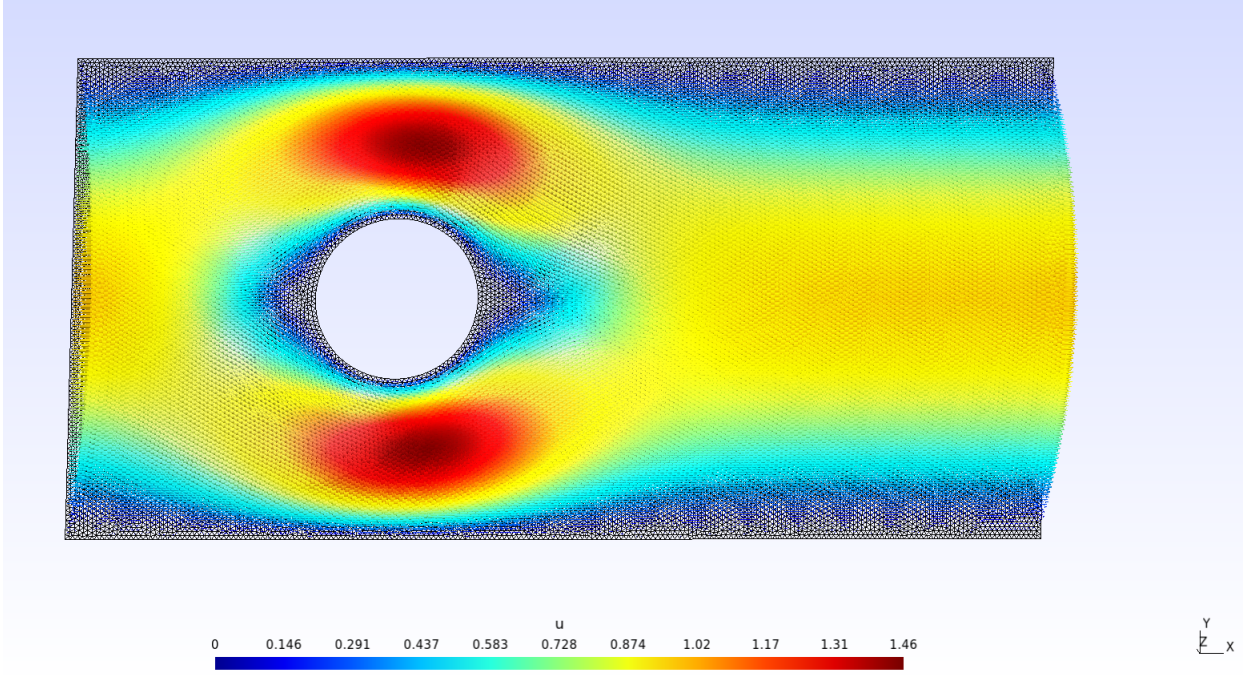


# BE SXS-Elements Finis Mixtes

Pierre RAYNAL, Jean-Pierre MANSOUR, Paulin HUGUET

Octobre 2023



## 1 Le problème de Stokes stationnaire 2D

L'objectif de ce BE est de résoudre numériquement le problème de Stokes stationnaire en 2 Dimensions sur le domaine  $\Omega$  délimité par ses frontières  $\Gamma_i$

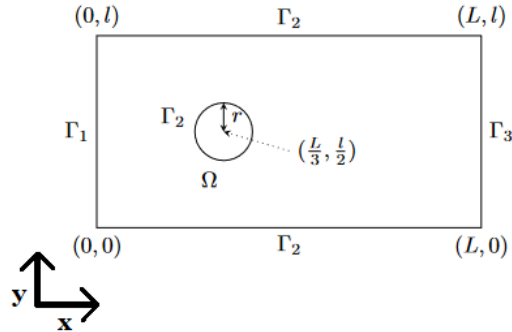


Figure 1: Domaine  $\Omega$  et ses frontières  $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$

On impose de plus les conditions suivantes sur les frontières du domaines :

- **Sur  $\Gamma_2$** , une condition de Dirichlet homogène d'adhérence qui impose une vitesse  $\mathbf{u}$  nulle autour du "trou",

- **Sur  $\Gamma_1$ ,** un profil de type Poiseuille quadratique en  $y$  (condition de Dirichlet non nulle sur  $\mathbf{u}$  ,
- **Sur  $\Gamma_3$ ,** un "flux aspirant" se traduisant par une condition de Fourier-Robin liant Pression  $P$  et champ de vitesse  $\mathbf{u}$  :

$$P = \nu \mathbf{u} \cdot \mathbf{n} ,$$

où  $\mathbf{n}$  est la normale à  $\Gamma_3$  tournée vers l'extérieur du système, et  $\nu$  est la viscosité cinématique du fluide,

- Aucune influence extérieure sur le système  $\Omega$  étudié.

Le système se met donc sous la forme suivante, (avec  $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ , et  $\mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$ ) :

$$\begin{aligned} -\nu \Delta \mathbf{u}(\mathbf{x}) + \nabla P(\mathbf{x}) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \forall \mathbf{x} \in \Omega, \\ \operatorname{div} \mathbf{u}(\mathbf{x}) &= 0, \forall \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}) &= \begin{pmatrix} 4y(1-y) \\ 0 \end{pmatrix}, \forall \mathbf{x} \in \Gamma_1, \\ \mathbf{u}(\mathbf{x}) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \forall \mathbf{x} \in \Gamma_2, \\ P(\mathbf{x}) &= \nu \mathbf{u} \cdot \mathbf{n}, \forall \mathbf{x} \in \Gamma_3. \end{aligned}$$

## 1.1 Etude mathématique du système

Dans cette section, l'idée est de donner un cadre mathématiquement rigoureux et précis (du moins presque, nous y revenons plus loin) de la formulation qui sera ensuite gérée numériquement par l'ordinateur via GETFEM.

Nous allons contrôler les conditions de Dirichlet (homogène et non-homogène) dans l'espace de fonctions test en effectuant un changement dans la formule variationnelle. Pour ceci, fixons arbitrairement une fonction  $\mathbf{u}_0 : \bar{\Omega} \rightarrow \mathbb{R}^2$  de classe  $C^\infty$  et vérifiant :

- $\mathbf{u}_0(\mathbf{x}) = \begin{pmatrix} 4y(1-y) \\ 0 \end{pmatrix}, \forall \mathbf{x} \in \Gamma_1$
- $\mathbf{u}_0 = 0$  sur  $\Gamma_2$  et dans un voisinage de  $\Gamma_3$  (en particulier,  $\frac{\partial \mathbf{u}_0}{\partial x} = 0$  sur  $\Gamma_3$ ).

On ne construit pas explicitement une telle fonction, mais il est clair que cela est faisable.

