Travaux pratiques TP $n°6$. Introduction aux méthodes de volumes finis (MVF)
Transport pur linéaire. Problème monodimensionnel

1 Schémas volumes finis pour le transport libre

Soit $a > 0$. On considère le problème de transport linéaire libre unidimensionnel

$$\partial_t u + a\,\partial_x u = 0, \quad x \in [0,1],\ t > 0$$

avec conditions aux limites **périodiques** et de condition initiale

$$u(t=0,x) = u^0(x) \quad \text{dans } [0,1]$$

avec $u^0 \in L^\infty \cap BV(0,1)$. La solution analytique de ce problème est

$$u(x,t) = u^0(x-at) \quad (\text{+ périodification}).$$

Pour le TP, on considèrera la donnée initiale suivante sur $[0,1]$:

$$u^0(x) = (\sin(6\pi x))_+\ 1_{(x\in[0,1/3])}(x) + (3x-1)1_{x\in[1/3,2/3[}(x) + 1_{(x\in[2/3,1[)}(x).$$

On va considérer différents schémas conservatifs volumes finis pour la résolution numérique de

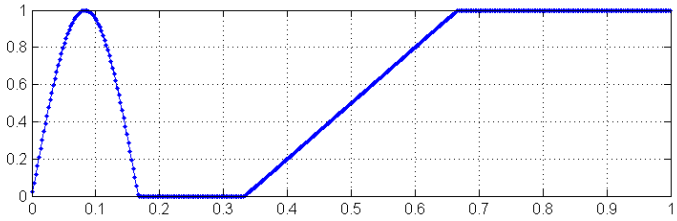

FIGURE 1 – Donnée initiale $u^0(x)$ pour le problème de transport libre.

ce problème. On prendra une discrétisation uniforme de l'intervalle $[0,1]$ constituée de volumes finis $I_j =]x_{j-1/2}, x_{j+1/2}[$, $x_{j+1/2} = jh$ centrés aux points $x_j = (j-1/2)h$, $j = 1, ..., N$, $h = \frac{1}{N}$. Un schéma de volumes finis explicite s'écrit

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{h}\left(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n\right),$$

où $\Phi_{j+1/2}^n$ représente un flux numérique à l'interface $x_{j+1/2}$ entre les cellules I_j et I_{j+1}. Dans ce qui suit on note

$$\nu = \Delta t \frac{|a|}{h}$$

le nombre de Courant.

Comparer les quatre schémas numériques suivants, pour les valeurs respectives de ν : $\nu = 0.1$, $\nu = 0.5$, $\nu = 0.9$ et $\nu = 1$. On effectuera des simulations sur un intervalle de temps $[0,T]$, $T = 4$, avec $N = 1600$:

1. Schéma de Lax-Friedrichs :

$$\Phi^n_{j+1/2} = a\frac{u^n_j + u^n_{j+1}}{2} - \frac{1}{2\nu}|a|(u^n_{j+1} - u^n_j) \ ;$$

2. Schéma décentré amont (upwind) pour $a > 0$:

$$\Phi^n_{j+1/2} = au^n_j.$$

   NB : en toute généralité sur a, le schéma upwind s'écrirait

$$\Phi^n_{j+1/2} = a\frac{u^n_j + u^n_{j+1}}{2} - \frac{1}{2}|a|(u^n_{j+1} - u^n_j) \ ;$$

3. Schéma de Lax-Wendroff (l'unique schéma à un pas d'ordre deux en espace et en temps) :

$$\Phi^n_{j+1/2} = a\frac{u^n_j + u^n_{j+1}}{2} - \frac{1}{2}\nu|a|(u^n_{j+1} - u^n_j) \ ;$$

4. Schéma de De Vuyst-Jaisson :

$$\Phi^n_{j+1/2} = a\frac{u^n_j + u^n_{j+1}}{2} - \frac{1}{2}\varphi(\nu)|a|(u^n_{j+1} - u^n_j) \ ; \tag{1}$$

   Dans (1), on expérimentera deux types de fonctions $\varphi(\nu)$:

$$\varphi(\nu) = \sqrt{\nu} \text{ et } \varphi(\nu) = \nu + \frac{1}{4}\left(1 - (2\nu - 1)^2\right).$$

Pour la mise en œuvre, on remarquera que les 4 schémas ci-dessus s'écrivent tous de la forme (1), et qu'il suffit en de coder le schéma (1) pour considérer tous les schémas mentionnés ci-dessus.

