

Filière "Systèmes énergétiques et marchés"
Année Universitaire 2009-2010

BE de Méthodes Numériques pour les EDP
Code pour l'équation d'advection linéaire en 1D

1 Position du problème - Principes de résolution numérique

On s'intéresse au problème suivant :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial f(\rho)}{\partial x} = 0 & \text{pour } t \geq 0, x \in [x_{min}, x_{max}] \\ \rho(x, t = 0) = \rho_0(x) & \text{pour } x \in [x_{min}, x_{max}] \end{cases} \quad (1)$$

où ρ désigne une quantité physique (par exemple la concentration d'un polluant) transportée à la vitesse constante a le long du segment $[x_{min}, x_{max}]$ avec $x_{max} = x_{min} + l_{dom}$. Le flux physique f est donc défini par $f(\rho) = a\rho$ où a est une constante que l'on supposera non-nulle.

La distribution initiale $\rho_0(x)$ sera soit un créneau (distribution discontinue) :

$$\rho_0(x) = \begin{cases} 0 & \text{si } x \leq 1 \\ 1 & \text{si } x \in]1, 2[\\ 0 & \text{si } x \geq 2 \end{cases} \quad (2)$$

soit une distribution gaussienne (régulière donc) :

$$\rho_0(x) = \exp(-2(x-3)^2) \quad (3)$$

On résout le problème (1) associé à la condition initiale (2) ou (3) en utilisant un schéma conservatif de flux numérique h et une intégration en temps de type Euler explicite. Le schéma de résolution peut donc se mettre sous la forme :

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x} (h_{i+\frac{1}{2}}^n - h_{i-\frac{1}{2}}^n) \quad (4)$$

où le flux numérique considéré est un flux dissipatif d'expression générale :

$$h_{i-\frac{1}{2}}^n = h(\rho_{i-1}^n, \rho_i^n) = \frac{1}{2}(f_i^n + f_{i-1}^n) - \frac{1}{2}q_{i-\frac{1}{2}}^n(\rho_i^n - \rho_{i-1}^n) \quad (5)$$

Le coefficient de dissipation q pourra être celui du schéma de Roe, de Lax ou de Lax-Wendroff (cf. notes de cours).

Le pas d'avancement en temps Δt est calculé de façon à satisfaire la condition de stabilité associée au schéma explicite à 3 points et 2 niveaux de temps utilisé soit :

$$\Delta t = CFL \frac{\Delta x}{|a|} \quad (6)$$

avec le nombre CFL choisi inférieur ou égal à 1 pour assurer la stabilité du schéma. Le choix du CFL et du nombre d'itérations temporelles $itermax$ détermine alors le temps final de la simulation.

Le schéma numérique ne s'applique qu'aux points intérieurs du maillage ; aux frontières du domaine de calcul on fait évoluer l'état physique ρ en appliquant des conditions aux limites. Ces conditions sont écrites en s'appuyant sur la théorie des caractéristiques : à la frontière amont ($x = x_{min}$ soit $i = 1$) du domaine, une vitesse d'onde positive correspond à une caractéristique entrante alors qu'une vitesse d'onde négative correspond à une caractéristique sortante (la proposition est inversée à la frontière aval ($x = x_{max}$ soit $i = I$)). Une caractéristique sortante signifie que la condition aux limites sera bien posée si on extrapole l'état à la frontière à partir de l'état au point intérieur voisin ; inversement, une caractéristique entrante conduit à imposer un état (fixé par la condition initiale) à la frontière du domaine.

2 Code de calcul

Les méthodes numériques ci-dessus sont implémentées dans un code *Scilab* mis à votre disposition dont on décrit succinctement l'organisation. Le "programme principal" correspond au script *scalaire.sce* qui fait lui-même appel aux autres scripts (*param.sce*, *model.sci*, ...). Ce programme principal commence donc par faire appel au script *param.sce* pour lire les paramètres de la simulation, qui sont fournis par l'utilisateur :

- a : vitesse d'onde
- x_{min} et l_{dom} pour définir le domaine de calcul
- $imax$ le nombre de points du maillage
- CFL le nombre CFL qui multiplie le pas de temps caractéristique de l'advection
- $itermax$ le nombre maximal d'itérations en temps
- $freq - ech$ la fréquence d'affichage des sorties graphiques (on ne souhaite pas nécessairement afficher la solution à chaque itération afin d'économiser en temps de simulation, dans la mesure où l'affichage graphique est souvent gourmand en temps de calcul processeur)
- $iinit$ une "clé" (de valeur 0 ou 1) que fixe l'utilisateur pour choisir la condition initiale du problème (créneau ou gaussienne)
- $ischema$ une "clé" de choix du schéma (l'expression du coefficient de dissipation q sera adaptée en fonction de la valeur de ce paramètre *ischema*)

Le programme principal charge ensuite les fonctions qui définissent le problème et seront utilisées dans la suite du programme :

- $f - phys$: fonction de flux physique
- $a - phys$: fonction de vitesse d'onde associée au flux physique, dont on doit donc vérifier la cohérence avec $f - phys$
- $q - num$: fonction de coefficient de dissipation (celui du schéma de Roe, de Lax ou de Lax-Wendroff)
- $rho - init$: fonction associée à la distribution initiale (créneau ou gaussienne)
- $rho - ex$: fonction solution exacte du problème d'advection scalaire ; elle utilise naturellement la fonction précédente et est donc placée après celle-ci.

On revient ensuite au programme principal pour définir le maillage (en fonction de $imax$, x_{min} , l_{dom}) et générer la distribution initiale du tableau $rho(i)$ qui contient les variables ρ_i^n (avec $n = 0$ pour l'état initial). On fait appel au script *conditions-limites.sce* pour calculer les états limites $rho(1)$ et $rho(imax)$ suivant les principes de la méthode des caractéristiques précédemment évoqués. On initialise la variable *temps* qui va stocker le temps physique écoulé depuis l'initialisation et on entre dans la boucle principale du code (boucle en temps physique). A l'intérieur de cette boucle (sur l'indice *iter* variant de 1 à *itermax*, qui correspond à la notation n de l'écriture papier du schéma),

on commence, à chaque itération, par calculer le pas de temps dans *pas-de-temps.sce*. Dans le cas du problème linéaire traité (a est une constante) ce pas de temps ne change pas en réalité d'une itération à l'autre mais il en irait autrement pour un problème non-linéaire (cf. équation du trafic qui sera résolue ultérieurement). On calcule ensuite le flux numérique dans le script *flux-numerique.sce*. On a fait le choix de stocker dans la case i du tableau *fnum* le flux numérique h calculé en $i + \frac{1}{2}$; par conséquent, ce flux numérique est calculé pour $i = 1, imax - 1$. Si on avait fait plutôt le choix (équivalent bien sûr) de stocker $h_{i-\frac{1}{2}}$ dans *fnum(i)* alors l'expression de *fnum(i)* ferait intervenir les états $rho(i)$, $rho(i - 1)$ et serait placée dans une boucle $i = 2, imax$. On utilise le flux numérique pour actualiser l'état physique *rho* dans toutes les cellules intérieures ($i = 2, imax - 1$) en traduisant l'expression (4) du schéma de discrétisation. On applique les conditions aux limites pour actualiser les états frontières. On calcule la nouvelle expression de la solution exacte et on trace solution numérique et solution exacte selon la fréquence d'affichage ou d'échantillonnage *freq - ech*. Le calcul s'arrête quand les *itermax* itérations ont été effectuées.