

Mise à jour de Broyden des sensibilités pour FEMU : que vaut cette troisième voie sans problème adjoint ni calcul perturbatif ?

Renaud Ferrier¹, Paul Baral¹

¹ Mines Saint-Étienne, Laboratoire Georges Friedel (UMR 5307)

Résumé — Dans le cadre de l'identification paramétrique par la méthode de mise à jour de modèles éléments finis (FEMU), on s'intéresse à une approche consistant à mettre à jour les sensibilités (dérivées partielles des observables) à l'aide de la méthode de Broyden. Cette approche promet en théorie un coût CPU par itération très faible (un seul calcul direct par itération) pour un coût d'implémentation extrêmement réduit (par rapport notamment à la mise en place d'un problème adjoint). On propose dans cette contribution d'évaluer l'approche sur des problèmes de complexité et de réalisme croissants.

Mots clés — Mise à jour de Broyden, FEMU, identification de paramètres, Réduction de temps de calcul, problèmes inverses

1 La clef de l'identification : le gradient

Soit un vecteur u , que l'on suppose être l'unique solution d'une équation différentielle pour un système continu, ou équation vectorielle de dimension finie pour un système discrétisé. Cette équation est notée de façon abstraite $\mathcal{F}(u; \theta) = 0$, où θ est un vecteur paramétrant l'équation \mathcal{F} . On cherche à identifier θ à partir d'une mesure, notée u_m . Si le modèle et la mesure étaient parfaits, on aurait $\Pi u = u_m$, où Π est l'opérateur d'observation.

1.1 La méthode FEMU

Il existe une grande variété de méthodes d'identification de paramètres, allant de la méthode des champs virtuels aux réseaux de neurones informés par la physique (PINNs) inverses. Dans cette diversité, la méthode de mise à jour de modèles éléments finis (FEMU) peut être considérée comme incontournable. Son principe est de formuler l'identification sous la forme du problème d'optimisation suivant :

$$\min_{\substack{u, \theta \\ \mathcal{F}(u; \theta) = 0}} \varphi(u), \quad \varphi(u) = \frac{1}{2} \|\Pi u - u_m\|^2 \quad (1)$$

où le problème $\mathcal{F}(u; \theta) = 0$ est la discrétisation par éléments finis d'un problème continu.

Le fait d'imposer de façon ferme le respect de la physique du problème, au sens discret, via la condition $\mathcal{F}(u; \theta) = 0$ a une vertu régularisante. Un autre attrait de la méthode est sa capacité à pouvoir s'appuyer de façon non-intrusive sur un solveur par éléments finis qui, une fois le vecteur paramétrique θ identifié, peut être utilisé pour résoudre des problèmes directs.

L'inconvénient principal de la méthode est son coût CPU. Ceci vient du fait que les méthodes classiquement utilisées pour résoudre le problème (1) sont itératives, et qu'à chaque itération il faut appeler le solveur par éléments finis à plusieurs reprises pour résoudre le problème direct $\mathcal{F}(u; \theta) = 0$.

On définit la fonction de résolution $\mathcal{S} : \theta \mapsto \{u \text{ tel que } \mathcal{F}(u; \theta) = 0\}$, et on réécrit par substitution le problème (1) sous la forme non-contrainte :

$$\min_{\theta} \psi(\theta), \quad \psi(\theta) = \varphi(\mathcal{S}(\theta)) \quad (2)$$

Les solveurs les plus communs, et les plus efficaces en terme de vitesse de convergence, pour résoudre ce type de problème font usage du gradient $\nabla \psi$, et parfois de la Hessienne $\nabla^2 \psi$. Pour calculer ces objets, on a le choix entre deux approches, à la croisée de deux compromis classiques : élégance

contre pragmatisme et performance contre rusticité. La première approche est la méthode de l'équation adjointe, qui peut s'établir via le théorème des multiplicateurs de Lagrange. La seconde est la méthode perturbative, qui peut être retrouvée à l'aide de la technique de substitution.