On tracera l'évolution des solutions numériques au cours du temps, on rafraîchira la solution numérique tous les 20 pas de temps, et on comparera à la solution exacte. Observer notamment le comportement des solutions dans les zones de régularité, de rupture de pente et de régions de discontinuité.

Dissipation d'entropie. Pour les lois de conservation, la dissipation d'entropie (on dit aussi "production" d'entropie) est un processus dissipatif, donc stabilisant pour le système. Pour l'équation de transport libre, toutes les fonctions strictement convexes de u sont des entropies. En multipliant l'équation de transport par $\eta'(u)$, η strictement convexe, on obtient la loi de conservation supplémentaire

$$\partial_t\eta(u) + \partial_x(a\eta(u)) = 0.$$

Dans ce cas, il y a visiblement conservation de l'entropie. Dans le cas de l'équation de transport-diffusion :

$$\partial_t u + \partial_x(au) - \varepsilon\partial^2_{xx}u = 0,$$

on aurait l'équation non-conservative de convection-diffusion non homogène sur l'entropie :

$$\partial_t\eta(u) + \partial_x\Big(a\eta(u) - \varepsilon\eta'(u)\partial_x u\Big) = -\varepsilon\eta''(u)(\partial_x u)^2 \leqslant 0.$$

Le flux

$$\psi(u, \partial_x u) = a\eta(u) - \varepsilon\eta'(u)\partial_x u$$

inclut le flux convectif et le flux visqueux. Le terme source (négatif)

$$d = d(u, \partial_x u) = -\varepsilon\eta''(u)(\partial_x u)^2$$

est appelé terme source de **dissipation d'entropie** est une mesure du procédé dissipatif sous-jacent dans le système.

D'un point de vue numérique, on peut utiliser la production d'entropie numérique comme mesure ou estimation de la diffusion numérique du schéma. Remarquer que tous les schémas ci-dessus peuvent s'écrire avec un flux de la forme

$$\Phi^n_{j+1/2} = a\, u^n_{j+1/2}$$

avec un état moyen $u^n_{j+1/2}$ aux interfaces. On peut donc définir une dissipation d'entropie numérique $d^{n,n+1}_j$ dans chaque cellule I_j de la manière suivante :

$$d^{n,n+1}_j = \frac{\eta(u^{n+1}_j) - \eta(u^n_j)}{\Delta t} + \frac{\psi^n_{j+1/2} - \psi^n_{j-1/2}}{h}$$

avec

$$\psi^n_{j+1/2} = a\,\eta(u^n_{j+1/2}).$$

Le schéma est dit **entropique** si, pour toute entropie $(\eta(\nu), \psi(u))$, η strictement convexe, on a

$$d^{n,n+1}_j \leqslant 0 \quad \forall j,\ \forall n.$$

Calculer et afficher les productions d'entropie pour les schémas ci-dessous. On effectuera les calculs pour $\nu = 0.45$ et $\eta(u) = u^2/2$. On observera notamment si la "dissipation numérique d'entropie" est toujours négative, ou non (défaut d'entropie).

2 Reconstruction MUSCL avec limiteur de pentes et ordre 2 en espace

Le schéma décentré amont (*upwind*) est le plus stable de tous les schémas (stabilité dans tous les L^p, inégalités d'entropie discrètes) mais est seulement d'ordre 1 et est considéré comme trop diffusif en pratique. Pour "monter" en ordre, on utilise une reconstruction MUSCL, P1-discontinue par maille :

$$\tilde{u}^n(x) = u^n_j + s^n_j(x - x_j) \quad \text{dans } I_j,$$

où s^n_j représente la pente dans la maille I_j à l'instant t^n. On note

$$u^{+,n}_j = u^n_j + \frac{h}{2}s^n_j,\ u^{-,n}_j = u^n_j - \frac{h}{2}s^n_j$$

les valeurs de la fonction aux extrémités droite et gauche de la cellule I_j. Pour une reconstruction monotone de la solution discrète au temps t^n, on **limite** la pente en prenant le minimum des pentes reconstruites à partir des voisins gauche et droit, soit

$$s^n_j = minmod\left(\frac{u^n_j - u^n_{j-1}}{h}, \frac{u^n_{j+1} - u^n_j}{h}\right)$$

où

$$minmod(a,b) = sgn(a)\ \max(0, sgn(ab))\ \min(|a|,|b|).$$

Le schéma décentré amont avec la reconstruction MUSCL s'écrit alors

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{h}\left(\Phi_{j+1/2}^n - \Phi_{j-1/2}^n\right)$$

avec $\Phi_{j+1/2}^n = \Phi(u_j^{+,n}, u_{j+1}^{-,n})$ où

$$\Phi(u,v) = a\frac{u+v}{2} - \frac{1}{2}|a|(v-u).$$

Programmer la méthode MUSCL avec la même donnée initiale que la question précédente. On utilisera un nombre de Courant égal à $\nu = 0.45$. On tracera la solution numérique à chaque pas de temps pour voir l'évolution.