Nous écrivons à présent les deux égalités sous forme variationnelle :

$$\begin{aligned} \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Gamma_3} \nu \left[ u_x \mathbf{n} - \frac{\partial \mathbf{u}}{\partial x} \right] \cdot \mathbf{v} - \int_{\Omega} p \operatorname{Div}(\mathbf{v}) &= - \int_{\Omega} \nu \nabla \mathbf{u}_0 : \nabla \mathbf{v}, \forall \mathbf{v} \in \mathbf{V} \\ \int_{\Omega} q \operatorname{Div}(\mathbf{u}) &= - \int_{\Omega} q \operatorname{Div}(\mathbf{u}_0), \forall q \in P \end{aligned}$$

L'idée est qu'un couple  $(\mathbf{u}, p)$  soit solution de cette formule variationnelle si et seulement si le couple  $(\bar{\mathbf{u}}, p)$  est solution du problème de Stokes présenté plus haut, où l'on a noté  $\bar{\mathbf{u}} := \mathbf{u} + \mathbf{u}_0$ . Il nous reste à correctement déterminer les espaces fonctionnels en jeu. Nous souhaitons que  $\partial_x \mathbf{u}$  ait une trace sur  $\Gamma_3$ , un choix judicieux est donc de prendre :

- $\mathbf{V} = \{ \mathbf{v} \in (H^1(\Omega))^2 : \mathbf{v}|_{\Gamma_1 \cup \Gamma_2} = 0 \text{ et } \partial_x \mathbf{v} \in (H^1(\Omega))^2 \}$  muni de la norme  $\| \cdot \|_{\mathbf{V}}^2 = \| \cdot \|_{H^1}^2 + \| \partial_x \cdot \|_{H^1}^2$  (qui en fait un espace de Hilbert);
- $P = L^2(\Omega)$ .

Nous montrons à présent (en partie) que le problème est bien posé et que la formulation variationnelle est équivalente au problème initiale.

Soit donc  $(\bar{\mathbf{u}}, p)$  un couple solution. Déjà pour commencer, on voit qu'en testant l'équation contre des fonctions test  $\mathbf{v} \in C_0^\infty(\Omega)^2$  et  $q \in C_0^\infty(\Omega)$ , on a bien au sens des distributions

$$\begin{cases} -\nu \Delta \bar{\mathbf{u}} + \nabla p = 0 \\ \text{Div}(\bar{\mathbf{u}}) = 0 \end{cases}$$

De plus les conditions au bord sur  $\Gamma_1$  et  $\Gamma_2$  sont vérifiées par nature de l'espace  $\mathbf{V}$ . Cependant en ce qui concerne la condition de Fourier-Robin sur  $\Gamma_3$ , nous mentionnerons que l'espace  $L^2$  ne permet pas *a priori* de définir correctement la trace de la pression  $p$  sur  $\Gamma_3$ . En cela, il existe un léger biais de rigueur dont nous nous satisferons.

Notez que nous ne cherchons pas ici à prouver l'unicité ou l'existence de solution(s) à la formule variationnelle mais simplement de justifier mathématiquement le parcours entre les deux facettes (EDP et Calcul des Variations) du problème de Stokes.

Par la suite, nous allons mettre au point le modèle numérique pour simuler toutes ces équations et comparer différentes méthodes de résolution numériques.

## 2 Mise au point des "Briques" de résolution

### 2.1 Formulation Variationnelle

Telle que formulé précédemment, le problème ne peut pas être discrétisé et résolu numériquement.

L'idée est alors de passer en formulation variationnelle pour trouver nos solutions. Les deux premières équations du système de Stokes deviennent (voir plus haut) :

$$\int_{\Omega} -\nu \Delta \mathbf{u} \cdot \mathbf{v} + \int_{\Omega} \nabla P \cdot \mathbf{v} = 0, \forall \mathbf{v} \in \mathbf{V}, \quad (1)$$

$$\int_{\Omega} -q \text{div}(\mathbf{u}) = 0, \forall q \in P. \quad (2)$$

La manière dont GETFEM est codé nous permet d'entrer immédiatement l'équation (2). Pour l'équation (1), l'intégration par parties fait réapparaître les conditions sur les frontières de  $\Omega$  :

$$\boxed{\int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{v} - \int_{\Omega} P \text{div}(\mathbf{v}) = \nu \int_{\partial\Omega} \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{v} - \int_{\partial\Omega} P \mathbf{n} \cdot \mathbf{v}}, \quad (3)$$

où ":" dénote le produit contracté :

$$\nabla \mathbf{u} : \nabla \mathbf{v} = \sum_i \sum_j \frac{\partial}{\partial x_j} u_i \frac{\partial}{\partial x_j} v_i.$$

Nous rentrons cette équation dans GETFEM telle quelle, pour nous permettre de prendre en compte la condition de Fourier-Robin dans les termes sur la frontière.

### 2.2 Conditions Limites

Pour prendre en compte les conditions de Dirichlet restantes, le module GETFEM nous permet de rentrer et spécifier les différents domaines où nous imposons  $\mathbf{u}$  sur notre système :

```
md.add_linear_term(mim, 'nu*Grad_u:Grad_Test_u -p*Div_Test_u')

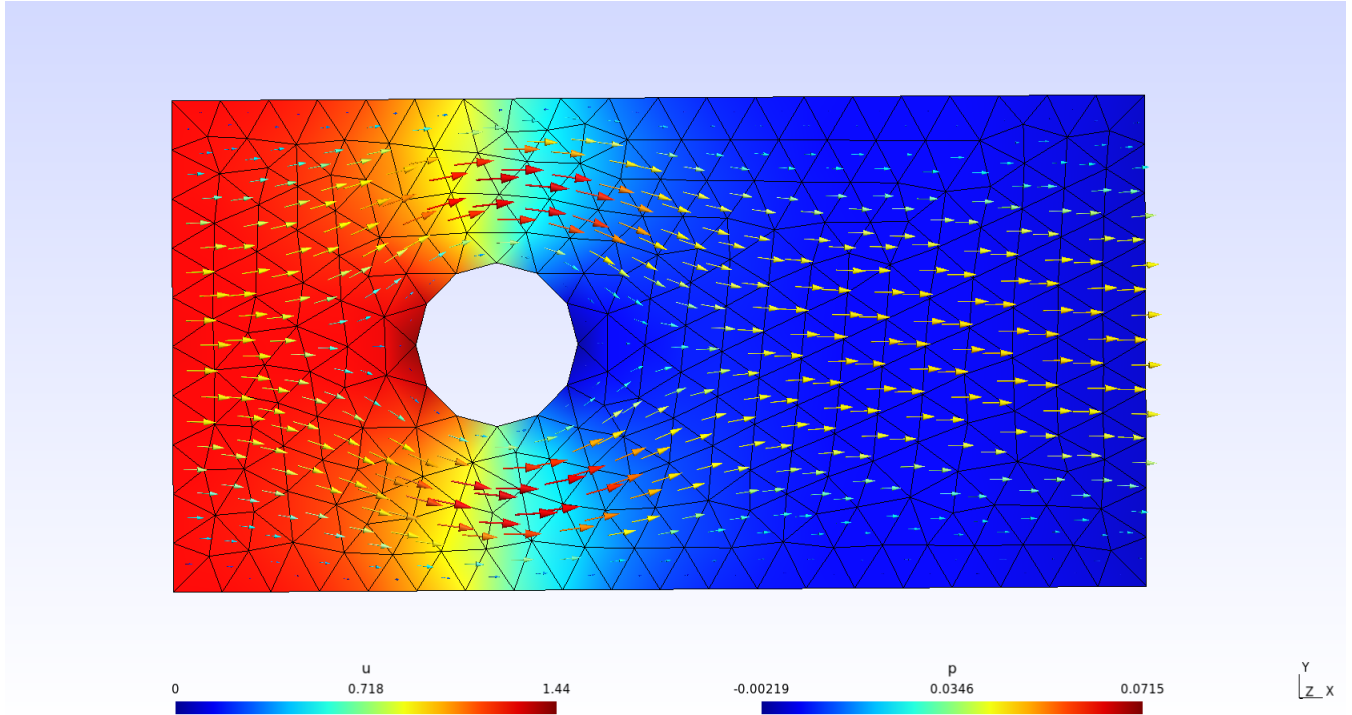
md.add_source_term_brick(mim, 'u', 'f')

md.add_Dirichlet_condition_with_simplification('u', LEFT_BOUND, 'h_inlet')
md.add_Dirichlet_condition_with_simplification('u', HOLE_BOUND, 'h_wall')
md.add_Dirichlet_condition_with_simplification('u', TOP_BOUND, 'h_wall')
md.add_Dirichlet_condition_with_simplification('u', BOTTOM_BOUND, 'h_wall')
```

```
md.add_linear_term(mim, '(-nu*Grad_u.Normal +
nu*(u.Normal)*Normal).Test_u ',RIGHT_BOUND)

md.add_linear_term(mim, 'Div_u*Test_p ')
```

