

Implémentation de la méthode $Q_1/VF9$ en dimension deux.

Introduction.

On présente dans ce chapitre une implémentation indépendante de la méthode multi-échelle, réalisée durant la première année du travail de thèse. L'objectif premier de la réalisation de cette maquette est de pouvoir prendre conscience des diverses problématiques associées à ce sujet de thèse. C'est pourquoi on s'affranchit autant que possible des contraintes techniques en imposant des hypothèses fortes au problème, pour se concentrer sur la méthode multi-échelle proprement dite.

En particulier, le domaine de travail est dans ce chapitre considéré comme un ouvert rectangulaire de \mathbb{R}^2 exclusivement. Il est découpé en macro-éléments K eux-aussi rectangulaires. Au niveau grossier, le problème est résolu par une méthode d'Éléments Finis Q_1 , décrite à la section §3.3.1.

Au niveau fin, les problèmes de cellules sont résolus par le biais d'une méthode de Volumes Finis $VF9$ [97], décrite à la section §3.2.2, sur des maillages fins quadrangulaires réguliers. Ce dernier choix permet de laisser de côté le problème de la discrétisation du milieu qui demande une attention particulière dans un cadre plus étendu et qui est l'objet du chapitre §6.

Le travail se fait ici séquentiellement, la parallélisation des travaux étant reportée à l'implémentation *MPCube* décrite au chapitre §5.

Le chapitre s'articule autour de deux axes. On présente tout d'abord l'implémentation des différentes étapes de la méthode, ainsi que sa validation par le biais d'un ou plusieurs exemples. On utilise ensuite cette maquette pour réaliser une simulation de diffusion au sein d'un cas pratique de matériaux cimentaires. Les résultats présentés dans ce chapitre ont été partiellement rédigés dans un rapport CEA [9].

4.1 Implémentation.

L'implémentation de la méthode $Q_1/VF9$ a été réalisée en langage Python (version 2.4 et supérieure) [155]. Comme on l'a décrit dans le chapitre précédent, la méthode multi-échelle se découpe en plusieurs étapes (génération du milieu, résolution des problèmes de cellules, assemblage de la base d'Éléments Finis, etc.), chaque étape utilisant les résultats de la précédente.

Ce type de chaîne de calcul peut s'avérer très vulnérable aux erreurs, surtout si le travail est fait *à la volée*, c'est-à-dire sans stockage de résultats temporaires. En effet, si on ne stocke que les résultats finaux, i.e la solution reconstruite fine $C_{H,h}$, une erreur intervenant à la dernière étape de la chaîne de calcul entraînera la perte de toutes les données précédemment calculées. Il sera alors nécessaire de reprendre le travail sur la chaîne de calcul depuis le début.

De plus, travailler *à la volée* réduit considérablement l'intérêt de la méthode multi-échelle. En effet, les problèmes de cellules et le problème grossier sont découplés : une fois les problèmes de cellules résolus, n'importe quel problème grossier peut être résolu. Il est donc vital de stocker les solutions des problèmes de cellules, ainsi que les données associées que sont les matrices locales Éléments Finis.

Dans cette optique, chaque étape de la méthode multi-échelle a été implémentée dans un module indépendant des autres. Ces modules sont ensuite liés par le biais du script Python SIMULATION.PY, qui joue donc le rôle de superviseur. À chaque lancement du script, les paramètres entrés permettent de spécifier quelle étape de la méthode on souhaite effectuer. Les résultats sont ensuite stockés pour un usage ultérieure, c'est-à-dire pour l'étape suivante de la méthode multi-échelle.

Les modules développés sont :

- FCTMILIEU permet de construire les milieux modélisant les matériaux cimentaires. Il utilise le module COMBS, développé par Erwan ADAM.
- FCTVF9 permet de résoudre les problèmes de cellules par la méthode $VF9$. Une fois le système linéaire construit, il fait appel à la librairie NUMERICAL PLATON [167] pour la résolution proprement dite.
- FCTBASE construit la base d'Éléments Finis à partir des solutions des problèmes de cellules.
- CREMK calcule les matrices locales de rigidité et de masse.
- CREASSEMBLAGE assemble et résout le problème grossier par une méthode d'Éléments Finis. La résolution du système linéaire se fait par le biais de la librairie NUMERICAL PLATON.
- CALSOLUTIONMICRO reconstruit, à partir de la solution grossière, la solution fine sur tout ou partie du domaine de travail Ω .
- CALHOMOGENE regroupe plusieurs routines permettant de calculer la diffusivité équivalente du milieu, telle que définie par (1.4).

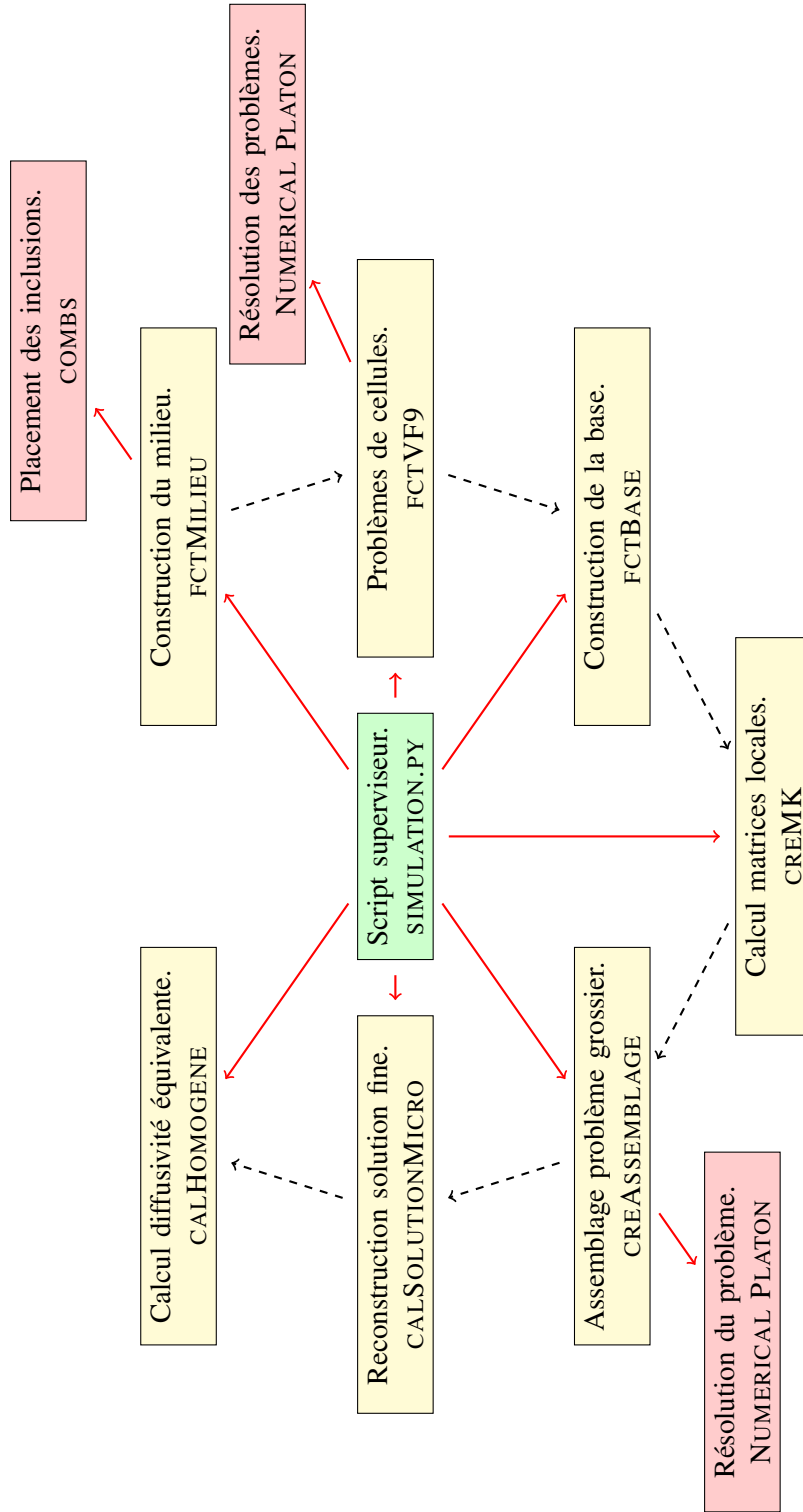


FIGURE 4.1 – Organigramme simplifié de la maquette Python implémentant la méthode $Q_1/VF9$. Les modules extérieurs au modèle sont représentés en rouge, ceux implémentés au cours de ces travaux sont en jaune. On représente en pointillés la *chaîne de calculs* qui forme la méthode multi-échelle. Les traits continus représentent une dépendance entre modules.

La Figure 4.1 présente l'organigramme liant ces modules. Y apparaissent en jaune les modules développés spécifiquement pour la méthode multi-échelle. Les modules indépendants, fruits d'un travail extérieur à ces travaux de thèse, sont représentés en rouge.

Dans la suite de cette section, on s'applique à valider l'implémentation de la méthode $Q_1/VF9$. On se concentre sur trois modules en particulier : FCTVF9, CREMK et CREASSEMBLAGE. Le module FCTMILIEU est explicité en détail à la section §4.3, en même temps que la modélisation des matériaux cimentaires. L'implémentation des autres modules ne révèle pas de difficultés particulières.

4.1.1 Qualification du module FCTVF9.

On valide dans cette section l'implémentation du schéma $VF9$. On résout pour cela plusieurs exemples du problème (3.8), que l'on reporte ici :

$$\begin{cases} -\nabla \cdot (A\nabla u) = 0 & \text{dans } \Delta, \\ u = u_D & \text{sur } \partial\Delta. \end{cases} \quad (4.1)$$

Δ est ici le domaine $]0, 1[\times]0, 1[$. u_D représente les conditions de Dirichlet du problème.

Pour chaque exemple, on travaille sur une série de discrétisations $(T_h(\Delta))_{h>0}$, où $T_h(\Delta)$ désigne un découpage du domaine Δ en carrés de côté h . On présente alors l'erreur L^2 discrète entre u_h la solution du problème approchée sur $T_h(\Delta)$ et u la solution exacte du problème (4.1). Si on note x_k le barycentre de l'élément $k \in T_h(\Delta)$, cette erreur est définie par :

$$e_2(u_h) = h \sqrt{\sum_{k \in T_h(\Delta)} (u_h(k) - u(x_k))^2} \quad (4.2)$$

Pour exprimer les taux de convergence, on utilise la notation de Landau $\mathcal{O}(h)$ pour désigner une fonction d'ordre h quand h tend vers 0.

