MIMSE : Analyse numérique approfondie

Freefem++, problème de Stokes

FreeFem++

- Page web : `http://www.freefem.org/ff++/`
- Documentation : `http://www.freefem.org/ff++/ftp/freefem++doc.pdf`
- Principe : FreeFem++ est un interpréteur de script (comme matlab, python, scilab...) dédié à la résolution approchée par éléments finis de problèmes mis sous forme variationnelle. Son langage est assez proche du C.
- On écrit donc des fichiers contenant des commandes, par exemple `exple.edp`, que l'on éxécute dans un terminal X en utilisant la commande suivante :
  `...$ FreeFem++ exple.edp`
- Quelques éditeurs de fichiers supportent une colorisation syntaxique pour FreeFem++, nedit, emacs...
- Il existe aussi un environnement de développement intégré (IDE) pour FreeFem++, voir `http://www.ann.jussieu.fr/~lehyaric/ffcs/learning.php`.

Bibliothèques d'algèbre linéaire

- BLAS (`http://www.netlib.org/blas/`) est en général installé sur tout système de calcul scientifique. C'est une bibliothèque aui implémente des opérations entre matrices et vecteurs. Sur chaque système, il existe des versions optimisées. Par exemple Intel fournit une implémentation de BLAS dans la bibliothèque mkl, `https://software.intel.com/en-us/intel-mkl`.
- UMFPACK est une bibliothèque qui fait partie de la suite logicielle `Suitesparse` (`http://faculty.cse.tamu.edu/davis/suitesparse.html`), utilisée en particuler dans `Matlab` (fonction « \ »). C'est une bibliothèque de résolution des systèmes linéaires creux par une méthode directe (décomposition LU). Elle utilise un système de stockage CSC (Compressed Sparse Column) des matrices.

Liste de cas tests

Les équations de Stokes stationnaire :

$$-\nu\Delta\mathbf{u} + \nabla p = \mathbf{f}, \quad \operatorname{div}(\mathbf{u}) = 0 \quad \text{dans } \Omega, \qquad \mathbf{u} = \mathbf{g} \quad \text{sur } \partial\Omega, \tag{1}$$

et instationnaire

$$\partial_t\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f}, \quad \operatorname{div}(\mathbf{u}) = 0 \quad \text{dans } \Omega, \qquad \mathbf{u} = \mathbf{g} \quad \text{sur } \partial\Omega, \tag{2}$$

et les équations de Navier-Stokes incompressibles

$$\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f}, \quad \operatorname{div}(\mathbf{u}) = 0 \quad \text{dans } \Omega, \qquad \mathbf{u} = \mathbf{g} \quad \text{sur } \partial\Omega. \tag{3}$$

Le problème discret associé au problème de Stokes stationnaire (1) est de la forme

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \tag{4}$$

où A est une matrice symétrique et définie positive, et B est une matrice dont le noyau est constitué des fonctions constantes, ie $\ker(B) = \mathbb{R}\mathbf{1}$, et $\mathbf{1} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$.

On rappelle que l'on recherche la solution de ce système qui vérifie $\int_\Omega p(x)dx = 0$, et que l'on peut trouver une solution approchée en résolvant le système pénalisé (pour $\epsilon > 0$)

$$\begin{pmatrix} A & B^T \\ B & -\epsilon \,\mathrm{Id} \end{pmatrix} \begin{pmatrix} U_\epsilon \\ P_\epsilon \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}. \tag{5}$$

On peut également obtenir une solution approchée en remarquant que U réalise le minimum de $J(U) = \frac{1}{2}U^T AU - U^T F$ sous la contrainte $BU = 0$. Dans ce cas, la pression apparait comme un multiplicateur de Lagrange, et on peut utiliser l'algorithme suivant (Uzawa) :

1. On se donne P^0, et $\mu^0 > 0$;
2. Pour $n = 0, 1...$, résoudre $AU^n = F - B^T P^n$ puis calculer $p^{n+1} = p^n + \mu^n BU^n$.

Le critère d'arrêt porte sur la différence $\|p^{n+1} - p^n\| = \mu\|BU^n\|$ qui doit tendre vers 0 pour $\mu > 0$ bien choisi.

TC1a : solutions affines en vitesse sur un triangle On prend pour Ω le triangle de sommets les points $(0, 0)$, $(1, 0)$ et $(1, 1)$. On choisit comme solution $u(x, y) = \begin{pmatrix} x & -y \end{pmatrix}$, et $p(x, y) = x + y + c$ (à une constante près), qui est bien solution du problème de Stokes (1) avec $f(x, y) = \begin{pmatrix} 1 & 1 \end{pmatrix}$.