1.2 Méthode de l'équation adjointe

On peut montrer que le gradient recherché est égal au gradient partiel par rapport à θ du Lagrangien associé au problème (1) évalué en un point où ce Lagrangien est stationnaire par rapport à u et au multiplicateur de Lagrange :

$$\mathcal{L}(u, \lambda; \theta) = \varphi(u) + \lambda^T \mathcal{F}(u; \theta), \quad \nabla \psi(\theta) = \nabla_{\theta} \mathcal{L}(u_0, \lambda_0; \theta) \quad \text{avec} \quad \begin{cases} \nabla_u \mathcal{L}(u_0, \lambda_0; \theta) = 0 \\ \nabla_{\lambda} \mathcal{L}(u_0, \lambda_0; \theta) = 0 \end{cases} \quad (3)$$

En développant, on établit que le système est triangulaire par blocs. $\nabla_{\lambda} \mathcal{L} = 0$ est équivalent à l'équation d'état (ou équation directe) $\mathcal{F}(u_0; \theta) = 0$, tandis que $\nabla_u \mathcal{L} = 0$ est équivalent à l'équation linéaire $\nabla_u \mathcal{F}(u_0; \theta)^T \lambda = -\nabla \varphi(u)$. Cette dernière équation, appelée équation adjointe, a une structure plutôt proche de celle de l'équation d'état, ce qui fait qu'elle peut souvent être résolue à l'aide du même solveur par éléments finis.

La méthode de l'équation adjointe conduit donc à résoudre seulement deux problèmes éléments finis par itération, dont l'un (au moins) est linéaire. Cependant, l'implémentation de cette résolution est d'autant moins triviale que le problème d'origine $\mathcal{F}(u_0; \theta) = 0$ est complexe, et que le solveur utilisé est un solveur industriel. Il en va de même pour l'évaluation de $\nabla_{\theta} \mathcal{L}$. C'est cette complexité d'implémentation qui limite la portée de cette approche.

La détermination de la Hessienne $\nabla^2 \psi$ est facultative, mais elle ouvre l'accès à des méthodes d'optimisation d'ordre 2, basées sur la méthode de Newton. Cette Hessienne est communément approchée à l'aide de la méthode BFGS, qui consiste en une mise à jour itérative de Broyden (voir section 2.1). Afin de lever toute ambiguïté, il convient de préciser que l'approche étudiée dans cette contribution est sans lien avec la méthode BFGS, bien qu'elle fasse également usage d'une mise à jour de Broyden, mais dans le cadre de la deuxième méthode de calcul de $\nabla \psi$ et $\nabla^2 \psi$, qui est l'approche perturbative.

1.3 Approche perturbative

À l'aide de la loi de composition des dérivées, on obtient les résultats suivants :

$$\begin{aligned} \nabla \psi(\theta) &= \nabla_{\theta} \mathcal{S}(\theta)^T \nabla_u \varphi(\mathcal{S}(\theta)) = \nabla_{\theta} \mathcal{S}(\theta)^T \Pi^T (\Pi \mathcal{S}(\theta) - u_m) \\ \nabla^2 \psi(\theta) &= \nabla_{\theta} \mathcal{S}(\theta)^T \Pi^T \Pi \nabla_{\theta} \mathcal{S}(\theta) + \nabla_{\theta}^2 \mathcal{S}(\theta)^T \Pi^T (\Pi \mathcal{S}(\theta) - u_m) \end{aligned} \quad (4)$$

où la notation $\nabla_{\theta}^2 \mathcal{S}(\theta)^T$ est très peu rigoureuse puisqu'il s'agit d'un tenseur d'ordre 3. Ce second terme de la Hessienne cumule deux défauts (en plus de la difficulté de l'écrire) : il est difficile à construire numériquement, et il a tendance à déstabiliser les algorithmes de résolution (il n'est pas défini-positif ni même symétrique). C'est pour toutes ces raisons qu'il est ignoré, et que l'on remplace en pratique $\nabla^2 \psi(\theta)$ par $\mathcal{H}(\theta) = \nabla_{\theta} \mathcal{S}(\theta)^T \Pi^T \Pi \nabla_{\theta} \mathcal{S}(\theta)$. La pseudo-méthode de Newton annulant $\nabla \psi(\theta)$ et utilisant cette Hessienne approchée est alors en fait une méthode de Levenberg-Marquardt (voir par exemple [1]).

Quoi qu'il en soit, un élément central de (4) est la matrice $\nabla_{\theta} \mathcal{S}(\theta)$, souvent notée \mathcal{J} car il s'agit de la Jacobienne de \mathcal{S} . Cette Jacobienne est usuellement déterminée par méthode perturbative, c'est à dire que pour chaque composante d de θ , on construit le vecteur paramétrique perturbé $\theta + h e_d$, où h est un réel ayant vocation à être petit, et e_d le vecteur élémentaire dans la direction d . Et on utilise la formule approchée suivante :

$$\frac{\partial \mathcal{S}}{\partial \theta_d} \simeq \frac{\mathcal{S}(\theta + h e_d) - \mathcal{S}(\theta)}{h} \quad (5)$$

Pour obtenir la Jacobienne $\nabla_{\theta} \mathcal{S}(\theta)$ complète, on doit donc évaluer la fonction \mathcal{S} en $n + 1$ points, n étant la dimension du vecteur paramétrique θ . Chaque évaluation de \mathcal{S} demandant de faire appel au

solveur éléments finis, le coût CPU de cette méthode peut être prohibitif dans les cas où l'espace paramétrique est de dimension relativement élevée. En revanche, une fois que le modèle direct a été construit, la méthode a l'avantage de la simplicité.