Voici un exemple de résultat après fonctionnement du programme :



### 3 Méthodes différentes et Influence des paramètres

En premier lieu, commençons par rappeler la situation algébrique dans laquelle nous nous trouvons. Nous cherchons à résoudre le système linéaire vectoriel suivant (d'inconnues  $U$  et  $P$ ) :

$$\begin{cases} AU + B^T P = F \\ BU = 0 \end{cases} \quad (4)$$

dans lequel :

- $A$  est une matrice carrée symétrique définie positive.
- $B$  est une matrice (à priori non carrée) telle que  $B^T$  soit injective (condition inf-sup);

Nous avons ici étudié trois méthodes différentes pour résoudre numériquement un tel système. Les voici :

**Méthode Globale** La première méthode (et la plus simple dans son idée) est de remarquer que le système (4) est équivalent à

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (5)$$

Que l'on peut résoudre "simplement" en suivant des algorithmes de résolutions linéaires. (La matrice de ce système sera appelée par la suite "matrice globale").

**Méthode de pénalisation** Soit  $\epsilon > 0$ , on résout l'équation (uniquement en  $U$  cette fois)

$$\left(A + \frac{1}{\epsilon} B^T B\right) U = F \quad (6)$$

Puis on en déduit  $P$  par la formule  $P = \frac{1}{\epsilon} BU$ .

Justifions rapidement que cette matrice  $M = A + \frac{1}{\epsilon} B^T B$  est bien inversible : soit  $U \in \ker M$ , alors en particulier

$$0 = \langle AU, U \rangle + \frac{1}{\epsilon} \langle B^T BU, U \rangle = \langle AU, U \rangle + \frac{1}{\epsilon} \|BU\|^2$$

$A$  étant positive, on a donc  $U \in \ker B$ . D'où  $AU = 0$ , puis par la condition *elker* on conclut que  $U = 0$  et que  $M$  est bien injective.  $\square$

(Remarque : on se rend ici compte que l'inversibilité de  $A$  n'est pas nécessaire pour le bon fonctionnement de cette méthode, la condition *elker* suffit. Ce qui ne pas le cas pour la méthode suivante.)

Le concept maître de cette méthode réside dans le fait suivant (découlant de théorèmes usuels de Calcul Différentiel). Pour  $\epsilon > 0$ , on pose l'énergie :

$$E_\epsilon(V) := \langle AV, V \rangle - \langle F, V \rangle + \frac{1}{2\epsilon} \|BV\|^2$$

alors le vecteur  $U_\epsilon$  solution du problème (6) est l'unique minimiseur de  $E_\epsilon$ . Or, pour  $\epsilon$  petit, la minimisation de  $E_\epsilon$  requiert une très faible norme pour  $BU_\epsilon$ , autrement dit, les écarts trop importants à l'hypersurface sur laquelle nous cherchons notre solution (ici  $\ker B$ ) sont pénalisés. Il s'agit donc bien d'une bonne approximation de la condition  $BU = 0$ .

**Méthode de dualité-pénalisation** Soit  $\gamma > 0$ , on résout le système suivant grâce à la méthode d'Uzawa :

$$\begin{cases} (A + \gamma B^T B)U + B^T P = F \\ BU = 0 \end{cases} \quad (7)$$

(Remarque : ici  $\gamma$  joue un rôle similaire à  $1/\epsilon$  dans la méthode de pénalisation.)

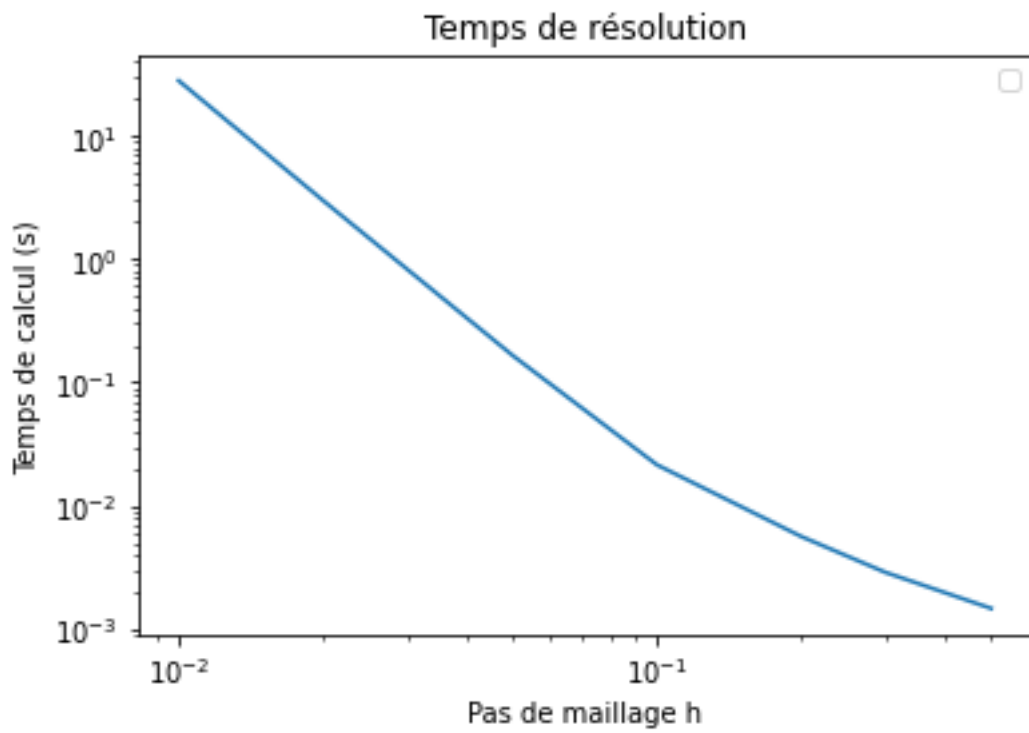
Nous munirons nos graphiques (voir ci-dessous) d'une régression exponentielle asymptotique (au voisinage de 0) qui permet d'estimer la corrélation entre  $h$  et la quantité mesurée.

