

Projet

Dalissier Eric

1 Mise en oeuvre de la méthode des éléments finis en 1D

Trouver u une fonction de l'ouvert connexe borné $\Omega \subset \mathbb{R} \mapsto \mathbb{R}$ telle que :

$$- \operatorname{div}(a \mathbf{grad}(u)) + (\mathbf{b} \mathbf{grad} u) + cu = f \text{ dans } \Omega \quad (1)$$

$$u = u_d \text{ sur } \Gamma \quad (2)$$

On multiplie par $v \in V_h$, avec $V_h = \{v \in H^1(\Omega) \text{ telque } v = v_d\}$ puis on intgre sur Ω :

$$\int_{\Omega} (-\operatorname{div}(a \mathbf{grad}(u))v + (\mathbf{b} \mathbf{grad} u)v + cuv)(x)dx = \int_{\Omega} (fv)(x)dx$$

Puis on fait une intégration par partie du premier membre :

$$-[(a \mathbf{grad}(u)v)(x)]_{\Omega} + \int_{\Omega} (a \mathbf{grad}(u) \mathbf{grad}(v))v + (\mathbf{b} \mathbf{grad} u)v + cuv)(x)dx = \int_{\Omega} (fv)(x)dx$$

Comme u_d est une constante connue, on a de mme pour v_d . Si on se place dans un cas aux limites de Dirichlet, on aura :

$$-[(a \mathbf{grad}(u)v)(x)]_{\Omega} = 0 \text{ car } u_d = v_d = 0$$

Donc dans ce cas là, la formulation variationnelle du problème (1) est :

$$\int_{\Omega} (a \mathbf{grad}(u) \mathbf{grad}(v))v + (\mathbf{b} \mathbf{grad} u)v + cuv)(x)dx = \int_{\Omega} (fv)(x)dx \quad (3)$$

Dans le cas contraire, on aura une formulation variationnelle :

$$\begin{aligned} \int_{\Omega} (a \mathbf{grad}(u) \mathbf{grad}(v))v + (\mathbf{b} \mathbf{grad} u)v + cuv)(x)dx = \\ = \int_{\Omega} (fv)(x)dx + \mathbf{grad}(u)(l1)v(l1) - \mathbf{grad}(u)(l0)v(l0) \end{aligned}$$

Ensuite on est amené à programmer *Mesh1D.cpp* qui nous rend un fichier *pas.dat* avec notre intervalle $[l0, l1]$, discrétisé soit de façon uniforme ($x_i = l0 + ihL$), soit de façon variable ($x_i = \sin(\varepsilon \frac{\pi}{2} ih) / \sin(\varepsilon \frac{\pi}{2})$). Avec $h = \frac{1}{N}$, $i = 0, \dots, N$, $\varepsilon = 3/4$, $L = \|l1 - l0\|$.

On veut ramener la résolution du problème (1) à une résolution d'un système matricielle de la forme $AX = F$. On a donc codé un programme *fctF.cpp*, qui à partir d'une fonction rentrée avant la compilation, nous rend un fichier *second.dat* qui contient en première colonne, l'intervalle discrétisée et en deuxième colonne la valeur de F , calculer à partir de la méthode des trapèzes.

La *méthode des trapèzes* :

$$\int_a^b f(x)dx \approx (b-a) \frac{f(b) + f(a)}{2}$$

On aurait pu utiliser la *méthode de Simpson* :

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6} (f(b) + 4f(\frac{a+b}{2}) + f(a))$$

Ici on a $x \in [q_{k-1}; q_{k+1}]$:

$$v_h(x) = v_h \omega_k^0(x) + v_h \omega_k^1(x)$$

avec :

$$\left\{ \begin{array}{ll} \omega_k^0(x) = \frac{q_{k+1} - x}{q_{k+1} - q_k} & (\omega_k^0(x))' = -\frac{1}{q_{k+1} - q_k} \quad \text{sur } [q_{k+1}; q_k] \\ \omega_k^1(x) = \frac{x - q_{k-1}}{q_k - q_{k-1}} & (\omega_k^1(x))' = \frac{1}{q_k - q_{k-1}} \quad \text{sur } [q_k; q_{k-1}] \end{array} \right.$$