La résolution des systèmes linéaires s'effectue *via* la librairie *Numerical Platon* [167], elle-même appelant la bibliothèque *PetsC* [154]. Plus précisément, on utilise le solveur itératif GMRES [161, 162] assorti d'un préconditionneur ILU [65].

Exemple 4.1 *On considère l'exemple défini par :*

$$A = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

$$u_D = xy$$

où a et b sont des réels strictement positifs. La solution du problème est alors :

$$u = xy$$

Le schéma *VF9* étant linéaire exact, on retrouve ici la solution exacte, à la précision machine près.

Exemple 4.2 On considère l'exemple défini par :

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$u_D = \sin(\pi x) \sinh(\pi y)$$

La solution du problème est alors :

$$u = \sin(\pi x) \sinh(\pi y)$$

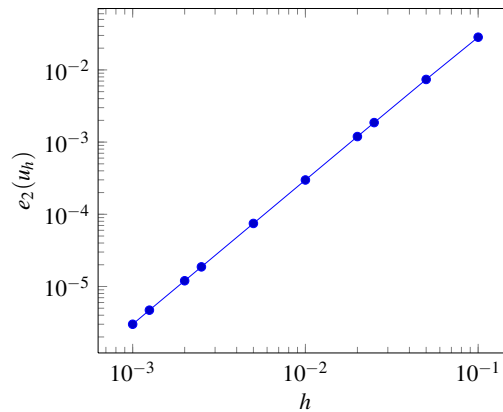


FIGURE 4.2 – Exemple 4.2 : représentation de $e_2(u_h)$ en fonction de h .

Les résultats de ce test sont présentés à la Figure 4.2. Ils montrent une convergence en $\mathcal{O}(h^{2.5})$, soit un ordre au-dessus de la convergence théorique de la méthode Volumes Finis *VF9*. Cette *super convergence* est habituelle lors de l'utilisation des Volumes Finis *VF9* sur un milieu homogène [97]. Il ne s'agit cependant pas d'une particularité de la méthode *VF9* mais plutôt des Volumes Finis en général. Le phénomène de super convergence est en effet observable chez un grand nombre de méthodes Volumes Finis [114].

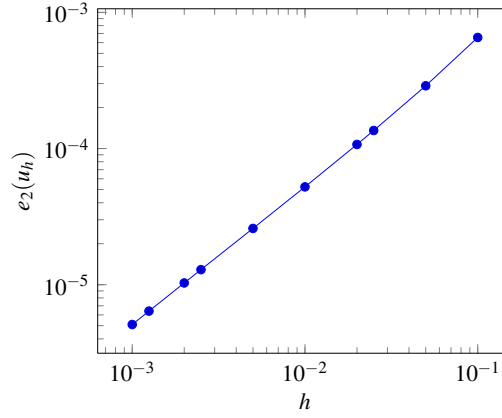
Exemple 4.3 On considère l'exemple défini par :

$$A = \begin{bmatrix} x^2 - y^2 + 3 & 0 \\ 0 & x^2 - y^2 + 3 \end{bmatrix}$$

$$u_D = \frac{1}{2} \ln(x^2 - y^2 + 3)$$

La solution du problème est alors connue :

$$u = \frac{1}{2} \ln(x^2 - y^2 + 3)$$

FIGURE 4.3 – Exemple 4.3 : représentation de $e_2(u_h)$ en fonction de h .

Les résultats de ce test sont présentés à la Figure 4.3. Il y apparaît clairement que la solution approchée converge vers la solution théorique, et ce avec un taux d'erreur en $\mathcal{O}(h)$.

En conclusion des bons résultats obtenus sur ces trois exemples, on peut considérer que la méthode *VF9* a été correctement implémentée au sein du module *FCTVF9*.

4.1.2 Qualification du module CREMK.

On valide dans cette section l'implémentation du module *CREMK* qui calcule les différents termes des matrices locales Éléments Finis : la matrice de raideur \mathbb{K}_K , la matrice de masse \mathbb{M}_K et la matrice de masse de bord $\mathbb{M}_{b,F}$, pour un élément $K \in T_H(\Omega)$ et une face $F \in \Gamma_{N,H}$ donnés. Ces quantités sont définies par :

$$\forall 1 \leq i, j \leq 4 \quad \mathbb{K}_K(i, j) = \frac{1}{2} \sum_{f \in \partial K} \left(\Phi_K^i(f) \int_f D\nabla \Phi_K^j \cdot \mathbf{n} + \Phi_K^j(f) \int_f D\nabla \Phi_K^i \cdot \mathbf{n} \right) \quad (4.3)$$

$$\forall 1 \leq i, j \leq 4 \quad \mathbb{M}_K(i, j) = \sum_{k \in K} |k| \Phi_K^i(k) \Phi_K^j(k) \quad (4.4)$$

$$\forall 1 \leq i, j \leq 4 \quad \mathbb{M}_{b,F}(i, j) = \sum_{f \in F} |f| \Phi_K^i(f) \Phi_K^j(f) \quad (4.5)$$

où $(\Phi_K^i)_{1 \leq i \leq 4}$ est la base d'Éléments Finis construite à partir des fonctions $(\Psi_K^i)_{1 \leq i \leq 4}$, solutions sur \hat{K} des problèmes de cellules. Une description précise de ces équations et un rappel des différentes notations est disponible à la section §3.3.1.

Sur une face $f \in \partial K$, le calcul de la matrice de raideur et la matrice de masse de bord fait intervenir deux quantités : $(\Phi_K^i(f))_{1 \leq i \leq 4}$ les valeurs interpolées en f et $(\int_f D\nabla \Phi_K^i \cdot \mathbf{n})_{1 \leq i \leq 4}$ les flux à travers f . La définition de ces deux quantités est étroitement liée à la position de f vis-à-vis de la cellule \hat{K} , le domaine de résolution des problèmes de cellules. Il est donc nécessaire de tester l'implémentation de ce module dans le cas où f est une face de la frontière $\partial \hat{K}$, c'est-à-dire quand le taux de sur-échantillonnage ρ est nul, et dans le cas où f est une face intérieure à la cellule \hat{K} , i.e. ρ est strictement positif.

Exemple 4.4 *On considère dans cet exemple que :*

$$D = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

$$K =]0, 1[\times]0, 1[$$

$$F = \{y = 0\}$$

où a et b sont des réels strictement positifs. Dans ce cas, les fonctions $(\Phi_K^i)_{1 \leq i \leq 4}$ sont explicitement connues, et ce pour toute valeur de ρ :

$$\Phi_K^1 = (1-x)(1-y)$$

$$\Phi_K^2 = x(1-y)$$

$$\Phi_K^3 = (1-x)y$$

$$\Phi_K^4 = xy$$

Les matrices \mathbb{K}_K , \mathbb{M}_K et $\mathbb{M}_{b,F}$ sont symétriques. Étant donnée la symétrie des fonctions $(\Phi_K^i)_{1 \leq i \leq 4}$, on a de plus :

$$\mathbb{K}_K(1,1) = \mathbb{K}_K(2,2) = \mathbb{K}_K(3,3) = \mathbb{K}_K(4,4),$$

$$\mathbb{K}_K(1,2) = \mathbb{K}_K(1,3) = \mathbb{K}_K(2,4) = \mathbb{K}_K(3,4),$$

$$\mathbb{K}_K(1,4) = \mathbb{K}_K(2,3),$$

des formules semblables existant pour les matrices \mathbb{M}_K et $\mathbb{M}_{b,F}$. Les matrices locales n'ont donc que trois éléments différents, correspondant aux coefficients $(1,1)$, $(1,2)$ et $(1,4)$.

Les Figures 4.4, 4.5 et 4.6 présentent l'erreur entre les valeurs théoriques et numériques des matrices locales sur ces trois coefficients. Il apparaît des simulations que les erreurs décroissent en $\mathcal{O}(h^2)$. Le calcul de la base locale $(\Phi_K^l)_{1 \leq l \leq n}$ à partir des fonctions $(\Psi_{\hat{K}}^l)_{1 \leq l \leq n}$ étant linéaire exact, l'erreur est indépendante de ρ .

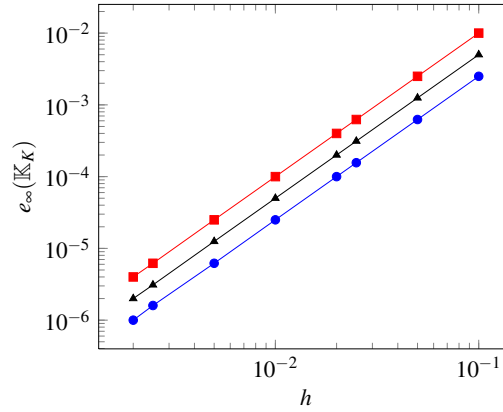


FIGURE 4.4 – Exemple 4.4 : erreur de calcul, en fonction de h , des coefficients $\mathbb{K}_K(1,1)$ (—●—), $\mathbb{K}_K(1,2)$ (—■—) et $\mathbb{K}_K(1,4)$ (—▲—). L'erreur décroît en $\mathcal{O}(h^2)$ et est indépendante de ρ .

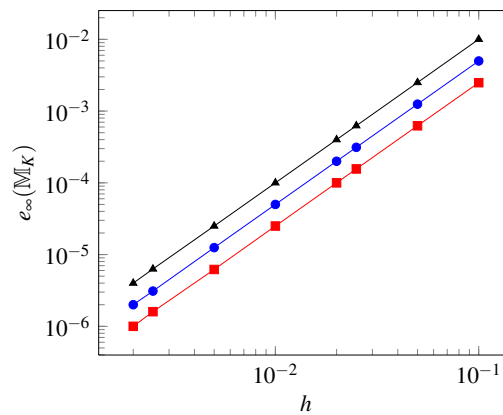


FIGURE 4.5 – Exemple 4.4 : erreur de calcul, en fonction de h , des coefficients $\mathbb{M}_K(1,1)$ (—●—), $\mathbb{M}_K(1,2)$ (—■—) et $\mathbb{M}_K(1,4)$ (—▲—). L'erreur décroît en $\mathcal{O}(h^2)$ et est indépendante de ρ .

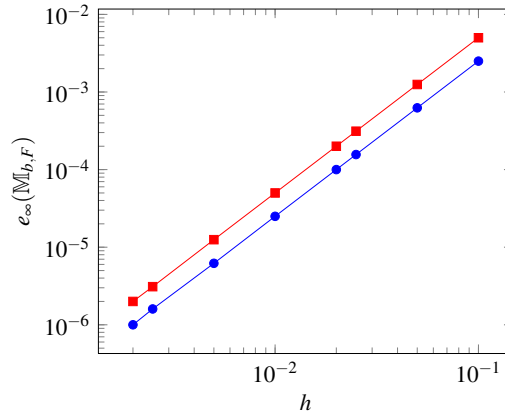


FIGURE 4.6 – Exemple 4.4 : erreur de calcul, en fonction de h , des coefficients $\mathbb{M}_{b,F}(1,1)$ (—●—) et $\mathbb{M}_{b,F}(1,2)$ (—■—). L'erreur décroît en $\mathcal{O}(h^2)$ et est indépendante de ρ .