### 3.1 Paramètre de maillage $h$

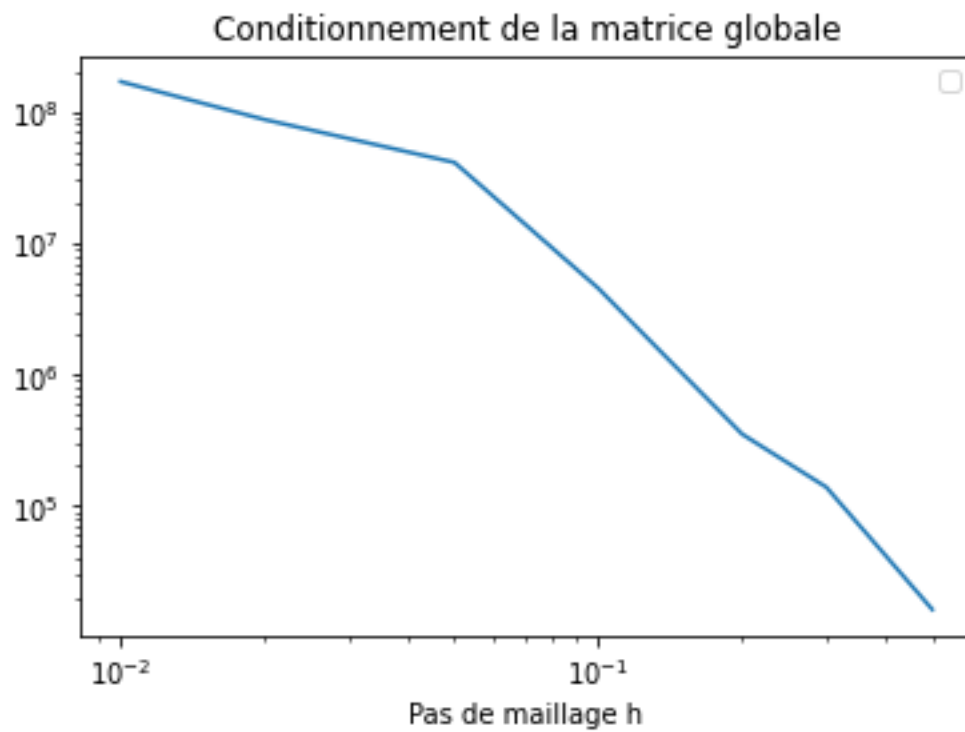
Nous étudions ici l'influence du paramètre  $h$  sur le temps de calcul, le conditionnement de la matrice, la charge mémoire et la norme  $L^2$  de la divergence de  $u$  (qui dans le cas optimal, se doit d'être nulle). Nous ne le précisons qu'une fois, mais tout nos graphiques seront tracés en échelle log-log.

Nous testons le programme avec les valeurs suivantes :

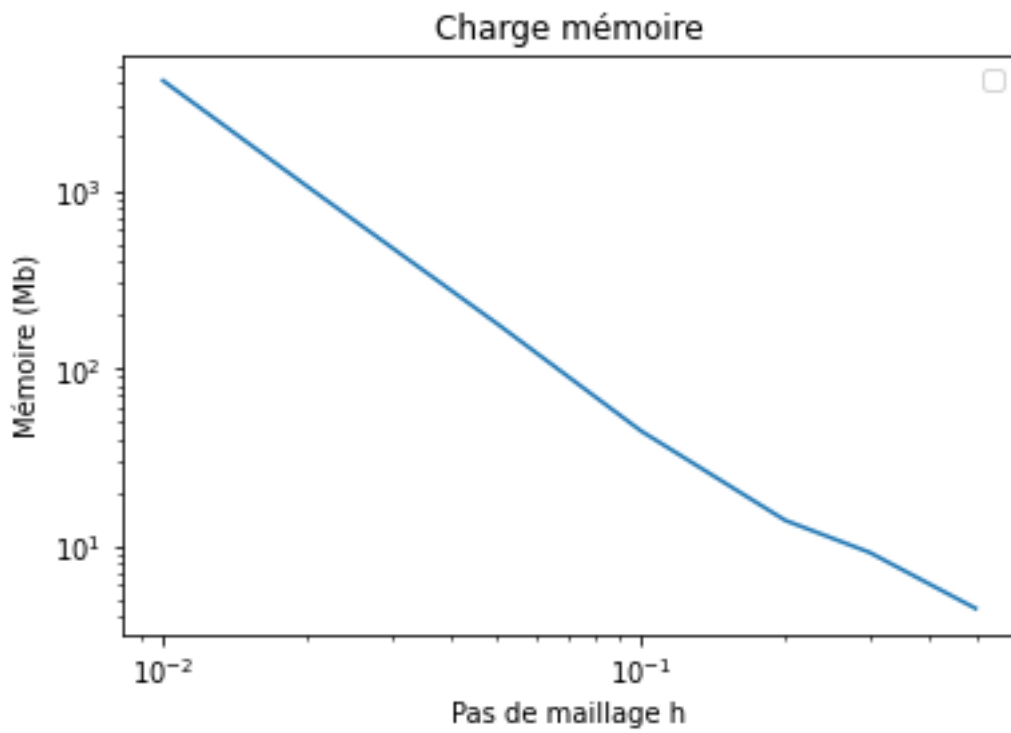
$$h \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$$



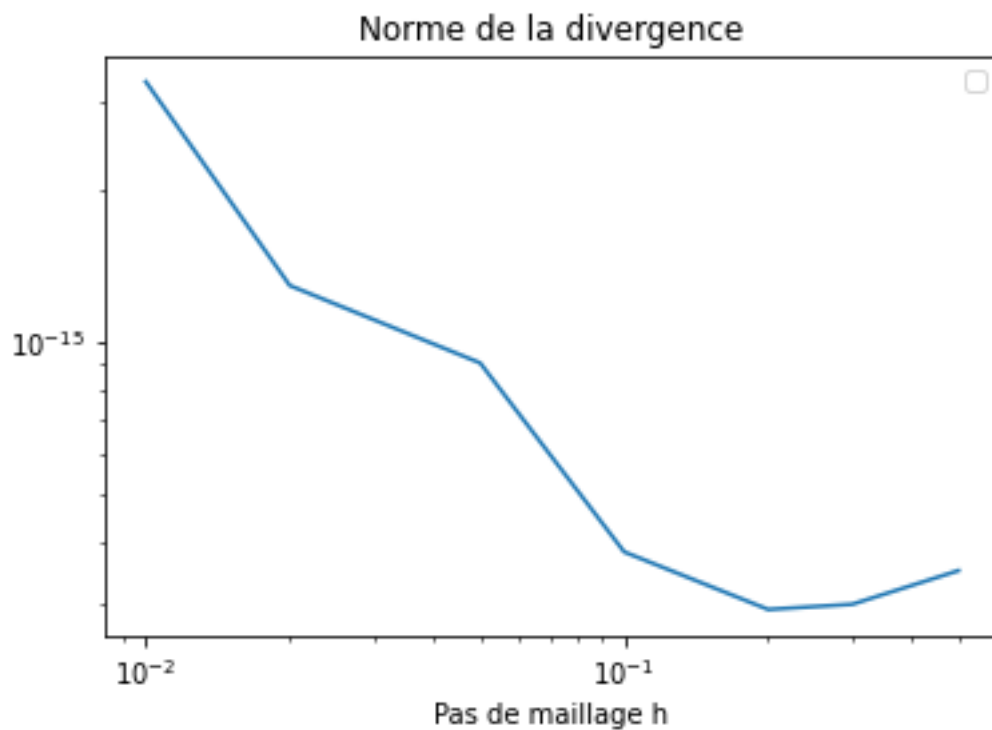
Régression exponentielle asymptotique :  $Y \sim h^{-2.57}$



