases} A \text{ if } x \in [x_0 - \sigma/2,\, x_0 + \sigma/2] \\ 0 \text{ else} \end{cases}
\end{array}$$

I/O Inputs :
Signification des entrées initiales.

- "Nx=?" : nombre de points en espace
- "Npas=?" : nombre d'itérations en temps
- "Nout=?" : chaque $Nout$ iterations, $u(t, x)$ sera exporté dans un fichier
- "C=?" : coefficient de diffusion
- "V=?" : coefficient d'advection
- "A=?" : Amplitude initiale, comme décrit en 2.3.2
- "x0=?": Paramètre x_0 défini dans 2.3.2
- "sigma=?": Paramètre σ défini dans 2.3.2
- "Dx=?" : valeur du pas Δx
- "Dt=?" : pas de temps Δt

Sorties :
À chaque éxecution du programme, $Nout$ fichiers de sortie sont générés **out_XXXX.dat**. Remarque, ces fichiers ne sont pas écrasés entre les runs. Chaque fichier est codé en format texte (ASCII) et contient $(Nx - 2)$ points avec les valeurs de u_i^j au temps $t = j * Nout$.

2.4 commandes utiles

- pour mesurer le temps d'éxécution, la commande **time** est utilisée :
  time ./bin/h2d_gcc.exe
  Remarque : une mesure ne peut être considérée comme fiable que si elle est supérieure à 10 secondes.

- tracer des données 1D avec gnuplot.

  ```
  gnuplot> plot "out_0000.dat" with lines
  ```

- with Octave

  - to read datas use function *load*
  - to plot datas $u(x, y)$, use function *surf*

    ```
    octave> data=load('H2D_0000.dat');
    octave> surf(data)
    ```

- download code in a terminal :
  wget https://github.com/GFuhr/MF_FCM6/zipball/master

- unzip archive :
  unzip master

- put unzipped files in a directory with a "friendly name" :
  mv GFuhr-MF_FCM6* *newname*

- change directory :
  cd

- create directory mkdir
  mkdir directory_name

- delete file :
  rm filename

- list file in a directory :
  ls directory_name

- list file in current directory :
  ls ./