

# Mini-projet : Simulation numérique d'un affaissement visqueux

LAVALLEY Julien  
RICHARD Pablo

Février 2019

## 1 Formulations variationnelles

**Question 1.** Soit  $V$  le sous-espace de  $(H^1(\Omega))^d$  défini par :

$$V = \{ \vec{v} \in (H^1(\Omega))^d \mid \vec{v} = \vec{0} \text{ sur } \Gamma_D \text{ et } -\operatorname{div}(\vec{v}) = 0 \text{ sur } \Omega \} \quad (1)$$

En partant de l'équation (1) du sujet on a, pour tout  $\vec{v}$  dans  $V$  :

$$\int_{\Omega} -(\Delta \vec{u}) \cdot \vec{v} \, dx + \int_{\Omega} \vec{\nabla} p \cdot \vec{v} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (2)$$

Où  $\Delta \vec{u}$  représente le vecteur  $\sum_{i=1}^d \nabla u_i \vec{e}_i$ .

On notera également par la suite  $\vec{\nabla} \vec{u}$  la matrice dont les coefficients sont  $(\vec{\nabla} \vec{u})_{ij} = \frac{\partial u_i}{\partial x_j}$ .

Par ailleurs, on désignera par  $A : B$  le produit scalaire matriciel  $\sum_{i,j} a_{ij} b_{ij}$ .

Il vient alors, par formule de Green sur le terme  $\int_{\Omega} -(\Delta \vec{u}) \cdot \vec{v} \, dx$  :

$$\int_{\Omega} (\vec{\nabla} \vec{u}) : (\vec{\nabla} \vec{v}) \, dx - \int_{\partial\Omega} ((\vec{\nabla} \vec{u}) \vec{n}) \cdot \vec{v} \, dx + \int_{\Omega} \vec{\nabla} p \cdot \vec{v} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (3)$$

Puis sur le terme  $\int_{\Omega} \vec{\nabla} p \cdot \vec{v} \, dx$  :

$$\int_{\Omega} (\vec{\nabla} \vec{u}) : (\vec{\nabla} \vec{v}) \, dx - \int_{\partial\Omega} ((\vec{\nabla} \vec{u}) \vec{n}) \cdot \vec{v} \, dx - \int_{\Omega} p \operatorname{div}(\vec{v}) \, dx + \int_{\partial\Omega} p \vec{n} \cdot \vec{v} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (4)$$

Les conditions de Dirichlet  $\vec{v} = \vec{0}$  sur  $\Gamma_D$  donnent :

$$\int_{\Omega} (\vec{\nabla} \vec{u}) : (\vec{\nabla} \vec{v}) \, dx - \int_{\Gamma_N} ((\vec{\nabla} \vec{u}) \vec{n}) \cdot \vec{v} \, dx - \int_{\Omega} p \operatorname{div}(\vec{v}) \, dx + \int_{\Gamma_N} p \vec{n} \cdot \vec{v} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (5)$$

On utilise enfin les conditions aux bords de Neumann où  $\sigma = \sigma_1$  :

$$\nabla \vec{u} = p \text{ Id sur } \Gamma_N \quad (6)$$

Pour obtenir l'équation :

$$\boxed{\int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{v}) \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx + \int_{\Omega} p \operatorname{div}(\vec{v}) \, dx} \quad (7)$$

On définit alors la forme bilinéaire symétrique :

$$a_1 : \begin{array}{ccc} V^2 & \rightarrow & \mathbf{R} \\ \vec{u}, \vec{v} & \mapsto & \int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{v}) \, dx \end{array} \quad (8)$$

et la forme linéaire :

$$l_1 : \begin{array}{ccc} V & \rightarrow & \mathbf{R} \\ \vec{v} & \mapsto & \int_{\Omega} \vec{f} \cdot \vec{v} \, dx + \int_{\Omega} p \operatorname{div}(\vec{v}) \, dx \end{array} \quad (9)$$

pour obtenir la formulation variationnelle (FV0) :

$$\boxed{(FV0) \quad \left| \quad \begin{array}{l} \text{Trouver } \vec{u} \text{ dans } V \text{ tel que :} \\ \forall \vec{v} \in V \quad a_1(\vec{u}, \vec{v}) = l_1(\vec{v}) \end{array} \right.} \quad (10)$$

Le problème peut ensuite être résolu en appliquant le théorème de Lax-Milgram. Jusqu'ici nous n'avons pas eu besoin de définir la norme  $\|\cdot\|_V$  de l'espace ambiant  $V$ . Puisque ce dernier ne contient pas de fonctions constantes non nulles sous l'hypothèse  $\Gamma_D$  **non vide et**  $\Omega$  **connexe**, nous disposons de l'inégalité de Poincaré. Ainsi, la norme associée à l'espace  $H_0^1(\Omega) = \{h \in H^1(\Omega) \mid h = 0 \text{ sur } \Gamma_D\}$ , définie comme étant la norme  $L^2$  du gradient :  $h \mapsto \left( \int_{\Omega} \|\vec{\nabla} h\|_{R^d}^2 \, dx \right)^{1/2}$  est bien une norme sur  $\Omega$  où elle contrôle la norme  $\|\cdot\|_{L^2(\Omega)}$ . On définit ainsi la norme  $\|\cdot\|_V$  sur  $V$  par :

$$\forall \vec{u} \in V, \|\vec{u}\|_V^2 = \sum_{i=1}^d \|u_i\|_{H_0^1(\Omega)}^2 = \sum_{i=1}^d \left( \int_{\Omega} \|\vec{\nabla} u_i\|_{R^d}^2 \, dx \right)^{1/2} \quad (11)$$

Pour appliquer le théorème de Lax-Milgram, il faut vérifier plusieurs points :

- $(V, \|\cdot\|_V)$  est bien un espace de Hilbert car la condition de divergence nulle le laisse fermé dans l'espace  $((H_0^1(\Omega))^d, \|\cdot\|_V)$  qui lui est hilbertien.
- $a_1$  est continue sur  $V^2$  pour  $\|\cdot\|_V$  par inégalité triangulaire et inégalité de Cauchy-Schwarz sur les intégrales puis sur les sommes.
- $a_1$  est coercive sur  $V^2$  pour  $\|\cdot\|_V$  par un calcul direct.
- $l_1$  est continue sur  $V$  pour  $\|\cdot\|_V$  par inégalité triangulaire, par inégalité de Cauchy-Schwarz sur les intégrales et enfin par continuité de l'application trace sur  $H^1(\Omega)$  vers  $L^2(\partial\Omega)$  pour leur norme respective et enfin par inégalité de Poincaré pour se ramener de la norme  $\|\cdot\|_{H^1(\Omega)}$  à la norme  $\|\cdot\|_V$ .