Notons que certains auteurs utilisent la méthode BFGS dans le cadre de cette approche perturbative. Une mise à jour de type Broyden est alors utilisée sur $\nabla\psi(\theta)$ déterminée par l'approche perturbative, pour en déduire la Hessienne. La méthode étudiée dans notre contribution est différente puisque, comme c'est expliqué à la section 2, la formule de Broyden sera appliquée sur $\mathcal{S}(\theta)$ et pas $\nabla\psi(\theta)$.

2 FEMU avec mise à jour de Broyden

2.1 Formule de Broyden

Dans [2], Broyden propose une méthode itérative permettant d'estimer une matrice \mathcal{J} approchant la Jacobienne $\nabla\mathcal{S}$ à partir d'évaluations successives de la fonction à valeurs vectorielles \mathcal{S} . La formule de mise à jour est la suivante :

$$\begin{aligned}\delta_i &= \theta_{i+1} - \theta_i \\ y_i &= \mathcal{S}(\theta_{i+1}) - \mathcal{S}(\theta_i) \\ \mathcal{J}_{i+1} &= \mathcal{J}_i + \frac{(y_i - \mathcal{J}_i\delta_i)\delta_i^T}{\delta_i^T\delta_i}\end{aligned}\tag{6}$$

Cette formule est en fait une généralisation de l'approximation de la dérivée par la pente de la corde entre θ_i et θ_{i+1} . Pour être valable, elle nécessite que ces deux points soient proches. En particulier, il est de bon ton que \mathcal{S} soit monotone entre θ_i et θ_{i+1} . Ceci est assuré par la première condition de Wolfe (relation 3.6a de [3]).

Deux articles plus récents [4, 5] permettent de prendre un peu plus de recul sur cette formule, notamment dans son contexte originel, qui est celui de la résolution du système non-linéaire $\mathcal{S}(\theta) = 0$ par méthode de quasi-Newton. Dans ce contexte, chaque \mathcal{J}_{i+1} approxime le $\nabla\mathcal{S}(\theta_{i+1})$ correspondant en supposant qu'il est proche de $\nabla\mathcal{S}(\theta_i)$. Cette méthode était donc initialement destinée à s'appliquer à des Jacobiennes carrées, mais rien n'interdit de l'utiliser dans le cas où $\nabla\mathcal{S}$ est rectangulaire.

Remarque 1 *C'est une variante de cette formule qui est utilisée dans l'algorithme BFGS permettant d'évaluer une Hessienne, qui n'est autre que la Jacobienne du gradient. Broyden donne d'ailleurs son B liminaire au sigle BFGS.*

Au vu du coût CPU de l'approche perturbative présentée dans la section 1.3, il paraît plus que tentant d'utiliser la formule de Broyden pour mettre à jour la matrice des sensibilités $\nabla_{\theta}\mathcal{S}(\theta)$ à chaque itération de l'approche perturbative. Et ainsi de se passer de l'équation (5), et avec elle de son lot d'appels au solveur éléments finis.

Cette idée n'est pas nouvelle. Certaines communautés semblent l'utiliser plutôt abondamment dans le cadre de l'algorithme de Levenberg-Marquardt [6, 7]. Cependant, à notre connaissance, il n'existe que peu de comparaisons numérique [8] ou théorique [9] entre des algorithmes utilisant la formule de Broyden d'une part et un re-calcul perturbatif complet via l'équation (5) d'autre part. Cette mise à jour de Broyden n'est par ailleurs jamais associée au mot-clef FEMU, ni mentionnée par la communauté mécanique. La présente contribution comblera ces manques.

L'application de la formule de Broyden permet de réduire drastiquement le coût CPU de chaque itération de l'algorithme d'optimisation. Il est cependant légitime de craindre les contreparties suivantes, principalement dues au fait que l'estimation des sensibilités est de moindre qualité :

- La nécessité d'effectuer davantage d'itérations pour converger ;
- La propension à converger vers des minima locaux ;
- Une instabilité accrue, ou la nécessité d'effectuer davantage de pas de recherche linéaire (*line-search*) pour atteindre la condition de Wolfe (nécessaire pour que la formule (6) soit pertinente).

