



Professeur : Angelo IOLLO

PROJET : PROBLÈMES INVERSES

31 janvier 2019

Gaëtan DESRUES
Ilyes MOUFID
Steven LAMPE

<https://github.com/GaetanDesrues/PbInverses.git>

Bordeaux INP - ENSEIRB-MATMECA
Département Mathématiques et Mécanique

Table des matières

1	Introduction du cas d'étude	2
2	Étude analytique	2
2.1	Existence d'un minimum	2
3	Sensitivity Analysis	3
4	Méthode de l'équation adjointe	5
4.1	Rappels	5
4.2	La méthode	5
5	Résultats	7
5.1	Sensitivity analysis	7
5.2	Méthode de l'équation adjointe	8
5.3	Comparaison des efficacités des méthodes	10
6	Conclusion	12

1 Introduction du cas d'étude

Soit $\Omega = [0, 1] \times [0, 1]$ le domaine d'étude, un entier $M \in \mathbb{N}^*$, une suite de coordonnées $(x_i)_{i \in \llbracket 0, M \rrbracket}$ de Ω et un ensemble de valeurs connues $(U_i^*)_{i \in \llbracket 0, M \rrbracket}$ de \mathbb{R} . Définissons de plus $(K_0, K_1, K_2) \in \mathbb{R} \times \mathbb{R}^2 \times S_2(\mathbb{R})$ avec $K_1 = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}$, $K_2 = \begin{pmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{pmatrix}$ et une fonction $g : \Omega \rightarrow \mathbb{R}$ tel que $\forall X \in \Omega, g(X) = K_0 + K_1^T X + X^T K_2 X$

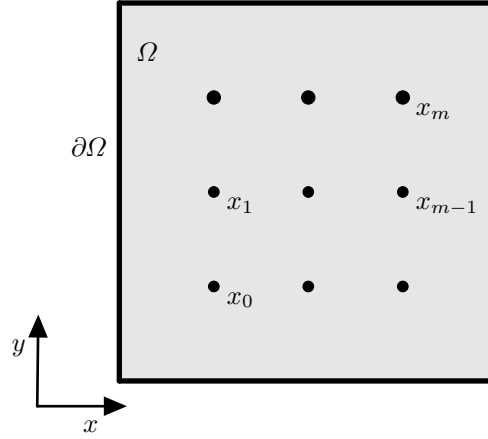


FIGURE 1 – Définition du domaine avec 9 capteurs

Soit $X_0 = (x_0, y_0) \in \Omega$, nous cherchons à minimiser une fonction f définie ainsi :

$$f : H^1(\Omega) \times \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$(u, X_0) \rightarrow \frac{1}{2} \sum_{m=1}^M (u(x_m) - u_m^*)^2 \quad (1)$$

dépendant donc de X_0 et d'une fonction u de $H^1(\Omega)$ qui est solution de l'équation suivante,

$$\begin{cases} -\Delta u(x, y) = g(x - x_0, y - y_0) & , \quad \forall (x, y) \in \Omega \\ u|_{\partial\Omega} = 0 \end{cases} \quad (2)$$

Dans un premier temps, nous avons cherché à savoir s'il y a existence de cette solution u et si oui sous quelle(s) condition(s). Pour cela nous nous sommes basés principalement sur des résultats démontrés en cours. Puis dans un second temps nous avons cherché à implémenter un algorithme permettant d'obtenir une solution approchée de ce problème à l'aide de FreeFem++, langage permettant de résoudre des EDP à l'aide de la méthode des éléments finis.

2 Étude analytique

2.1 Existence d'un minimum

Théorème d'existence d'un minimum

Soit K un fermé de \mathbb{R}^n , $f : K \rightarrow \mathbb{R}$ une fonction propre, continue et coercive.
Alors $\max_{x \in K} f(x)$ admet au moins une solution.

Remarque : Pour K borné l'argument de coercivité n'est pas nécessaire.