**Question 2.** Dans le cas d'une discrétisation P1 sans souci de divergence, on dispose pour chaque triangle  $T$  du maillage de trois fonctions de base de P1 notées  $\lambda_1^T$ ,  $\lambda_2^T$  et  $\lambda_3^T$  affines par morceaux sur  $\Omega$  et affines sur  $T$ , qui valent chacune 1 sur leur nœud respectif et 0 sur le bord opposé à celui-ci. Si l'on veut par la suite respecter les conditions de divergence nulle il faut se restreindre à deux fonctions de base par triangle que l'on pourrait choisir en trouvant 2 combinaisons linéaires des  $\lambda_i^T$  respectant chacune la condition de divergence. Cela nécessite toutefois de recalculer ces combinaisons pour chaque triangle du maillage ce qui peut être **rapidement couteux en calculs**, surtout si l'on est amené à modifier le maillage au cours d'une simulation comme effectué à partir de la question 11.

**Question 3.** On ajoute maintenant la pression dans l'inconnue qui devient alors le couple  $(\vec{u}, p)$ . L'espace associé  $W$  devient alors :

$$W = \{(\vec{v}, q) \in (H^1(\Omega))^{d+1} \mid \vec{v} = \vec{0} \text{ sur } \Gamma_D\} \quad (12)$$

Le choix de la norme associée à  $W$  n'est pour l'instant pas évident et nécessite de voir le rôle des composantes  $p$  et  $q$  dans la formulation variationnelle finale.

Soit donc  $(\vec{v}, q)$  dans  $W$ . En partant de l'équation (1) du sujet, on obtient sur  $\Omega$  :

$$-\Delta \vec{u} \cdot \vec{v} + \vec{\nabla} p \cdot \vec{v} = \vec{f} \cdot \vec{v} \quad (13)$$

$$-q \operatorname{div}(\vec{u}) = 0 \quad (14)$$

Puis, en intégrant la somme de ces deux équations :

$$\int_{\Omega} (-\Delta \vec{u} \cdot \vec{v} + \vec{\nabla} p \cdot \vec{v} - q \operatorname{div}(\vec{u})) \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (15)$$

On applique l'équation (7), valable dans le cas  $\sigma = \sigma_1$  pour obtenir :

$$\boxed{\int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Omega} p \operatorname{div} \vec{v} \, dx - \int_{\Omega} q \operatorname{div} \vec{u} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx} \quad (16)$$

On définit alors la forme bilinéaire **symétrique** :

$$a_2 : \begin{array}{ccc} W^2 & \rightarrow & \mathbf{R} \\ (\vec{u}, p), (\vec{v}, q) & \mapsto & \int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Omega} p \operatorname{div} \vec{v} \, dx - \int_{\Omega} q \operatorname{div} \vec{u} \, dx \end{array} \quad (17)$$

et la forme linéaire cette fois-ci indépendante de la pression :

$$l : \begin{array}{ccc} W & \rightarrow & \mathbf{R} \\ (\vec{v}, q) & \mapsto & \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \end{array} \quad (18)$$

pour obtenir la formulation variationnelle (FV1) :

$$(FV1) \quad \left| \begin{array}{l} \text{Trouver } (\vec{u}, p) \text{ dans } W \text{ tels que :} \\ \forall (\vec{v}, q) \in W \quad a_2((\vec{u}, p), (\vec{v}, q)) = l((\vec{v}, q)) \end{array} \right. \quad (19)$$

Afin de définir complètement l'espace de Hilbert de (FV1) il faut le munir d'une norme. Celle-ci doit permettre au moins la continuité des deux formes  $a_2$  et  $l$ . On voit que l'ajout de la pression dans l'inconnue intervient dans le problème variationnel seulement au niveau de la forme bilinéaire  $a_2$ , sans ses dérivées. Aussi nous choisissons la norme suivante sur  $W$  :

$$\|(\vec{u}, p)\|_V^2 = \|p\|_{L^2(\Omega)}^2 + \sum_{i=1}^d \|u_i\|_{H_0^1(\Omega)}^2 = \|p\|_{L^2(\Omega)}^2 + \sum_{i=1}^d \left( \int_{\Omega} \|\vec{\nabla} u_i\|_{\mathbb{R}^d}^2 dx \right)^{1/2} \quad (20)$$

La forme bilinéaire  $a_2$  est clairement symétrique. Ainsi le problème de l'affaissement visqueux pour des contraintes du type  $\sigma = \sigma_1$  admet une **interprétation énergétique** :

$$E_2 : (\vec{u}, p) \in W \mapsto \frac{1}{2} \int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{u}) dx - \int_{\Omega} p \operatorname{div} \vec{u} dx - \int_{\Omega} \vec{f} \cdot \vec{u} dx \quad (21)$$

**Question 4.** Soit  $\vec{u}$  dans  $C^2(\Omega, \mathbb{R}^d)$  un champ de vecteurs à divergence nulle,  $p$  dans  $C^1(\Omega, \mathbb{R})$  et  $i$  dans  $\llbracket 1, d \rrbracket$ . Au vu de la régularité de  $\vec{u}$  et  $p$ , la quantité  $-\sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij}$  existe et vaut :

$$-\sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij} = \sum_{j=1}^d \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - p \delta_{ij} \right) \quad (22)$$

$$-\sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij} = \frac{\partial p}{\partial x_i} - \sum_{j=1}^d \frac{\partial^2 u_i}{\partial x_j^2} - \sum_{j=1}^d \frac{\partial}{\partial x_j} \frac{\partial u_j}{\partial x_i} \quad (23)$$

Or, par théorème de Schwarz,  $\vec{u}$  étant de classe  $C^2$ , on a :  $\forall j \in \llbracket 1, d \rrbracket \quad \frac{\partial}{\partial x_j} \frac{\partial u_j}{\partial x_i} = \frac{\partial}{\partial x_i} \frac{\partial u_j}{\partial x_j}$ . Ainsi :

$$-\sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij} = \frac{\partial p}{\partial x_i} - \Delta u_i - \frac{\partial}{\partial x_i} \operatorname{div}(\vec{u}) \quad (24)$$

Et donc :

$$-\sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij} = \frac{\partial p}{\partial x_i} - \Delta u_i \quad (25)$$

**Question 5.** Avec le résultat précédent, en considérant le gradient  $\vec{\nabla}$  comme un vecteur colonne tel que  $(\vec{\nabla})^T = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d})$  on obtient :

$$\sigma_2 \vec{\nabla} = \left( (\vec{\nabla})^T \sigma_2^T \right)^T = \sum_{i=1}^d \sum_{j=1}^d \frac{\partial}{\partial x_j} \sigma_{2,ij} \vec{e}_i = \Delta \vec{u} - \vec{\nabla} p \quad (26)$$

Or :

$$\sigma_2 \vec{\nabla} = \left( (\vec{\nabla})^T \sigma_2^T \right)^T = \left( (\vec{\nabla})^T (\nabla^T \vec{u} + \nabla \vec{u} - p Id) \right)^T = \left( (\vec{\nabla})^T \nabla^T \vec{u} \right)^T + \Delta \vec{u} - \vec{\nabla} p \quad (27)$$

Donc sur  $\Omega$  :

$$\Delta^* \vec{u} := \left( (\vec{\nabla})^T \nabla^T \vec{u} \right)^T = \vec{0} \quad (28)$$

Ainsi, par formule de Green :

$$0 = \int_{\Omega} \Delta^* \vec{u} \cdot \vec{v} \, dx = - \int_{\Omega} (\nabla^T \vec{u}) : (\nabla \vec{v}) \, dx + \int_{\partial\Omega} \left( (\nabla^T \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx \quad (29)$$