Exemple 4.5 On reprend dans cet exemple la solution de l'Exemple 4.2.

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$K =]0, 1[\times]0, 1[$$

$$F = \{y = 1\}$$

$$\beta = \sin(\pi x) \sinh(\pi y)$$

La solution du problème est alors :

$$\Psi = \sin(\pi x) \sinh(\pi y)$$

On peut alors calculer les termes (4.3)-(4.5) en posant $\Phi_K^i = \Phi_K^j = \Psi$. On présente à la Figure 4.7 l'erreur entre valeurs théoriques et numériques de ces trois termes en fonction de h . Les coefficients $\mathbb{K}(\Psi, \Psi)$ et $\mathbb{M}(\Psi, \Psi)$ convergent respectivement en $\mathcal{O}(h)$ et $\mathcal{O}(h^2)$. Le coefficient $\mathbb{M}_b(\Psi, \Psi)$ est calculé de manière exacte.

4.1.3 Qualification du module CREASSEMBLAGE.

On valide dans cette section l'implémentation du module CREASSEMBLAGE qui assemble et résout le problème d'Éléments Finis grossier. Il ne s'agit pas ici de tester la méthode multi-échelle $Q_1/VF9$ dans son intégralité, mais seulement d'assurer que la méthode d'Éléments Finis a été correctement implémentée. Pour cela, on suppose que la diffusivité D est égale à la matrice *Identité*, ce qui correspond au cadre de l'Exemple 4.4 avec $(a, b) = (1, 1)$. La base $(\Phi^I)_{I \in \mathcal{I}}$ est alors explicitement connue, et correspond aux Éléments Finis Q_1 classiques.

On va maintenant résoudre plusieurs exemples du problème de diffusion suivant :

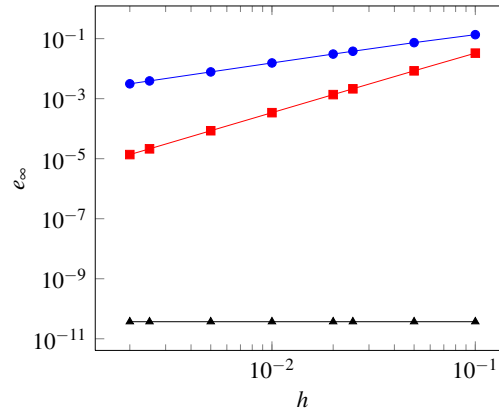


FIGURE 4.7 – Exemple 4.5 : erreur de calcul, en fonction de h , des coefficients $\mathbb{K}(\Psi, \Psi)$ (—●—), $\mathbb{M}(\Psi, \Psi)$ (—■—) et $\mathbb{M}_b(\Psi, \Psi)$ (—▲—) pour $\rho = 0$.

$$\begin{cases} -\Delta C = f & \text{dans } \Omega, \\ C = g_D & \text{sur } \Gamma_D, \\ D \nabla C \cdot \mathbf{n} = g_N & \text{sur } \Gamma_N, \end{cases} \quad (4.6)$$

où (Γ_D, Γ_N) est une partition de $\Gamma = \partial\Omega$ la frontière du domaine Ω . f , g_D et g_N sont respectivement le terme source, la condition aux limites de Dirichlet et la condition aux limites de Neumann.

Dans tout ce qui suit, on suppose que $\Omega =]0, 1[\times]0, 1[$. La discrétisation $T_H(\Omega)$ de Ω se compose de macroéléments carrés de longueur H .

Exemple 4.6 On considère dans cet exemple que :

$$\begin{aligned} f &= 0 \\ \Gamma_D &= \partial\Omega \\ g_D &= \sin(\pi x) \sinh(\pi y) \end{aligned}$$

La solution du problème est alors :

$$C = \sin(\pi x) \sinh(\pi y)$$

Exemple 4.7 On considère dans cet exemple que :

$$\begin{aligned} f &= 0 \\ \Gamma_D &= \{0, 1\} \times [0, 1] \\ g_D &= \sin(\pi x) \sinh(\pi y) \\ \Gamma_N &= [0, 1] \times \{0, 1\} \\ g_N &= \frac{\partial}{\partial \mathbf{n}} (\sin(\pi x) \sinh(\pi y)) \end{aligned}$$

4.2. VALIDATION DE LA MÉTHODE $Q_1/VF9$ SUR UN EXEMPLE THÉORIQUE.83

La solution du problème est alors :

$$C = \sin(\pi x) \sinh(\pi y)$$

On choisit volontairement de se restreindre en n'imposant les conditions de Neumann qu'en deux bords disjoints de Ω afin de s'affranchir des problèmes informatiques liés aux coins du domaine.

Exemple 4.8 On considère dans cet exemple que :

$$\begin{aligned} f &= \sin(\pi x) \sin(\pi y) \\ \Gamma_D &= \partial\Omega \\ g_D &= 0 \end{aligned}$$

La solution du problème est alors :

$$C = \frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y)$$

Le critère de validation choisit ici est l'erreur maximale entre valeurs calculées et théoriques aux noeuds $I \in \mathcal{I}$:

$$e_\infty(C_H) = \max_{I \in \mathcal{I}} |C_H(I) - C(I)| \quad (4.7)$$

Il ne s'agit pas du maximum de l'erreur entre solution théorique et calculée car on ne recalcule pas la solution $C_{H,h}$ au niveau fin. De cette manière, la discrétisation des macroéléments K n'influence le résultat que par le biais des matrices de raideurs et de masses, influence que l'on néglige dans cette partie.

On présente à la Figure 4.8 les courbes d'erreurs des Exemples 4.6, 4.7 et 4.8 en fonction de H .

4.2 Validation de la méthode $Q_1/VF9$ sur un exemple théorique.

4.2.1 Convergence théorique.

On se place dans cette section dans le cadre de l'homogénéisation théorique périodique. Soit d une fonction périodique sur $Y = [0, 1]^2$, et $\varepsilon > 0$ un paramètre d'échelle, D est défini par :

$$D^\varepsilon = d\left(\frac{x}{\varepsilon}\right)$$

Le problème de diffusion à résoudre est alors :

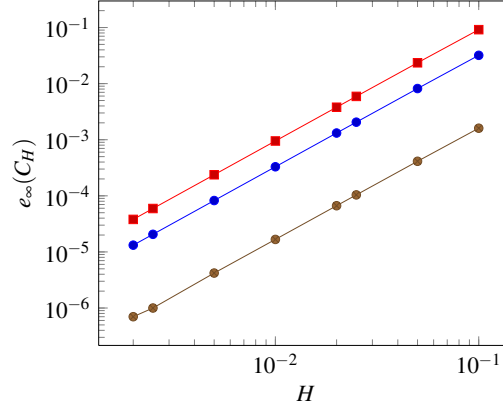


FIGURE 4.8 – Erreur $e_\infty(C_H)$ en fonction de H pour les Exemples 4.6 (—●—), 4.7 (—■—) et 4.8 (—●—).

$$\begin{cases} -\nabla \cdot (D^\varepsilon \nabla C^\varepsilon) = f & \text{dans } \Omega, \\ C^\varepsilon = g_D & \text{sur } \Gamma_D, \\ D^\varepsilon \nabla C^\varepsilon \cdot \mathbf{n} = g_N & \text{sur } \Gamma_N, \end{cases} \quad (4.8)$$

Comme on l'a défini au chapitre §3, (Γ_D, Γ_N) forme une partition de $\Gamma = \partial\Omega$ la frontière du domaine Ω . f , g_D et g_N sont respectivement le terme source, la condition aux limites de Dirichlet et la condition aux limites de Neumann du problème.

Dans ce cadre, il est possible d'estimer l'erreur entre C^ε , la solution théorique du problème (4.8) et $C_{H,h}^\varepsilon$, la solution numérique obtenue par la méthode multi-échelle $Q_1/VF9$ [28, 90, 120, 121].

Pour $h < \varepsilon < H$, on a :

$$\|C^\varepsilon - C_{H,h}^\varepsilon\|_2 \leq A_1 \left(H + \sqrt{\frac{\varepsilon}{H} + \frac{h}{\varepsilon}} \right) \quad \text{si } \rho = 0, \quad (4.9)$$

$$\|C^\varepsilon - C_{H,h}^\varepsilon\|_2 \leq A_2 \left(H + \frac{\varepsilon}{H} + \frac{h}{\varepsilon} \right) \quad \text{si } \rho > 0. \quad (4.10)$$

A_1 et A_2 sont des constantes ne dépendant d'aucun des paramètres H , ε et h .

L'influence du taux de sur-échantillonnage ρ sur A_2 est mal connu. Les travaux de *Efendiev et al.* [90] donnent à penser qu'on peut réécrire (4.10) sous la forme :

$$\|C^\varepsilon - C_{H,h}^\varepsilon\|_2 \leq A_{2,1}H + \frac{A_{2,2}}{\rho} \frac{\varepsilon}{H} + A_{2,3} \frac{h}{\varepsilon} \quad (4.11)$$

mais le problème reste le même, puisque la dépendance en ρ des constantes $A_{2,1}$, $A_{2,2}$ et $A_{2,3}$ n'est pas connue. Au vu des simulations numériques sur le sujet, il semble cependant que $A_{2,1}$, $A_{2,2}/\rho$ et $A_{2,3}$ tendent vers des constantes non nulles quand ρ augmente [90, 120, 121].

4.2. VALIDATION DE LA MÉTHODE $Q_1/VF9$ SUR UN EXEMPLE THÉORIQUE.85

Pour étudier la convergence de la méthode $Q_1/VF9$, il est donc nécessaire de faire tendre H , ε et h vers 0, tout en supposant que l'on a :

$$h \ll \varepsilon \ll H \quad (4.12)$$

Il est très difficile de remplir numériquement l'ensemble de ces conditions. C'est pourquoi les résultats présentés dans les sections suivantes se concentrent sur l'évolution de l'erreur quand *un seul* paramètre tend vers 0, les autres étant constants. Les courbes présentées ne sont donc en aucun cas des courbes de *convergence*.