$$\begin{aligned} \int_{q_{k-1}}^{q_{k+1}} f v_h(x) dx &\approx (q_k - q_{k-1}) \frac{(v_h f)(q_k) + (v_h f)(q_{k-1})}{2} + (q_{k+1} - q_k) \frac{(v_h f)(q_{k+1}) + (v_h f)(q_k)}{2} \\ &= \frac{(q_k - q_{k-1})}{2} (f(q_{k-1}) \frac{q_{k-1} - q_{k-1}}{q_k - q_{k-1}} + f(q_k) \frac{q_k - q_{k-1}}{q_k - q_{k-1}}) \\ &\quad + \frac{(q_{k+1} - q_k)}{2} (f(q_k) \frac{q_{k+1} - q_k}{q_{k+1} - q_k} + f(q_{k+1}) \frac{q_{k+1} - q_{k+1}}{q_{k+1} - q_k}) \\ &= \frac{(q_k - q_{k-1})}{2} f(q_k) + \frac{(q_{k+1} - q_k)}{2} f(q_k) \end{aligned}$$

Une fois que \mathbf{F} est construit, il reste à construire la matrice A de taille $N \times N$. On écrit d'abord un programme qui va l'initialiser à zéro, puis après on va lui rajouter des N matrices élémentaires. Car en partant de la formulation variationnelle (3), on remarque :

$$\int_{\Omega} (a \mathbf{grad}(u) \mathbf{grad}(v))v + (\mathbf{b} \mathbf{grad}u)v + cuv)(x)dx = \int_{\Omega} (fv)(x)dx$$

On a pour $x \in [q_{k-1}; q_k]$:

$$v_h(x) = v_h(q_{k-1})\lambda_k^0(x) + v_h(q_k)\lambda_k^1(x)$$

avec :

$$\left\| \begin{array}{ll} \lambda_k^0(x) = \frac{q_k - x}{q_k - q_{k-1}} & (\lambda_k^0(x))' = -\frac{1}{q_k - q_{k-1}} \\ \lambda_k^1(x) = \frac{x - q_{k-1}}{q_k - q_{k-1}} & (\lambda_k^1(x))' = \frac{1}{q_k - q_{k-1}} \end{array} \right.$$

En remplaçant dans la formulation variationnelle :

$$\begin{aligned} & \sum_{k=0}^{N+1} \int_{q_{k-1}}^{q_k} a(v_h(q_{k-1})\lambda_k^0(x) + v_h(q_k)\lambda_k^1(x))'(u_h(q_{k-1})\lambda_k^0(x) + u_h(q_k)\lambda_k^1(x))' + \\ & \sum_{k=0}^{N+1} \int_{q_{k-1}}^{q_k} b(u_h(q_{k-1})\lambda_k^0(x) + u_h(q_k)\lambda_k^1(x))'(v_h(q_{k-1})\lambda_k^0(x) + v_h(q_k)\lambda_k^1(x)) + \\ & \sum_{k=0}^{N+1} \int_{q_{k-1}}^{q_k} c(v_h(q_{k-1})\lambda_k^0(x) + v_h(q_k)\lambda_k^1(x))(u_h(q_{k-1})\lambda_k^0(x) + u_h(q_k)\lambda_k^1(x)) = F \end{aligned}$$

Donc :

$$\begin{aligned} & \sum_{k=0}^{N+1} a \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \int_{q_{k-1}}^{q_k} \begin{bmatrix} -\frac{1}{q_k - q_{k-1}} \\ \frac{1}{q_k - q_{k-1}} \end{bmatrix} \left[-\frac{1}{q_k - q_{k-1}}, \frac{1}{q_k - q_{k-1}} \right] dx [u_h(q_{k-1}), u_h(q_k)] + \\ & \sum_{k=0}^{N+1} b \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \int_{q_{k-1}}^{q_k} \begin{bmatrix} \frac{q_k - x}{q_k - q_{k-1}} \\ \frac{q_k - q_{k-1}}{x - q_{k-1}} \end{bmatrix} \left[-\frac{1}{q_k - q_{k-1}}, \frac{1}{q_k - q_{k-1}} \right] dx [u_h(q_{k-1}), u_h(q_k)] + \\ & \sum_{k=0}^{N+1} c \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \int_{q_{k-1}}^{q_k} \begin{bmatrix} \frac{q_k - x}{q_k - q_{k-1}} \\ \frac{x - q_{k-1}}{q_k - q_{k-1}} \end{bmatrix} \left[\frac{q_k - x}{q_k - q_{k-1}}, \frac{x - q_{k-1}}{q_k - q_{k-1}} \right] dx [u_h(q_{k-1}), u_h(q_k)] = F \end{aligned}$$