Nous cherchons l'existence d'un minimum pour la fonction f ci-dessous à l'aide du théorème précédent.

$$\begin{aligned} f : H^1(\Omega) \times \Omega &\rightarrow \mathbb{R} \\ (u, X_0) &\rightarrow \frac{1}{2} \sum_{m=1}^M (u(x_m) - u_m^*)^2 \end{aligned} \quad (3)$$

Tout d'abord, il est à noter que nous nous plaçons sur l'ensemble $\Omega = [0, 1] \times [0, 1]$ qui est un compact de \mathbb{R}^2 car c'est un produit cartésien de deux compacts. Ainsi, Ω est fermé et borné de \mathbb{R}^2 .

On définit la fonction suivante φ tel que :

$$\begin{aligned} \varphi : H^1(\Omega) \times \Omega &\rightarrow \mathbb{R} \\ X &\rightarrow \frac{1}{2} \sum_{m=1}^M (u(X) - u_m^*)^2 \end{aligned} \quad (4)$$

Montrons que φ est une fonction propre. Tout d'abord φ ne doit pas prendre la valeur " $-\infty$ ". Soit u solution de l'équation 2,

$$\forall X \in \Omega, \quad \varphi = \frac{1}{2} \sum_{m=1}^M (u(X) - u_m^*)^2 \geq 0 \quad (5)$$

Ainsi la fonction φ ne peut pas prendre de valeur négatives et donc ne peut pas tendre vers " $-\infty$ ".

Maintenant montrons que le domaine effectif de φ est non vide.

Soit $X \in \partial\Omega$, et u solution de l'équation 2 $\implies u(X) = 0$ car $u|_{\partial\Omega} = 0$.

Alors $\varphi(X) = \frac{1}{2} \sum_{m=1}^M (u_m^*)^2 \neq +\infty \implies$ le domaine effectif de φ est non vide.

De cette manière φ est une fonction propre.

Soit u solution de (2) $\implies u$ est dérivable sur Ω et donc continue sur Ω .

φ est donc une fonction continue. La fonction f est ainsi une fonction propre et continue par rapport au domaine Ω .

D'après le théorème d'existence d'un minimum, la fonction f admet au moins un minimum.

3 Sensitivity Analysis

Nous cherchons une solution u du problème (2), donc nous allons tout d'abord construire un système linéaire permettant d'obtenir une solution approchée. Pour cela, nous nous donnons une discrétisation spatiale de notre domaine Ω et en utilisant la méthode des éléments finis pour approcher la solution de l'équation (2), il vient un système linéaire :

$$\begin{aligned} MU &= b \\ \text{avec } U &= \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} K_0 + K_1^T(X_1 - X_0) + (X_1 - X_0)^T K_2(X_1 - X_0) \\ K_0 + K_1^T(X_2 - X_0) + (X_2 - X_0)^T K_2(X_2 - X_0) \\ \vdots \\ \vdots \\ K_0 + K_1^T(X_{N-1} - X_0) + (X_{N-1} - X_0)^T K_2(X_{N-1} - X_0) \\ K_0 + K_1^T(X_N - X_0) + (X_N - X_0)^T K_2(X_N - X_0) \end{pmatrix} \end{aligned}$$

où N représente le nombre total de degrés de liberté choisis, les X_i représentent les coordonnées des degrés de liberté du domaine et les u_i les approximations de la solution u en ces points X_i .

De plus, l'objectif principal est de minimiser f en fonction de X_0 . Une condition nécessaire est d'avoir $\vec{\nabla} f = \vec{0}$, c'est à dire :

$$\begin{cases} \frac{\partial f}{\partial x_0} = 0 \\ \frac{\partial f}{\partial y_0} = 0 \end{cases} \quad (6)$$

$$\quad (7)$$

Intéressons nous tout d'abord à (6) :

$$\frac{\partial f}{\partial x_0} = \frac{\partial u}{\partial x_0} \frac{\partial f}{\partial u} = \frac{\partial u}{\partial x_0} \sum_{m=1}^M (u(x_m) - u_m^*)$$