4.2.2 Validation.

On présente ici les résultats de la méthode $Q_1/VF9$ sur un exemple théorique simple. Le but premier est ici de retrouver le comportement attendu de la méthode des Éléments Finis multi-échelles, tel que décrit à la section précédente. C'est pourquoi le milieu étudié ne présente qu'une variation modérée de la diffusivité : il n'est pas question ici de tester la robustesse de la méthode.

Exemple 4.9 On résout sur $\Omega = [0, 1]^2$ le problème suivant :

$$\begin{cases} -\nabla(D\nabla C) = f & \text{dans } \Omega, \\ C = 0 & \text{sur } \Gamma_D = \partial\Omega. \end{cases} \quad (4.13)$$

f est une fonction en créneau, c'est-à-dire égale à 1 au sein de la sphère de centre $(x_0, y_0) = (0.2, 0.4)$ et de rayon 0.05 et nulle partout ailleurs.

Avec $p = 50$, on pose :

$$D(x, y) = (1.1 + \sin(2\pi px) \sin(2\pi py)) I_d \quad (4.14)$$

Le champs de diffusivité est représenté sur une portion du domaine Ω à la Figure 4.9 (à gauche).

On divise le milieu Ω en $M = m \times m$ macro-éléments K carrés de côté $H = 1/m$, formant la discrétisation grossière $T_H(\Omega)$. Chaque macro-élément K est ensuite divisé en $N = n \times n$ micro-éléments carrés de côté $h = 1/(mn)$, formant la discrétisation $T_h(K)$. Avec ρ le taux de sur-échantillonnage, la discrétisation $T_h(\hat{K})$ d'une cellule \hat{K} comporte alors $(1 + 2\rho)^2 n^2$ micro-éléments.

Afin d'obtenir une solution de référence C_r , on résout le problème (4.13) via une méthode directe de Volumes Finis à 4 millions d'éléments. En posant $h_r = 5.10^{-4}$, la discrétisation $T_{h_r}(\Omega)$ du milieu Ω comporte donc 2000×2000 micro-éléments. Une représentation de C_r est présentée à la Figure 4.9 (à droite).

On compare ensuite les solutions numériques $C_{H,h}$ de l'Exemple 4.9, comportant $m^2 \times n^2$ micro-éléments, à C_r . Soit $k_r \in T_{h_r}(\Omega)$, pour tout $h > h_r$, on définit $\Upsilon_h(k_r)$ l'ensemble des éléments $k \in T_h(K)$ ayant une intersection non-vide avec k_r :

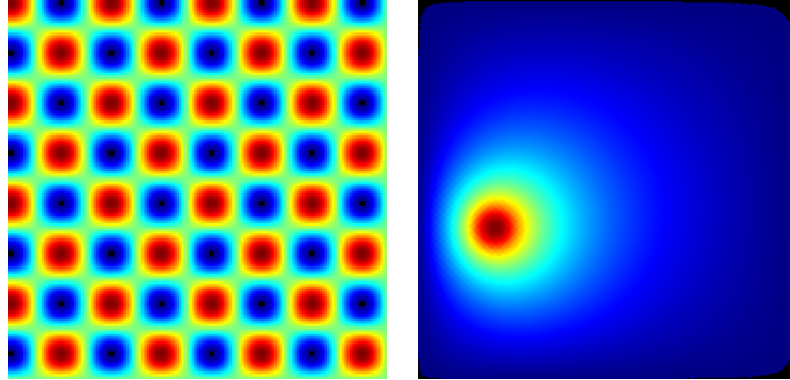


FIGURE 4.9 – Champs de diffusivité (zoom) et solution de référence C_r pour l'Exemple 4.9.

$$\Upsilon_h(k_r) = \{k \in T_h(K), K \in T_H(\Omega), k \cap K \neq \emptyset\} \quad (4.15)$$

On note $n_h(k_r)$ le cardinal de $\Upsilon_h(k_r)$. L'erreur L_2 discrète est alors calculée par :

$$e_2(C_{H,h}) = h_r \sqrt{\sum_{k_r \in T_{h_r}(\Omega)} \left[C_r(k_r) - \frac{1}{n_h(k_r)} \sum_{k \in \Upsilon_h(k_r)} C_{H,h}(k) \right]^2} \quad (4.16)$$

On peut définir une quantité équivalente sur la densité de flux $D\nabla C_{H,h}$:

$$e_2(D\nabla C_{H,h}) = h_r \sqrt{\sum_{k_r \in T_{h_r}(\Omega)} \left[D_r \nabla C_r(k_r) - \frac{1}{n_h(k_r)} \sum_{k \in \Upsilon_h(k_r)} D_h \nabla C_{H,h}(k) \right]^2} \quad (4.17)$$

La Figure 4.10 présente les courbes d'erreurs sur la concentration $C_{H,h}$ et la densité de flux $D\nabla C_{H,h}$ en fonction de h , pour différentes valeurs fixées de H . On note que les deux erreurs stagnent rapidement. Diminuer sans fin le pas de résolution des problèmes locaux est donc inutile si le nombre de degré de liberté du maillage grossier (i.e $m = 1/H$) est limité. Dans les deux cas présentés, on note que $h = 1e - 3$ suffit amplement à atteindre la zone limite.

Bien que les erreurs en elles-mêmes augmentent avec ρ , le comportement générale de l'erreur reste lui pratiquement inchangé. .

Sur la Figure 4.11, on montre cette fois l'évolution de l'erreur quand H diminue, en imposant $h = 0.02H$ (à gauche) ou $h = 0.01H$ (à droite). Il apparaît que l'erreur n'est pas une fonction décroissante de H , mais qu'elle passe par un minimum avant de remonter. En effet, si H devient trop petit, la taille des problèmes de cellules

4.2. VALIDATION DE LA MÉTHODE $Q_1/VF9$ SUR UN EXEMPLE THÉORIQUE.87

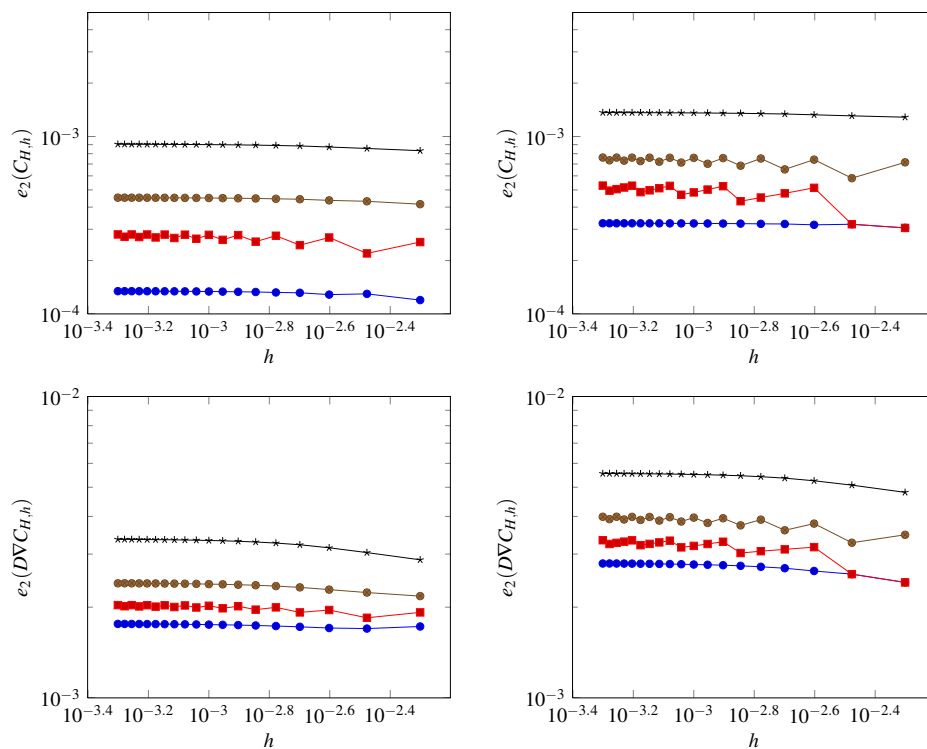


FIGURE 4.10 – Exemple 4.9 : évolution des erreurs $e_2(C_{H,h})$ (en haut) et $e_2(D\nabla C_{H,h})$ (en bas) pour $H = 0.01$ (gauche) et $H = 0.05$ (droite). Sur chaque courbe, on a tracé l'erreur en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—).

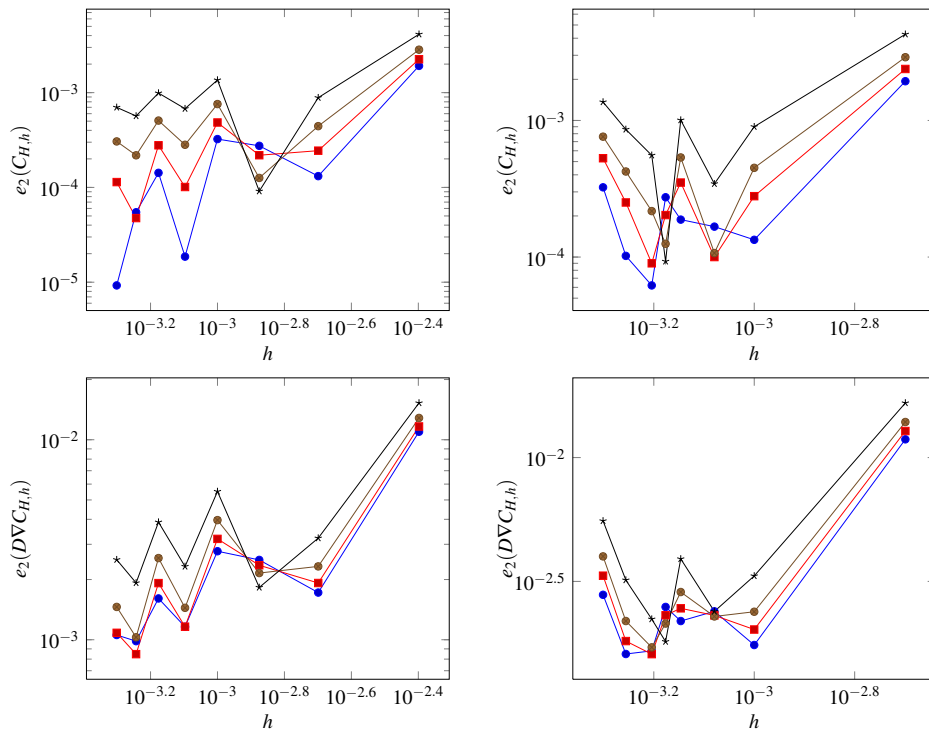


FIGURE 4.11 – Exemple 4.9 : évolution des erreurs $e_2(C_{H,h})$ (en haut) et $e_2(D^2 \nabla C_{H,h})$ (en bas) pour $h = 0.02H$ (gauche) et $h = 0.01H$ (droite). Sur chaque courbe, on a tracé l'erreur en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—).

devient trop petite par rapport aux oscillations du milieu et l'homogénéisation de la cellule n'est plus possible. Il faut en effet que chaque cellule englobe 4 à 5 cellules périodiques du milieu pour que la méthode soit efficace. Dans le cas de l'Exemple 4.9, cela revient à choisir H supérieur à 0.08.