Soit, avec les conditions de Dirichlet :

$$\int_{\Gamma_N} \left( (\nabla^T \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx = \int_{\Omega} (\nabla^T \vec{u}) : (\nabla \vec{v}) \, dx \quad (30)$$

Et donc dans le cas  $\sigma = \sigma_2$  on a :

$$- \int_{\Gamma_N} \left( (\nabla \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx = \int_{\Gamma_N} \left( (\nabla^T \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx - \int_{\Gamma_N} p \vec{n} \cdot \vec{v} \, dx \quad (31)$$

$$- \int_{\Gamma_N} \left( (\nabla \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx = \int_{\Omega} (\nabla^T \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Gamma_N} p \vec{n} \cdot \vec{v} \, dx \quad (32)$$

On reprend l'équation (5) obtenue en question 1 avant que l'on ait traduit les conditions de Neumann et l'on additionne l'équation (14) intégrée sur  $\Omega$ . Soit  $(\vec{v}, q)$  dans  $W$  on a :

$$\int_{\Omega} (\nabla \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Gamma_N} \left( (\nabla \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx + \int_{\Gamma_N} p \vec{n} \cdot \vec{v} \, dx - \int_{\Omega} p \operatorname{div} \vec{v} \, dx - \int_{\Omega} q \operatorname{div} \vec{u} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx \quad (33)$$

Finalement, on utilise l'équation (32) :

$$\boxed{\int_{\Omega} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Omega} p \operatorname{div} \vec{v} \, dx - \int_{\Omega} q \operatorname{div} \vec{u} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx} \quad (34)$$

La forme linéaire reste la même qu'en question 3. Toutefois, la forme bilinéaire devient :

$$a_3^* : \begin{array}{ccc} W^2 & \rightarrow & \mathbf{R} \\ (\vec{u}, p), (\vec{v}, q) & \mapsto & \int_{\Omega} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v}) \, dx - \int_{\Omega} p \, \text{div} \vec{v} \, dx - \int_{\Omega} q \, \text{div} \vec{u} \, dx \end{array} \quad (35)$$

Ce qui donne la formulation variationnelle intermédiaire (FV2\*) :

$$(FV2*) \quad \left| \quad \begin{array}{l} \text{Trouver } (\vec{u}, p) \text{ dans } W \text{ tels que :} \\ \forall (\vec{v}, q) \in W \quad a_3^*((\vec{u}, p), (\vec{v}, q)) = l((\vec{v}, q)) \end{array} \right. \quad (36)$$

La forme bilinéaire  $a_3^*$  ne semble pas être symétrique en apparence. Il n'y donc a priori pas d'interprétation énergétique associée à la contrainte  $\sigma = \sigma_2$ . Par ailleurs,  $\mathbf{a}_3^*$  n'est pas **coercive**. En effet, dans l'expression de la quantité  $a_3^*((\vec{u}, p), (\vec{u}, p))$  interviennent deux intégrales : l'une dépend exclusivement des dérivées de  $\vec{u}$  et l'autre met en jeu à la fois  $p$  et  $\text{div}(\vec{u})$ . Aussi est-il possible d'annuler la quantité  $a_3^*((\vec{u}, p), (\vec{u}, p))$  en choisissant convenablement la pression  $p$  tout en ayant fixé la vitesse  $\vec{u}$  non nulle de façon à ce que la quantité  $\|(\vec{u}, p)\|_V$  soit elle aussi non nulle. Il n'est donc pas possible d'appliquer directement le théorème de Lax-Milgram.

**Question 6.** Soit  $A$  et  $B$ , deux matrices carrées  $d \times d$ . Considérons  $A$  symétrique et  $B$  antisymétrique. On a alors :

$$2 \sum_{i,j=1}^d A_{ij} B_{ij} = \sum_{i=1}^d \sum_{j=1}^d A_{ij} B_{ij} + \sum_{i=1}^d \sum_{j=1}^d A_{ji} B_{ji} \quad (37)$$

$$2 \sum_{i,j=1}^d A_{ij} B_{ij} = \sum_{i=1}^d \sum_{j=1}^d A_{ij} B_{ij} - \sum_{i=1}^d \sum_{j=1}^d A_{ij} B_{ij} = 0 \quad (38)$$

d'où

$$\boxed{A : B = \sum_{i,j=1}^d A_{ij} B_{ij} = 0} \quad (39)$$

Ainsi, pour tout  $\vec{u}$  et  $\vec{v}$  dans  $H^1(\Omega)^d$  on a :

$$(\nabla \vec{u} + \nabla^T \vec{u}) : \nabla \vec{v} = (\nabla \vec{u} + \nabla^T \vec{u}) : \left( \frac{1}{2}(\nabla \vec{v} + \nabla^T \vec{v}) + \frac{1}{2}(\nabla \vec{v} - \nabla^T \vec{v}) \right) \quad (40)$$

$$(\nabla \vec{u} + \nabla^T \vec{u}) : \nabla \vec{v} = \frac{1}{2}(\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v} + \nabla^T \vec{v}) + \frac{1}{2}(\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v} - \nabla^T \vec{v}) \quad (41)$$

Or, les matrices  $(\nabla \vec{u} + \nabla^T \vec{u})$  et  $(\nabla \vec{v} - \nabla^T \vec{v})$  sont respectivement symétrique et anti-symétrique. En appliquant le résultat (39) on obtient :

$$(\nabla \vec{u} + \nabla^T \vec{u}) : \nabla \vec{v} = \frac{1}{2} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v} + \nabla^T \vec{v}) \quad (42)$$

Ainsi la formulation variationnelle précédente peut se reformuler comme suit :

$$\boxed{\frac{1}{2} \int_{\Omega} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v} + \nabla^T \vec{v}) \, dx - \int_{\Omega} p \, \text{div} \vec{v} \, dx - \int_{\Omega} q \, \text{div} \vec{u} \, dx = \int_{\Omega} \vec{f} \cdot \vec{v} \, dx} \quad (43)$$

Et donc la forme bilinéaire  $a_3^*$  est en fait égale à la forme bilinéaire **symétrique** suivante :

$$a_3 : \begin{array}{ccc} W^2 & \rightarrow & \mathbf{R} \\ (\vec{u}, p), (\vec{v}, q) & \mapsto & \frac{1}{2} \int_{\Omega} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{v} + \nabla^T \vec{v}) \, dx - \int_{\Omega} p \, \text{div} \vec{v} \, dx - \int_{\Omega} q \, \text{div} \vec{u} \, dx \end{array} \quad (44)$$

Ce qui donne la formulation variationnelle (FV2) :

$$\boxed{(FV2) \quad \left| \begin{array}{l} \text{Trouver } (\vec{u}, p) \text{ dans } W \text{ tels que :} \\ \forall (\vec{v}, q) \in W \quad a_3((\vec{u}, p), (\vec{v}, q)) = l((\vec{v}, q)) \end{array} \right.} \quad (45)$$

La forme  $a_3$  n'est toujours pas coercive, d'après le raisonnement effectué en question 5. Elle est cependant symétrique, ce qui nous permet d'interpréter le problème en terme de minimisation énergétique sur  $W$  :

$$\boxed{E_3 : (\vec{u}, p) \in W \mapsto \frac{1}{4} \int_{\Omega} (\nabla \vec{u} + \nabla^T \vec{u}) : (\nabla \vec{u} + \nabla^T \vec{u}) \, dx - \int_{\Omega} p \, \text{div} \vec{u} \, dx - \int_{\Omega} \vec{f} \cdot \vec{u} \, dx} \quad (46)$$

$\vec{u}$  et  $p$  sont bien tous les deux des inconnues et chacune apporte une contribution à l'énergie  $E_3$ . Afin de simplifier la méthode des éléments finis, on a choisi de relacher la condition de divergence nulle sur  $\vec{u}$  dans de travail  $W$ , mais celle-ci se retrouve heureusement au travers du terme  $\frac{1}{2} \int_{\Omega} p \, \text{div} \vec{u}$  dans l'énergie.

