

---

# Implémentation de l'équation de Navier-Stokes instationnaire 2D

---

*Auteur:*

Elise Grosjean

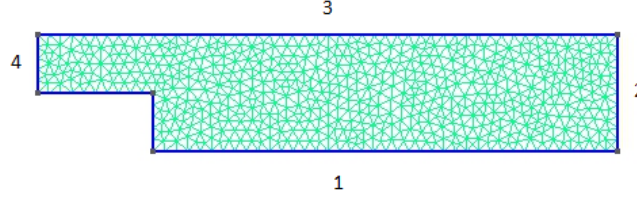
November 18, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Maillage sous gmsh</b>	<b>2</b>
<b>3</b>	<b>Formulation variationnelle</b>	<b>3</b>
<b>4</b>	<b>Discrétisation</b>	<b>4</b>
4.1	Fonctions de base de $P_1 - P_2$ . . . . .	4
4.2	Ecriture de $a_h$ et $b_h$ sans les termes de bords . . . . .	5
4.2.1	Transformation affine . . . . .	5
4.3	Les conditions aux bords . . . . .	7
<b>5</b>	<b>Freefem++</b>	<b>7</b>
<b>6</b>	<b>C++</b>	<b>8</b>
<b>7</b>	<b>Résultats</b>	<b>10</b>
7.1	Stokes . . . . .	10
7.1.1	Validation du code . . . . .	10
7.1.2	Problème de Stokes sur l'exemple de la marche . . . . .	10
7.2	Navier-Stokes . . . . .	11

# 1 Introduction

Le domaine de calcul  $\Omega \in \mathbf{R}^2$  est une marche descendante de  $\frac{1}{2}$  dans un canal de hauteur 1 avec une viscosité  $\nu$  de 1/100 et 1/400 avec un profil d'écoulement en entrant parabolique de vitesse maximal unitaire (bord 4), une vitesse nulle sur les parties haute et basse, et une condition de type Neumann en sortie c'est-à-dire ( $\nu \partial_n u = pn$ ).



Soit donc un fluide incompressible de vitesse  $\mathbf{u} \in H^1(\Omega)^2$  et de pression  $p \in L^2(\Omega)$  solution de l'équation de Navier-Stokes:

$$\begin{cases} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= 0 \quad (\Omega) \\ \nabla \cdot \mathbf{u} &= 0 \quad (\Omega) \\ \mathbf{u}|_{\Gamma} &= \mathbf{u}_{\Gamma} \quad (\partial\Omega) \end{cases}$$

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} \end{pmatrix}$$

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = u_1 \frac{\partial \mathbf{u}}{\partial x} + u_2 \frac{\partial \mathbf{u}}{\partial y}$$

$$\Delta \mathbf{u} = \nabla \cdot \nabla \mathbf{u} = \frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} = \begin{pmatrix} \frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} \\ \frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} \end{pmatrix}$$

$$\nabla \cdot \mathbf{u} = \frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y}$$

$$\nabla \cdot (\nabla u^T) = \nabla(\nabla \cdot u)$$

$$\nabla \cdot (\nabla \cdot u I) = \nabla(\nabla \cdot u)$$

## 2 Maillage sous gmsh

Le moteur CAO OpenCascade permet de mailler très simplement sous gmsh.

La ligne DefineConstant permet de modifier les valeurs (ici de largeur et d'hauteur) sur l'interface graphique. Pour définir un point dans gmsh, on utilise la syntaxe suivante:

Point(indice) = {x,y,z,h}; On fait pareil pour une ligne, et après avoir défini toutes les lignes, on peut créer une Line Loop qui correspond au bord avec l'indice des lignes, puis une surface avec l'indice de la Line Loop. Les physical Lines (ou Surfaces) sont les labels que l'on veut attribuer. Voici pour l'exemple de la marche (voir la figure de l'introduction), un code sans OpenCascade, puis un avec OpenCascade.

```
DefineConstant [
    L={5,Min 3, Max 10, Step 1,Name "00Longueur",Visible 1},
    l={1,Min 1, Max 2, Step 0.5,Name "00Largeur",Visible 1}
];
h=0.2;
```

```

Point(1) = {0, 1/2, 0,h};
Point(2) = {1, 1/2, 0,h};
Point(3) = {1, 0, 0,h};
Point(4) = {L, 0, 0,h};
Point(5) = {L, 1, 0,h};
Point(6) = {0, 1, 0,h};

Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,5};
Line(5) = {5,6};
Line(6) = {6,1};

Line Loop(1) = {1,2,3,4,5,6};
Plane Surface (1) = {1};
Physical Surface (0)={1};
Physical Line(4)={6};
Physical Line(1)={1,2,3};
Physical Line(2)={4};
Physical Line(3)={5};