Remarque 4.1 *Comme on l'a expliqué à la section §4.2.1, les courbes présentées ici ne sont pas des courbes de convergence de la méthode $Q_1/VF9$, mais des courbes d'évolution de l'erreur en fonction d'un paramètre (h , H ou $\alpha = h/H$), les autres étant fixés.*

Ces simulations valident correctement l'implémentation Python de la méthode $Q_1/VF9$. Si on fait varier l'une ou l'autre des deux valeurs h et H , l'erreur diminue jusqu'à atteindre un minimum qui dépend du second paramètre resté fixe. Le sur-échantillonnage ρ , quant à lui, permet de diminuer les valeurs maximales des erreurs, mais au prix d'une moins bonne précision, surtout lorsque le milieu est grossièrement discrétisé («grandes» valeurs de H et h .) Lorsque H et h sont suffisamment petits, par contre, les résultats sont légèrement en faveur du sur-échantillonnage.

4.3 Application aux matériaux cimentaire.

4.3.1 Génération d'un matériau cimentaire via le module COMBS.

Comme on l'a décrit à la section §3.6, les matériaux cimentaires sont modélisés dans le cadre des travaux de cette thèse par un ensemble d'inclusions de formes diverses enchâssées dans un milieu homogène. Pour construire un échantillon d'un matériau cimentaire, il est donc nécessaire de placer ces inclusions dans le milieu. Un algorithme spécifique, le module COMBS a été développé par *Erwan Adam* afin de s'acquitter de cette tâche.

Soit une liste détaillée, précisant la taille, la forme et le nombre des inclusions voulues, le module COMBS va placer aléatoirement ces inclusions au sein du domaine. En sortie du module, on obtient un fichier décrivant chaque inclusion et sa position. À chaque inclusion est également associé le nom de la phase à laquelle elle appartient.

Il est possible de préciser plusieurs options, en fonction du milieu que l'on souhaite générer. On peut ainsi demander au module de construire un milieu périodique ; les inclusions débordant d'un côté du domaine réapparaissant de l'autre ; ou au contraire interdire aux inclusions d'approcher les frontières du domaine. Il est de plus possible de paramétrer la distance minimale séparant deux inclusions.

Enfin, il arrive que le module n'arrive pas à placer toutes les inclusions voulues, soit parce que le domaine est plein, soit parce que la place libre est «difficile» à trouver. Dans ce dernier cas, l'expérience montre que l'on peut espérer atteindre un taux de remplissage de près de 80% en volume si on laisse suffisamment de temps à l'algorithme (généralement plus de 12h).

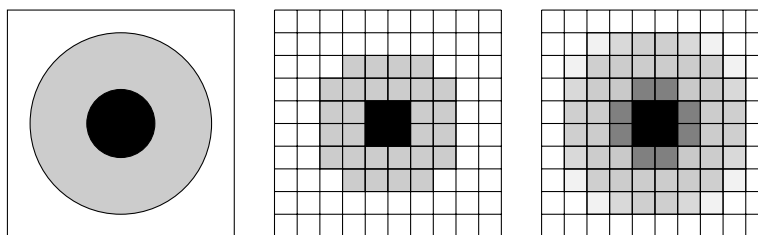


FIGURE 4.12 – Pour un milieu donné (à gauche), deux méthodes d’attribuer une diffusivité aux éléments du maillage fin : par *dichotomie* (au milieu) et par *moyenne* (à droite).

La première version de ce module ne permettait que de placer des inclusions sphériques à une ou plusieurs couches [21]. C’est cette version qui est utilisée, dans le cadre $2D$, pour générer les milieux utilisés par l’implémentation présentée dans ce chapitre. Les dernières versions, en faisant appel à la plate-forme *SALOME* [163], permettent maintenant de placer des formes quelconques en deux et trois dimensions [70].

4.3.2 Discrétisation du milieu.

4.3.2.1 Principes.

Comme on l’a précisé au début de ce chapitre, la maquette Python présentée ici travaille exclusivement avec des maillages rectangulaires réguliers en dimension deux aux deux niveaux : grossier et fin. A l’échelle fine, une cellule \hat{K} rectangulaire est donc découpée en rectangles k et une unique valeur de diffusivité est attribuée à chaque rectangle. Cette discrétisation du milieu est très pratique, puisque les connectivités entre les divers éléments (sommets/éléments, éléments/éléments, arêtes/éléments . . .) sont explicitement connus. Un problème se pose cependant dans le cas où l’on veut discrétiser des inclusions sphériques ou en couronne par un ensemble de rectangles. La question est de savoir quelle valeur de diffusivité attribuer aux rectangles à cheval sur la *matrice* et une, voire plusieurs, inclusions.

Dans un premier temps, on a choisi de départager le maillage par *dichotomie* : seuls les rectangles inclus dans une inclusion se voyant attribuer la diffusivité de l’inclusion. Les autres, partiellement dans l’inclusion ou non, sont considérés comme appartenant à la matrice, et reçoivent la diffusivité correspondante. Un exemple de milieu discrétisé par dichotomie est visible à la Figure 4.12, au milieu.

Cette caractérisation a pour elle le mérite de la simplicité, mais ne permet qu’une description grossière du milieu. En effet, cette méthode découpe les inclusions, les réduisant à leurs seuls rectangles internes, ce qui entraîne une perte de masse. C’est particulièrement flagrant dans le cas des petites inclusions, dont la taille est de l’ordre du pas du maillage. Les inclusions se retrouvent alors mo-

délimitées par un simple rectangle, ou encore, pour la taille juste supérieure, par une croix.

Il est également impossible de représenter correctement deux inclusions très proches l'une de l'autre. Quelle que soit la distance réelle entre deux inclusions, cette méthode impose une distance effective égale au pas h du maillage fin $T_h(\hat{K})$.

Pour toutes ces raisons, une discrétisation par *moyenne* a été développée. On attribue, pour chaque microélément k , une valeur de diffusivité au prorata des aires des différentes phases (matrice, inclusions) le traversant.

Soit $(\phi_i)_{1 \leq i \leq p}$ l'ensemble des phases d'un rectangle k , $(D_i)_{1 \leq i \leq p}$ les diffusivités correspondantes, et $(a_i)_{1 \leq i \leq p}$ l'aire occupée par ces phases au sein du rectangle. On définit alors D_k la diffusivité du rectangle par moyenne arithmétique :

$$D_k = \frac{1}{\sum_{1 \leq i \leq p} a_i} \sum_{1 \leq i \leq p} a_i D_i \quad (4.18)$$

Un exemple de milieu discrétisé par filtrage est visible à la Figure 4.12, à droite.

Il est à noter que cette méthode n'est utilisable que dans le cadre particulièrement avantageux où l'on se trouve. Il suffit ici de déterminer les aires d'intersections entre des sphères et des rectangles, ce qui est explicité dans la suite de cette section. Le problème est autrement plus difficile en 3 dimension, ou si la forme des inclusions n'est pas aussi simple. C'est pour cette raison que cette technique n'est pas utilisée au sein de l'implémentation *MPCube*, comme on l'explique au chapitre §6.

4.3.2.2 Caractérisation de l'intersection d'une sphère et d'un rectangle.

Soit C un cercle de rayon r , R un rectangle de dimensions (h_x, h_y) et $I(R, C)$ leur intersection éventuellement vide, dont on cherche à calculer l'aire. On décompose $\mathcal{S}(R)$, l'ensemble des sommets du rectangle R , en $\mathcal{S}_i(R)$ et $\mathcal{S}_e(R)$, sous-ensembles des sommets intérieurs stricts, respectivement extérieurs large, au cercle C . On note $\mathcal{A}(R)$ l'ensemble des arêtes de R , et $\mathcal{I}(a)$ l'ensemble, éventuellement vide, des points d'intersection de l'arête $a \in \mathcal{A}(R)$ et du cercle C .

On pose :

$$q = \text{Card}(\mathcal{S}_i(R)) \quad (4.19)$$

On caractérise alors l'intersection $I(R, C)$ en fonction des valeurs de q . Dans le cas où $q = 0$, c'est-à-dire quand le cercle C ne contient aucun sommet de R , on distingue deux cas, selon que le centre de C soit dans R ou non. Au total, on distingue donc six types d'intersection qui sont présentés à la Figure 4.13.

Chaque intersection est ensuite découpée en un ensemble de triangles rectangles, de trapèzes et de cordes. Les aires de ces figures géométriques sont facilement

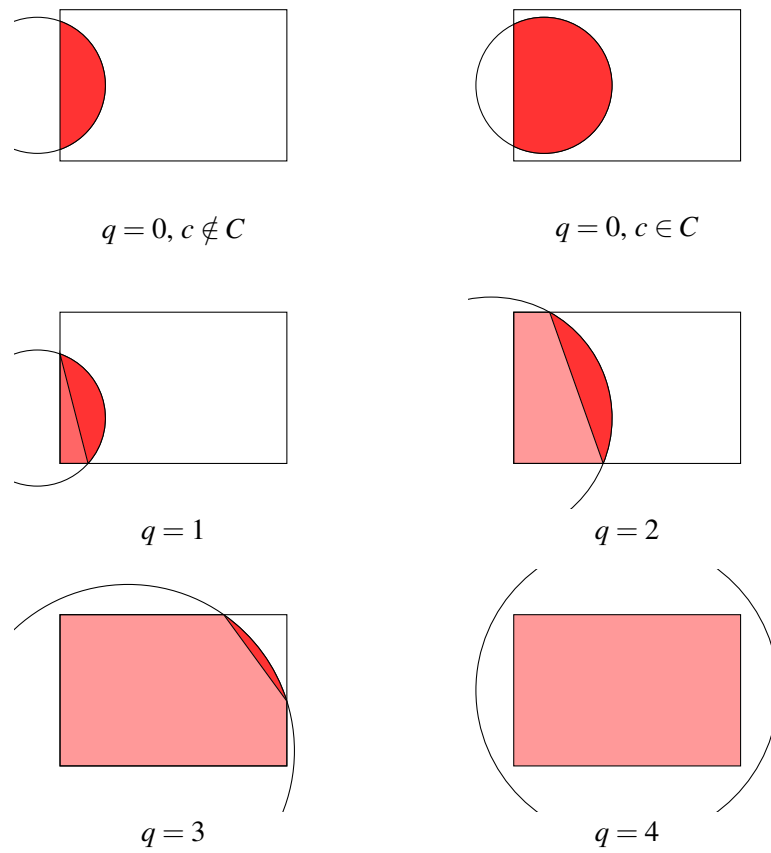


FIGURE 4.13 – Intersections d'un cercle C et d'un rectangle R en fonction du nombre q de sommets R à l'intérieur de C . Les zones délimitées par l'Algorithme 4.1 afin de calculer l'aire d'intersection apparaissent en différentes teintes de rouge.

calculables et sont rappelées à la Figure 4.14. Finalement, l'aire de l'intersection $I(R, C)$ s'obtient par sommation, ou retranchement, des aires des zones concernées.