Régression exponentielle asymptotique :  $Y \sim h^{-2.40}$



Régression exponentielle asymptotique :  $Y \sim h^{-1.7}$



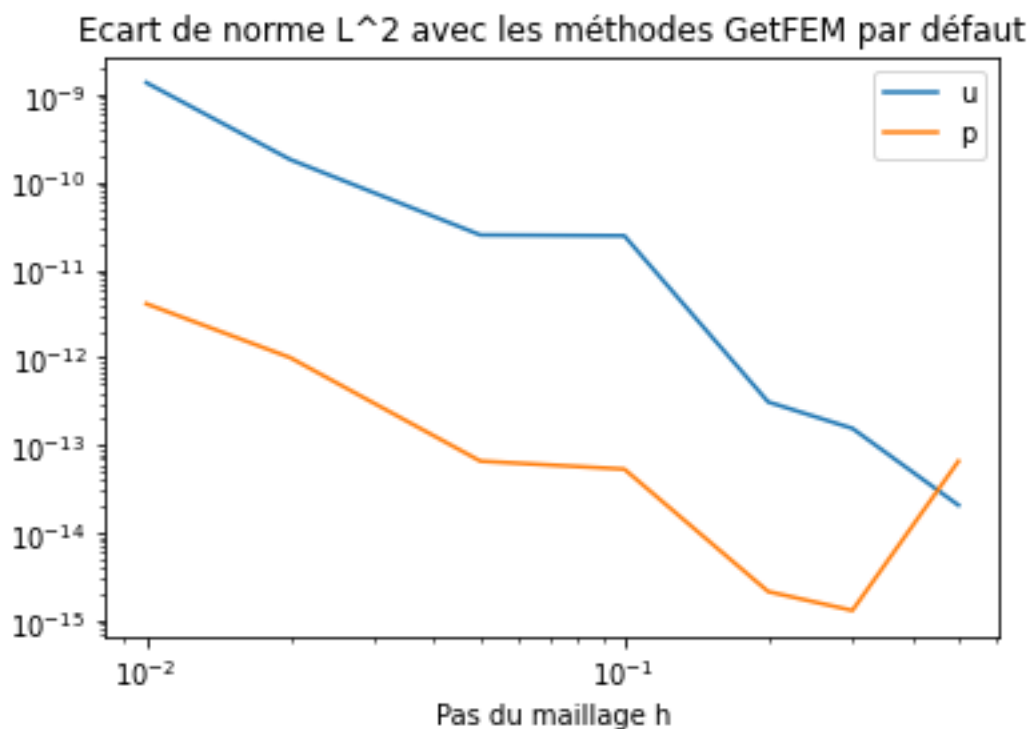
Comme l'on peut naturellement s'y attendre, le temps de calcul, la charge mémoire et le conditionnement augmentent quand le pas d'espace  $h$  diminue. C'est tout à fait cohérent avec le fait que le système linéaire à résoudre se trouve avoir une taille augmentée (l'un des prix à payer pour plus de précision).

Quelque chose d'intéressant à noter est le fait que ces quantités se comportent en  $O(h^2)$  environ pour  $h$  petit. C'est satisfaisant, vu que nous travaillons en dimension 2 et que le nombre d'éléments du maillage (et donc la taille de la matrice) est quadratique en  $h$ .

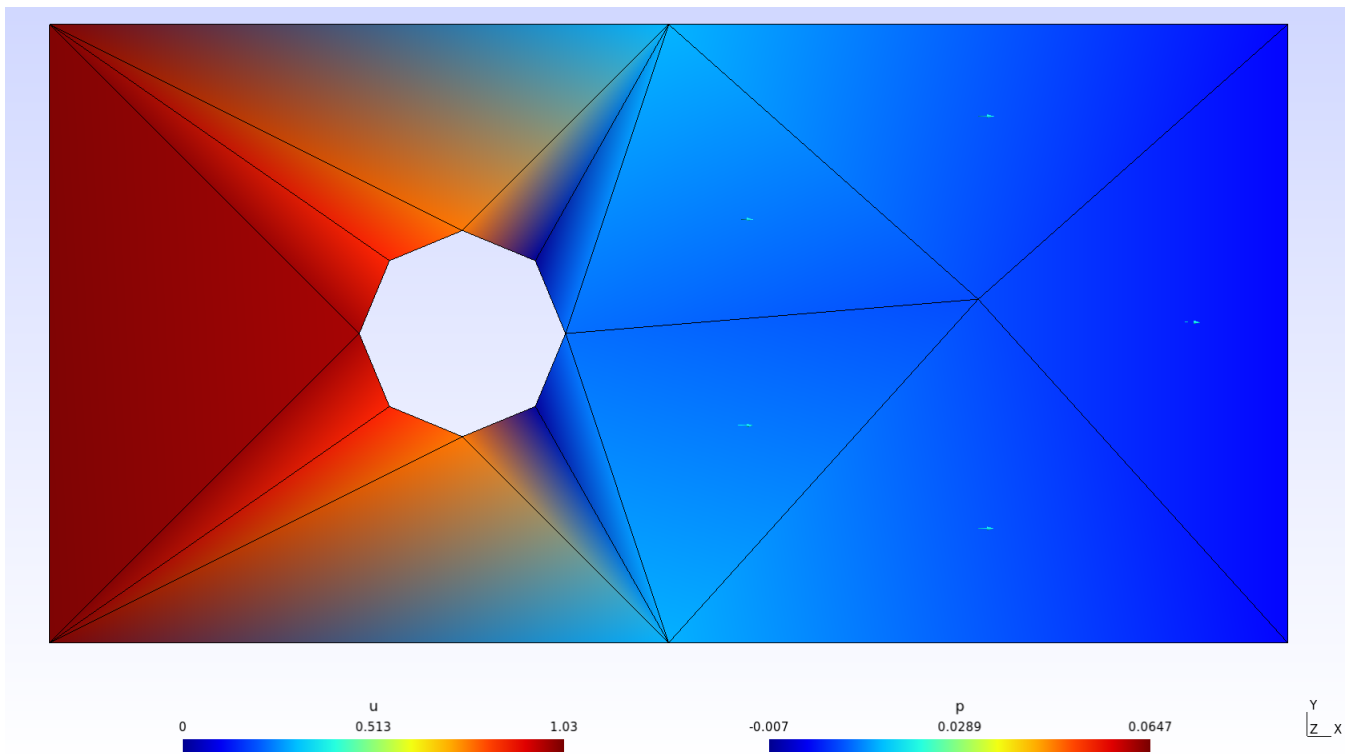
Cependant, le dernier graphique ne correspond pas à l'intuition initiale. En effet, il semble que le caractère incompressible de  $u$  soit de moins en moins bien approché pour de petites valeurs de  $h$ . Une explication confortable serait la taille des quantités considérées (de l'ordre de  $10^{-15}$ ) qui est proche du zéro machine. Pourtant l'explication n'est pas parfaitement satisfaisante, on peut imaginer un facteur tier inconnu qui viendrait perturber le calcul (peut-être un défaut dans l'algorithme d'approximation de la norme ?).

