

TP n° 8 – 24 janvier 2017

Résolution par MVF des équations de Saint-Venant en dimension 2. Schéma à flux caractéristique.

Fichiers de maillage : dam0 . amdba, dam1 . amdba, dam2 . amdba.

Les équations des eaux peu profondes (“shallow water equations” en anglais) ou équations de Saint-Venant modélisent les écoulements de fluides incompressibles à frontière libre, en hypothèse de faible profondeur d’eau. Elles sont très utiles en ingénierie environnementale, géosciences, etc. Elles permettent par exemple de simuler des écoulements proche des côtes (tsunamis, marées) ou dans les rivières. On s’intéresse ici au problème de la rupture d’un barrage.

Ces équations dérivent des équations d’Euler incompressibles en tenant compte des conditions aux limites cinétiques et cinématiques au fond et à la surface libre de l’écoulement. Lorsque l’on néglige les forces de Coriolis et les forces de frottement au fond, les lois de conservation de la masse et de la quantité de mouvement s’écrivent :

$$\frac{\partial V(x, y, t)}{\partial t} + \frac{\partial F_1(V(x, y, t))}{\partial x} + \frac{\partial F_2(V(x, y, t))}{\partial y} = G(x, y, V(x, y, t)) \quad (1)$$

$$V(x, y, 0) = V_0(x, y), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in [0, T]$$

avec

$$V = \begin{pmatrix} h \\ hu_1 \\ hu_2 \end{pmatrix}, \quad F_1(V) = \begin{pmatrix} hu_1 \\ hu_1^2 + g \frac{h^2}{2} \\ hu_1 u_2 \end{pmatrix}, \quad F_2(V) = \begin{pmatrix} hu_2 \\ hu_1 u_2 \\ hu_2^2 + g \frac{h^2}{2} \end{pmatrix}$$

et

$$G(x, y, V) = \begin{pmatrix} 0 \\ -g h \partial_x z \\ -g h \partial_y z \end{pmatrix}.$$

Ici $h(x, y, t)$ et $\mathbf{u}(x, y, t) = (u_1(x, y, t), u_2(x, y, t))$ sont respectivement la hauteur et la vitesse du fluide au point (x, y) et à l’instant t . La fonction $z = z(x, y)$ représente la hauteur du fond au point (x, y) calculée par rapport à une hauteur de référence et $g = 9.81 \text{ m.s}^{-2}$ la gravité. Les équations de Saint-Venant peuvent être encore écrites

$$\partial_t h + \nabla \cdot (h\mathbf{u}) = 0,$$

$$\partial_t (h\mathbf{u}) + \nabla \cdot (h\mathbf{u} \otimes \mathbf{u}) + \nabla p = -g h \nabla z(x, y)$$

avec

$$p = p(h) = \frac{1}{2} g h^2.$$

SIMPLIFICATION. Dans le présent TP, pour simplifier on considère un fond plat i.e. $z = cst$ et donc un terme source G nul.

La matrice jacobienne du problème est donnée par l'expression suivante pour $V = (h, hu_1, hu_2)^T$ et $\boldsymbol{\nu} = (\nu_1, \nu_2) \in \mathbb{R}^2$

$$A(\boldsymbol{\nu}, V) \equiv \sum_{i=1}^2 \frac{\partial F_i}{\partial V} \nu_i = \begin{pmatrix} 0 & \nu_1 & \nu_2 \\ (c^2 - u_1^2)\nu_1 - u_1 u_2 \nu_2 & 2u_1 \nu_1 + u_2 \nu_2 & u_1 \nu_2 \\ (c^2 - u_2^2)\nu_2 - u_1 u_2 \nu_1 & u_2 \nu_1 & 2u_2 \nu_2 + u_1 \nu_1 \end{pmatrix}$$

avec la vitesse de propagation des ondes de gravité

$$c = \sqrt{gh}.$$

Les valeurs propres (réelles) de cette matrice sont

$$\lambda_1 = \mathbf{u} \cdot \boldsymbol{\nu} - c, \quad \lambda_2 = \mathbf{u} \cdot \boldsymbol{\nu}, \quad \lambda_3 = \mathbf{u} \cdot \boldsymbol{\nu} + c$$