En particulier, l’archive [8] documente numériquement le comportement de l’algorithme sur des problèmes mathématiques modèles. Les résultats semblent montrer la viabilité de l’approche sur la plupart de problèmes. L’objectif de notre contribution est de présenter des expériences numériques, sur données synthétiques ou réelles, afin de documenter le comportement de cette méthode sur des problèmes intéressant la communauté des mécaniciens.

2.2 De l’importance des valeurs initiale et finale de la Jacobienne

La formule exposée dans la section 2.1 permet de mettre à jour l’estimateur de $\nabla_{\theta}\mathcal{S}(\theta)$ à partir d’un estimateur à l’itération précédente. Se pose alors la question de l’initialisation : que vaut \mathcal{J}_0 ? On peut proposer différentes réponses, qui seront évaluées lors de la présentation orale. Mais la plus simple d’entre elles, celle qui est utilisée dans les exemples numériques de ce document, consiste à effectuer une étude de sensibilité complète à l’aide de la formule (5) à l’itération 0.

Par ailleurs, dans le cadre d’une ré-interprétation Bayésienne de la méthode, et en notant C_m la covariance du bruit de mesure, il est courant d’utiliser $\mathcal{H}(\theta)^{-1} = (\nabla_{\theta}\mathcal{S}(\theta)^T \Pi^T C_m^{-1} \Pi \nabla_{\theta}\mathcal{S}(\theta))^{-1}$ comme approximation de la matrice de covariance de θ . Dans ce contexte, si on estime que la Jacobienne approchée par mise à jour de Broyden est insuffisamment précise, il sera nécessaire d’effectuer, en fin d’optimisation, une nouvelle étude de sensibilité complète à l’aide de la formule (5).

3 Cas-test à données synthétiques

Ce premier cas-test utilise des données synthétiques et un solveur éléments finis maison associé au logiciel de maillage GMSH [10]. Ceci permet notamment de comparer l’approche proposée avec mise à jour de Broyden non-seulement à la méthode perturbative avec re-calcul complet de \mathcal{J} , mais aussi à la méthode de l’équation adjointe associée au solveur BFGS.

3.1 Présentation du problème

On s’intéresse au problème de l’identification de la perméabilité hétérogène d’un milieu poreux à partir de données redondantes (pression et débit) en quatre buses situées en chacun des coins du domaine. En dehors des quatre buses, les bords du domaine sont supposés étanches. La perméabilité cible, utilisée pour générer les données synthétiques, vaut 1 dans la zone périphérique et 2 dans l’inclusion circulaire. La géométrie et le maillage utilisés sont affichés sur la Figure 1.

Le problème résolu pour générer les données est celui de l’équation (7). Il s’agit de l’équation de Darcy pour la pression $p \in H^1(\Omega)$, où Ω est un ouvert de \mathbb{R}^2 , doté de ses bords Γ_d (pression imposée sur la surface des buses) et Γ_n (vitesse normale nulle sur le reste du bord) et de normale sortante \underline{n} . $k \in L^2(\Omega)$ est une perméabilité inhomogène.

Le problème direct donnant corps à l’équation abstraite $\mathcal{F}(u; \theta)$ des sections précédente est celui de l’équation (8). Ici, la pression est notée u et la perméabilité à identifier est θ . On utilise le débit Q supposé mesuré en chaque buse pour calculer une vitesse normale moyenne V sur la buse, qui sert de condition de Neumann sur la surface de ces buses. Sur le reste du bord, la vitesse normale est toujours fixée nulle.

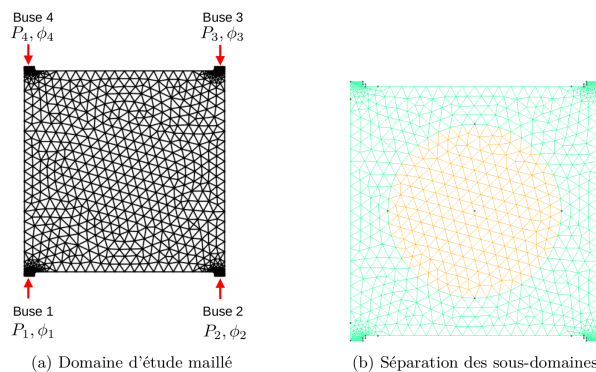


FIGURE 1 – Géométrie du problème et discrétisation par éléments finis

Il s'agit donc d'un problème de Neumann, pour lequel on fixe la constante d'intégration en imposant la moyenne des pressions sur la surface des buses.