TC1b : solutions affines en vitesse sur un carré On prend pour Ω le carré $(0, 1) \times (0, 1)$. On choisit comme solution $u(x, y) = \begin{pmatrix} x & y \end{pmatrix}$, et $p(x, y) = x + y + c$ (à une constante près), qui est bien solution du problème de Stokes (1) avec $f(x, y) = \begin{pmatrix} 1 & 1 \end{pmatrix}$.

TC2a : écoulement de Poiseuille Sur le domaine $\Omega = (0, L) \times (-0.5, 0.5)$, on cherche à retrouver la solution de Poiseuille du problème de Stokes (1), $u = \begin{pmatrix} 1/4 - y^2 & 0 \end{pmatrix}$ et $p = -2x$ (à une constante près). On prend comme conditions aux limites $u = 0$ sur les bords horizontaux $y \in \{\pm 0.5\}$, et on peut varier les données aux bords $x = 0$ et $x = L$, par exemple $u = \begin{pmatrix} 1 & 0 \end{pmatrix}$ ou bien $u = \begin{pmatrix} 1/4 - y^2 & 0 \end{pmatrix}$.

TC2b : écoulement pulsé dans un tube Sur le domaine $\Omega = (0, L) \times (-0.5, 0.5)$, on recherche la forme du profil lorsque le forçage en entrée est pulsé (ie comme dans le problème de Poiseuille mais périodique en temps). Pour le problème de Stokes instationnaire (2), on prend comme conditions aux limites $u = 0$ sur les bords horizontaux $y \in \{\pm 05\}$, et $u = \begin{pmatrix} u_\infty \sin^2(\omega t) & 0 \end{pmatrix}$ sur les bords $x = 0$ et $x = L$.

On doit retrouver des profils de vitesse typique ayant une forme de W.

TC3a : cavité entrainée On s'intéresse maintenant au cas-test suivant : $\Omega = (0, 1) \times (0, 1)$ et les conditions aux limites $u = 0$ sur les bords $\{x = 0\}$, $\{x = 1\}$, $\{y = 0\}$ et $u = \begin{pmatrix} u_\infty & 0 \end{pmatrix}$ sur le bord supérieur $\{y = 1\}$.

Pour le problème de Stokes (1), la solution n'est pas connue analytiquement.

TC3b : cavité entrainée pulsée On s'intéresse maintenant au cas-test suivant : $\Omega = (0, 1) \times (0, 1)$ et les conditions aux limites $u = 0$ sur les bords $\{x = 0\}$, $\{x = 1\}$, $\{y = 0\}$ et $u = \begin{pmatrix} u_\infty \sin(\omega t) & 0 \end{pmatrix}$ sur le bord supérieur $\{y = 1\}$.

Pour le problème de Stokes instationnaire (2), la solution n'est pas connue analytiquement.

TC4 : marche inversée On cherche la solution du problème de Stokes dans le domaine tronqué $\Omega = (0, 1) \times (0, 1) \backslash (0, L) \times (0, H)$ avec $L, H \in (0, 1)$. Les conditions aux limites sont $u = \begin{pmatrix} u_\infty & 0 \end{pmatrix}$ en $x = 0$, $u = \begin{pmatrix} (1 - H)u_\infty & 0 \end{pmatrix}$ en $x = 1$ (pour maintenir le même débit en entrée et en sortie), et $u = 0$ partout ailleurs.

Pjt1 : Schéma MAC, pénalisation, solveur direct

Programmer en Fortran 90 (ou C/C++) la résolution du problème de Stokes avec des conditions aux limites de Dirichlet non homogènes sur le carré $\Omega = (0,1) \times (0,1)$. On utilisera la bibliothèque logicielle UMFPACK pour la résolution des systèmes linéaires par une méthode directe, ce qui suppose que les matrices soient stockées dans le format appelé CSC pour *Compressed Sparse Column.*

1. Vérifier l'implémentation du calcul de la matrice A du problème discret pour le problème de Poisson $-\Delta u = f$ avec des conditions de Dirichlet non homogènes $u = g$ sur Ω en utilisant des solutions manufacturées. Exple $u(x,y) = 1+ax+by$, puis $u(x,y) = ax^2+by^2$, etc.
2. Assembler la matrice globale du système de Stokes avec une pénalisation $\epsilon > 0$, et assembler le vecteur second membre en tenant compte de la condition de Dirichlet.
3. Tester l'implémentation avec un écoulement à vitesse constante, $u = (u1, u2)$ quelconque au bord et $f = (0,0)$ puis $f = (0,-g)$. La solution est donc $u = (u1, u2)$ partout et $p = cste$ puis $p = -gz + cste$. On doit retrouver exactement cette solution. Étudier les comportement pour les cas tests TC1a et TC1b.
4. Étudier le comportement du schéma (convergence en maillage, précision sur vitesse et pression, valeur de $\int_\Omega p(x)dx$) pour l'écoulement de Poisseuille, cas test TC2a pour différents profils de vitesse en entrée et sortie. Quel est le rôle du coefficient de pénalisation ? Comment le choisir au fur et à mesure du raffinement ?