Pour terminer sur l'influence du paramètre  $h$ , on propose ici un dernier graphique rassemblant l'écart  $L^2$  entre les solutions (pour la vitesse  $\mathbf{u}$  et la pression  $p$ ) calculées par le code vu et utilisé durant ce BE et la routine GetFEM par défaut.

On montre également le résultat obtenu pour des valeurs élevées de  $h$  (ici,  $h = 1$ ). Bien sûr, ceci est plus de l'ordre de la curiosité numérique que concret :







### 3.2 Paramètre de pénalisation $\epsilon$

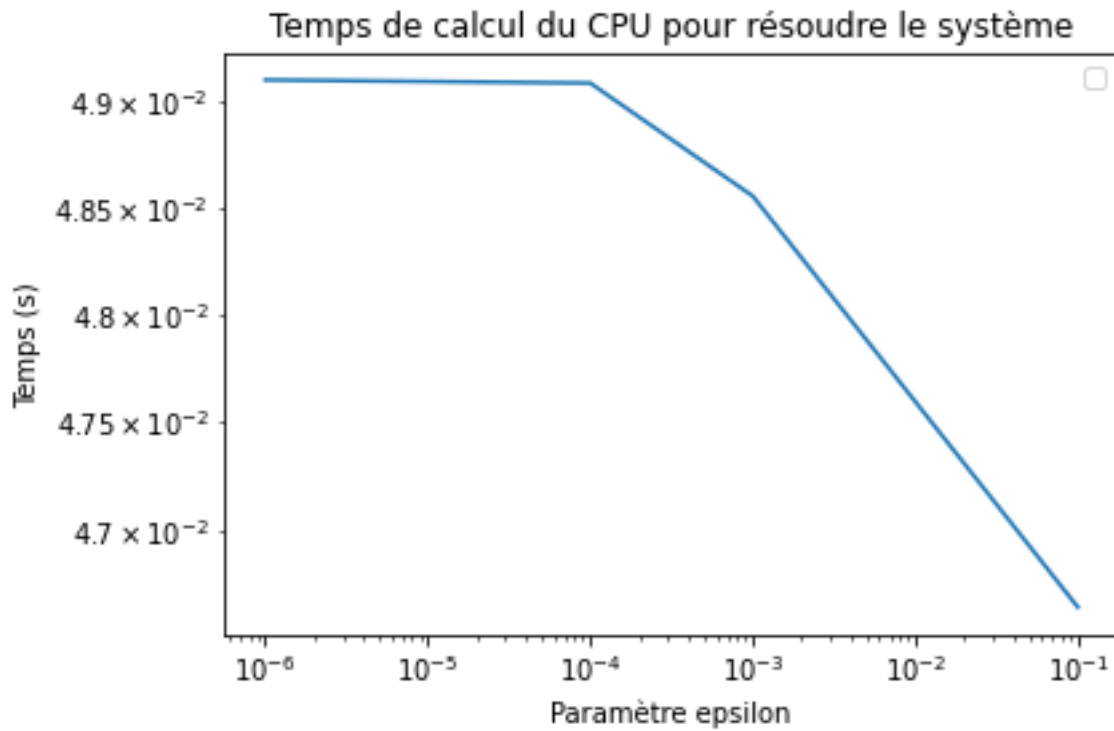
On étudie à présent l'influence du paramètre  $\epsilon$  sur la méthode de pénalisation. On s'attend à un effet violent sur le conditionnement car la matrice  $A + \frac{1}{\epsilon} B^T B$  est pleine à priori et donc aucun algorithme de *sparsing* ne fonctionne.

Nous utilisons les valeurs suivantes :

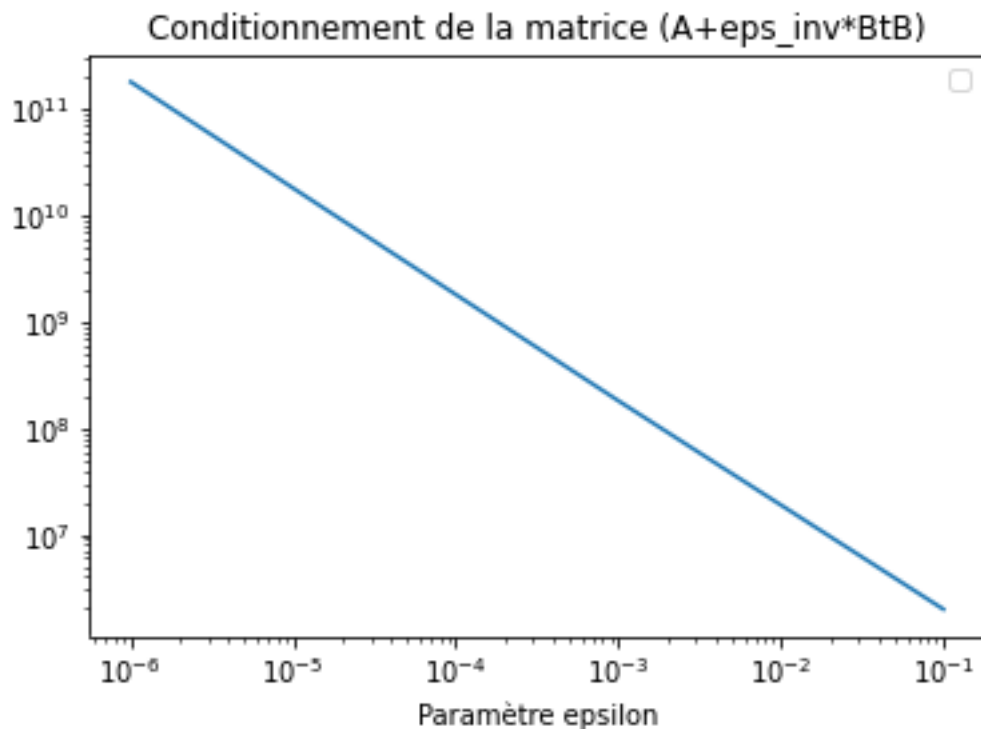
$$\epsilon \in \{10^{-1}, 10^{-3}, 10^{-4}, 10^{-6}\}$$

Pour cette étude, nous fixons le pas d'espace à  $h = 0.1$ .