Nous ne connaissons pas $\frac{\partial u}{\partial x_0}$, mais nous pouvons peut être en chercher une approximation en utilisant le système linéaire $MU = b$ et en dérivant par rapport à x_0 . En faisant cela, il vient un système linéaire dont l'inconnue est $\frac{\partial}{\partial x_0} U$:

$$M \frac{\partial}{\partial x_0} U = \frac{\partial}{\partial x_0} b \quad (8)$$

De plus, pour obtenir $\frac{\partial}{\partial x_0} b$, il suffit de calculer $\frac{\partial g}{\partial x_0}$. Pour l'équation (7), la même démarche est utilisée et nous calculons donc par la même occasion $\frac{\partial g}{\partial y_0}$ ci-après.

$$\frac{\partial g}{\partial x_0} = -k_1 - 2k_{1,1}(x - x_0) - 2k_{1,2}(y - y_0)$$

$$\frac{\partial g}{\partial y_0} = -k_2 - 2k_{2,1}(x - x_0) - 2k_{2,2}(y - y_0)$$

Ainsi avec ces deux équations, il nous est possible d'obtenir $\frac{\partial}{\partial x_0} U$ et $\frac{\partial}{\partial y_0} U$. Puis, avec ces deux vecteurs, nous pouvons calculer $\frac{\partial f}{\partial x_0}$ et $\frac{\partial f}{\partial y_0}$. A partir de ces données, il nous est maintenant possible de chercher le minimum de f .

Au niveau de l'implémentation numérique, nous partons d'un couple d'éléments (x_0^0, y_0^0) pris arbitrairement dans Ω . Puis nous calculons (6) et (7). Nous cherchons à obtenir $\|\nabla f\| < \varepsilon$ où ε est pris petit (inférieur à 10^{-5}). Si cette conditions n'est pas satisfaite à l'itération k , nous modifions les valeurs de (x_0^{k+1}, y_0^{k+1}) en utilisant les valeurs (6) et (7) et un coefficient α choisi arbitrairement et représentant le coefficient de descente. Cela revient à :

$$\begin{cases} x_0^{n+1} = x_0^n - \alpha \frac{\partial f}{\partial x_0} \\ y_0^{n+1} = y_0^n - \alpha \frac{\partial f}{\partial y_0} \end{cases} \quad (9)$$

$$\quad (10)$$

4 Méthode de l'équation adjointe

4.1 Rappels

Rappel du problème

Soient $\Omega = [0, 1]^2$, m le nombre de capteurs, n le nombre de degrés de liberté associé à la méthode utilisée, M et b la matrice et le second membre du système linéaire obtenu pour le problème (2) et $U \in \mathbb{R}^n$ la solution de ce système linéaire :

$$MU = b \quad (11)$$

Le problème est de trouver le point $X_0 \in \Omega$ tel que :

$$X_0 = \min_{U \in \mathbb{R}^n} F(U)$$

avec

$$F = \frac{1}{2} (U_i - U_m^*)^T (U_i - U_m^*) \delta_{im}$$

On introduit les vecteurs $V \in \mathbb{R}^n$ tel que pour tout $i \in \llbracket 1, n \rrbracket$:

$$V_i = (U_i - U_m^*) \delta_{im}$$

et $\lambda \in \mathbb{R}^n$ défini par l'équation adjointe

$$M^T \lambda = V \quad (12)$$

De plus, on rappelle le développement de Taylor à l'ordre 1 :

$$f(a + h) = f(a) + \nabla f(a)h + o(h)$$

4.2 La méthode

On considère une petite perturbation $\delta U \in \mathbb{R}^n$ de la solution U telle que

$$\delta U = \delta_{x_0} U + \delta_{y_0} U$$

On introduit cette perturbation dans la relation (11) qui devient

$$M(U + \delta U) = b + \delta b$$

Puis en développant et en réutilisant (11), il vient

$$M\delta U = \delta b \quad (13)$$

On calcule ensuite δF , fonction F perturbée :

$$(\delta F)_i = (U_i - U_m^*) \delta U_i \delta_{im} = (V^T \delta U)_i$$

Soit

$$\delta F = V^T \delta U \quad \text{et} \quad \delta F^T = \delta U^T V$$

En multipliant à gauche l'équation (12) par δU^T , on retrouve δF grâce à la formule précédente

$$\delta U^T M^T \lambda = \delta U^T V = \delta F^T \quad (14)$$

Pour résumé, nous avons les relations suivantes :

$$\begin{cases} M\delta U = \delta b \\ \delta F = \lambda^T M\delta U \end{cases} \quad \begin{matrix} (15) \\ (16) \end{matrix}$$