Pjt2 : Schéma MAC, pénalisation, solveur itératif

Programmer en Fortran 90 (ou C/C++) la résolution du problème de Stokes avec des conditions aux limites de Dirichlet non homogènes sur le carré $\Omega = (0,1) \times (0,1)$. On programmera la méthode du gradient conjuguée pour la résolution des systèmes linéaires en utilisant, soit le stockage « band »et la bibliothèque BLAS, soit le format appelé CSR pour *Compressed Sparse Row* et un produit matrice vecteur adapté.

1. Vérifier l'implémentation du calcul de la matrice A du problème discret pour le problème de Poisson $-\Delta u = f$ avec des conditions de Dirichlet non homogènes $u = g$ sur Ω en utilisant des solutions manufacturées. Exple $u(x,y) = 1+ax+by$, puis $u(x,y) = ax^2+by^2$, etc.
2. Assembler la matrice globale du système de Stokes avec une pénalisation $\epsilon > 0$, et assembler le vecteur second membre en tenant compte de la condition de Dirichlet.
3. Tester l'implémentation avec un écoulement à vitesse constante, $u = (u1, u2)$ quelconque au bord et $f = (0,0)$ puis $f = (0,-g)$. La solution est donc $u = (u1, u2)$ partout et $p = cste$ puis $p = -gz + cste$. On doit retrouver exactement cette solution. Étudier les comportement pour les cas tests TC1a et TC1b.
4. Étudier le comportement du schéma (convergence en maillage, précision sur vitesse et pression, valeur de $\int_\Omega p(x)dx$) pour l'écoulement de Poisseuille, cas test TC2a pour différents profils de vitesse en entrée et sortie. Quel est le rôle du coefficient de pénalisation ? Comment le choisir au fur et à mesure du raffinement ? On pourra aussi étudier l'influence du critère d'arrêt des itérations de gradient conjugué, et comment choisir ce critère de manière cohérence avec le paramètre de régularisation...

Pjt3 : Schéma MAC, méthode d'Uzawa, solveur itératif

Programmer en Fortran 90 (ou C/C++) la résolution du problème de Stokes avec des conditions aux limites de Dirichlet non homogènes sur le carré $\Omega = (0,1) \times (0,1)$. On programmera la méthode du gradient conjuguée pour la résolution des systèmes linéaires en utilisant, soit le stockage « band »et la bibliothèque BLAS, soit le format appelé CSR pour *Compressed Sparse Row* et un produit matrice vecteur adapté.

1. Vérifier l'implémentation du calcul de la matrice A du problème discret pour le problème de Poisson $-\Delta u = f$ avec des conditions de Dirichlet non homogènes $u = g$ sur Ω en utilisant des solutions manufacturées. Exple $u(x,y) = 1+ax+by$, puis $u(x,y) = ax^2+by^2$, etc.
2. Assembler séparément les matrices discrètes A (pour $u \mapsto -\nu\Delta u$) et B (pour $u \mapsto -\operatorname{div}(u)$) du système de Stokes, et assembler le vecteur second membre en tenant compte de la condition de Dirichlet.
3. Implémenter une résolution par la méthode itérative de Uzawa avec un paramètre $\mu > 0$. Le système linéaire sera résolu par une méthode de gradient conjugué.
4. Tester l'implémentation avec un écoulement à vitesse constante, $u = (u1, u2)$ quelconque au bord et $f = (0,0)$ puis $f = (0,-g)$. La solution est donc $u = (u1, u2)$ partout et $p = cste$ puis $p = -gz + cste$. On doit retrouver exactement cette solution. Étudier les comportement pour les cas tests TC1a et TC1b. On étudiera l'influence de μ sur la convergence, la cohérence du critère d'arrêt du gradient conjugué avec ce choix.
5. Étudier le comportement du schéma (convergence en maillage, précision sur vitesse et pression, valeur de $\int_\Omega p(x)dx$) pour l'écoulement de Poisseuille, cas test TC2a pour différents profils de vitesse en entrée et sortie.

Pjt4 : comparaisons de méthodes d'éléments finis, pénalisation

Programmer dans Freefem++ la résolution du problème de Stokes avec une pénaisation, avec des conditions aux limites de Dirichlet non homogènes dans un domaine Ω. On implémentera une version qui utilise un solveur direct (UMFPACK) et une version qui utilise la méthode du gradient conjugué (CG).