**Question 7.** Les formes bilinéaires  $a_2$  et  $a_3$  issues des formulations variationnelles respectives (FV1) et (FV2) sont différentes de :

$$\forall (\vec{u}, p), (\vec{v}, q) \in W^2 \quad (a_3 - a_2)((\vec{u}, p), (\vec{v}, q)) = (a_3^* - a_2)((\vec{u}, p), (\vec{v}, q)) \quad (47)$$

$$(a_3 - a_2)((\vec{u}, p), (\vec{v}, q)) = \int_{\Omega} (\nabla^T \vec{u}) : (\nabla \vec{v}) \, dx \quad (48)$$

Soit, d'après l'équation (30) :

$$(a_3 - a_2)((\vec{u}, p), (\vec{v}, q)) = \int_{\Gamma_N} \left( (\nabla^T \vec{u}) \vec{n} \right) \cdot \vec{v} \, dx \quad (49)$$

Si l'une des solutions de (FV1) ou (FV2), notée  $\vec{u}^*$ , est telle que  $(\nabla^T \vec{u}^*) \vec{n} = \vec{0}$  presque partout sur  $\Gamma_N$  ce qui est équivalent à  $\int_{\Gamma_N} \|(\nabla^T \vec{u}^*) \vec{n}\|_{\mathbb{R}^d}^2 \, dx = 0$  alors elle est solution des deux formulations variationnelles. Réciproquement, si  $\int_{\Gamma_N} \|(\nabla^T \vec{u}^*) \vec{n}\|_{\mathbb{R}^d}^2 \, dx \neq 0$ , alors les formes bilinéaires  $a_2$  et  $a_3$  sont strictement différentes pour certains  $\vec{v}$  et ne peuvent être égales simultanément à la même forme linéaire  $\int_{\Omega} \vec{f} \cdot \vec{v} \, dx$  et donc  $\vec{u}^*$  n'est pas solution de l'autre formulation variationnelle. De façon plus synthétique :

$(FV1) \text{ et } (FV2) \text{ sont équivalentes } \iff$ $\forall \vec{u}^* \text{ solution de } (FV1) \text{ ou } (FV2) \quad (\nabla^T \vec{u}^*) \vec{n} = \vec{0} \text{ p.p sur } \Gamma_N$	(50)
--	------

En particulier, dès que  $\Gamma_N$  est de mesure nulle, par exemple  $\Gamma_N = \emptyset$ , les formulations variationnelles (FV1) et (FV2) sont équivalentes.



## 2 Implémentation numérique

**Question 8.** On implémente en *FreeFem++* les deux formulations variationnelles. On se place dans le cadre de la question 10 où  $\Omega$  est le carré unité  $\Omega = ]0, 1[^2$  et  $\vec{f} = -\vec{e}_2$  représente la force de gravité.

```
// Constants
// Dimension constants
real L          = 1;
real l          = 1;

// Discretization constants
int Nx          = 8;
int Ny          = 8;

// Triangular Mesh with references

int dirichlet = 1;
int neumann = 0;
int[int] labs = [ dirichlet , neumann , neumann , neumann ];
mesh Th = square(Nx,Ny,[L*x,l*y],label=labs);
plot(Th, wait=1, cmm="Mesh");

// FE Spaces and functions
fespace Vh(Th,P2);
Vh u1,u2,v1,v2;

fespace Ph(Th,P1);
Ph p,q;

// Variational formulations

//FV1
problem LaplaceAbsorbDirichlet1 ([u1,u2,p],[v1,v2,q])
= int2d(Th)(dx(u1)*dx(v1) + dy(u1)*dy(v1) \
+ dx(u2)*dx(v2) + dy(u2)*dy(v2))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2)))
- int2d(Th)(-v2)
+ on(dirichlet, u1=0, u2=0);

//FV2
problem LaplaceAbsorbDirichlet2 ([u1,u2,p],[v1,v2,q])
= int2d(Th)(2*dx(u1)*dx(v1) + 2*dy(u2)*dy(v2) + dy(u1)*dy(v1)
```

```

+ dx(u2)*dx(v2) + dy(u1)*dx(v2) + dx(u2)*dy(v1))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2)))
- int2d(Th)(-v2)
+ on(dirichlet , u1=0, u2=0);

```

```
// Solution
```

```
LaplaceAbsorbDirichlet1;
```

```

plot([u1,u2],coef=0.5,wait=1,fill=1,value=10, \
cmm="Vitesse_par_FV1",ps="FV1_vitesse.eps");
plot(p,fill=1,wait=1,value=10, \
cmm="Pression_par_FV1",ps="FV1_pression.eps");

```

```
LaplaceAbsorbDirichlet2;
```

```

plot([u1,u2],coef=0.5,wait=1,fill=1,value=10, \
cmm="Vitesse_par_FV2",ps="FV2_vitesse.eps");
plot(p,fill=1,wait=1,value=10, \
cmm="Pression_par_FV2",ps="FV2_pression.eps");

```

**Question 9.** On se place dans un cas particulier. Soit  $\Omega = ]0, 1[^2$ , on choisit  $\vec{f} = \vec{0}$  sur  $\Omega$ . On impose  $\vec{u} = \vec{w}$  sur  $\Gamma_D = \partial\Omega$  où  $\vec{w}$  est donné par :

$$\vec{w} = w_1 \vec{e}_1 + w_2 \vec{e}_2 \quad (51)$$

Avec, pour  $(x, y)$  dans  $\Omega$  :

$$\begin{aligned} w_1(x, y) &= \cos(\pi x) \sin(\pi y) \\ w_2(x, y) &= -\sin(\pi x) \cos(\pi y) \end{aligned} \quad (52)$$

On voit qu'en prenant  $p = \pi \sin(\pi x) \sin(\pi y)$  sur  $\Omega$  on a :

$$\begin{aligned} -\Delta \vec{w} + \vec{\nabla} p &= \vec{f} = \vec{0} \text{ sur } \Omega \\ -\text{div}(\vec{u}) &= 0 \text{ sur } \Omega \\ \vec{w} &= \vec{u} \text{ sur } \partial\Omega \end{aligned} \quad (53)$$

