Résolution d'un écoulement en milieu poreux

Le but du projet est de résoudre un écoulement de Darcy en milieu poreux, avec une perméabilité *k* variable, dans un domaine 2D, par la méthode EFM vue en cours.

Le domaine 2D est carré : [0,1]x[0,1]

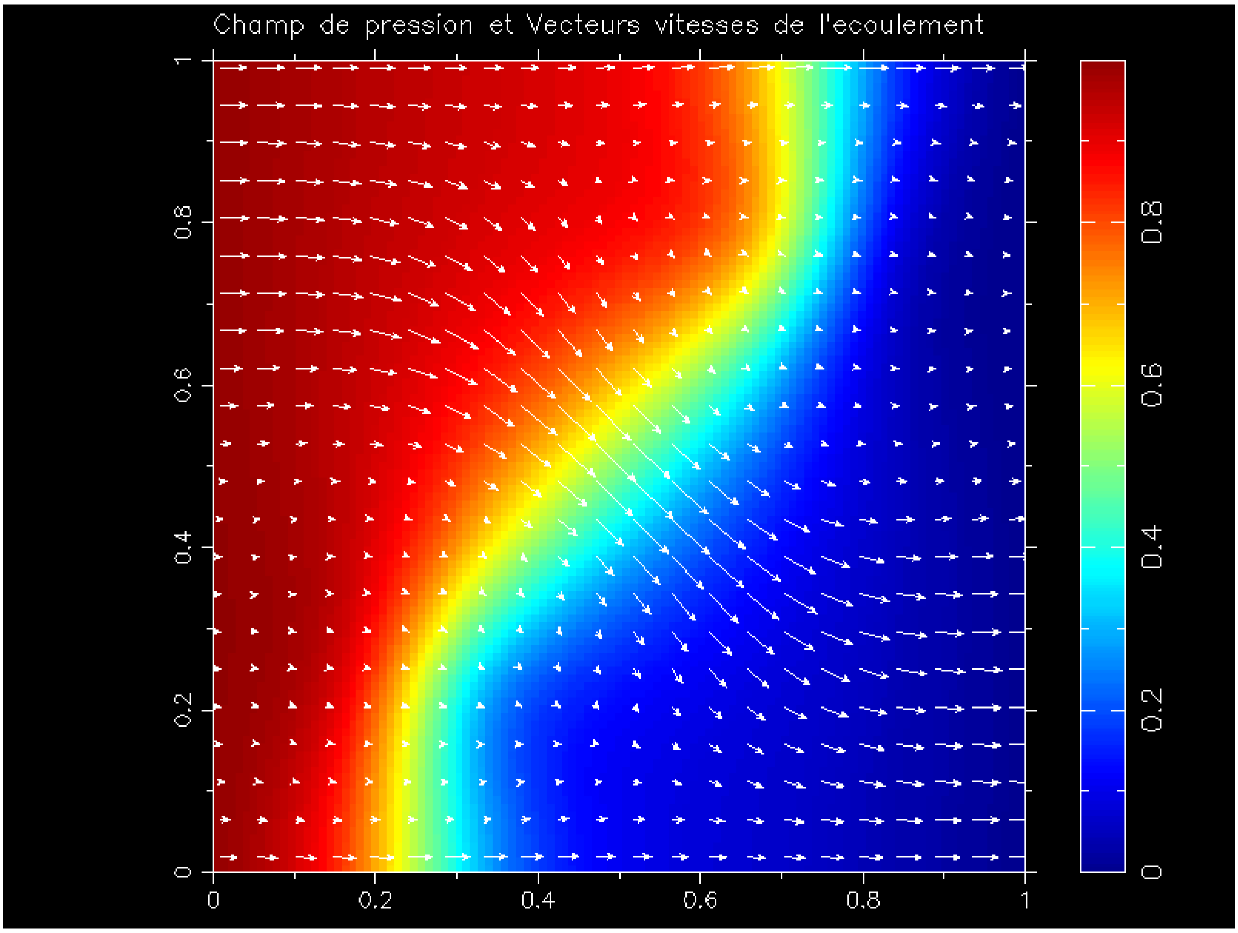
La perméabilité $k(x,y)$ est **donnée** ; la pression $p(x,y)$ est l'**inconnue**.

Les conditions limites sont :

– imperméabilité sur les bords sup. et inf. : $v = 0$ (Neumann)

– pression constante imposée aux extrémités (1 à gauche ; 0 à droite) (Dirichlet)

Le maillage est n par n, de type structuré.



On envisage le projet en plusieurs phases.

1°/ Résolution en stockage dense

a) Rappel des relations entre le couple (i,j) d'une maille dans l'espace physique
et l'indice k de la même maille dans le vecteur des inconnues.
Génération du système linéaire de taille : remplissage de la matrice et du second membre.

b) Vérification que le système est SPD (symétrique et défini positif).
Résolution par `LAPACK` (routine appropriée) et calcul du résidu
(mesure et tracé du temps CPU en fonction du nombre total d'inconnues).
Recherche du facteur le plus limitant : mémoire ou temps CPU, quand on augmente la résolution.

c) Vérification du résultat par le tracé des cartes de pressions et de vecteurs vitesse.

2°/ Résolution en stockage creux

a) génération de la matrice au format `COO`
(puis conversion au format `CSC`, routine `coo2csc` fournie)

b) résolution par `CHOLMOD` (bibliothèque SuiteSparse, interface F90 fournie) et calcul du résidu
(mesure et tracé du temps CPU en fonction du nbr total d'inconnues)

recherche du facteur le plus limitant : mémoire ou temps CPU, quand on augmente la résolution.

c) vérification du résultat par le graphique, comme en phase n°1.

3°/ Mise en œuvre du parallélisme par décomposition de domaine

a) renuméroter le domaine de manière à faire apparaître deux parties indépendantes, reliées par une interface ;
Quel est le vecteur permutation associé ? (on en aura besoin pour revenir au plan physique)
Envisager, par la suite, de partager en 4 parties ; est-il facile de généraliser à *s* sous-domaines ?

b) génération des blocs de la matrice sur différents processeurs.

c) mettre en œuvre la méthode du complément de Schur
montrer (ou vérifier) que tous les blocs carrés sont SPD et les blocs rectangulaires sont transposés 2 à 2 ; simplifier alors l'écriture de la méthode.

d) mettre en œuvre le parallélisme (MPI) :
les blocs carrés diagonaux principaux B_i sont factorisés sur des processeurs différents par `CHOLMOD`
le système final faisant intervenir le complément de Schur S est également résolu via `CHOLMOD`
(mesure et tracé du temps CPU en fonction du nbr total d'inconnues)
Quelle est l'accélération obtenue (la tracer en fonction du nombre *s* de sous-domaines) ?

Jusqu'à quelle résolution peut-on monter ?

Vérification du résultat par le graphique, comme en phase n°1.