```

Avec OpenCascade, on peut écrire le code suivant: (Avec pour créer un rectangle la syntaxe Rectangle(indice)={x,y,z,largeur,hauteur} où (x,y,z) sont les coordonnées du point en bas à gauche du rectangle, et BooleanUnion permet de concaténer des surfaces)

```

SetFactory("OpenCASCADE"); //On indique qu'on utilise le moteur OpenCascade

DefineConstant [
    L={5,Min 3, Max 10, Step 1,Name "00Longueur",Visible 0},
    l={1,Min 1, Max 2, Step 0.5,Name "00Largeur",Visible 0}
];
h=0.2;
Mesh. CharacteristicLengthMin = h;
Mesh. CharacteristicLengthMax = h;

Rectangle(1)= {0,0.5,0,1,0.5};
Rectangle(2)= {1,0,0,L,1};
BooleanUnion{Surface{1};Delete;}{Surface{2};Delete;}

Physical Surface (0)={1,2};
Physical Line(4)={2};
Physical Line(1)={1,6,5};
Physical Line(2)={7};
Physical Line(3)={8};

```

### 3 Formulation variationnelle

J'utilise pour résoudre cette équation la méthode des caractéristiques avec des éléments finis de Taylor-Hood P2-P1.

On approche le terme d'accélération  $\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}$  par  $\frac{\mathbf{u}^{n+1} - \mathbf{u}^n \circ X^n}{\Delta t}$  où :

$$\begin{cases} \mathbf{u}^{n+1}(\mathbf{x}) &= \mathbf{u}(\mathbf{x}, \Delta t(n+1)) \\ X^n(\mathbf{x}) &= X^n(t^n; t^{n+1}, \mathbf{x}) \end{cases}$$

La caractéristique approchée  $X^n(\mathbf{x})$  correspond à la position à  $t^n$  d'une particule qui sera en  $\mathbf{x}$  à  $t^{n+1}$ , et  $\frac{dX(t;t^{n+1},\mathbf{x})}{dt} = u^n(X(t;t^{n+1},\mathbf{x}))$  ( $t^n \leq t < t^{n+1}$ ) et  $X(t^{n+1};t^{n+1},\mathbf{x}) = \mathbf{x}$ .  
On obtient donc le schéma temporel d'ordre 1 suivant:

$$\begin{cases} \mathbf{u}^{n+1} - \nu \Delta t \Delta \mathbf{u}^{n+1} + \Delta t \nabla p^{n+1} &= \mathbf{u}^n \circ X^n \\ \nabla \cdot \mathbf{u} &= 0 \end{cases}$$

Si on le multiplie par  $\mathbf{v} \in H^1(\Omega)^2$  et on intègre dans  $\Omega$ , on obtient la formulation variationnelle mixte suivante:

$$\int_{\Omega} \mathbf{u}^{n+1} \mathbf{v} + \nu \Delta t \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} - \nu \Delta t \int_{\partial\Omega} \partial_n \mathbf{u}^{n+1} \mathbf{v} - \Delta t \int_{\Omega} p^{n+1} \nabla \cdot \mathbf{v} + \Delta t \int_{\partial\Omega} (p^{n+1} \cdot n) \mathbf{v} = \int_{\Omega} \mathbf{u}^n (X^n) \mathbf{v}$$

$$\text{où } \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} = \int_{\Omega} \nabla u_1 \cdot \nabla v_1 + \nabla u_2 \cdot \nabla v_2$$

On doit chercher une pression de moyenne nulle pour assurer son unicité car pour une constante quelconque  $C$ , si  $p$  est solution,  $p + C$  l'est aussi. On pose donc comme espaces  $L_0^2(\Omega) = \{q \in L^2(\Omega), \int_{\Omega} q \, dx = 0\}$ . Pour les conditions aux bords pour la vitesse, on a des conditions de Dirichlet homogènes seulement sur  $\Gamma_1$  et  $\Gamma_3$ , donc on définit l'espace  $H_{d,0}^1(\Omega)^2 = \{v \in H^1(\Omega)^2, v|_{\Gamma_1} = v|_{\Gamma_3} = 0\}$

On pose:

$$\begin{aligned} \alpha &= \frac{1}{\Delta t}, \\ a(\mathbf{u}^{n+1}, \mathbf{v}) &= \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} - \nu \int_{\partial\Omega} \partial_n \mathbf{u}^{n+1} \mathbf{v}, \\ b(\mathbf{v}, p^{n+1}) &= - \int_{\Omega} p^{n+1} \nabla \cdot \mathbf{v} + \int_{\partial\Omega} (p^{n+1} \cdot n) \mathbf{v}, \\ l(v) &= \int_{\Omega} \mathbf{u}^n (X^n) \mathbf{v} \end{aligned}$$

On veut chercher  $(\mathbf{u}^{n+1}, p^{n+1}) \in H_{d,0}^1(\Omega)^2 \times L_0^2(\Omega)$  tels que

$\begin{aligned} \forall \mathbf{v} \in H_{d,0}^1(\Omega)^2, \alpha(\mathbf{u}^{n+1}, \mathbf{v}) + a(\mathbf{u}^{n+1}, \mathbf{v}) + b(\mathbf{v}, p^{n+1}) &= \alpha l(\mathbf{v}) \\ \forall q \in L_0^2(\Omega), b(\mathbf{u}^{n+1}, q) &= 0 \end{aligned}$
---

## 4 Discrétisation

On veut réécrire la formule variationnelle sous la forme  $\mathbf{AX}=\mathbf{F}$ .