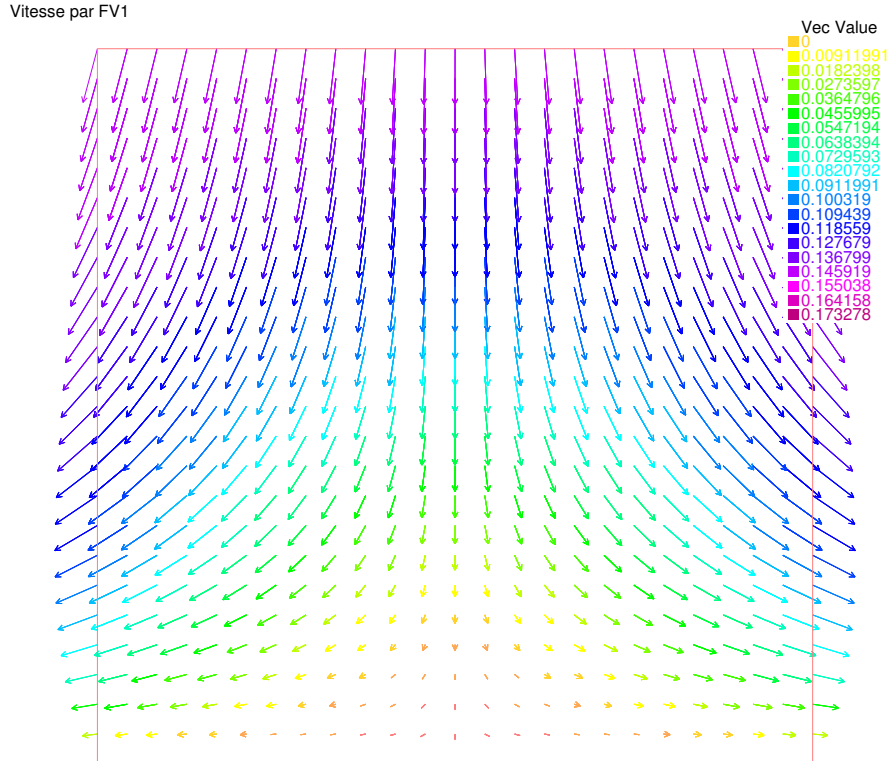
Donc le couple  $(\vec{w}, p)$  est une solution au problème particulier considéré ici. On regarde la convergence des différentes solutions calculées par FreeFem++ (code en annexe) selon la formulation variationnelle choisie ainsi que le mode d'élément finis choisi pour  $\vec{u}$  et  $p$ . **Puisque  $\Gamma_N$  est vide, le problème n'admet de solution unique pour la pression :** seul son gradient intervient dans le problème initial. Aussi nous étudierons la convergence

de la norme  $\|p\|_{H_0^1(\Omega)} = \|\vec{\nabla} p\|_{L^2}$  de la pression  $p$ . Pour un maillage donné on obtient via le code fourni en annexe :

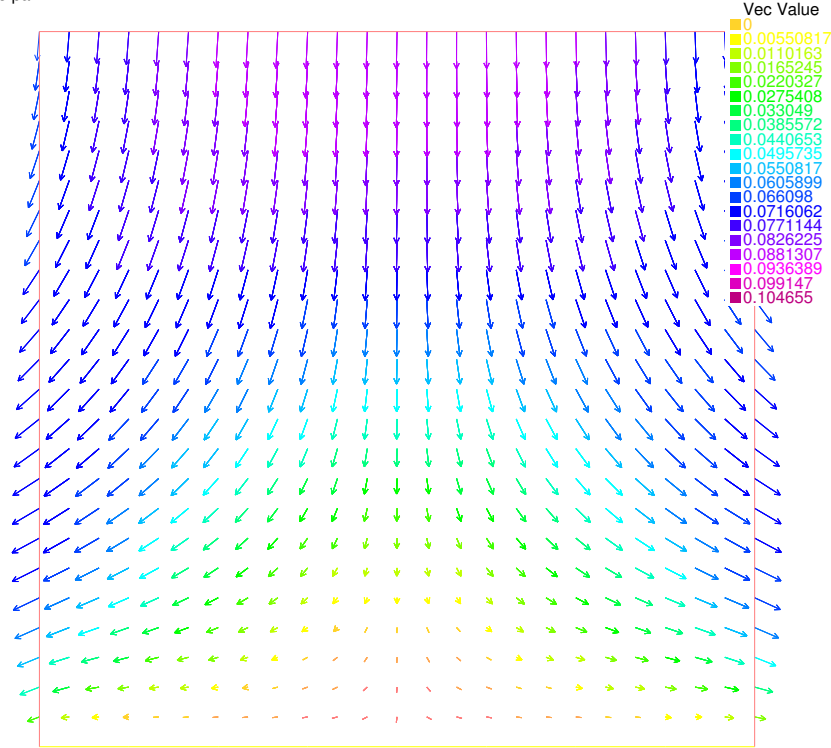
	$(P^1, P^0)$	$(P^1, P^1)$	$(P^1b, P^1)$	$(P^2, P^1)$
$\ \vec{u}_{FV1} - \vec{w}\ _{L^2(\Omega)^d}$	$4,98.10^3$	$9,74.10^{12}$	7,53	6,03
$\ \vec{u}_{FV2} - \vec{w}\ _{L^2(\Omega)^d}$	$1,14.10^{13}$	$2,80.10^{12}$	$2,7.10^{-2}$	$2,81.10^{-2}$
$\ div(\vec{u}_{FV1})\ _{L^2}$	$7,23.10^4$	$1,68.10^{14}$	$2,18.10^2$	$2,41.10^2$
$\ div(\vec{u}_{FV2})\ _{L^2}$	$1,58.10^{14}$	$3,62.10^{13}$	$1,49.10^{-1}$	$5,51.10^{-3}$
$\ \vec{\nabla} p_{FV1} - \vec{\nabla} p\ _{L^2(\Omega)^d}$	6,96	$3,64.10^{32}$	$1,15.10^4$	$7,51.10^3$
$\ \vec{\nabla} p_{FV2} - \vec{\nabla} p\ _{L^2(\Omega)^d}$	6,96	$8,29.10^{31}$	$1,99.10^1$	$1,67.10^1$

On voit aisément que **seuls les deux derniers choix**  $(P^1b, P^1)$  et  $(P^2, P^1)$  **semblent pertinents**. L'espace  $P^1 + \text{bulle}$  est l'espace  $P^1$  augmenté d'un degré de liberté pour chaque triangle  $T$ . Sur ce dernier, on a déjà les 3 fonctions de base de  $P^1$ , affines par morceaux sur  $\Omega$  et affines sur  $T$ , notées  $\lambda_1^T$ ,  $\lambda_2^T$  et  $\lambda_3^T$  qui valent chacune 1 sur leur nœud respectif et 0 sur le bord opposé à celui-ci. Le degré de liberté supplémentaire vient d'une quatrième fonction  $\lambda_b^T$ , appelée "bulle". Elle est nulle sur  $\partial T$  et positive sur  $T$ . Concentrée sur son barycentre, on choisit généralement la bulle  $\lambda_b^T = \lambda_1^T \lambda_2^T \lambda_3^T$ . Cette technique permet de donner du poids au barycentre de chaque triangle en plus de ses nœuds.