et les vecteurs propres à droite et à gauche (normalisés) sont donnés par

$$\begin{aligned} \mathbf{r}_1 &= \begin{pmatrix} 1 \\ u_1 - c\nu_1 \\ u_2 - c\nu_2 \end{pmatrix} & \mathbf{r}_2 &= \begin{pmatrix} 0 \\ -\nu_2 \\ \nu_1 \end{pmatrix} & \mathbf{r}_3 &= \begin{pmatrix} 1 \\ u_1 + c\nu_1 \\ u_2 + c\nu_2 \end{pmatrix} \\ \boldsymbol{\ell}_1 &= -\frac{1}{2c} \begin{pmatrix} -c - \mathbf{u} \cdot \boldsymbol{\nu} \\ \nu_1 \\ \nu_2 \end{pmatrix} & \boldsymbol{\ell}_2 &= \begin{pmatrix} u_1 \nu_2 - u_2 \nu_1 \\ -\nu_2 \\ \nu_1 \end{pmatrix} & \boldsymbol{\ell}_3 &= \frac{1}{2c} \begin{pmatrix} c - \mathbf{u} \cdot \boldsymbol{\nu} \\ \nu_1 \\ \nu_2 \end{pmatrix}. \end{aligned}$$

On a

$$A(\boldsymbol{\nu}, V) = R(\boldsymbol{\nu}, V) \Lambda(\boldsymbol{\nu}, V) (L(\boldsymbol{\nu}, V))^T,$$

avec $L = \text{col}(\boldsymbol{\ell}_k)$, $\Lambda = \text{diag}(\lambda_k)$, $R = \text{col}(\mathbf{r}_k)$.

Enfin les définitions de F_1 et de F_2 entraînent que le flux normal dans la direction $\boldsymbol{\nu}$ vaut

$$F(V) \cdot \boldsymbol{\nu} = \begin{pmatrix} h(\mathbf{u} \cdot \boldsymbol{\nu}) \\ hu_1(\mathbf{u} \cdot \boldsymbol{\nu}) + \frac{1}{2}gh^2\nu_1 \\ hu_2(\mathbf{u} \cdot \boldsymbol{\nu}) + \frac{1}{2}gh^2\nu_2 \end{pmatrix}$$

L'objet de cette séance est de résoudre ce système hyperbolique par la méthode des volumes finis à flux caractéristique (VFFC). Cela consiste à chercher une approximation constante par morceaux (approximation de la moyenne de la solution sur chaque volume fini K) telle que

$$\begin{aligned} V_K^{n+1} &= V_K^n - \frac{\Delta t}{|K|} \sum_{L \in \mathcal{N}(K)} |K \cap L| \Phi(V_K^n, V_L^n, \boldsymbol{\nu}_{K,L}) \\ &\quad - \frac{\Delta t}{|K|} \sum_{\partial K \subset \partial \Omega} |\partial K \cap \partial \Omega| \Phi(V_K^n, \partial \Omega, \boldsymbol{\nu}_{K,\partial \Omega}) \end{aligned}$$

où le flux numérique sur les arêtes intérieures est défini par

$$\begin{aligned} \Phi(V_K^n, V_L^n, \boldsymbol{\nu}_{K,L}) &= \frac{F(V_K^n) \cdot \boldsymbol{\nu}_{K,L} + F(V_L^n) \cdot \boldsymbol{\nu}_{K,L}}{2} \\ &\quad - \text{sgn}(M_{KL}^n) \frac{F(V_L^n) \cdot \boldsymbol{\nu}_{K,L} - F(V_K^n) \cdot \boldsymbol{\nu}_{K,L}}{2} \end{aligned}$$

avec

$$M_{KL}^n = A \left(\nu_{K,L}, \frac{V_K^n + V_L^n}{2} \right).$$

Pour les arêtes de bord, deux cas se présentent :

– cas du mur (arêtes de zone 1) : on prendra en 1ère approximation ($\boldsymbol{\nu} = \boldsymbol{\nu}_{K,\partial\Omega}$)

$$\Phi(V_K^n, \partial\Omega, \boldsymbol{\nu}) = \begin{pmatrix} 0 \\ \frac{1}{2} g (h_K^n)^2 \nu_1 \\ \frac{1}{2} g (h_K^n)^2 \nu_2 \end{pmatrix}$$

– cas des bords artificiels dits numériques (arêtes de zone 2) : on prendra en 1ère approximation et pour l'intervalle de temps considéré ($\boldsymbol{\nu} = \boldsymbol{\nu}_{K,\partial\Omega}$)

$$\Phi(V_K^n, \partial\Omega, \boldsymbol{\nu}) = \begin{pmatrix} (h\mathbf{u})_K^n \cdot \boldsymbol{\nu} \\ (\mathbf{u}_K^n)_{|_1} (h\mathbf{u})_K^n \cdot \boldsymbol{\nu} + \frac{1}{2} g (h_K^n)^2 \nu_1 \\ (\mathbf{u}_K^n)_{|_2} (h\mathbf{u})_K^n \cdot \boldsymbol{\nu} + \frac{1}{2} g (h_K^n)^2 \nu_2 \end{pmatrix}$$

Géométrie. Le domaine de calcul est le carré $[0, 200] \times [0, 200]$ avec un “barrage” en $x = 100$. A $t = 0$, le barrage rompt sur une partie entre $y = 95$ et $y = 170$.