1. Pour quelques cas test, étudier les résultats obtenus sur un maillage fixé avec les méthodes de discrétisation suivantes : P1-P0, P1-P1, P1b-P1, Pk+1-Pk pour les cas disponibles en Freefem++. Comparer les résolutions par la méthode directe et par la méthode itérative, étudier l'effet du choix du paramètre de pénalisation, du critère d'arrêt du gradient conjugué.
2. Pour les cas test TC1a, TC1b, TC2a, sur une famille de maillage de pas de plus en plus fin, calculer les erreurs (au sens L^2, H^1) $E_u = \left(\int_\Omega |\mathbf{u} - \mathbf{u}_h|^2\right)^{1/2}$ sur la vitesse et $E_p = \left(\int_\Omega |p - p_h|^2\right)^{1/2}$ sur la pression (attention à la normalisation des pressions p et p_h). Tracer les graphes des erreurs et étudier les erreurs de convergence des méthodes qui donnent un résultat jugé satisfaisant. De même, on comparera méthode itérative et directe, choix de la pénalisation, critère d'arrêt du gradient conjugué. On cherchera en particulier à choisir le paramètre de pénalisation en fonction du pas du maillage.
3. Essayer de calculer des solutions pour les cas tests TC3a et TC4 sur plusieurs maillages de pas différents. Comparer et commenter les différentes solutions.

Pjt5 : comparaisons de méthodes d'éléments finis, méthode d'Uzawa

1. Programmer dans Freefem++ la résolution du problème de Stokes avec des conditions aux limites de Dirichlet non homogènes dans un domaine Ω. On cherchera à construire un implémentation de l'algorithme d'Uzawa dans ce cadre. On pourra obtenir 2 versions : une version qui utilise un solveur direct (UMFPACK) et une version qui utilise la méthode du gradient conjugué (CG).
2. Pour quelques cas test, étudier les résultats obtenus sur un maillage fixé avec les méthodes de discrétisation suivantes : P1-P0, P1-P1, P1b-P1, Pk+1-Pk pour les cas disponibles en Freefem++. Comparer les résolutions par la méthode directe et par la méthode itérative, étudier l'effet du choix du paramètre μ de a méthode d'Uzawa, du critère d'arrêt du gradient conjugué.

3. Pour les cas test TC1a, TC1b, TC2a, sur une famille de maillage de pas de plus en plus fin, calculer les erreurs (au sens L^2, H^1) $E_u = \left(\int_\Omega |\mathbf{u} - \mathbf{u}_h|^2\right)^{1/2}$ sur la vitesse et $E_p = \left(\int_\Omega |p - p_h|^2\right)^{1/2}$ sur la pression (attention à la normalisation des pressions p et p_h). Tracer les graphes des erreurs et étudier les erreurs de convergence des méthodes qui donnent un résultat jugé satisfaisant. De même, on comparera méthode itérative et directe, choix du pas μ dans la méthode d'Uzawa, critère d'arrêt du gradient conjugué.
4. Essayer de calculer des solutions pour les cas tests TC3a et TC4 sur plusieurs maillages de pas différents. Comparer et commenter les différentes solutions.

Pjt6 : méthodes d'éléments finis pour des écoulement instationnaires

On cherche à résoudre dans Freefem++ le problème de Stokes instationnaire (2) dans un domaine Ω avec des conditions aux limites de Dirichlet non homogènes et variables en temps. Les cas tests visés sont les cas TC2b et TC3b.

1. Programmer la résolution par une méthode d'Euler implicite en temps, avec un terme de pénalisation pour fixer la pression. Valider l'implémentation, par exemple en utilisant un écoulement permanent pour lequel la solution exacte est connue (TC1a, TC1b, TC2a). On utilisera des éléments finis P1b-P1, P2-P1, et P3-P2 (s'ils sont disponibles). On pourra ensuite (si assez de temps), programmer le schéma de Crank-Nicholson.
2. Essayer de calculer des solutions pour les cas tests TC3a et TC4 sur plusieurs maillages de pas différents. Comparer et commenter les différentes solutions.
3. Pour le cas test TC3a, on peut faire varier la longueur L du domaine et essayer de déterminer expérimentalement l'allure du profil de vitesse en aval dans le domaine.
4. Pour le cas test TC4, on essayer de comprendre ce qui se passe quand on fait varier la fréquence, en gardant le vitesse maximale relativement faible (faible Reynolds).

Sujet	Groupe
Pjt 3	Jacquet, Raven
Pjt 5	Lariviere, Robyn
Pjt 4	Rouault, Raymond
Pjt 6	Windpassinger, Marchais
Pjt 4 ou 5	Lebret, Dasprès, Renard
Pjt 1 ou 2	Foune, Zahid

TABLE 1 – Répartition