**Question 10.** La résolution du problème en éléments finis  $P^2$  pour les composantes de  $\vec{u}$  donne les résultats suivants selon la formulation variationnelle choisie :



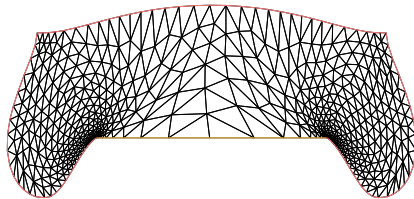
Vitesse par FV2



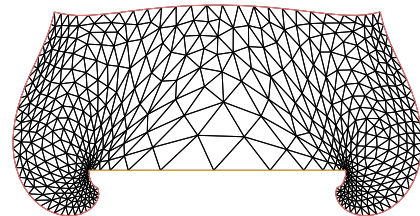
On observe des vitesses plus élevées pour (FV1) que pour (FV2), d'un facteur 1,7. Cela implique donc un gradient de vitesse plus grand en moyenne, puisque la condition de Dirichlet impose une vitesse nulle au bord inférieur. On peut interpréter cela par le fait que la présence du tenseur gradient transposé dans les contraintes de Neumann permet de mieux amortir la condition de Dirichlet sur le champ de vitesse en influant plus sur la vitesse aux points lointains (ici le bord supérieur).

**Question 11.** Via la commande `FreeFem++ movemesh`, on peut simuler l'écoulement du fluide. On peut alors comparer l'évolution de l'écoulement selon les contraintes appliquées :  $\sigma_1$  ou  $\sigma_2$ . Le code associé aux résultats suivants est en annexe. Dans la simulation, le remaillage automatique est déjà pris en compte.

Ecoulement apres 5 secondes avec FV1



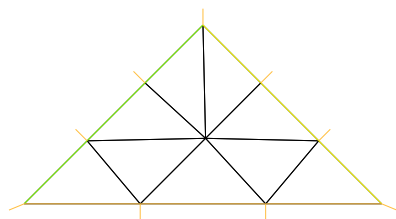
Ecoulement apres 5 secondes avec FV2



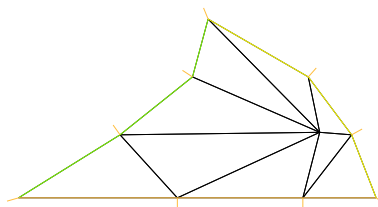
On aboutit à deux formes finales clairement différentes. Globalement le fluide "s'affaisse" moins sous  $\sigma_2$  que sous  $\sigma_1$  : la vitesse est plus faible comme vu en question 10. Mais localement, le fluide se déforme davantage sous  $\sigma_2$ .

**Question 12.** Si l'on applique uniquement des mouvements de maillage lors de la simulation, il se peut que, par l'écoulement du fluide, le maillage se torde, de sorte qu'au moins un triangle du maillage se trouve inversé. C'est-à-dire que lors du **mouvement non homogène** des trois points de ce triangle de leur position initiale vers leur position finale, ces derniers passent par une phase d'alignement. Il y a alors plusieurs triangles du maillage dont les bords se croisent, ce qui contredit la définition même d'un maillage. Dans ce cas, FreeFem++ nous retourne l'erreur : `Exec error : Error move mesh triangles was reverse`. On illustre ci-dessous une transformation non-homogène du maillage. L'erreur arriverait si l'on faisait sortir le nœud central du bord triangulaire déformé.

Maillage initial



Transformation non homogène du maillage



// code question 12

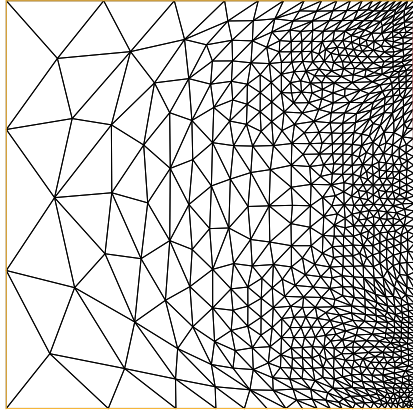
```
border a(t=0,1){x=t;y=0;label = 1;};
border b(t=0,1){x=1-t/2;y=t/2;label = 2;};
border c(t=0,1){x=0.5-t/2;y=0.5-t/2;label = 3;};
mesh Th = buildmesh (a(3)+b(3)+c(3));
plot(Th, wait=1, cmm="Maillage_initial", ps="q12_forme.eps");
```

```
func f = 0.35*exp(-10*(x-0.5)^2-20*(y-0.18)^2);
fespace Vh(Th,P2);
Vh fh = f;
```

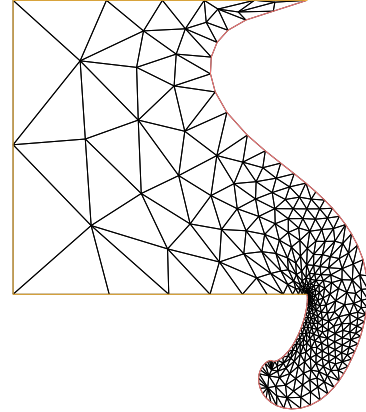
```
Th = movemesh(Th,[x,y]+[f,0]);
plot(Th, wait=1, cmm="Transformation_non_homogene_du_maillage", ps="q12_deforme.eps");
```

**Question 13.** Les codes utilisés pour les simulations suivantes sont en annexe. Lorsque le bord de Dirichlet correspond à tous les bords du carré sauf le bord droit on obtient le résultat suivant sous les contraintes  $\sigma = \sigma_2$  :

Debut d'écoulement avec bord droit libre

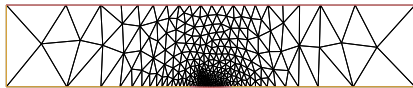


Écoulement par le bord droit après 15 secondes

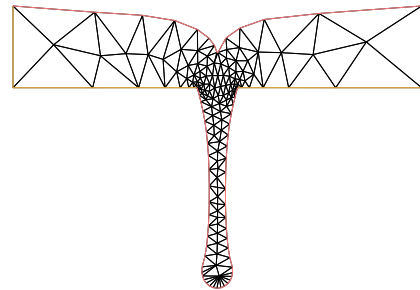


On peut également simuler l'écoulement d'une goutte de fluide depuis un petit trou dans le plafond sous les contraintes  $\sigma = \sigma_2$  :