En combinant ces deux équations nous obtenons une relation liant δF à δb et λ :

$$\delta F = \lambda^T \delta b \quad \Longleftrightarrow \quad \delta_{x_0} F + \delta_{y_0} F = \lambda^T \delta_{x_0} b + \lambda^T \delta_{y_0} b$$

ceci conduit à :

$$\begin{cases} \delta_{x_0} F = \lambda^T \delta_{x_0} b \\ \delta_{y_0} F = \lambda^T \delta_{y_0} b \end{cases} \quad \begin{matrix} (17) \\ (18) \end{matrix}$$

Comme nous cherchons le minimum de F , nous cherchons à minimiser δF . De la même manière que pour la méthode précédente, numériquement nous cherchons à avoir $|\delta F| < \varepsilon$. Pour cela nous partons d'un couple (x_0, y_0) choisi arbitrairement, puis nous le modifions à chaque itération pour diminuer la valeur de $|\delta F|$ en utilisant les relations (17) et (18) :

$$\begin{cases} x_0^{n+1} = x_0^n - \alpha \delta_{x_0} F \\ y_0^{n+1} = y_0^n - \alpha \delta_{y_0} F \end{cases} \quad \begin{matrix} (19) \\ (20) \end{matrix}$$

5 Résultats

Après avoir déterminé les équations à résoudre pour les deux méthodes utilisées, nous vous présentons dans cette partie nos résultats.

5.1 Sensitivity analysis

Dans un premier temps, nous avons cherché à trouver un coefficient de décente α convenable, coefficient que nous prenons constant durant toute la durée d'une exécution du programme, pour obtenir la solution recherchée, c'est à dire le bon couple (x_0, y_0) . Nous avons remarqué que prendre le coefficient trop grand faisait diverger le programme, tandis que le prendre trop petit augmentait considérablement le nombre d'itérations et donc le temps mis par le programme pour obtenir la bonne solution. De plus, la limite pour laquelle α fait exploser la solution dépend du cas traité (nombre de mailles, nombre de capteurs...), donc nous l'avons ajusté selon les cas, pris entre 1 et 10^{-2} .

Ensuite, nous avons cherché pour un maillage donné et une position donnée (x_0, y_0) , l'influence du nombre de capteurs sur l'efficacité du programme à retrouver (x_0, y_0) . Pour cela nous avons pris comme position initiale $(x_0^0, y_0^0) = (0.1, 0.9)$ et comme position recherchée $(x_0, y_0) = (0.5, 0.9)$ et nous avons tracé l'évolution de la position (x_n^0, y_n^0) .