Algorithme 4.1 Aire d'intersection entre le cercle C de rayon r et le rectangle R de côté (h_x, h_y) .

```

1: Si  $q = 0$  alors
2:   Si  $C \in R$  alors
3:      $Aire \leftarrow \pi r^2 - \sum_{a \in \mathcal{A}(R)} C_r(r, l_i(a))$ .
4:   sinon
5:      $Aire \leftarrow \sum_a C_r(r, l_i(a))$ .
6:   Fin Si
7: sinon Si  $q = 1$  alors
8:    $Aire \leftarrow T(l_i(a_1), l_i(a_4)) + C_r(r, L) - C_r(r, l_i(a_2)) - C_r(r, l_i(a_4))$ .
9: sinon Si  $q = 2$  alors
10:   $Aire \leftarrow T_r(l_i(a_1), l_i(a_3), l_i(a_4)) + C_r(r, L) - C_r(r, l_i(a_2))$ .
11: sinon Si  $q = 3$  alors
12:   $Aire \leftarrow h_x h_y - T(l_e(a_1), l_e(a_2)) + C_r(r, L)$ .
13: sinon Si  $q = 4$  alors
14:   $Aire \leftarrow h_x h_y$ .
15: Fin Si

```

D'un point de vue algorithmique, afin de calculer proprement ces surfaces, on commence par ordonner les sommets et les arêtes de R selon le sens trigonométrique. On fait ensuite de même avec les points $(\mathcal{S}(a))_{a \in \mathcal{A}(R)}$, en prenant pour origine l'arête sortante de C , c'est-à-dire l'arête $a = [S_1 S_2] \in \mathcal{A}(R)$ telle que :

$$S_1 \in \mathcal{S}_i(R) \quad \text{et} \quad S_2 \in \mathcal{S}_e(R) \quad (4.20)$$

Dans les cas $n = 0$ et $n = 4$, aucune arête ne satisfait cette condition. L'origine dans ces deux cas est sans importance et n'importe quelle point de départ convient donc. On définit et calcule ensuite les quantités suivantes :

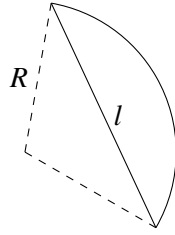
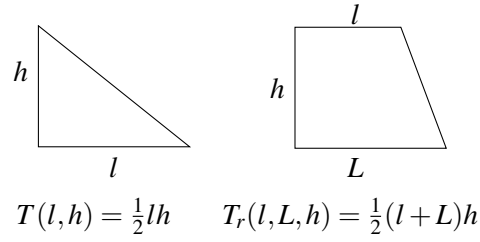
- L : distance entre le premier et le dernier point d'intersection.
- $l_i(a)$, respectivement $l_e(a)$, est la longueur de la portion de a à l'intérieur, respectivement à l'extérieur, de C .

On calcule alors l'aire de $I(R, C)$ en suivant l'Algorithme 4.1.

4.3.3 Un exemple de pâte cimentaire.

4.3.3.1 Définition de l'exemple.

On considère ici un milieu reprenant la physionomie de la pâte cimentaire. On a essayé de respecter au mieux les caractéristiques réelles (proportions, valeurs de



$$C_r(R, l) = \frac{R}{2} \left(R \arccos\left(1 - \frac{l^2}{2R^2}\right) - l \sqrt{1 - \frac{l^2}{4R^2}} \right)$$

FIGURE 4.14 – Aires de surfaces élémentaires. On note $T(l, h)$, $T_r(l, h)$ et $C_r(R, l)$ les aires respectives du triangle, du trapèze et de la corde représentés ici.

diffusivité, tailles et formes des inclusions) de la pâte cimentaire, mais certaines valeurs ont été adaptées au cas bidimensionnel. C'est le cas notamment du taux de remplissage qui est inférieur à la normale. Quoiqu'il en soit, l'intérêt de cette expérience reste entier : tester la robustesse de la méthode sur un cas réaliste.

On travaille sur l'espace $\Omega = [0, 1] \times [0, 1]$, qui représente un carré d' 1mm^2 de pâte cimentaire. Cet espace se compose d'un milieu homogène représentant la porosité capillaire, de diffusivité $D_0 = 2.2 \cdot 10^{-9} \text{m}^2 \cdot \text{s}^{-1}$, dans lequel on place plusieurs milliers d'inclusions réparties de la façon suivante :

- 1900 Sphères homogènes ($d_3 = 9 \cdot 10^{-12} \text{m}^2 \cdot \text{s}^{-1}$, $r_3 \in [0.001, 0.013]$) en LD C-S-H (*Low Density C-S-H*).
- 1700 Sphères homogènes ($d_2 = 1 \cdot 10^{-12} \text{m}^2 \cdot \text{s}^{-1}$, $r_2 \in [0.001, 0.013]$) en HD C-S-H (*High Density C-S-H*).
- 3500 Sphères homogènes ($d_1 = 1 \cdot 10^{-20} \text{m}^2 \cdot \text{s}^{-1}$, $r_1 \in [0.001, 0.013]$) représentant la portlandite, les aluminates et les anhydres.
- 200 Sphères multi-couches (rayon $r_4 \in [0.013, 0.028]$) composées d'un coeur de portlandite (ou d'aluminates, d'anhydre) occupant 40% du rayon, et de deux coquilles une de HD C-S-H (de $0.4r_4$ à $0.9r_4$) et une de LD C-S-H (de $0.9r_4$ à r_4).

Ces inclusions sont placées aléatoirement dans le domaine Ω par le module COMBS, décrit à la section §4.3.1.

Exemple 4.10 On résout le problème grossier suivant :

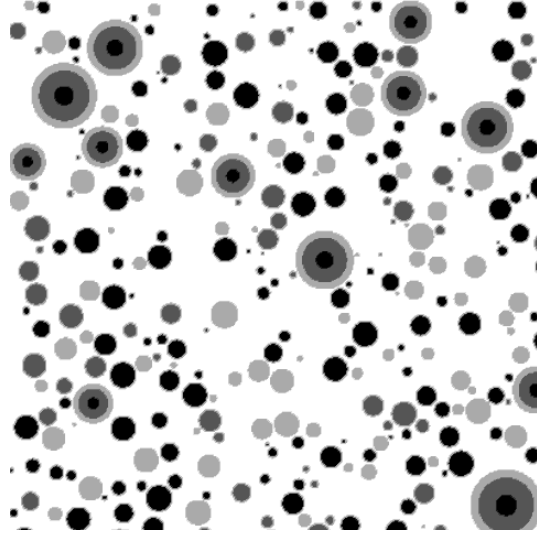


FIGURE 4.15 – Représentation de la pâte cimentaire modélisée : porosité capillaire (blanc), LD C-S-H (gris clair), HD C-S-H (gris foncé), portlandite, aluminates et anhydres (noir).

$$\begin{cases} -\nabla(D\nabla C) = 0 & \text{dans } \Omega, \\ C = \delta_{\{x=0\}} & \text{sur } \Gamma_D = \{0, 1\} \times [0, 1], \\ D\nabla C \cdot \mathbf{n} = 0 & \text{sur } \Gamma_N = [0, 1] \times \{0, 1\}. \end{cases} \quad (4.21)$$

où δ représente le symbole de Kronecker.

Dans le milieu, on impose donc à la concentration d'être égale à 1 en $x = 0$ et à 0 en $x = 1$. Le flux à travers les parois verticales est nul et il n'y a pas de terme source. Si on note $L = 1\text{mm}$ la longueur entre les deux parois horizontales de Ω et A^- son arête inférieure, on définit la *diffusivité équivalente du milieu* D^* par :

$$D^* = \frac{1}{L} \int_{A^-} D\nabla C \cdot \mathbf{n}. \quad (4.22)$$

Le problème grossier et le calcul de la diffusivité équivalente imitent les mesures expérimentales de diffusivité équivalente [48, 49, 63, 73].

4.3.3.2 Simulations.

On résout numériquement l'Exemple 4.10 pour plusieurs jeux de maillages grossiers et fins. On reprend les notations de la section §4.2 : on divise le milieu Ω en $M = m \times m$ macro-éléments K carrés de côté $H = 1/m$, formant ainsi la discrétisation grossière $T_H(\Omega)$. Chaque macro-élément K est ensuite divisé en $N = n \times n$ micro-éléments carrés de côté $h = 1/(mn)$, formant la discrétisation fine $T_h(K)$.

Avec ρ le taux de sur-échantillonnage, la discrétisation fine $T_h(\hat{K})$ d'une cellule \hat{K} comporte alors $(1 + 2\rho)^2 n^2$ micro-éléments.

Afin d'obtenir une solution de référence C_r , on résout l'Exemple 4.10 via une méthode directe de Volumes Finis à 4 millions d'éléments. En posant $h_r = 5 \cdot 10^{-4}$, la discrétisation $T_{h_r}(\Omega)$ du milieu Ω comporte donc 2000×2000 micro-éléments.

On compare ensuite les solutions numériques $C_{H,h}$ de l'Exemple 4.10, comportant $m^2 \times n^2$ micro-éléments, à C_r . On utilise pour cela les définitions des erreurs L_2 discrètes (4.16) et (4.17), portant respectivement sur les quantités $C_{H,h}$ et $D_h \nabla C_{H,h}$.

Remarque 4.2 *Comme pour l'Exemple 4.9, les courbes présentées ici ne sont pas des courbes de convergence de la méthode $Q_1/VF9$, mais des courbes d'évolution de l'erreur en fonction d'un paramètre (h , H ou $\alpha = h/H$) donné, les autres étant fixés. On se reportera à la section §4.2.1 pour plus de détails à ce sujet.*

4.3.3.3 Influence de la discrétisation fine $T_h(\hat{K})$.

Les courbes présentées à la Figure 4.16 montrent l'évolution des erreurs L^2 discrètes sur la concentration ($e_2(C_{H,h})$, en haut), le gradient ($e_2(\nabla C_{H,h})$, au milieu) et le flux ($e_2(D_h \nabla C_{H,h})$, en bas) quand h diminue, pour différentes valeurs fixées de H . Deux valeurs de m sont ici représentées : $H = 0.1$ (à gauche) et $H = 0.02$ (à droite).