Debut de l'écoulement au travers d'un plafond trou

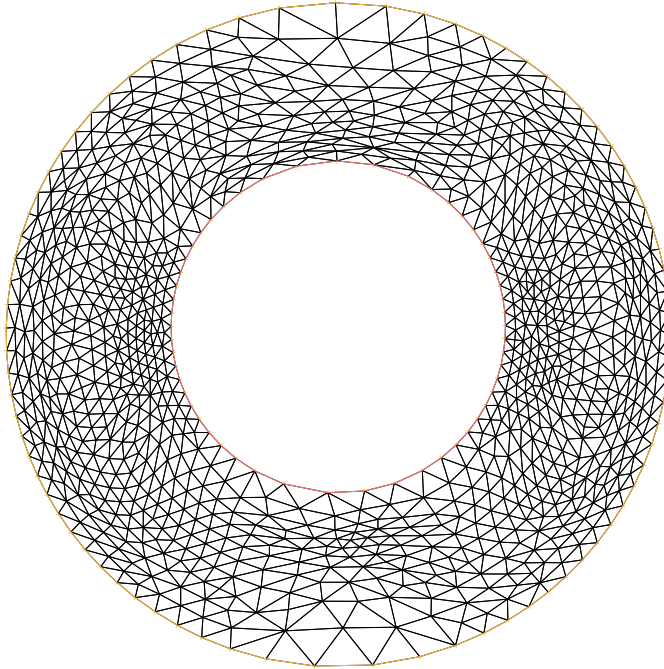


Écoulement au travers d'un plafond trou après 50 secondes

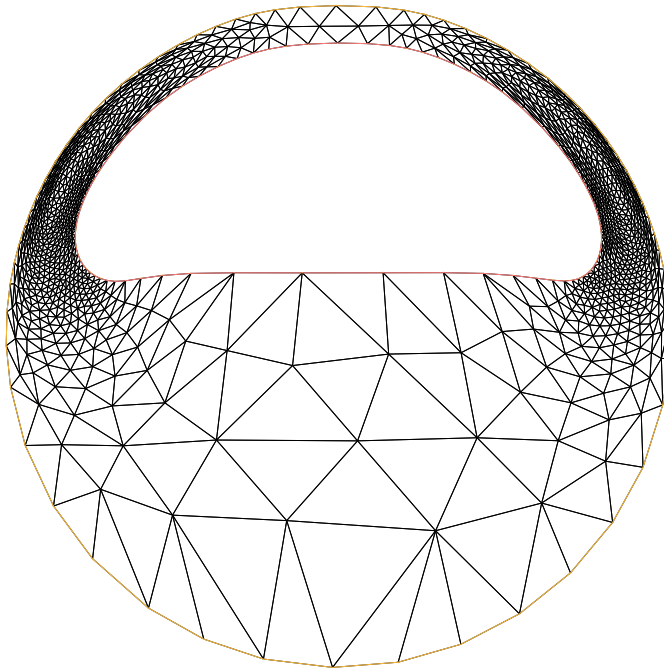


*Ou encore la présence d'une bulle d'air emprisonnée au milieu du fluide, sous les contraintes  $\sigma = \sigma_1$  qui fournissent le même résultat que les contraintes  $\sigma = \sigma_2$  mais étant beaucoup plus rapides à résoudre pour FreeFem++ :*

Bulle d'air initiale



Bulle d'air apres 100 secondes



### 3 Annexe - codes FreeFem++

— Code de la question 9 :

```
// Constants
// Dimension constants
real L          = 1;
real l          = 1;

// Discretization constants
int Nx          = 25;
int Ny          = 25;

// Triangular Mesh with references

int dirichlet = 1;
int[int] labs = [dirichlet, dirichlet, dirichlet, dirichlet];
mesh Th = square(Nx, Ny, [L*x, l*y], label=labs);

real alpha = pi;
func a1 = cos(alpha*x)*sin(alpha*y);
func a2 = -sin(alpha*x)*cos(alpha*y);

// FE Spaces and functions
fespace Vh(Th, P2);
Vh u1, u2, v1, v2;
Vh a1h = a1;
Vh a2h = a2;

fespace Ph(Th, P1);
Ph p, q;

// Variationnal formulations

//FV1
problem LaplaceAbsorbDirichlet1([u1, u2, p], [v1, v2, q])
= int2d(Th)(dx(u1)*dx(v1) + dy(u1)*dy(v1)
+ dx(u2)*dx(v2) + dy(u2)*dy(v2))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2))) // pas de forces
+ on(dirichlet, u1=a1h, u2=a1h);

//FV2
```



```

problem LaplaceAbsorbDirichlet2 ([u1,u2,p],[v1,v2,q])
= int2d(Th)(2*dx(u1)*dx(v1) + 2*dy(u2)*dy(v2) + dy(u1)*dy(v1)
+ dx(u2)*dx(v2) + dy(u1)*dx(v2) + dx(u2)*dy(v1))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2))) // pas de forces
+ on(dirichlet, u1=a1h, u2=a2h);

// Solution
LaplaceAbsorbDirichlet1;

plot([u1,u2],coef=0.5,wait=1,fill=1,value=10);
plot([a1h,a2h],coef=0.5,wait=1,fill=1,value=10);

real err1 = (int2d(Th)((u1-a1h)^2+(u2-a2h)^2))^0.5; //erreur L2 de u
real err2 = (int2d(Th)((dx(p)+(alpha^2)*a1h)^2
+(dy(p)-(alpha^2)*a2h)^2))^0.5; //erreur L2 de grad p
real err3 = (int2d(Th)((dx(u1)+dy(u2))^2))^0.5; //norme L2 de div(u)
cout << "*****" << endl;
cout << "Erreur_L2_de_u_pour_FV1:_ " << err1 << endl;
cout << "Erreur_L2_de_grad_p_pour_FV1:_ " << err2 << endl;
cout << "Norme_L2_de_div(u)_pour_FV1:_ " << err3 << endl;
cout << "*****" << endl;

LaplaceAbsorbDirichlet2;

err1 = (int2d(Th)((u1-a1h)^2+(u2-a2h)^2))^0.5; //erreur L2 de u
err2 = (int2d(Th)((dx(p)+(alpha^2)*a1h)^2
+(dy(p)-(alpha^2)*a2h)^2))^0.5; //erreur L2 de grad p
err3 = (int2d(Th)((dx(u1)+dy(u2))^2))^0.5; //norme L2 de div(u)
cout << "*****" << endl;
cout << "Erreur_L2_de_u_pour_FV2:_ " << err1 << endl;
cout << "Erreur_L2_de_grad_p_pour_FV2:_ " << err2 << endl;
cout << "Norme_L2_de_div(u)_pour_FV2:_ " << err3 << endl;
cout << "*****" << endl;
plot([u1,u2],coef=0.5,wait=1,fill=1,value=10);
plot([a1h,a2h],coef=0.5,wait=1,fill=1,value=10);

```

— Code de la question 11 :

```
// Constants
// Dimension constants

real L          = 1;
real l          = 1;

// Discretization constants
int Nx          = 20;
int Ny          = 10;
int steps       = 100;      // nombre d'etapes temporelles
real dt         = 0.05;     // intervalle de temps entre chaque etape

// Triangular Mesh with references
int dirichlet = 1;
int neumann = 0;
int[int] labs = [ dirichlet , neumann , neumann , neumann ];
mesh Th1 = square(Nx,Ny,[L*x,l*y],label=labs);
mesh Th2 = square(Nx,Ny,[L*x,l*y],label=labs);

// FE Spaces and functions
fespace Vh1(Th1,P2);
fespace Vh2(Th2,P2);
Vh1 u11,u12,v11,v12;
Vh2 u21,u22,v21,v22;

fespace Ph1(Th1,P1);
fespace Ph2(Th2,P1);
Ph1 p1,q1;
Ph2 p2,q2;

// Variationnal formulations

//FV1
problem LaplaceAbsorbDirichlet1([u11,u12,p1],[v11,v12,q1])
= int2d(Th1)(dx(u11)*dx(v11) + dy(u11)*dy(v11)
+ dx(u12)*dx(v12) + dy(u12)*dy(v12))
- int2d(Th1)(p1*(dx(v11)+dy(v12)))
- int2d(Th1)(q1*(dx(u11)+dy(u12)))
- int2d(Th1)(-v12)
+ on(dirichlet, u11=0 ,u12=0);

//FV2
```

```

problem LaplaceAbsorbDirichlet2 ([u21,u22,p2],[v21,v22,q2])
= int2d(Th2)(2*dx(u21)*dx(v21) + 2*dy(u22)*dy(v22) + dy(u21)*dy(v21)
+ dx(u22)*dx(v22) + dy(u21)*dx(v22) + dx(u22)*dy(v21))
- int2d(Th2)(p2*(dx(v21)+dy(v22)))
- int2d(Th2)(q2*(dx(u21)+dy(u22)))
- int2d(Th2)(-v22)
+ on(dirichlet , u21=0, u22=0);