$$\begin{cases} \operatorname{div}(k \underline{\operatorname{grad}}(p)) = 0 & \text{dans } \Omega \\ k \frac{\partial p}{\partial \underline{n}} = 0 & \text{sur } \Gamma_n \\ p = P_d & \text{sur } \Gamma_d \end{cases} \quad (7)$$

$$\begin{cases} \operatorname{div}(\theta \underline{\operatorname{grad}}(u)) = 0 & \text{dans } \Omega \\ \theta \frac{\partial u}{\partial \underline{n}} = V & \text{sur } \partial\Omega \\ \int_{\Gamma_d} u dS = \int_{\Gamma_d} u dS \end{cases} \quad (8)$$

Les données sont générées en utilisant un problème de Dirichlet (7) (pression imposée aux buses), et la perméabilité est identifiée à l'aide d'un problème de Neumann (8) (vitesse moyenne imposée aux buses) qui ne lui est pas strictement équivalent car la vitesse normale n'est en réalité pas uniforme dans l'épaisseur des buses. Ceci nous met d'emblée face à un problème d'identification n'ayant pas forcément de solution au sens de l'égalité, mais uniquement au sens de la minimisation. Cependant, afin d'éviter tout soupçon de crime inverse, le problème servant à générer les données utilise un maillage éléments finis différent (environ 4 fois plus d'éléments, mais pas de façon hiérarchique). Par ailleurs on testera l'effet de l'ajout d'un bruit blanc gaussien sur le vecteur de mesure.

Le vecteur de mesure est de taille 4. Il s'agit des pressions moyennées sur la surface des quatre buses : $u_m = [P_1 \ P_2 \ P_3 \ P_4]^T$. L'opérateur de mesure Π est construit de façon à ce que $u_m = \Pi p$.

Le problème inverse (continu) résolu par la méthode FEMU s'écrit de la façon suivante :

$$\min_{\substack{\theta \in L^2(\Omega) \\ u \in H^1(\Omega) \text{ respecte (8)}}} \frac{1}{2} \|\Pi u - u_m\|^2 \quad (9)$$

En pratique, le champ θ lui-même est discrétisé. Dans un premier temps, on suppose que l'on connaît la géométrie de l'inclusion circulaire visible sur la Figure 1, et que l'on ignore seulement les perméabilités des deux domaines. Dans ce cas, on n'identifie que deux paramètres. Dans un second temps, on identifie une valeur différente de θ par élément.

Dans les expériences numériques de cette section, la résolution s'est avérée être suffisamment stable pour que l'on ait pas besoin de *linesearch* pour la méthode perturbative. La méthode de l'adjoint, quant à elle, a tendance à davantage reposer sur les heuristiques d'optimisation, et on n'a pas pris le risque de la priver de son *linesearch*.

3.2 Cas de l'identification de deux paramètres

Pour ce cas-test simple, le comportement de l'algorithme est similaire avec et sans ajout d'un bruit gaussien synthétique sur la mesure. Pour que la longueur du document reste raisonnable, on présentera ici uniquement les résultats dans le cas sans bruit.

On trace sur la Figure 2a l'évolution de la fonction-coût (9) avec le numéro de l'itération courante pour les deux variantes de la méthode perturbative et la méthode de l'adjoint. Comme le coût CPU d'une itération n'est pas le même suivant les méthodes, on trace sur la Figure 2b les mêmes courbes en fonction du nombre d'appels au solveur éléments finis (pour le problème direct ou adjoint, y-compris au cours du *linesearch*). Enfin, on trace sur la Figure 2c les mêmes courbes en fonction du temps d'exécution du programme implémenté en Python sur une machine dotée d'un processeur Intel i7@3GHz. Notons que ce dernier graphe a l'avantage de prendre en compte toutes les opérations de chaque méthode, mais il a l'inconvénient d'être tributaire des contingences de l'implémentation, du processeur et même des processus masqués en cours au moment de l'exécution.

En terme de nombre d'itérations nécessaire à la convergence, la méthode perturbative complète est la plus performante. La méthode avec mise à jour de Broyden est légèrement moins rapide en raison de l'approximation sur la Jacobienne. Enfin, la méthode de l'adjoint est la plus lente car elle ne construit la Hessienne que de façon indirecte via BFGS. En terme de nombre d'appels au solveur, la méthode de l'adjoint est encore plus désavantagée à cause de son *linesearch*. Ce résultat confirme la supériorité de l'approche perturbative dans ce cas. La méthode avec mise à jour de Broyden est celle qui fait le moins appel au solveur. En ce qui concerne le temps total d'exécution, il semble ici être quasiment exclusivement piloté par les appels au solveur, ce qui fait que la méthode avec mise à jour de Broyden