### 4.1 Fonctions de base de $P_1 - P_2$

Soit les ensembles  $V_h = \{v_h \in \mathcal{C}^0(\Omega); v_h|_T \in \mathbf{P}^2, \forall T \in \mathcal{T}_h\}$  et  $M_h = \{q_h \in \mathcal{C}^0(\Omega); q_h|_T \in \mathbf{P}^1, \forall T \in \mathcal{T}_h\}$ . On introduit les fonctions de base  $(\varphi_j)_{j=0\dots nbv-1}$  tel que  $\varphi_j \in M_h$  et  $\forall i = 0\dots nbv-1$ ,  $\varphi_j(s_i) = \delta_{ij}$  et  $(\psi_j)_{j=0\dots nbe+nbv-1}$  tel que  $\psi_j \in V_h$  et pour tous sommets ou milieux d'arêtes  $s_i$ ,  $\varphi_j(s_i) = \delta_{ij}$ . On obtient sur le triangle de référence  $\hat{K}$  les fonctions de forme:

$$\begin{aligned} \hat{\varphi}_0(\varsigma, \eta) &= 1 - \varsigma - \eta \\ \hat{\varphi}_1(\varsigma, \eta) &= \varsigma \\ \hat{\varphi}_2(\varsigma, \eta) &= \eta \\ \hat{\psi}_0 &= 2\hat{\varphi}_0(\hat{\varphi}_0 - 1/2) \\ \hat{\psi}_1 &= 2\hat{\varphi}_1(\hat{\varphi}_1 - 1/2) \\ \hat{\psi}_2 &= 2\hat{\varphi}_2(\hat{\varphi}_2 - 1/2) \end{aligned}$$

$$\begin{aligned}\hat{\psi}_3 &= 4\hat{\varphi}_0\hat{\varphi}_1 \\ \hat{\psi}_4 &= 4\hat{\varphi}_1\hat{\varphi}_2 \\ \hat{\psi}_5 &= 4\hat{\varphi}_2\hat{\varphi}_0\end{aligned}$$

## 4.2 Ecriture de $a_h$ et $b_h$ sans les termes de bords

Soit nbt, nbv, nbe, le nombre respectivement de triangles, de sommets, et d'arêtes de notre maillage triangulaire  $\mathcal{T}_h$  de  $\Omega$ .

En négligeant les termes de bord pour le moment, que l'on prendra en compte plus tard (par hypothèse les contraintes sur le bord de sortie s'annulent), on peut discrétiser  $a$  et  $b$  en  $a_h$  et  $b_h$ , où

$$\begin{cases} a_h(\psi_i, \psi_j) = \nu \sum_{k=0}^{nbt-1} \int_{T_k} \sum_{i,j=0\dots 5} \nabla \psi_j : \nabla \psi_i \\ b_h(\psi_i, \varphi_j) = - \sum_{k=0}^{nbt-1} \int_{T_k} \sum_{i=0\dots 2, j=0\dots 5} (\varphi_j(x) \nabla \cdot \psi_i(x)) \end{cases}$$

### 4.2.1 Transformation affine

On passe d'un triangle  $K$  (de sommets locaux  $s_0, s_1, s_2$ ) à un triangle de référence  $\hat{K}$  par une transformation affine.  $\hat{K}$  correspond au triangle des sommets  $\hat{s}_0 = (0, 0)$ ,  $\hat{s}_1 = (0, 1)$ , et  $\hat{s}_2 = (1, 0)$ .

Notons  $\mathcal{A}_T(\varsigma, \eta) = \hat{\phi}_0(\varsigma, \eta)s_0 + \hat{\phi}_1(\varsigma, \eta)s_1 + \hat{\phi}_2(\varsigma, \eta)s_2$  cette transformation affine. Pour calculer les dérivées partielles selon  $x$  et  $y$  de  $\hat{\psi}_i$ , on utilise la formule:

$$\begin{pmatrix} \frac{\partial \psi_i^K}{\partial x} \\ \frac{\partial \psi_i^K}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial \varsigma}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \varsigma}{\partial y} & \frac{\partial \eta}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial \hat{\psi}_i}{\partial \varsigma} \\ \frac{\partial \hat{\psi}_i}{\partial \eta} \end{pmatrix}.$$