```

```

for (int i=1;i<=steps;i++) // debut de la boucle sur les maillages
{
LaplaceAbsorbDirichlet1;
Th1 = movemesh(Th1,[x,y]+dt*[u11,u12]);
Th1 = adaptmesh(Th1,u11^2+u12^2);

LaplaceAbsorbDirichlet2;
Th2 = movemesh(Th2,[x,y]+dt*[u21,u22]);
Th2 = adaptmesh(Th2,u21^2+u22^2);

cout << "Step_" << i << "_done"<<endl;
}

```

```

plot(Th1, wait=1, cmm="Ecoulement_avec_FV1");
plot(Th2, wait=1, cmm="Ecoulement_avec_FV2");

```

— Code de la question 13 où le bord droit est libre :

```
// Constants
// Dimension constants
real L          = 1;
real l          = 1;

// Discretization constants
int Nx          = 20;
int Ny          = 20;
int steps       = 150;      // nombre d'etapes temporelles
real dt         = 0.1;      // intervalle de temps entre chaque etape

// Triangular Mesh with references
int dirichlet = 1;
int neumann = 0;
int[int] labs = [dirichlet, neumann, dirichlet, dirichlet];
mesh Th = square(Nx, Ny, [L*x, l*y], label=labs);

// FE Spaces and functions
fespace Vh(Th, P2);
Vh u1, u2, v1, v2;

fespace Ph(Th, P1);
Ph p, q;

// Variationnal formulations

//FV2
problem LaplaceAbsorbDirichlet2([u1, u2, p], [v1, v2, q])
= int2d(Th)(2*dx(u1)*dx(v1) + 2*dy(u2)*dy(v2) + dy(u1)*dy(v1)
+ dx(u2)*dx(v2) + dy(u1)*dx(v2) + dx(u2)*dy(v1))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2)))
- int2d(Th)(-v2)
+ on(dirichlet, u1=0, u2=0);

for (int i=1; i<=steps; i++) // debut de la boucle sur les maillages
{
  LaplaceAbsorbDirichlet2;
  Th = movemesh(Th, [x, y]+dt*[u1, u2]);
  Th = adaptmesh(Th, u1^2+u2^2);
  plot(Th, wait=0, cmm="Mesh");
  cout << "Step_" << i << "_done"<<endl;
}
```

— Code de la question 13 où le plafond est troué :

```
// Constants
// Dimension constants
real L          = 1;          // largeur du plafond
real l          = 0.2;        // epaisseur initiale du fluide
real h          = 0.1;        // largeur du trou

// Discretization constants
int steps       = 250;        // nombre d'etapes temporelles
real dt         = 0.2;        // intervalle de temps entre chaque etape

// Triangular Mesh with references
int dirichlet = 1;
int neumann = 0;
border top(t=L,0){x=t;y=l;label = neumann;};
border left(t=l,0){x=0;y=t;label = dirichlet;};
border right(t=0,l){x=L;y=t;label = dirichlet;};
border hole(t=(L-h)/2,(L+h)/2){x=t;y=0;label = neumann;};
border groundLeft(t=0,(L-h)/2){x=t;y=0;label = dirichlet;};
border groundRight(t=(L+h)/2,1){x=t;y=0;label = dirichlet;};
mesh Th = buildmesh (top(6) + right(4) + left(4)
+ hole(5) + groundLeft(4) + groundRight(4));

// FE Spaces and functions
fespace Vh(Th,P2);
Vh u1,u2,v1,v2;
fespace Ph(Th,P1);
Ph p,q;

// Variationnal formulation - FV2
problem LaplaceAbsorbDirichlet2([u1,u2,p],[v1,v2,q])
= int2d(Th)(2*dx(u1)*dx(v1) + 2*dy(u2)*dy(v2) + dy(u1)*dy(v1)
+ dx(u2)*dx(v2) + dy(u1)*dx(v2) + dx(u2)*dy(v1))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2)))
- int2d(Th)(-v2)
+ on(dirichlet, u1=0, u2=0);

for (int i=1;i<=steps;i++){ // debut de la boucle sur les maillages
LaplaceAbsorbDirichlet2;
Th = movemesh(Th,[x,y]+dt*[u1,u2]);
Th = adaptmesh(Th,u1^2+u2^2);
plot(Th, wait=0, cmm="Ecoulement_par_un_trou_dans_le_plafond...");
cout << "Step_" << i << "_done"<<endl;}
```

— Code de la question 13 avec la bulle d'air :

```
// Constants
// Dimension constants
real L          = 1;
real l          = 1;

// Discretization constants
int Nx          = 20;
int Ny          = 20;
int steps       = 100;      // nombre d'etapes temporelles
real dt         = 1;      // intervalle de temps entre chaque etape

// Triangular Mesh with references
int dirichlet = 1;
int neumann = 0;
border Cext(t=0,2*pi){x=cos(t); y=sin(t); label = dirichlet;};
border Cint(t=0,2*pi){x=0.5*cos(t); y=0.5*sin(t); label = neumann;};
mesh Th = buildmesh (Cext(50)+Cint(-50));

// FE Spaces and functions
fespace Vh(Th,P2);
Vh u1,u2,v1,v2;
fespace Ph(Th,P1);
Ph p,q;

// Variationnal formulations

//FV1
problem LaplaceAbsorbDirichlet1 ([u1,u2,p],[v1,v2,q])
= int2d(Th)(dx(u1)*dx(v1) + dy(u1)*dy(v1) \
+ dx(u2)*dx(v2) + dy(u2)*dy(v2))
- int2d(Th)(p*(dx(v1)+dy(v2)))
- int2d(Th)(q*(dx(u1)+dy(u2)))
- int2d(Th)(-v2)
+ on(dirichlet , u1=0 ,u2=0);

for (int i=1;i<=steps;i++) // debut de la boucle sur les maillages
{
LaplaceAbsorbDirichlet1; // calcul plus rapide avec ces contraintes
Th = movemesh(Th,[x,y]+dt*[u1,u2]);
Th = adaptmesh(Th,u1^2+u2^2);
plot(Th, wait=0, cmm="Mesh");
cout << "Step_" << i << "_done"<<endl;
}
```