On pourra aussi essayer le limiteur de pente de Sweby défini par

$$sweby(a,b) = sgn(a)\ \max(0, sgn(ab))\ \max\left(\min(|a|, \beta|b|),\ \min(\beta|a|, |b|)\right),$$

où $\beta \in [1,2]$. Le cas $\beta = 1$ correspond au limiteur "minmod". Le cas $\beta = 2$ correspond au limiteur appelé "superbee".

3 Structures de données 2D pour les volumes finis triangulaires

Fichiers nécessaires : `disq0.amdba`, `face_number.m`

1) Écrire un script qui lit le fichier `disq0.amdba`, trace le maillage (`M`) correspondant et appelle la fonction **face_number()** :

```
M = face_number(M);
```

qui (si `M` est la variable contenant la structure maillage EF) :

- numérote les arêtes de 1 à `M.nba`.
- associe à chaque arête les numéros des 2 éléments adjacents par ordre croissant des numéros : `M.fac_elm(nf,i)` est le numéro du i-ème élément adjacent à l'arête `nf`. S'il s'agit d'une arête sur la frontière du domaine alors `M.fac_elm(nf,2) = 0`.
- calcule pour chaque arête `nf` la normale unitaire de l'arête orientée de `M.fac_elm(nf,1)` vers `M.fac_elm(nf,2)` : `M.fac_nor(nf,1:2)`.
- calcule pour chaque arête `nf` le milieu de l'arête : `M.fac_gra(nf,1:2)`.
- définit pour chaque arête `nf` un numéro de zone : `M.fac_zon(nf)`.
- calcule pour chaque arête `nf` la longueur de l'arête : `M.fac_mes(nf)`.
- calcule pour chaque triangle `ie` sa surface `M.elm_mes(ie)`.
- calcule pour chaque triangle `ie` le centre de gravité `M.elm_gra(ie,1:2)`.
- associe à chaque arête `nf` les 2 numéros des sommets de l'arête : `M.fac_som(nf,i)` est le numéro du ième sommet de l'arête `nf`. Ces 2 sommets sont rangés de telle sorte que la normale à l'arête, orientée de l'élément `M.fac_elm(nf,1)` vers l'élément `M.fac_elm(nf,2)` soit définie par

```
[ M.som_coo(M.fac_som(:,2),2) - M.som_coo(M.fac_som(:,1),2) ;
 -M.som_coo(M.fac_som(:,2),1) + M.som_coo(M.fac_som(:,1),1)]
```

En conclusion, la fonction `face_number()` permet d'enrichir les structures de données éléments finis (EF) pour traiter les volumes finis (VF) et les schémas numériques volumes finis associés.

3) **Passage données VF centrées aux triangles vers des données EF aux sommets du maillage**.

Sur papier : écrire l'algorithme qui à partir d'une donnée définie triangle par triangle `sol_t` calcule une donnée approchée (interpolée) en chaque sommet `sol_s` par la formule suivante :

$$sol\_s(S_{is}) = \frac{\sum_{K \ni S_{is}} |K| sol\_t(K)}{\sum_{K \ni S_{is}} |K|} \qquad \forall\, S_{is} \text{ sommet du maillage} \tag{2}$$

Écrire en Matlab une fonction `tri_to_som` de la forme

```
function [sol_s] = tri_to_som(mesh, sol_t)
```

qui à partir d'une donnée définie triangle par triangle `sol_t` calcule une donnée approchée (interpolée) en chaque sommet `sol_s` par la formule de moyenne donnée plus haut.

Pour valider la méthode, tracer le champ EF P1 de la fonction

$$f(x, y) = xy$$

sur le maillage `disq0.amdba`. Calculer le champ VF discret `f_t`, constant par maille, de f, avec $f_K = f(x_K)$ où x_K est le centre de gravité du triangle K et calculer son interpolé `f_s` sur les sommets du maillage via la fonction `tri_to_som()` :

```
f_s = tri_to_som(mesh, f_t);
```

Tracer le champ `f_s` et comparer au champ EF P1.