On a  $\nabla_{x,y} \psi_i^K(x, y) = B_K \nabla_{\varsigma, \eta} \hat{\psi}_i(\varsigma, \eta)$  avec  $B_K = (J_K^T)^{-1}$

Pour obtenir  $B_K$ , on utilise la jacobienne  $J_K = \begin{pmatrix} \frac{\partial x}{\partial \varsigma} & \frac{\partial y}{\partial \varsigma} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix}$

Comme  $\nabla_{\varsigma, \eta} \hat{\psi}_j(\varsigma, \eta) = (J_K)^T \nabla_{x,y} \psi_j^K(x, y)$ , on a donc:

$$B_K = (J_K^T)^{-1} = \frac{1}{\det(J_K)} \begin{pmatrix} y_3^K - y_1^K & y_1^K - y_2^K \\ x_1^K - x_3^K & x_2^K - x_1^K \end{pmatrix}$$

De plus,  $\det(J_K) = 2|T_K|$  et donc:

$$\begin{aligned} \int_{T_K} f(x) dx &= |\det(J_K)| \int_{\hat{K}} \hat{f}(\hat{x}) d\hat{x} \\ &= 2|T_K| \int_{\hat{K}} \hat{f}(\hat{x}) d\hat{x} \end{aligned}$$

Par la précédente transformation affine, on obtient donc :

$$\begin{cases} a_h = 2\nu \sum_{k=0}^{nbt-1} |T_K| \int_{\hat{K}} \sum_{i,j=0\dots 5} \nabla \hat{\psi}_j^T B_K^T B_K \nabla \hat{\psi}_i \\ b_h = -2 \sum_{k=0}^{nbt-1} |T_K| \int_{\hat{K}} \sum_{i=0\dots 2, j=0\dots 5} (\hat{\varphi}_j B_K \nabla \hat{\psi}_i) \end{cases}$$

On choisit une méthode d'intégration numérique à sept points d'après la règle de Gauss-Hammer, exacte pour les polynômes de degré inférieur ou égal à 5. (cf tableau ci-contre)

points ( $a_s$ )	poids ( $\omega_s$ )
$(1/3, 1/3)$	0.225
$(\frac{6-\sqrt{15}}{21}, \frac{6-\sqrt{15}}{21})$	$\frac{155-\sqrt{15}}{1200}$
$(\frac{6-\sqrt{15}}{21}, \frac{9+2\sqrt{15}}{21})$	$\frac{155-\sqrt{15}}{1200}$
$(\frac{9+2\sqrt{15}}{21}, \frac{6-\sqrt{15}}{21})$	$\frac{155-\sqrt{15}}{1200}$
$(\frac{6+\sqrt{15}}{21}, \frac{6+\sqrt{15}}{21})$	$\frac{155+\sqrt{15}}{1200}$
$(\frac{6+\sqrt{15}}{21}, \frac{9-2\sqrt{15}}{21})$	$\frac{155+\sqrt{15}}{1200}$
$(\frac{9-2\sqrt{15}}{21}, \frac{6+\sqrt{15}}{21})$	$\frac{155+\sqrt{15}}{1200}$

$$\int_{\hat{K}} \hat{f}(\hat{x}) = |\hat{K}| \sum_{s=0\dots 6} \omega_s \hat{f}(a_s)$$

On obtient donc:

$$\begin{cases} a_h(\psi_i, \psi_j) = \nu \sum_{k=0}^{nbt-1} |T_K| \sum_{i,j=0\dots 5} \sum_{s=0\dots 6} \omega_s \nabla \hat{\psi}_j^T(a_s) B_K^T B_K \nabla \hat{\psi}_i(a_s) \\ b_h(\psi_i, \varphi_j) = - \sum_{k=0}^{nbt-1} |T_K| \sum_{i=0\dots 2, j=0\dots 5} \sum_{s=0\dots 6} \omega_s (\hat{\varphi}_j(a_s) B_K \nabla \hat{\psi}_i(a_s)) \end{cases}$$

Pour rappel:

$$\begin{aligned} \hat{\psi}_i &= \hat{\varphi}_i(2\hat{\varphi}_i - 1) \text{ et } \nabla \hat{\psi}_i = (4\hat{\varphi}_i - 1) \nabla \hat{\varphi}_i \text{ pour } i=0,1,2 \\ \hat{\psi}_3 &= 4\hat{\varphi}_1\hat{\varphi}_2 \text{ et } \nabla \hat{\psi}_3 = 4(\hat{\varphi}_1 \nabla \hat{\varphi}_2 + \hat{\varphi}_2 \nabla \hat{\varphi}_1) \\ \hat{\psi}_4 &= 4\hat{\varphi}_0\hat{\varphi}_2 \text{ et } \nabla \hat{\psi}_4 = 4(\hat{\varphi}_0 \nabla \hat{\varphi}_2 + \hat{\varphi}_2 \nabla \hat{\varphi}_0) \\ \hat{\psi}_5 &= 4\hat{\varphi}_0\hat{\varphi}_1 \text{ et } \nabla \hat{\psi}_5 = 4(\hat{\varphi}_0 \nabla \hat{\varphi}_1 + \hat{\varphi}_1 \nabla \hat{\varphi}_0) \end{aligned}$$