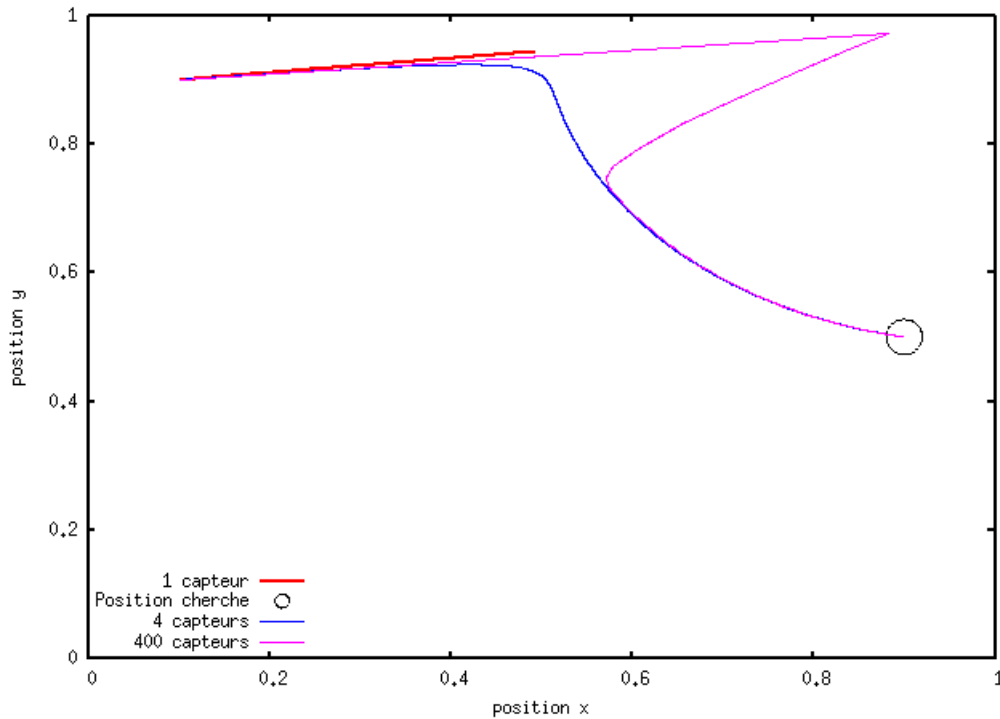


FIGURE 2 – Évolution de la position (x_n^0, y_n^0) dans Ω pour différents nombres de capteurs avec un critère d'arrêt $\epsilon = 10^{-5}$, une position initiale $(0.1, 0.9)$ et une position recherchée $(0.5, 0.5)$.

Tout d'abord il est intéressant de noter sur la figure (2) que pour un seul capteur (courbe rouge) la solution ne converge pas vers la position souhaitée (cercle noir) mais vers une autre solution. Au delà de un capteur, la solution converge vers la position théorique. En effet, nous avons montré dans la section 2 qu'il existe au moins une solution à notre problème, cependant si le nombre de capteurs est insuffisant, il semblerait qu'il existe plusieurs solutions

au problème. Cependant, au vu de la configuration des capteurs, nous n'avons pas testé avec deux et trois capteurs pour déterminer exactement la limite pour obtenir l'unicité de la solution. Nous supposons donc qu'il est suffisant d'avoir plus de 4 capteurs pour converger vers la bonne solution.

De plus nous pouvons observer que pour 400 capteurs, la position s'éloigne de la trajectoire prise pour 4 capteurs, et ceci est dû au trop grand pas de descente et à l'erreur plus grande car mieux calculée.

Ensuite, nous avons étudié le nombre d'itérations nécessaires pour converger en fonction du nombre de capteurs.

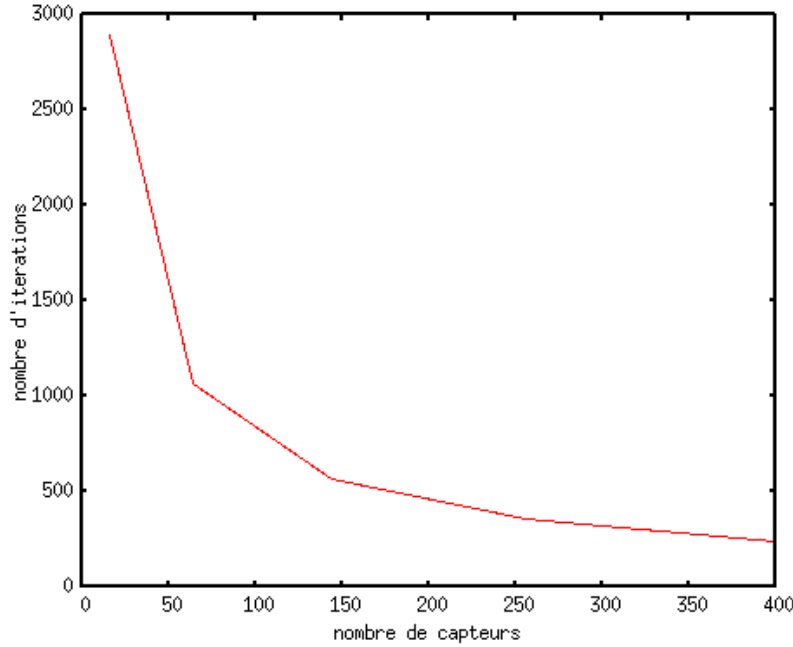


FIGURE 3 – Évolution du nombre d'itérations en fonction du nombre de capteurs avec un critère d'arrêt $\varepsilon = 10^{-5}$.