Pour les deux valeurs faibles du recouvrement, $\rho = 0$ (en bleu) et $\rho = 0.05$ (en rouge), les erreurs sont sensiblement égales. Tout au plus le recouvrement non nul donne-t'il des résultats légèrement meilleurs sur le gradient et le flux. Ces erreurs diminuent jusqu'à atteindre un seuil minimum qui dépend de H .

Par contre, un recouvrement plus élevé comme $\rho = 0.10$ (en marron) et surtout $\rho = 0.20$ (en noir), a une influence très négative sur la méthode. Tout d'abord, quand H est grand, les résultats sont moins bons que ceux de la méthode classique. Mais surtout l'erreur diverge quand H tend vers 0 : la méthode est instable.

4.3.3.4 Influence de la discrétisation grossière $T_H(\Omega)$.

On représente à la Figure 4.16 l'évolution des erreurs L^2 discrètes sur la concentration ($e_2(C_{H,h})$, en haut), le gradient ($e_2(\nabla C_{H,h})$, au milieu) et le flux ($e_2(D_h \nabla C_{H,h})$, en bas) quand h diminue, pour un rapport $\alpha = h/H$ fixés. Deux rapports sont ici représentés : $\alpha = 0.02$ (à gauche) et $\alpha = 0.01$ (à droite).

On note que, pour la concentration et le gradient, le cas sans recouvrement (en bleu) stagne très rapidement à un niveau minimum dépendant de α . Le cas $\rho = 0.05$ (en rouge) suit la même allure, mais avec des valeurs légèrement inférieures. Le cas $\rho = 0.10$ (en marron) donne un résultat mitigé, précurseur des très mauvais résultats du cas à fort recouvrement $\rho = 0.20$ (en noir). Dans ce dernier cas en effet, la méthode diverge.

Si on se penche maintenant sur le flux, on note que la méthode stagne, et ce quel que soit le taux de sur-échantillonnage.

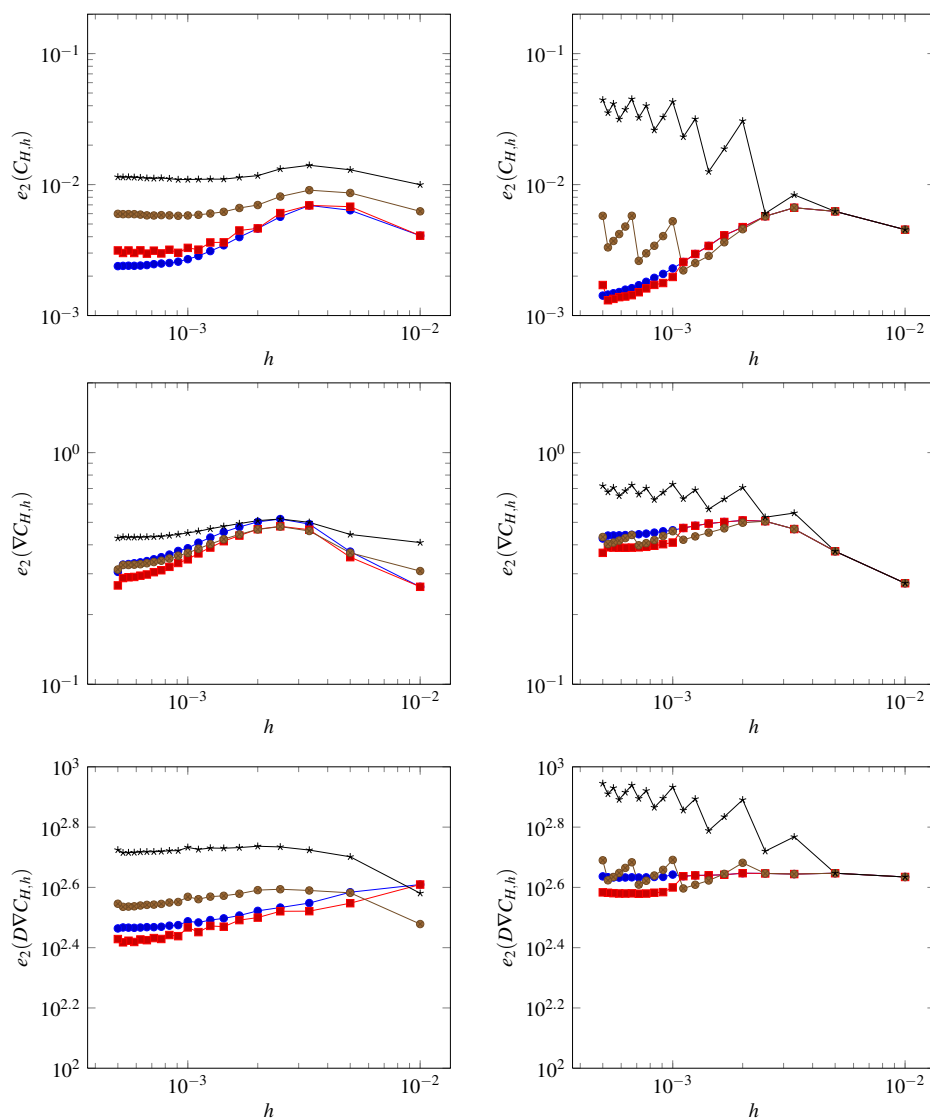


FIGURE 4.16 – Exemple 4.10 : $e_2(C_{H,h})$ (haut), $e_2(\nabla C_{H,h})$ (milieu) et $e_2(D\nabla C_{H,h})$ (bas) pour $H = 0.1$ (gauche) et $H = 0.02$ (droite). Sur chaque courbe, on a tracé l'évolution de l'erreur en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—).

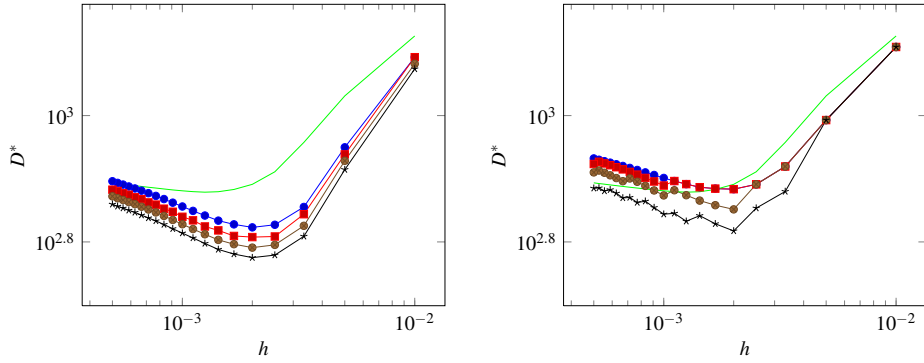


FIGURE 4.17 – Exemple 4.10 : évolution de la diffusivité équivalente D^* pour $H = 0.1$ (gauche) et $H = 0.02$ (droite). Sur chaque courbe, on a tracé l'évolution de D^* en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—). La courbe verte correspond au coefficient calculé à partir de la solution directe.

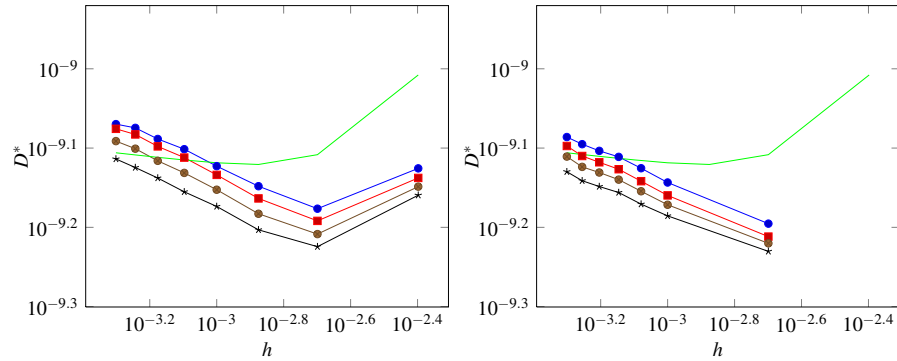


FIGURE 4.18 – Exemple 4.10 : évolution de la diffusivité équivalente D^* pour $h = 0.02H$ (gauche) et $h = 0.01H$ (droite). Sur chaque courbe, on a tracé l'évolution de D^* en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—). La courbe verte correspond au coefficient calculé à partir de la solution directe.