Une fois les matrices calculés :

$$\begin{aligned} & \sum_{k=0}^{N+1} a \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \int_{q_{k-1}}^{q_k} \begin{bmatrix} \frac{1}{(q_k - q_{k-1})^2} & -\frac{1}{(q_k - q_{k-1})^2} \\ -\frac{1}{(q_k - q_{k-1})^2} & \frac{1}{(q_k - q_{k-1})^2} \end{bmatrix} dx [u_h(q_{k-1}), u_h(q_k)] + \\ & \sum_{k=0}^{N+1} b \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \int_{q_{k-1}}^{q_k} \begin{bmatrix} \frac{(q_k - q_{k-1})^2}{q_{k-1} - x} & \frac{(q_k - q_{k-1})^2}{x - q_{k-1}} \\ \frac{(q_k - q_{k-1})^2}{q_{k-1} - x} & \frac{(q_k - q_{k-1})^2}{x - q_{k-1}} \end{bmatrix} dx [u_h(q_{k-1}), u_h(q_k)] + \\ & \sum_{k=0}^{N+1} c \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \begin{bmatrix} \frac{(q_k - q_{k-1})}{3} & \frac{(q_k - q_{k-1})}{6} \\ \frac{(q_k - q_{k-1})}{6} & \frac{(q_k - q_{k-1})}{3} \end{bmatrix} [u_h(q_{k-1}), u_h(q_k)] = F \end{aligned}$$

Finalement :

$$\begin{aligned}
& \sum_{k=0}^{N+1} a \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \begin{bmatrix} \frac{1}{q_k - q_{k-1}} & -\frac{1}{q_k - q_{k-1}} \\ -\frac{1}{q_k - q_{k-1}} & \frac{1}{q_k - q_{k-1}} \end{bmatrix} [u_h(q_{k-1}), u_h(q_k)] + \\
& \sum_{k=0}^{N+1} b \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} [u_h(q_{k-1}), u_h(q_k)] + \\
& \sum_{k=0}^{N+1} c \begin{bmatrix} v_h(q_{k-1}) \\ v_h(q_k) \end{bmatrix} \begin{bmatrix} \frac{(q_k - q_{k-1})}{3} & \frac{(q_k - q_{k-1})}{6} \\ \frac{(q_k - q_{k-1})}{6} & \frac{(q_k - q_{k-1})}{3} \end{bmatrix} [u_h(q_{k-1}), u_h(q_k)] = F
\end{aligned}$$

On a ainsi les matrices élémentaires en sommant les matrices 2x2, on aura :

$$A_k = \begin{bmatrix} \frac{a}{(q_k - q_{k-1})} - \frac{b}{2} + \frac{c(q_k - q_{k-1})}{3} & \frac{-a}{(q_k - q_{k-1})} + \frac{b}{2} + \frac{c(q_k - q_{k-1})}{6} \\ \frac{-a}{(q_k - q_{k-1})} - \frac{b}{2} + \frac{c(q_k - q_{k-1})}{6} & \frac{a}{(q_k - q_{k-1})} + \frac{b}{2} + \frac{c(q_k - q_{k-1})}{3} \end{bmatrix}$$

Ensuite, il ne restait plus qu'à programmer une fonction qui somme tout les A_k . Le c++ offre la possibilité plus commode de définir un opérateur pour cela. On l'a noté $+=$.

Avant de passer à la résolution de notre système linéaire $Ax = F$, on veut prendre en compte les conditions aux limites de Dirichlet. Pour cela on met à zéro $A_{1,2}$ et $A_{N,N-1}$.

Dans la dernière étape, on programme les fonctions *LU* et *Solve* pour résoudre ce système linéaire. On peut faire cela car A est tridiagonale.