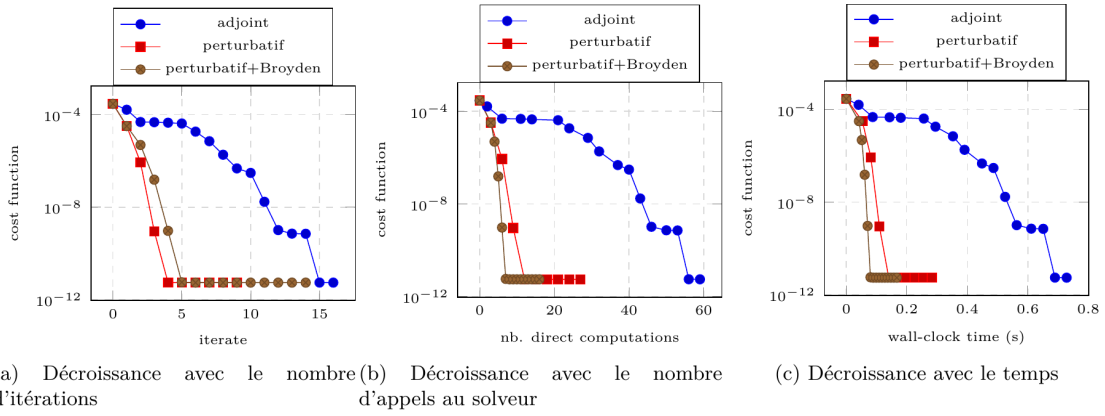


FIGURE 2 – Décroissance de la fonction-coût pour l'identification de deux paramètres sans bruit ajouté

est la plus rapide.

Dans le cadre d'un problème inverse, il est indispensable de ne pas se focaliser sur la décroissance de la fonction coût, mais d'aussi examiner l'erreur (distance à la vraie solution) quand elle est disponible. Pour s'assurer que les trois algorithmes convergent vers la même solution, on affiche sur le Tableau 1 les valeurs trouvées pour les paramètres de perméabilité et la valeur finale de la fonction coût.

TABLE 1 – Solution à convergence pour l'identification de deux paramètres sans bruit ajouté

	perturb.	perturb.+Broyden	adjoint+BFGS	référence
θ_1 (bord)	0.99192639	0.99192639	0.99192639	1
θ_2 (inclusion)	1.95146743	1.95146748	1.95146743	2
min. fonct. coût	$5.4827995887 \cdot 10^{-12}$	$5.4827995883 \cdot 10^{-12}$	$5.4827995862 \cdot 10^{-12}$	0

On observe que, si l'identification de $\{\theta_1, \theta_2\}$ n'est pas parfaite, aucun des trois algorithmes n'aboutit à une meilleure solution que les autres. Quelques itérations en plus ou en moins sur l'un des algorithmes pourraient facilement faire varier le 8ème chiffre après la virgule, qui n'est donc pas numériquement significatif. Il est évidemment encore moins physiquement significatif.

3.3 Cas de l'identification de champ

Sur le maillage utilisé, identifier un champ constant par élément revient à identifier 1436 paramètres scalaires. C'est un cas d'école dans lequel l'approche perturbative est extrêmement coûteuse en temps de calcul, et la méthode de l'adjoint est à privilégier si possible. Dans notre cas, le modèle direct est suffisamment simple pour qu'il soit possible, en stockant les matrices élémentaires, d'effectuer les 1437 calculs directs de l'équation (5) en un temps raisonnable. Un tel artifice d'implémentation est impensable dans le cadre d'un modèle de complexité industrielle, d'autant plus si le calcul est fait via un code éléments finis fermé. Mais dans le cadre de cette étude, il nous permettra de confronter les trois mêmes méthodes que précédemment.

Tenter d'identifier correctement 1436 paramètres en utilisant uniquement 4 mesures est impossible (on le vérifie numériquement). C'est pourquoi deux choses se sont avérées indispensables :

- Supposer que l'on dispose d'une mesure de champ complète de la pression. En pratique, celle-ci est obtenue par interpolation à partir de la solution synthétique complète. Dans ce cas, l'opérateur d'observation Π est l'identité.
- Régulariser les gradients du champ cherché, via un opérateur de gradient discret G calculé par éléments finis sur une triangulation de Delaunay des points de Gauss. La fonction-coût du problème (9) se voit rajouter le terme $\frac{\mu}{2} \|Gk\|^2$, où $k \in \mathbb{R}^{1436}$ est le vecteur paramétrique, et $\mu = 10^{-6}$ est un paramètre de régularisation dont on n'étudiera pas l'impact.