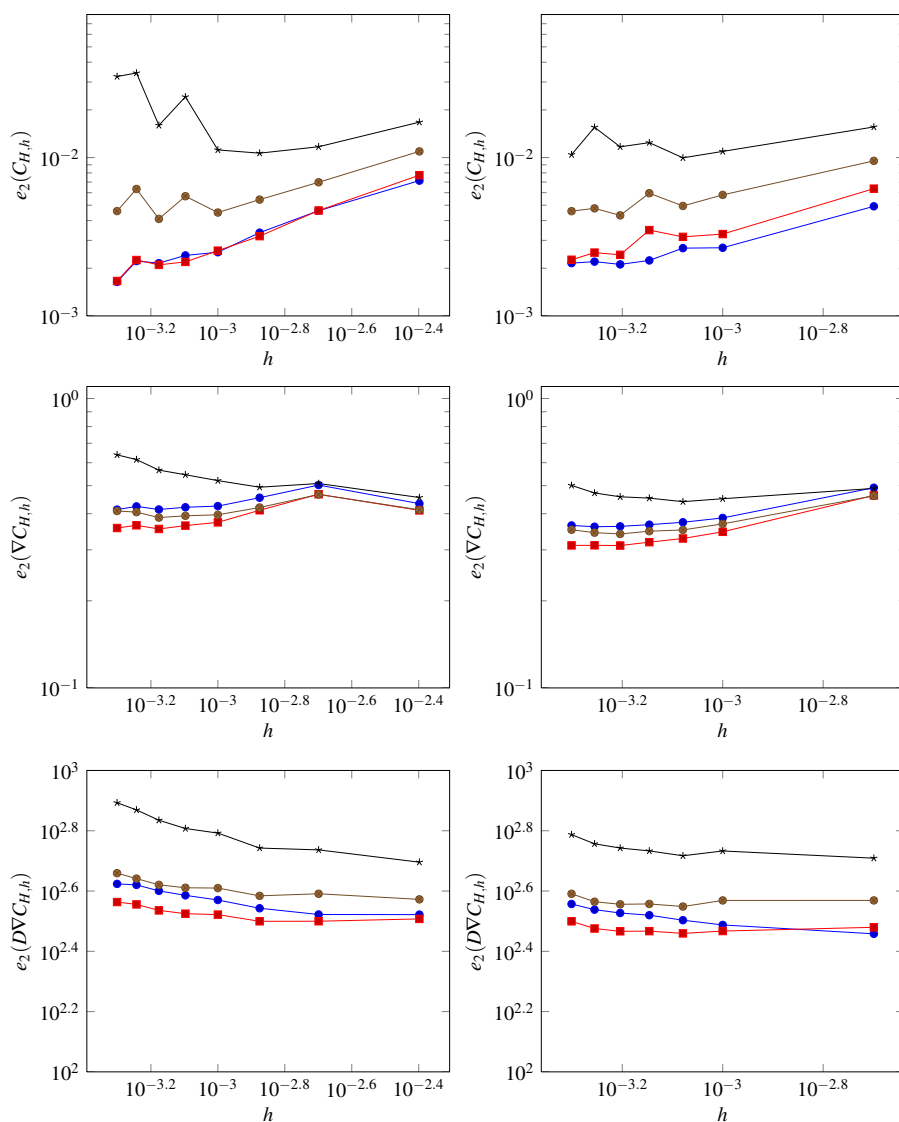


FIGURE 4.19 – Exemple 4.10 : erreurs $e_2(C_{H,h})$ (haut), $e_2(\nabla C_{H,h})$ (milieu) et $e_2(D\nabla C_{H,h})$ (bas) pour $h = 0.02H$ (gauche) et $h = 0.01H$ (droite). Sur chaque courbe, on a tracé l'évolution de l'erreur en fonction de h pour quatre valeurs de ρ : 0 (—●—), 0.05 (—■—), 0.10 (—●—) et 0.20 (—*—).

4.3.3.5 Évolution de la diffusivité équivalente du milieu.

Les diffusivités équivalentes D^* issues des solutions calculées par la méthode MsFEM sont présentées à la Figure 4.17 (évolution en fonction de h à H fixé) et à la Figure 4.18 (évolution en fonction de h à $\alpha = h/H$ fixé). On note que la diffusivité suit la même évolution quelle que soit la valeur choisie pour le taux de sur-échantillonnage ρ . Par contre, si à la Figure 4.17 cette évolution correspond bien à celle des valeurs obtenues par résolution directe (en vert sur les courbes), ce n'est pas le cas de la Figure 4.18. On ne peut donc réellement considérer qu'il y a convergence de la diffusivité équivalente, même si, au final, l'écart entre les valeurs calculées ne dépasse pas 10%.

Cet écart est sans doute dû aux erreurs dans l'évaluation du flux évoquées dans les paragraphes précédents. Le coefficient D^* étant en effet calculé à partir du flux, il est logique qu'une erreur sur l'un se reporte sur l'autre.

4.3.3.6 Repartition locale de l'erreur.

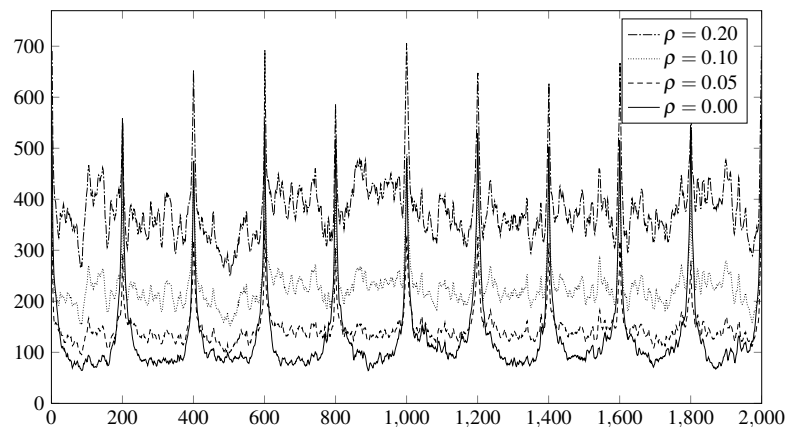


FIGURE 4.20 – Coupe de la densité de flux à travers le domaine Ω : différence entre la solution méthode multi-échelle $D_h \nabla C_{H,h} \cdot \mathbf{n}$ pour $(H, h) = (0.1, 5 \cdot 10^{-4})$, et la solution de référence $D_r \nabla C_r \cdot \mathbf{n}$.

La Figure 4.20 présente une coupe de l'erreur, sur le flux $D \nabla C$, entre le cas $(H, h) = (0.1, 5 \cdot 10^{-4})$ et la solution de référence. On note que l'erreur est trois à quatre fois plus importante le long des frontières de macro-éléments.

Ce phénomène de *couche limite*, décrit dans [90], devrait normalement être limité par le sur-échantillonnage, ce qui n'est pas le cas ici. Choisir $\rho = 0.05$ permet bien de réduire les valeurs maximales de l'erreur d'environ 30%, mais augmente aussi la valeur minimale de l'erreur, si bien, qu'en moyenne, l'erreur augmente. Quand aux cas $\rho = 0.1$ et $\rho = 0.2$, ils donnent de moins bons résultats sur tous les tableaux.

Le phénomène de *couche limite* n'étant pas limité ici, il est compréhensible que l'erreur sur la densité de flux augmente avec le nombre de *frontières*, c'est-à-dire, en pratique, avec le nombre M de macro-éléments.

4.4 Conclusions.

Ce chapitre présente une implémentation autonome de la méthode multi-échelle, réalisée durant la première année du travail de thèse. Cette maquette Python, une fois validée, a été utilisée pour résoudre des problèmes de diffusion : un cas théorique et un autre, plus complexe, modélisant une pâte cimentaire 2D. De ces résultats, on a pu mettre en lumière plusieurs problèmes.

Tout d'abord, il apparaît sur les cas théoriques que l'erreur de la méthode diminue avec H , c'est-à-dire quand le nombre de cellules augmente. Une plus grande division du domaine permet donc d'obtenir de meilleurs résultats. Même s'il ne s'agit pas à proprement parler d'une convergence de la méthode $Q_1/VF9$, comme expliqué à la section §4.2.1, c'est toutefois un bon résultat

La méthode de recouvrement [79] permet normalement de corriger certains défauts des méthodes multi-échelles. Ses avantages dans le cadre de l'homogénéisation théorique sont bien connus. En reprenant les notations de la section §4.2.1, on a :

$$\|C^\varepsilon - C_{H,h}^\varepsilon\|_2 \leq A_1 \left(H + \sqrt{\frac{\varepsilon}{H}} + \frac{h}{\varepsilon} \right) \quad \text{si } \rho = 0, \quad (4.23)$$

$$\|C^\varepsilon - C_{H,h}^\varepsilon\|_2 \leq A_2 \left(H + \frac{\varepsilon}{H} + \frac{h}{\varepsilon} \right) \quad \text{si } \rho > 0. \quad (4.24)$$

Le lien théorique entre la convergence de la méthode multi-échelle et la valeur du taux de sur-échantillonnage ρ est cependant mal connu, car on ne sait expliciter la dépendance de A_2 en ρ . Tout au plus sait-on qu'augmenter ρ au-delà d'un certain point ne fait pas ni converger ni diverger la méthode multi-échelle, la constante A_2 se stabilisant rapidement à une valeur constante non nulle.

En pratique cependant, lorsque H diminue, l'erreur a tendance à stagner pour le cas de la pâte cimentaire. Les forts sauts de diffusivité du milieu, dont l'ordre de grandeur peut atteindre 10^8 , sont sûrement à l'origine de ces problèmes. Il semble difficile, voir impossible, de choisir optimalement les différents paramètres de la méthode dans un cas aussi raide. En conséquence de quoi, la méthode de recouvrement est ici inefficace. Pire un recouvrement trop important détériore considérablement les résultats.

Plusieurs pistes ont été considérées pour tenter de régler ce problème.

Tout d'abord, on a augmenté la taille des systèmes étudiés. Augmenter la précision de la discrétisation permet en effet une capture plus fine des effets de couche-

limites via la méthode de recouvrement. Comme les calculs présentés dans ce chapitre atteignent les capacités maximales d'un ordinateur séquentiel, il a été nécessaire de changer le cadre d'implémentation de la méthode $Q_1/VF9$ (programme Python sur un ordinateur) pour une interface plus robuste. Le choix s'est porté sur le code *MPCube* [61], basé sur le noyau de calcul *Trio-U* [171]. Cette nouvelle implémentation est présentée au chapitre §5.

Profitant de ce changement d'implémentation, on a considérablement élargi le cadre de travail, l'ambition affichée étant de pouvoir traiter des matériaux cimentaires $2D$ et $3D$. Afin de modéliser au mieux les géométries des matériaux cimentaires, la contrainte sur les maillages a été supprimée. Les maillages fins ne sont plus quadrangulaires réguliers, mais quelconques. Ils doivent donc être générés avant toute simulation. Le chapitre §6 contient les méthodes développées au sein de la plate-forme *SALOME* [163] afin de pouvoir discrétiser rapidement et automatiquement les domaines de travail, c'est-à-dire l'ensemble des cellules \hat{K} . Il décrit ainsi comment l'on obtient, à partir d'une description d'un matériau cimentaire, les maillages nécessaires aux simulations *MPCube*.

Afin de faire face à l'importante raideur des problèmes de diffusion au sein des matériaux cimentaires, on a remplacé, à l'échelle fine, la méthode des Volumes Finis *VF9* par la méthode des Volumes Finis *VDiam*. Introduite par *Coudière et al.* [79], cette méthode est depuis utilisée par le CEA pour travailler sur des milieux à forts contrastes et/ou à fortes anisotropies comme les matériaux cimentaires. Elle est décrite à la section §3.2.3.

Il apparaît qu'une non-conformité trop importante des Éléments Finis pénalise les résultats de la méthode $Q_1/VF9$, en particulier au niveau de la résolution du problème grossier. Il semble donc judicieux de remplacer, à l'échelle grossière, la méthode d'Éléments Finis classique par une méthode plus adaptée aux fonctions non-conformes. Le choix s'est porté sur une méthode discontinue de Galerkin, plus précisément une méthode de *pénalité intérieure*, ou IP [29, 82]. Elle est décrite à la section §3.3.2.

De ces deux choix de méthodes, on a construit deux couplages grossier/fin : le couplage Éléments Finis et *VDiam*, que l'on note $Q_1/VDiam$, et le couplage Galerkin discontinu et *VDiam*, noté *GD/VDiam*. Les résultats obtenus avec ces deux couplages sont présentés au chapitre §7.