---



En observant les échelles de temps, on remarque que la durée du calcul est essentiellement inchangée par les variations de  $\epsilon$ . Ceci est très cohérent car la structure de la matrice  $A + \frac{1}{\epsilon} B^T B$  n'est pas modifiée par la valeur du paramètre. L'algorithme de résolution prend donc des temps très similaires à s'effectuer.



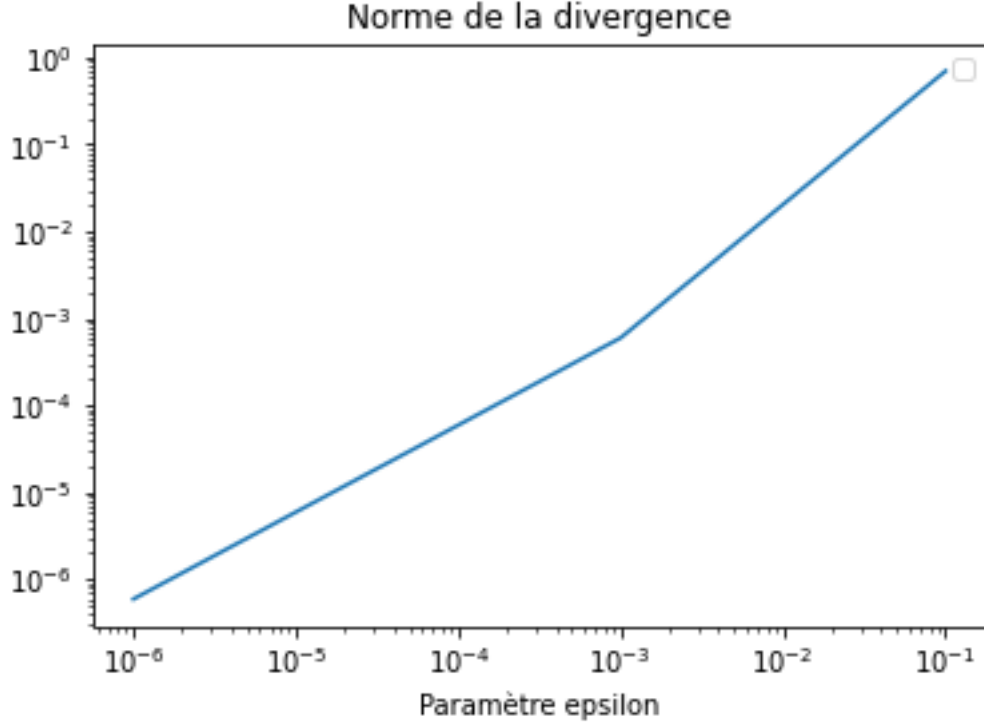
Le conditionnement quand à lui, est immédiatement lié à la valeur de epsilon (en réalité, la dépendance est quasi-linéaire). Et on aperçoit de nettes augmentations de celui-ci lorsque  $\epsilon$  est petit. On peut justifier ça ainsi :

Notons  $M_\epsilon = A + \frac{1}{\epsilon} B^T B$ .  $M_\epsilon$  étant symétrique, on a :

$$\text{cond}(M_\epsilon) = \rho(M_\epsilon) \rho(M_\epsilon^{-1})$$

où  $\rho$  est l'opérateur rayon spectral. Or, pour  $\epsilon$  petit, il est assez clair que  $\rho(M_\epsilon) \sim \frac{1}{\epsilon} \rho(B^T B)$  et  $\rho(M_\epsilon^{-1}) \sim \rho(A^{-1})$ . (Une preuve rigoureuse est possible, mais sans doute superflue ici)

On peut donc en conclure :  $\text{cond}(M_\epsilon) \sim \frac{1}{\epsilon} \rho(B^T B) \rho(A^{-1})$ .



La norme de la divergence de notre approximation finale est beaucoup diminuée par un  $\epsilon$  petit. Ceci découle immédiatement de la discussion faite plus haut sur la méthode de pénalisation : par construction, son objectif est de trouver le minimiseur d'une énergie  $E_\epsilon$  qui pénalise fortement les écarts à  $\ker B$ .

On peut se poser la question de l'utilisation mémoire. Bien sûr (comme pour le temps de calcul), la matrice à stocker  $M_\epsilon$  est de même nature peu importe la valeur de  $\epsilon$  : on obtient une valeur de 154.28 Mb. Ceci est à comparer avec les 44.20 Mb nécessaires pour la méthode globale (avec  $h = 0.1$ ).

Ceci nous prouve le défaut principale de cette méthode : la matrice  $B^T B$  étant pleine a priori, aucun outil sophistiqué de *sparsing* ne permet de diminuer la charge mémoire dont le programme a besoin.

### 3.3 Paramètre de dualité-pénalisation $\gamma$

Comme pour la méthode de pénalisation, on fixe  $h = 0.1$ . On s'intéresse ici à l'influence de  $\gamma > 0$  sur la méthode de dualité-pénalisation d'Usawa. Une chose à retenir, c'est que  $\gamma$  joue ici un rôle similaire à  $1/\epsilon$  dans le cas précédent.

Nous nous servons des valeurs :