La Figure 3 présente, pour le cas sans bruit de mesure, les courbes de convergence des trois méthodes

étudiées en fonction du nombre d'itérations, du nombre d'appels au solveur éléments finis et enfin du temps d'exécution du programme.

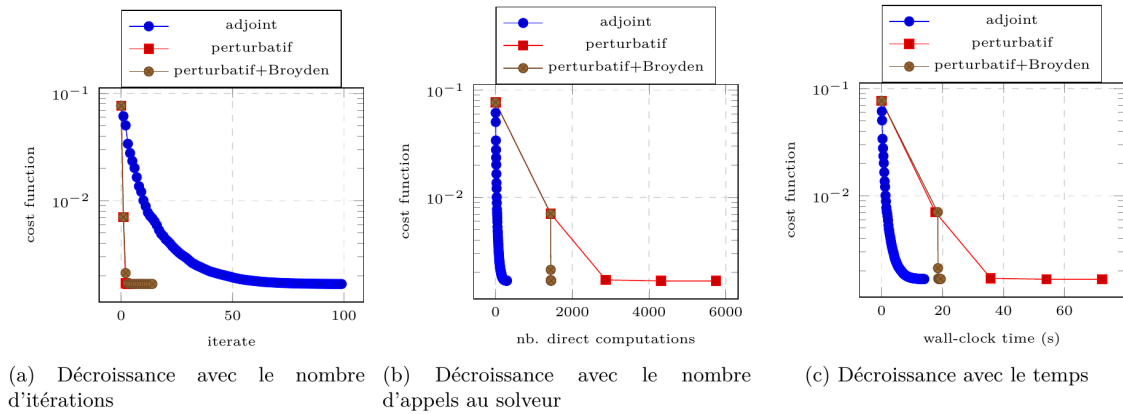


FIGURE 3 – Décroissance de la fonction-coût pour l'identification de champ sans bruit ajouté

Les trois méthodes conduisent à un niveau de stagnation de la fonction coût quasi-identique. En terme de nombre d'itérations pour atteindre la convergence, on observe des résultats similaires à ceux de la section 3.2, c'est à dire que la méthode de l'adjoint est la plus lente tandis que la méthode perturbative est la plus rapide. Le fait de remplacer le re-calcul perturbatif par une mise à jour de Broyden dégrade légèrement la vitesse de convergence en terme de nombre d'itérations, mais pas davantage que sur l'exemple de la section 3.2 alors que le nombre de paramètres est bien supérieur. En revanche, en ce qui concerne le nombre d'appels au solveur éléments finis, la méthode perturbative est de loin la plus gourmande. L'approche de Broyden permet ici d'économiser un nombre très important d'appels au solveur à partir de la deuxième itération (pour la première itération, une analyse de sensibilité complète est faite). Pour ce qui est du temps d'exécution total, on s'aperçoit ici qu'il n'est pas piloté exclusivement par les appels au solveur. En effet, la résolution d'un système linéaire impliquant une Hessienne pleine de taille 1437 est probablement plus coûteuse en temps de calcul que la résolution du problème direct. La méthode de l'adjoint est sans surprise la plus adaptée pour ce problème, mais la mise à jour de Broyden des sensibilités permet un gain en temps de calcul très intéressant.

On compare sur le Tableau 2 les valeurs minimales atteintes par la fonction-coût pour différents niveaux de bruit.

TABLE 2 – Valeurs atteintes à convergence par la fonction coût en fonction du niveau de bruit et de la méthode de minimisation

bruit	perturb.	perturb.+Broyden	adjoint+BFGS
0	0.0016629	0.0016753	0.0016674
1%	0.00212998	0.00214381	0.00213379
10%	0.0445277	0.0446184	0.0445361

Ici, et contrairement à la section 3.2, on observe des différences numériquement significatives entre les fonctions coûts à convergence pour les différentes méthodes. Pour tous les niveaux de bruit, la méthode perturbative est la meilleure, suivie par la méthode de l'adjoint. La méthode avec mise à jour de Broyden conduit systématiquement aux moins bons résultats. Ces écarts sont très resserrés, et dans beaucoup de circonstances ils seraient probablement considérés comme physiquement non-significatifs. Cependant, leur existence même permet d'illustrer le fait que les trois méthodes convergent vers des minima locaux légèrement différents.

Une autre façon d'illustrer cette différence est de tracer les cartographies finales de perméabilité obtenues par les trois méthodes. C'est fait sur la Figure 4 pour le cas à 10% de bruit (le seul pour lequel la différence est visible). On observe que le champ obtenu avec l'algorithme perturbatif et la mise à jour de Broyden se distingue des deux autres champs par sa forme moins régulière. Ceci semble attester d'une convergence légèrement moins bonne de l'algorithme.