Conditions initiales. Initialement, les hauteurs d’eau sont $h(x, y) = 10$ si $x \leq 100$ (amont du barrage) et $h(x, y) = 5$ si $x > 100$ (aval du barrage) et le fluide est au repos (vitesses nulles).

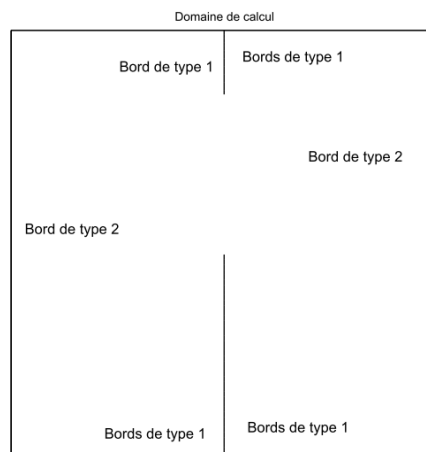


FIGURE 1 – Domaine de calcul

Structures de données. Pour représenter informatiquement les inconnues V , on prendra des tableaux à 2 indices : pour le volume K , $V(1, K)$ représentera $h(K)$, $V(2, K)$ représentera $(hu_1)(K)$, $V(3, K)$ représentera $(hu_2)(K)$.

1°) Écrire une fonction

```
function [Vnew] = init_sw(mesh)
```

qui initialise la solution (attention, il y a 3 composantes !) et la fonction

```
function [Vnew] = conv_sw(mesh, Vold, dt)
```

qui à partir de l'approximation de la solution à $n\Delta t$ (variable `Vold`), calcule l'approximation à l'instant $t^{n+1} = (n+1)\Delta t$ de la solution i.e. V^{n+1} (variable `Vnew`) à l'aide du schéma volumes finis.

On effectuera une boucle **sur les arêtes** et on traitera séparément les **arêtes internes** et les **arêtes de bord**. Dans un premier temps, on calculera la matrice signe SM dite "signe de M " de la façon suivante (en tirant avantage de la routine `eig()` de Matlab) :

```
[V,D] = eig(M);  
SM = V * sign(D) / V; % Matrice signe
```

2°) Modifier le script du TP précédent (TP 7) qui lit le maillage, le trace, initialise la solution, trace la surface 3D de la hauteur initiale et qui résout le problème entre 0 et $T=6.8$ avec le pas Δt .

A chaque pas de temps, on affichera la ligne suivante (fonction `fprintf` de Matlab)

```
it temps max(h) max(q1) max(q2) min(h) min(q1) min(q2)
```

Ce programme tracera aussi :

- la surface 3D de la hauteur finale (à T) vue quasiment de face en l'orientant à l'aide de `view(55, 30)`,
- les lignes de niveau de la hauteur (avec `tri_contour()` et `hautr = 5 : 0.2 : 10`).

On appliquera ce programme aux maillages successifs :

- `dam0.amdba` avec $\Delta t = 0.4$;
- `dam1.amdba` avec $\Delta t = 0.1$;
- `dam2.amdba` avec $\Delta t = 0.05$.

3°) Reprendre la fonction `conv_sw()` en utilisant les expressions des éléments propres de la matrice jacobienne A et le fait que

$$F(V) \cdot \nu = \sum_{k=1}^3 \ell_k \cdot (F(V) \cdot \nu) \, r_k,$$

$$\text{sgn}(M_{KL}^n) = R_{KL}^n \, \text{diag}(\text{sign}(\lambda_i(M_{KL}^n))) \, (L_{KL}^n)^T,$$

$$\text{sgn}(M_{KL}^n) (F(V) \cdot \nu) = \sum_{k=1}^3 \text{sign}(\lambda_k(M_{KL}^n)) \, \ell_k \cdot (F(V) \cdot \nu) \, r_k. \quad (2)$$

En pratique et en général, on code le schéma VFFC au moyen de l'expression (2).