BE - Problèmes inverses: inversion de la forme d'onde

Youssef Diouane youssef.diouane@isae-supaero.fr
ISAE - SUPAERO
Toulouse, France.

Le but de ce BE est de mettre en oeuvre différentes techniques pour la résolution des problèmes inverses. Le travail est à réaliser en binôme. Un compte-rendu est à déposer sur le LMS avant la date limite : le vendredi 17/12/2021.

Un intérêt majeur en géophysique est la détermination de certaines propriétés physiques des sous-sols, afin d'évaluer leurs richesses en substance minérale, gazeuse ou pétrolière. En effet, la connaissance de la masse volumique ou de la vitesse de propagation au sein d'un champ géophysique 3D, nous permet d'avoir une idée sur la structure de ce sous-sol. Ce BE se limite à l'étude d'un cas simplifié en 1D. On va établir une loi de comportement de ce domaine de la manière suivante : excité par une source, le domaine se comporte comme un milieu propagatif. Sa réponse se manifeste sous forme de propagation d'une onde de pression p se propageant avec une vitesse c, et d'après les lois de la mécanique, les équations aux dérivées partielles de propagation en domaine temporel sont :

$$\begin{aligned} \Delta p(z,t|z_s) - \frac{1}{[c(z)]^2}\frac{\partial^2}{\partial t^2}p(z,t|z_s) &= s(z,t|z_s) \quad \text{dans }]0,L[\times]0,T[\\ p(z,0|z_s) = \frac{\partial p}{\partial t}(z,0|z_s) &= 0 \quad \text{dans }]0,L[\times\{0\} \\ p(0,t) &= 0 \quad \text{dans } \{0\}\times]0,T[\end{aligned} \tag{1}$$

où $p(z,t|z_s)$ représente le champ de pression à la position z à l'instant t d'une source placée à z_s, $c(z)$ est la vitesse de propagation et $s(z,t|z_s)$ est la fonction source. Formellement, l'équation (1) peut s'écrire sous la forme :

$$p = F(c,s), \tag{2}$$

où F représente un opérateur non-linéare donné. En utilisant la fonction de Green $G(r,t|z',0)$ associée à l'équation (1), la solution $p(z,t|z_s)$, peut s'écrire sous la forme :

$$p(z,t|z_s) = \int_\Omega G(z,t|z',0) * s(z',t|z_s)dz', \tag{3}$$

où $*$ représente l'operateur de convolution en temps. L'équation (2) (ou bien (3)) représente la solution du problème direct. L'équation (2) a l'avantage de permettre l'utilisation de la solution du problème direct sans expliciter l'outil numérique (ici l'algorithme différences finies). Quant à l'equation (3), elle permet de démontrer facilement les différentes formules

nécessaires (gradient, Jacobien, Hessien, ...). Pour ce BE, on suppose connaître la fonction source (e.g., de type ondelette de Ricker) :

$$s(z,t|z_s) \quad = \quad \delta_0(z-z_s)(1-\pi^2 f^2(t-\tau)^2)\exp\left(-\pi^2 f^2(t-\tau)^2\right),$$

où δ_0 est la fonction Dirac en 0, f est la fréquence de pic et τ un décalage temporel. On fixe dans un premier instant, la fréquence de pic à $f = 5$Hz. Afin de simuler numériquement l'équation (2), on se donne un maillage uniforme de $[0, L]$, de pas dx (i.e., $dx = L/(n_z + 1)$, avec $n_z \in \mathbb{N}$), un maillage uniforme de $[0, T]$ de pas dt (i.e., $dt = T/(n_t + 1)$, avec $n_t \in N$). On notera $z_j = jdx$, $j = 0, \ldots, n_z + 1$ les points du maillage spatial et t_k , $k = 0, \ldots, n_t + 1$ les instants du maillage temporel. On approche le système par différences finies. Le fichier MatLab `ForwardProblem.m` vous fournit cette implémentation du problème direct :

```
[p_calc,p_all] = ForwardProblem(n_z,n_t,s,c,zs,dx,dt,p0,p1,print)
```

où n_z est le nombre de points en espace, n_t le nombre de pas de temps, $s \in \mathbb{R}^{n_t}$ le termee source, $c \in \mathbb{R}^{n_z}$ le vecteur vitesse, les vecteurs $p_0 \in \mathbb{R}^{n_z}$ et $p_1 \in \mathbb{R}^{n_z}$ représentent respectivement les pressions aux instants initiaux t_0 et t_1. L'argument `print` est un boolean qui permet d'afficher le champ de pression total `p_all`. le terme source s est calculé via l'appel à au fichier MatLab `SourceTerm.m` :

```
[s]=SourceTerm(n_t,f,dt,print)
```