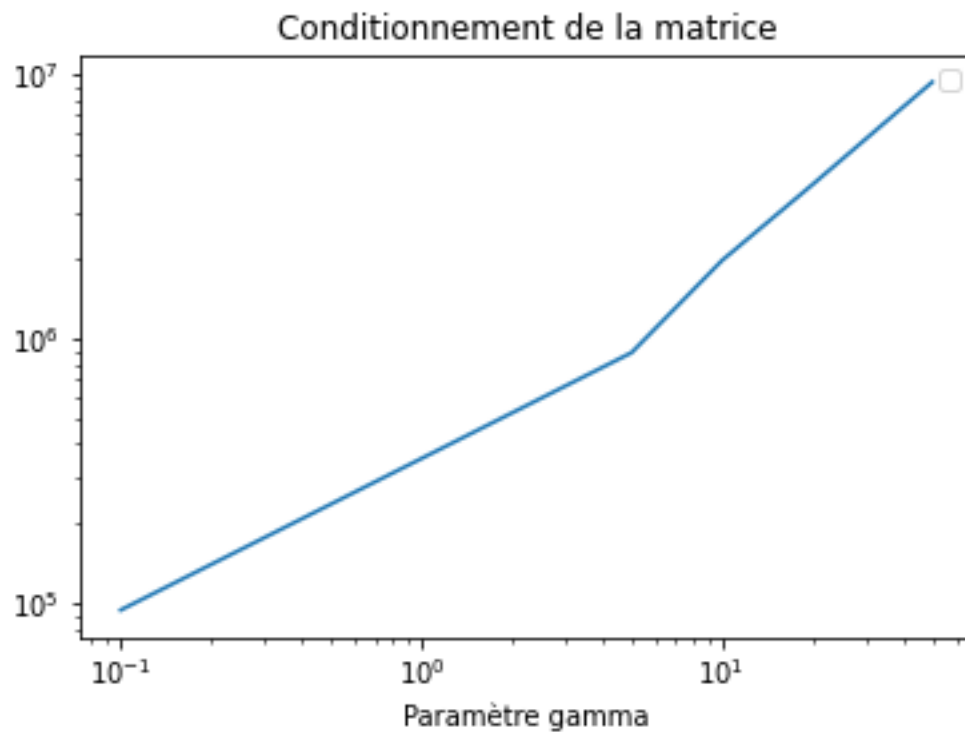
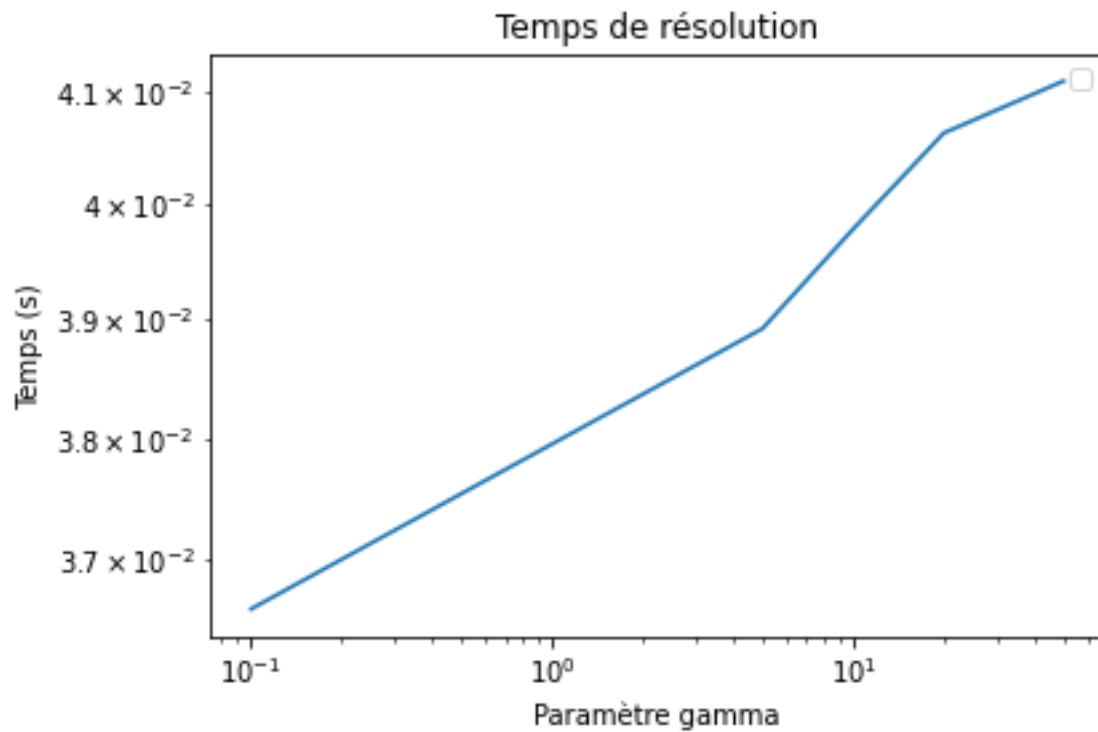
$$\gamma \in \{0.1, 5, 10, 20, 50\}$$

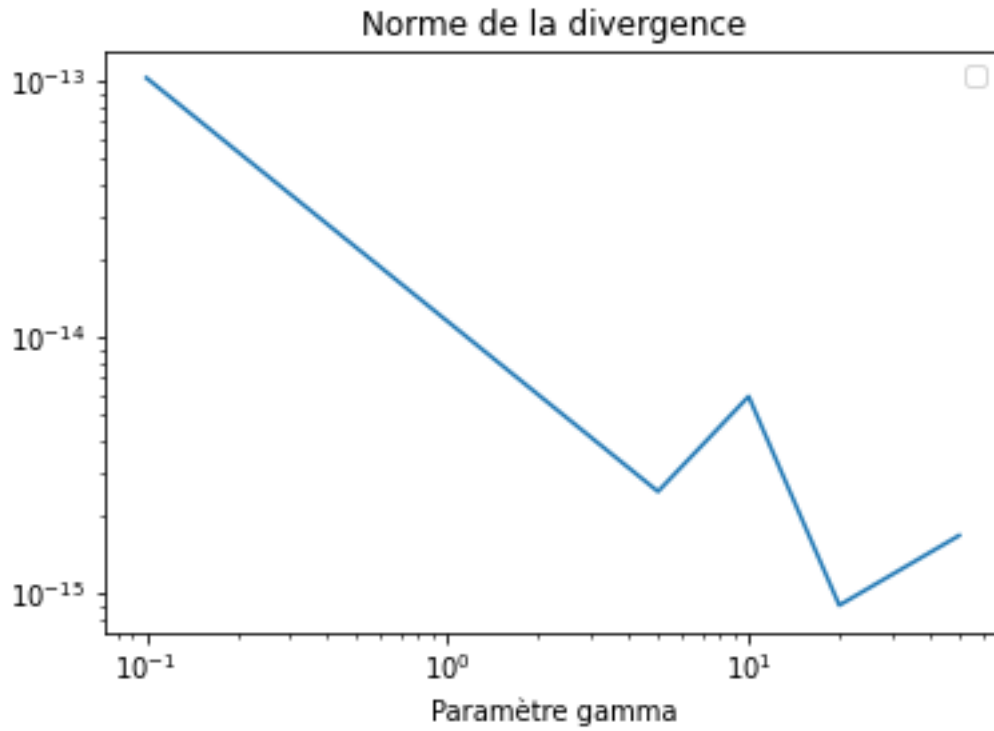
Très similairement à la méthode de pénalisation, les effets du paramètre  $\gamma$  se reconnaissent (voir ci-dessous). Le temps de calcul est inchangée, le conditionnement explose en  $\gamma$  grand, et la norme de la divergence se contrôle de mieux en mieux.

Un point qui vaut plus la peine d'être soulevé est la mémoire : en effet, comme précédemment, cette dernière n'évolue pas en fonction de  $\gamma$  mais est fixe à une valeur de  $154.28 + 125.63$  Mb. Les

premiers 154.28 Mb sont dûs au stockage de la matrice (pleine)  $A + \gamma B^T B$  (très exactement égale à la valeur retenue plus tôt), et les 125.63 Mb sont quant à eux dûs au stockage de la matrice d'Usawa. Le prix mémoire est donc augmenté.

---





### 3.4 Conclusion

Nous listons ici pour finir les avantages et inconvénients des différentes méthodes vues.

**Méthode globale** Cette méthode a l'avantage d'être directe et de ne faire intervenir que des matrices creuses, elle est donc peu coûteuse en charge mémoire.

**Méthode de pénalisation** L'astuce de pénaliser les écarts au noyau de  $B$  permet de se séparer de l'une des équation pour une précision pourtant importante, permettant ainsi de diminuer la taille des matrices considérées. Son principal défaut est sa perte de stabilité et son coût en mémoire.

**Dualité-pénalisation** Permet de bénéficier à la fois des avantages de la méthode par pénalisation et de celle d'Usawa. Elle est la plus précise concernant la norme de la divergence mais souffre tout de même du défaut de devoir stocker la matrice d'Usawa et donc d'augmenter (quasiment doubler !) la charge mémoire.