2 Deux Schéma en temps

On propose deux schémas en temps de l'équation de Black-Scholes *sous forme divergente* :
Schéma implicite

$$\frac{u^n - u^{n-1}}{\delta t} - \frac{\partial}{\partial x} \left(\frac{\sigma^2 x^2}{2} \frac{\partial u^n}{\partial x} \right) - \beta x \frac{\partial u^n}{\partial x} + r u^n = 0$$

Schéma semi-implicite

$$\frac{u^n - u^{n-1}}{\delta t} - \frac{\partial}{\partial x} \left(\frac{\sigma^2 x^2}{2} \frac{\partial u^n}{\partial x} \right) - \beta x \frac{\partial u^{n-1}}{\partial x} + r u^n = 0$$

Avec comme condition initiale sur le temps : $u^0 = (K - x)^+$

Il faut discrétiser en espace ces 2 équations. Pour cela on prendra la base de V_h formée par les fonctions chapeaux. Soit $v_h \in V_h$ quelconque :

$$v_h = \sum_{j=0}^{N+1} v_h(x_j) w_j$$

On a donc pour le schéma implicite : $\forall x \in [q_{k-1}; q_k]$

$$\begin{aligned} & \sum_{j=0}^{N+1} \frac{(u_h^n(x_j)w_j(x) - u_h^{n-1}(x_j)w_j(x))}{\delta t} + \\ & \sum_{j=0}^{N+1} \int_{q_{k-1}}^{q_k} \left(\frac{\sigma^2 x^2}{2} (v_h^n(q_{k-1})\lambda_k^0(x) + v_h^n(q_k)\lambda_k^1(x))' (u_h^n(q_{k-1})\lambda_k^0(x) + u_h^n(q_k)\lambda_k^1(x))' + \right. \\ & \quad \left. \beta x (v_h^n(q_{k-1})\lambda_k^0(x) + v_h^n(q_k)\lambda_k^1(x))' (u_h^n(q_{k-1})\lambda_k^0(x) + u_h^n(q_k)\lambda_k^1(x))' + \right. \\ & \quad \left. r(u_h^n(q_{k-1})\lambda_k^0(x) + u_h^n(q_k)\lambda_k^1(x)) (v_h^n(q_{k-1})\lambda_k^0(x) + v_h^n(q_k)\lambda_k^1(x)) \right) dx = 0 \end{aligned}$$

avec :

$$\left\| \begin{aligned} \lambda_k^0(x) &= \frac{q_k - x}{q_k - q_{k-1}} & (\lambda_k^0(x))' &= -\frac{1}{q_k - q_{k-1}} \\ \lambda_k^1(x) &= \frac{x - q_{k-1}}{q_k - q_{k-1}} & (\lambda_k^1(x))' &= \frac{1}{q_k - q_{k-1}} \end{aligned} \right.$$

De façon analogue, on trouve la formulation variationnelle discrétisée en espace :

$$\begin{aligned} & \sum_{j=0}^{N+1} \frac{(u_h^n(x_j)w_j(x) - u_h^{n-1}(x_j)w_j(x))}{\delta t} + \\ & \sum_{j=0}^{N+1} \int_{q_{k-1}}^{q_k} \left(\frac{\sigma^2 x^2}{2} (v_h^n(q_{k-1})\lambda_k^0(x) + v_h^n(q_k)\lambda_k^1(x))' (u_h^n(q_{k-1})\lambda_k^0(x) + u_h^n(q_k)\lambda_k^1(x))' + \right. \\ & \quad \left. \beta x (v_h^{n-1}(q_{k-1})\lambda_k^0(x) + v_h^{n-1}(q_k)\lambda_k^1(x))' (u_h^{n-1}(q_{k-1})\lambda_k^0(x) + u_h^{n-1}(q_k)\lambda_k^1(x))' + \right. \\ & \quad \left. r(u_h^n(q_{k-1})\lambda_k^0(x) + u_h^n(q_k)\lambda_k^1(x)) (v_h^n(q_{k-1})\lambda_k^0(x) + v_h^n(q_k)\lambda_k^1(x)) \right) dx = 0 \end{aligned}$$