La matrice élémentaire A (carrée de taille 15) est de la forme:

$$\begin{pmatrix} C & 0 & B_1 \\ 0 & C & B_2 \\ B_1^T & B_2^T & -I_\varepsilon \end{pmatrix}$$

$$\begin{aligned} \text{Avec } C &= (C_{i,j})_{i,j}, \text{ où } C_{i,j} = \alpha \int_{\Omega} \psi_j \psi_i + \nu \int_{\Omega} \frac{\partial \psi_j}{\partial x} \frac{\partial \psi_i}{\partial x} + \frac{\partial \psi_j}{\partial y} \frac{\partial \psi_i}{\partial y} \\ B_{1,i,j} &= - \int_{\Omega} \frac{\partial \psi_i}{\partial x} \varphi_j \text{ et } B_{2,i,j} = - \int_{\Omega} \frac{\partial \psi_i}{\partial y} \varphi_j \end{aligned}$$

On veut calculer les matrices élémentaires de rigidité  $\int_{T_K} \frac{\partial \psi_i}{\partial x_k} \frac{\partial \psi_j}{\partial x_l}$

$$\text{Si note } J = \begin{pmatrix} y_3^K - y_1^K & y_1^K - y_2^K \\ x_1^K - x_3^K & x_2^K - x_1^K \end{pmatrix}, B_K = (J_K^T)^{-1} = \frac{J}{\det(J_K)}$$

$$\text{Comme } \nabla_{x,y} \psi_i^K(x, y) = B_K \nabla_{\varsigma, \eta} \hat{\psi}_i(\varsigma, \eta),$$

On obtient:

$$C_{i,j} = \alpha |T_K| \sum_{s=0\dots 6} \omega_s \hat{\psi}_j \hat{\psi}_i + \frac{\nu}{4 * |T_K|} \sum_{s=0\dots 6} \omega_s \nabla \hat{\psi}_j(a_s) : J^T J \nabla \hat{\psi}_i(a_s)$$

$$B_{1,i,j} = -\frac{1}{2} \sum_{s=0\dots 6} \omega_s (J[0][0] \partial_x \hat{\psi}_i(a_s) + J[0][1] \partial_y \hat{\psi}_i(a_s)) \hat{\varphi}_j(a_s)$$

et

$$B_{2,i,j} = -\frac{1}{2} \sum_{s=0\dots 6} \omega_s (J[1][0] \partial_x \hat{\psi}_i(a_s) + J[1][1] \partial_y \hat{\psi}_i(a_s)) \hat{\varphi}_j(a_s)$$

$$\text{On pose } X = \begin{pmatrix} U_1 \\ U_2 \\ P \end{pmatrix}$$

Alors le système se réécrit sous forme matricielle:  $AX=F$

où  $F_i = \alpha \int_{\Omega} u^n(X^n) \psi_i$

Pour ce dernier terme, on calcule

$$F_i = \alpha \sum_{T_k} |T_k| \sum_{s=0 \dots 6} \omega_s \hat{\psi}_i(a_s) \cdot u^n(X^n(a_s))$$

$$X(\tau; t, x) = x + \int_t^{\tau} u(X(s; t, x), s) ds$$

Donc

$$X(t^n; t^{n+1}, a_s) = a_s - \int_{t^n}^{t^{n+1}} u(X(s; t^{n+1}, a_s), s) dx$$

On approche ensuite l'intégrale par la méthode d'Euler rétrograde, et on obtient:

$$X^n(a_s) = a_s - \Delta t u^n(a_s)$$

où

$$u^n(a_s) = \sum_{i=0 \dots 5} \psi_i(a_s) u_i^n$$

Il faut donc trouver à quel triangle appartient  $X^n(a_s)$ .

### 4.3 Les conditions aux bords

On a  $u|_{\Gamma_1} = u|_{\Gamma_3} = 0$  ainsi qu'une condition de neumann sur  $\Gamma_2$  s'annulant lors de la formulation variationnelle et un maximum de 1 sur  $\Gamma_4$ .

Si un triangle possède deux arêtes sur deux bords différents, il ne doit pas y avoir de conflit sur les conditions.

On choisit donc une parabole pour l'entrée du canal. On prend  $g(x, y) = 16(1 - y)(y - 0.5)$  qui est bien de maximum 1 en  $y = 0.75$  ( et s'annule bien en  $y = 1/2$  et  $y = 1$ ). En c++, on utilise la méthode des TGV qui implique simplement de multiplier par une très grande valeur les coefficients de la diagonale des noeuds qui appartiennent au bord ainsi que le coefficient du second membre correspondant.