Nous pouvons observer que le nombre d'itérations diminue lorsque le nombre de capteurs augmente. Ceci est tout à fait en accord avec la réalité : lorsqu'il y a beaucoup de capteurs la position souhaitée a plus de chance d'être trouvée rapidement. De plus, nous pouvons observer que l'efficacité varie fortement pour un faible nombre de capteurs, mais lorsque le nombre de capteurs est très grand, le nombre d'itérations nécessaire pour l'obtention de la bonne position reste proche.

5.2 Méthode de l'équation adjointe

Pour cette méthode, le coefficient de descente α sera compris entre 2 et 10^{-2} en fonction des cas. Nous avons effectué la même étude que pour la méthode précédente en cherchant la position $(x_0, y_0) = (0.5, 0.5)$ et en prenant comme position initiale $(x_0^0, y_0^0) = (0.1, 0.9)$.

D'après la figure 4, nous obtenons des résultats similaires à la méthode précédente. En effet, pour un capteur, la position calculée converge vers une position qui n'est pas celle voulue. De plus lorsqu'on augmente le nombre de capteurs, la solution converge bien vers la position souhaitée.

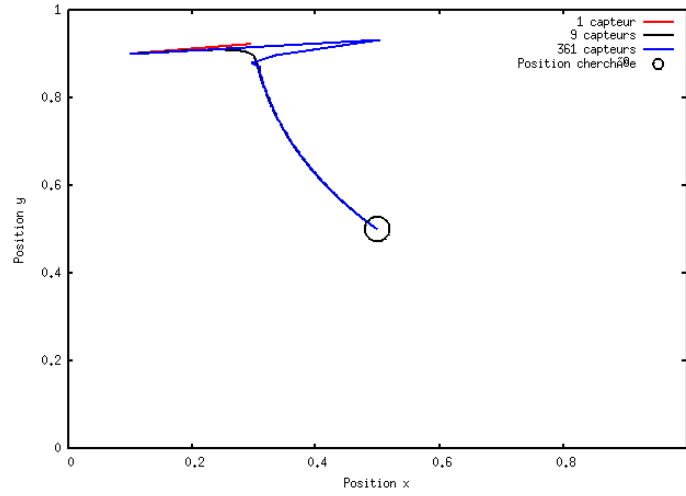


FIGURE 4 – Évolution de la position (x_0^n, y_0^n) dans Ω pour différents nombres de capteurs avec un critère d'arrêt $\epsilon = 10^{-5}$, une position initiale $(0.1, 0.9)$ et une position recherchée $(0.5, 0.5)$.

L'exemple précédent à pour position cherchée une position où $x_0 = y_0$, mais pour une position cherchée avec x_0 différent de y_0 , nous ne convergions pas vers la bonne solution. En effet, en prenant comme position cherchée $(0.9, 0.5)$ et une position initiale égale à $(0.1, 0.1)$, les résultats ne sont plus corrects.