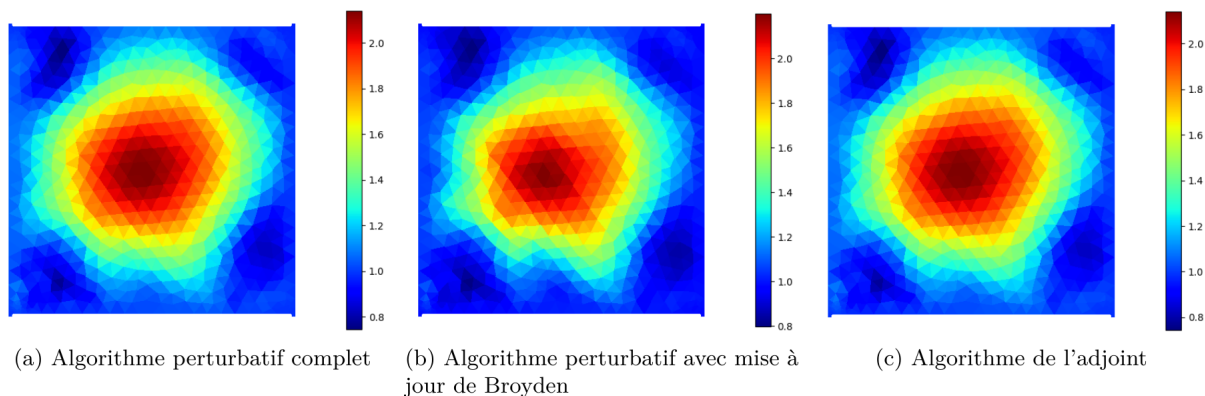


FIGURE 4 – Cartographie du champ identifié à convergence dans le cas à 10% de bruit

4 Cas-test à données expérimentales

La section précédente a permis de montrer le potentiel de cette approche par mise à jour de Broyden pour s'affranchir (partiellement) de la dépendance au nombre de paramètres à identifier. La différence entre la solution avec mise à jour ou avec re-calculation complète est systématiquement nulle ou marginale.

Pourtant, le lecteur peut encore se demander, et c'est légitime, si une telle approche est bien adaptable à un cas réel. Peut-on mettre en œuvre cet algorithme de façon non-intrusive via un solveur industriel ? Peut-on s'en servir pour traiter des données expérimentales ? Oui aux deux. Rien ne s'y oppose en théorie (voir la section 2). Et pour prouver que rien ne s'y oppose en pratique non-plus, on s'intéressera, dans la contribution orale, à l'identification des paramètres d'une loi élasto-plastique à partir d'un essai de traction sur une éprouvette percée en aluminium. La mesure est faite par corrélation d'images via le logiciel d'EikoSim, et les calculs directs sont faits via le solveur Abaqus.

Quant à savoir si les résultats seront aussi encourageants que ceux de la section précédente, la réponse sera apportée lors de la contribution orale.

Références

- [1] Jérôme Molimard, Rodolphe Le Riche, Alain Vautrin, and Jung-Ryul Lee. Identification of the four orthotropic plate stiffnesses using a single open-hole tensile test. *Experimental mechanics*, 45(5) :404–411, 2005.
- [2] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92) :577–593, 1965.
- [3] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68) :7, 1999.
- [4] Charles G Broyden. On the discovery of the “good Broyden” method. *Mathematical programming*, 87(2) :209–213, 2000.
- [5] Andreas Griewank. Broyden updating, the good and the bad. *Optimization Stories, Documenta Mathematica. Extra Volume : Optimization Stories*, pages 301–315, 2012.
- [6] Stefan Finsterle and Michael B Kowalsky. A truncated Levenberg–Marquardt algorithm for the calibration of highly parameterized nonlinear models. *Computers & geosciences*, 37(6) :731–738, 2011.
- [7] Henri P Gavin. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering Duke University August*, 3 :1–23, 2019.
- [8] Mark K Transtrum and James P Sethna. Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization. *arXiv preprint arXiv :1201.5885*, 2012.
- [9] Alexandra Smirnova, Benjamin Sirb, and Gerardo Chowell. On stable parameter estimation and forecasting in epidemiology by the Levenberg–Marquardt algorithm with broyden's rank-one updates for the jacobian operator. *Bulletin of mathematical biology*, 81(10) :4210–4232, 2019.
- [10] Christophe Geuzaine and Jean-François Remacle. Gmsh : A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11) :1309–1331, 2009.