## 5 Freefem++

Sur Freefem++, pour prendre en compte la condition,  $\int_{\Omega} p = 0$ , puisque p doit être de moyenne nulle pour assurer l'unicité, on pose  $\nabla \cdot u = \varepsilon p$ , ce qui implique que  $\int_{\Omega} p = \frac{1}{\varepsilon} \int_{\Omega} \nabla \cdot u = 0$ .

Pour rappel, la formulation variationnelle s'écrit avec  $\alpha = 1/\Delta t$ :

$$\alpha \int_{\Omega} u^{n+1} v + \nu \int_{\Omega} \nabla u^{n+1} \nabla v - \nu \int_{\partial \Omega} \partial_n u^{n+1} v - \int_{\partial \Omega} p^{n+1} \nabla v + \int_{\partial \Omega} (p^{n+1} \cdot n) v = \alpha \int_{\Omega} u^n(X^n) v$$

La fonction convect de Freefem s'occupe de calculer  $X^n$ .

```
problem NS ([u1,u2,p],[v1,v2,q]) =
int2d(Th)( alpha*(u1*v1+u2*v2)
+ nu*(dx(u1)*dx(v1)+dy(u1)*dy(v1)+ dx(u2)*dx(v2)+dy(u2)*dy(v2))
- p*q*(0.000001)
- p*dx(v1) - p*dy(v2)
- dx(u1)*q - dy(u2)*q)
+on(1,3,u1=0,u2=0)
```



```

+on(4,u1=16(1-y)(,u2=0)
-int2d(Th,optimize=1)(alpha*convect(u1p,-dt,u1p)*v1
-alpha*convect(u2p,-dt,u2p)*v2);

```

Voici les graphiques obtenus avec Freefem pour le problème de Stokes:

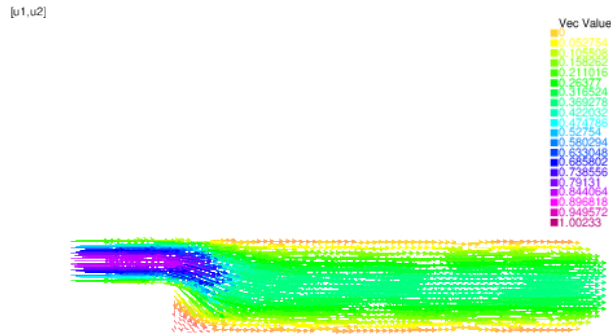


Figure 1: vitesses

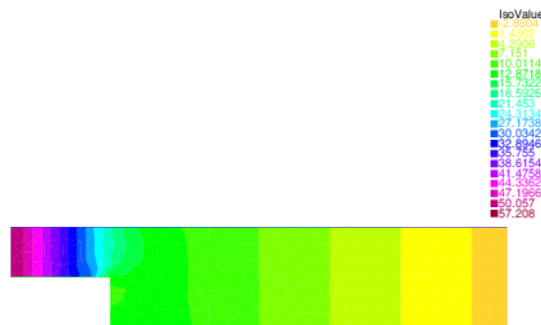


Figure 2: pression

## 6 C++

J'ai commencé par résoudre le problème stationnaire de Stokes, puis j'ai itéré sur plusieurs pas de temps en résolvant le problème instationnaire en utilisant la solution précédente.

La classe Mesh\_gmsh.cpp lit le maillage et crée l'ensemble total des points (les sommets et les milieux des arêtes) avec un numéro global et un label indiquant si le point se trouve au bord ou non. On crée ensuite la matrice élémentaire. Pour cela, on a besoin des fonctions de base:

**double lambda(int i,R2 P),** (calcule  $\hat{\varphi}_i(P.x, P.y)$ )

**int PartialLamb(int i,int ind,R2 P),** (calcule  $\nabla \hat{\varphi}_i(P.x, P.y)$ )

**double Phi(int i,R2 P),** (calcule  $\hat{\psi}_i(P.x, P.y)$ )

**double PartialPhi(int i,int ind,R2 P),** (calcule  $\nabla \hat{\psi}_i(P.x, P.y)$ )

Ensuite on calcule  $B_K^T B_K$  pour obtenir la matrice élémentaire.

Pour la matrice globale, on utilise une map: `typedef map< pair<int,int>,double> MatMap`.