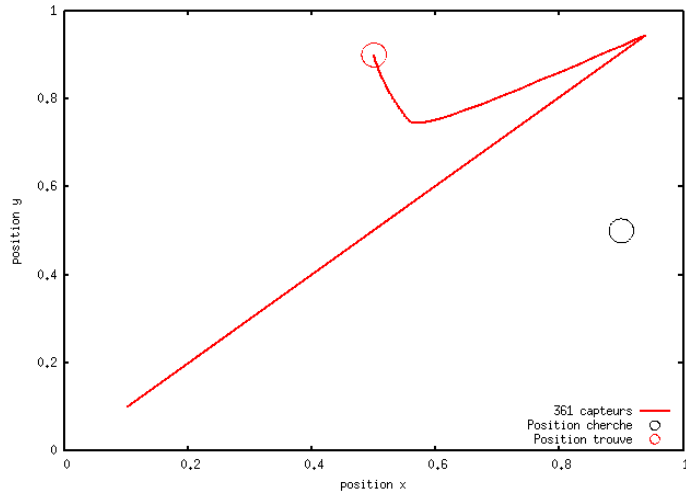


FIGURE 5 – Évolution de la position (x_0^n, y_0^n) dans Ω pour différents nombres de capteurs avec un critère d'arrêt $\epsilon = 10^{-5}$, une position initiale $(0.1, 0.1)$, une position recherchée $(0.9, 0.5)$ et la position trouvé $(0.5, 0.9)$.

Comme le montre la figure 5, la solution converge vers (y_0, x_0) , le symétrique de la solution cherchée par rapport à $y = x$. Nous n'avons pas réussi à trouver l'erreur dans le programme qui induit cela. Et nous avons observé ce phénomène pour tous nos autres cas testés. Mais nous avons tout de même étudié l'efficacité de cette méthode en faisant le même test que précédemment, c'est à dire d'étudier le nombre d'itérations pour converger pour différents nombres de processeurs.

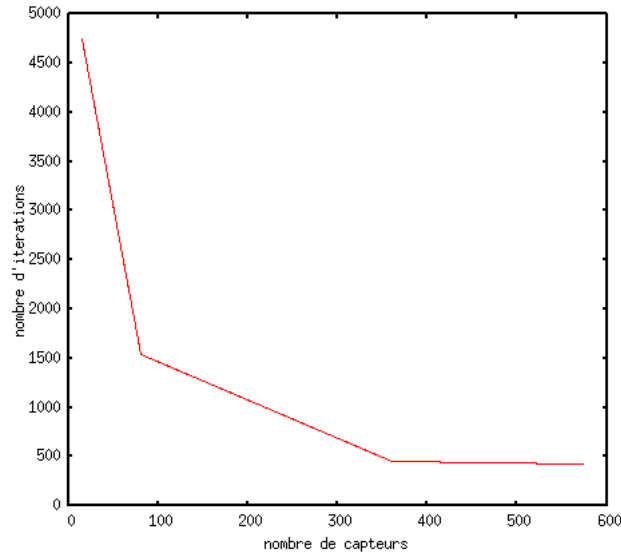


FIGURE 6 – Évolution du nombre d'itérations en fonction du nombre de capteurs avec un critère d'arrêt $\varepsilon = 10^{-5}$.

Nous pouvons observer des résultats semblables à ceux obtenus pour la méthode de l'analyse de sensibilité. Le nombre d'itérations diminue lorsque le nombre de capteurs augmente. Cependant nous pouvons observer qu'il y a légèrement plus d'itérations que pour la première méthode. Pour mieux observer cette différence nous avons étudié la différence des résultats obtenus avec les deux méthodes dans la partie suivante.

5.3 Comparaison des efficacités des méthodes

Nous avons tout d'abord affiché le nombre d'itérations effectué pour des même nombres de capteurs et avec les deux méthodes.