Cela permet de stocker uniquement les valeurs non nuls.

```
for(int il=0;il <15;il++){
    for(int jl=0;jl <15;jl++){
        if(il <6){
            i=Th(k, il );
        }
        else if(il <12){
            i=Th(k, il -6)+n; //n est le degre de liberte P2
        }
        else{
            i=Th(k, il -12)+2*n;
        }
        if(jl <6){
            j=Th(k, jl );
        }
        else if(jl <12){
            j=Th(k, jl -6)+n;
        }
        else{
            j=Th(k, jl -12)+2*n;
        }
        if(fabs(A[il][jl])>1e-30){
            M[make_pair(i,j)]+=A[il][jl];
        }
    }
}
```

La surcharge d'opérateur `Th(k,il)` renvoie le numéro global du *il*-ième sommet du triangle *k*, si *il* < 3 et sinon il renvoie le numéro global du *il*-3 milieu des arêtes du triangle *k*.

On ajoute ensuite les conditions au bord en parcourant les indices locaux et en récupérant les labels, et en appliquant la méthode TGV.

J'ai utilisé ensuite le solver **UMFPACK** pour résoudre le système, après avoir réécrit la matrice globale sous forme d'une matrice de morse.

Pour le problème instationnaire, j'ai créé un vecteur de vecteurs d'entiers de la taille du nombre de triangles pour enregistrer le numéro de ses triangles voisins. Après avoir calculer les points de quadrature d'un triangle, je calcule  $u^n(a_s) = \sum_{i=0..5} \psi_i(a_s) u_i^n$  avec la fonction **double vitesseInterpolee(vector<double> un, R2 PtInterp)** et en itérant sur les voisins du triangle, on peut connaître à quel voisin appartient  $X^n(a_s)$  (pour un pas de temps suffisamment petit) avec la fonction **int RecupVoisins(Mesh2d Th, int triangle, R2 PtInterp)**, et ainsi construire le second membre. Pour visualiser la solution sur Freefem, on doit permuter la solution obtenue pour que la numérotation corresponde à celle de Freefem.

## 7 Résultats

### 7.1 Stokes

#### 7.1.1 Validation du code

J'ai obtenu les graphiques suivant pour un carré de taille 10,10 avec des conditions de dirichlet ( $g(x,y)=4y(1-y)$ ) sur tous les bords excepté le bord droit:

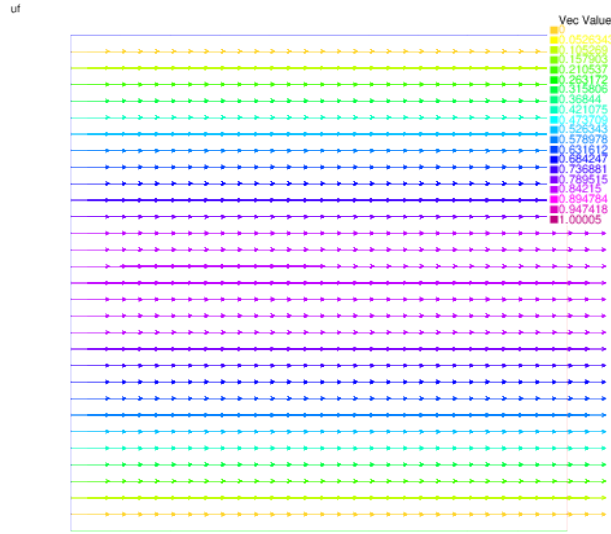


Figure 3: vitesses u1 et u2 pour Stokes sur un carré 10,10

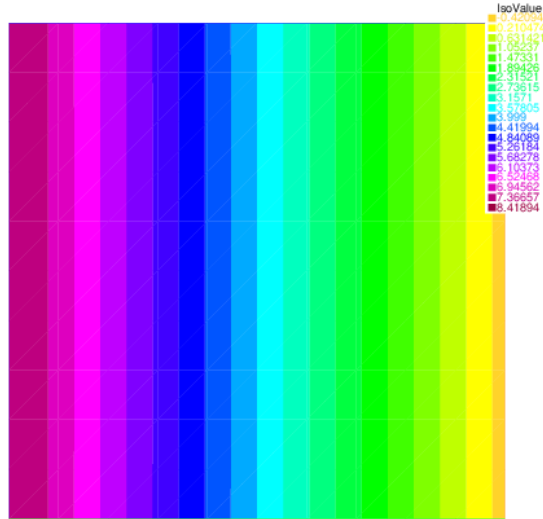


Figure 4: pression pour Stokes sur un carré 10,10

#### 7.1.2 Problème de Stokes sur l'exemple de la marche

Voici les graphiques que j'ai obtenu pour le problème de Stokes sur la marche:

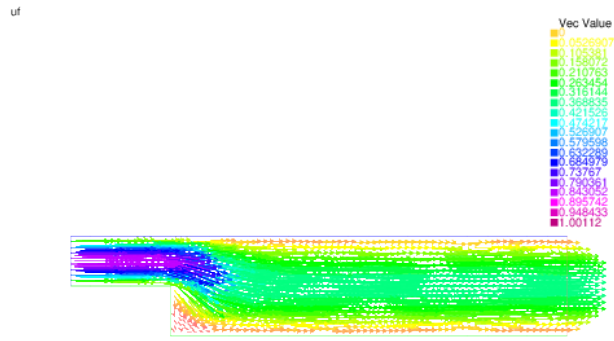


Figure 5: vitesses  $u_1$  et  $u_2$  pour la marche avec Stokes

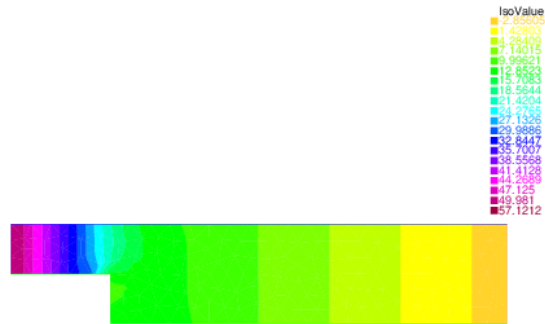
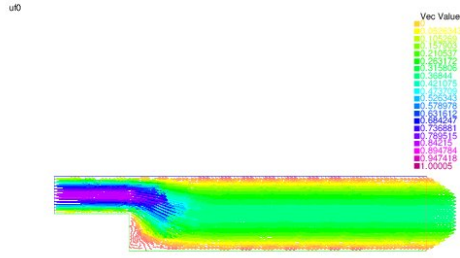


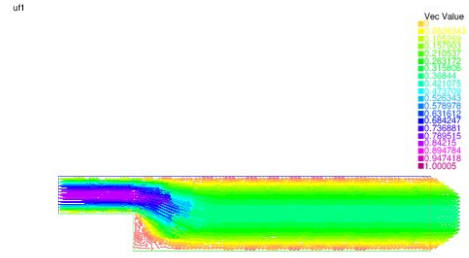
Figure 6: pression pour la marche avec Stokes

## 7.2 Navier-Stokes

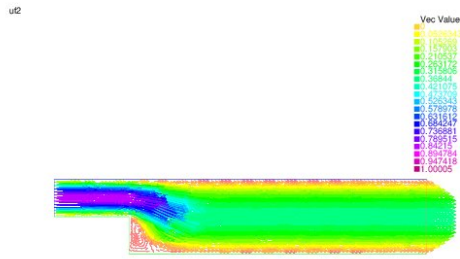
J'obtiens les graphiques suivant respectivement pour  $\nu = 0.01$  et  $\nu = 0.0025$ .



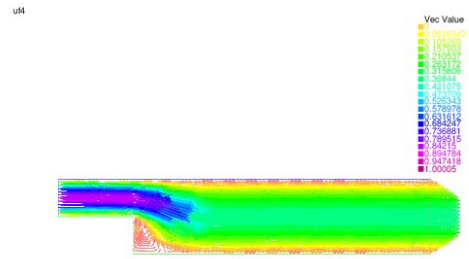
(a) pas de temps 0



(b) pas de temps 1

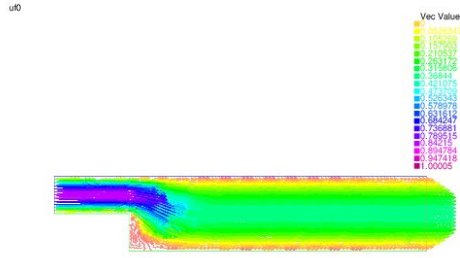


(c) pas de temps 2

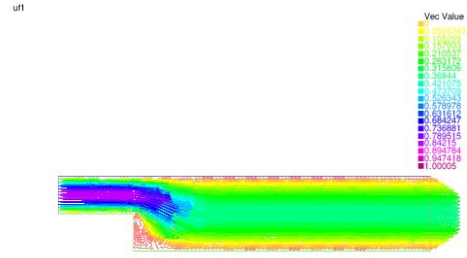


(d) pas de temps 3

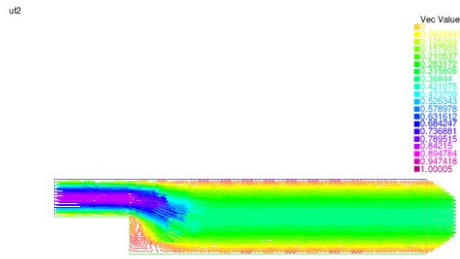
Figure 7: Navier Stokes  $\nu = 0.01$



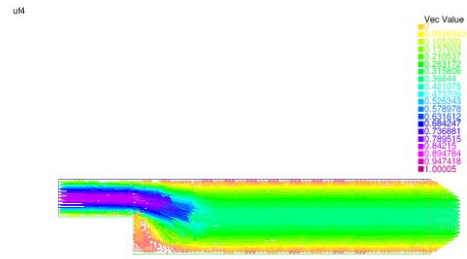
(a) pas de temps 0



(b) pas de temps 1



(c) pas de temps 2



(d) pas de temps 3

Figure 8: Navier Stokes  $\nu = 0.0025$