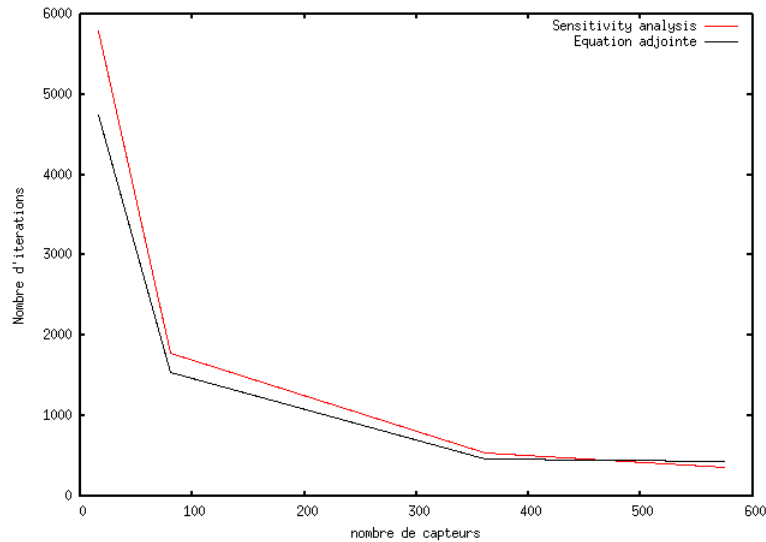


FIGURE 7 – Évolution du nombre d'itérations en fonction du nombre de capteurs avec un critère d'arrêt $\varepsilon = 10^{-5}$ pour les deux méthodes.

La figure précédente appuie ce qui a été dit précédemment : le nombre d'itération est légèrement inférieur pour la méthode de l'équation adjointe sauf lorsque le nombre de capteurs est grand. Nous avons de plus étudié le temps d'exécution des deux méthodes pour différents nombres de capteurs afin de comparer leur efficacité.

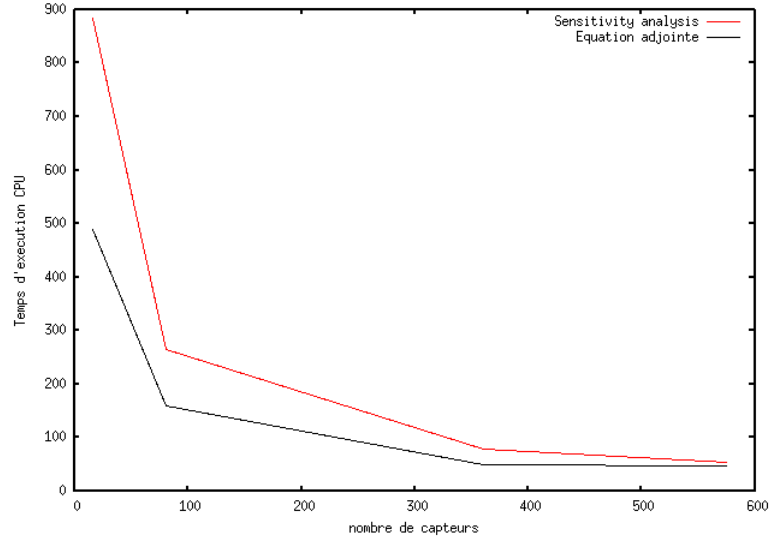


FIGURE 8 – Temps d'exécution en secondes du programme pour différents nombres de capteurs avec $\alpha = 0.5$ et un critère d'arrêt $\varepsilon = 10^{-5}$.

Nous pouvons observer que la méthode de l'équation adjointe est plus rapide que la méthode de l'analyse de sensibilité, avec un rapport proche de deux. Ceci est compréhensible en raison du fait que le nombre d'itérations est similaire, mais la méthode de l'équation adjointe demande de résoudre un seul système linéaire, contrairement à l'analyse de sensibilité qui requiert deux résolutions. Cependant pour un grand nombre d'itérations les temps sont semblables.

6 Conclusion

Ce projet nous a permis de programmer deux méthodes de résolution pour un problème inverse. A partir de l'émission d'une source, nous avons été capable de retrouver les coordonnées de cette source dans le domaine considéré à l'aide de la méthode de l'analyse de sensibilité. Cependant, les résultats n'ont pas été totalement satisfaisants pour la méthode utilisant l'équation adjointe.

De plus, nous avons pu observer sur quelques résultats que, comme attendu, la convergence est plus rapide lorsque le nombre de capteurs augmente. Nous avons de plus comparé les deux méthodes, et nous avons pu observer que la seconde méthode de l'équation adjointe est plus rapide, en raison du nombre de systèmes à résoudre.

Il reste néanmoins à s'intéresser aux propriétés de la solution lorsque les capteurs sont répartis aléatoirement dans le domaine et non pas de façon organisée, ainsi qu'une étude plus poussée sur l'unicité de la solution.