Le schéma d'inversion est inspiré de la méthode adjointe (vue en cours). On définit donc la fonction résidu par :

$$\delta p(z_g,t|z_s) \quad = \quad p_{\text{obs}}(z_g,t|z_s) - p_{\text{calc}}(z_g,t|z_s),$$

où z_g est un vecteur indiquant la position du récepteur, $p_{\text{obs}}(z_g,t|z_s)$ et $p_{\text{calc}}(z_g,t|z_s)$ sont, respectivement, les données observées et calculees. Notre objectif est la minimisation de cette fonction objectif représentative de la fonction residu (4) en modifiant le profil de vitesse c à chaque itération. La fonction objectif est de la forme suivante :

$$E(c) \quad = \quad \frac{1}{2}\sum_s \sum_g \int (\delta p(z_g,t|z_s))^2 dt. \tag{4}$$

Vu la taille du problème, une méthode d'optimisation de gradient local est utilisée pour la minimisation de la fonction objectif (4). Le gradient de la fonction objectif par rapport à une perturbation de profil de vitesse c est donné par

$$g(z) \quad = \quad \frac{1}{c(z)^3}\sum_s \sum_g \int \dot{p}_{\text{calc}}(z,t|z_s) p_{\text{back}}(z,t|z_g,z_s) dt. \tag{5}$$

où $\dot{p}$ représente la dérivée en temps de p, et $p_{\text{back}}(z,t|z_s)$ la rétro-propagation de la fonction résidu, i.e.,

$$p_{\text{back}}(z,t|z',z_s) = G(z,-t|z_g,0) * \delta p(z_g,t|z_s).$$

On suppose que le recépteur et la source se situent au même endroit (i.e., $z_g = z_s$). Les données observées représentent des mesures in situ. Dans le cadre de ce BE, pour simplifier

la tâche, on construit ces données en se basant sur un profil de vitesse exact c_{true} qu'on cherche à inverser (éventuellement en ajoutant un bruit). Un tel scénario reste cohérent avec le cas réaliste (i.e., mesures in situ) ; on est dans le cadre du *crime inverse*. Voici un exemple de code Matlab pour générer des données observées :

```
n_t = 1500; dx = 5; dt = 0.001; n_z = 301;print=1;
c_true = 2000*ones(n_z,1);c_true(51:80) = 1600; c_true(121:150) = 2500;
z_s = 5; % The source position
f = 5; % The peak frequency
s = SourceTerm(nt,f,dt,print);
p0 = zeros(nz,1);p1 = p0; p0(zs) = s(1); p1(zs) = s(2);
[p_true,p_all] = ForwardProblem(n_z,n_t,s,c_true,zs,dx,dt,p0,p1,print);
p_obs=p_true; % Possibly we can add noise
```

Pour simplifier, on garde les mêmes notations qu'avant pour la fonction objectif (et son gradient) dans l'espace discrétisé : $E : \mathbb{R}^{n_z} \to \mathbb{R}$ et $g : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$. Le fichier MatLab `CostFun_FWI.m` vous fournit l'implémentation de la fonction objectif et de son gradient :

```
[varargout]=CostFun_FWI(c,varargin)
```

Pour assurer la stabilité numérique du schéma, on pourra imposer sur le profil de vitesse c au cours de l'optimisation la condition suivante : $\min(c_{\text{true}}) \leq c \leq \max(c_{\text{true}})$.

Question 1 *En utilisant un profil de vitesse c_{true}, simuler vos données observées p_{obs}. Tracer la courbe de la fonction $\alpha \to E(c_0 + \alpha(c_1 - c_0))$ avec $\alpha \in [-2, 2]$ où c_0 et c_1 sont deux vitesses choisis (on pourra choisir par exemple $c_0 = c_{true}$ et $c_1 = c_0 + \delta c$). Analyser l'impact des différents paramètres (e.g., c_{true}, fréquences de pic f, le bruit, ...). Que peut-on en conclure ?*

Un algorithme d'optimisation est déterminé par les stratégies de choix des directions de descente successives, puis par le pas qui sera effectué dans la direction choisie. L'idée est de remplacer la fonction f par un modèle local plus simple, dont la minimisation nous donnera une direction de descente. Soit c_k le profil de vitesse à l'itéré courant, on remplace E au voisinage de c_k par son développement de Taylor au premier ordre :

$$E(c_k + d) \sim E(c_k) + g_k^T d,$$

où g_k représente le gradient de la fonction objectif en c_k (i.e., $g_k = \nabla_{c_k} E$). Pour minimiser E, une direction de descente possible est donc la direction de plus profonde descente définie par :

$$d_k = -g_k.$$

Le choix de la direction de plus profonde descente définit une famille d'algorithmes appelés algorithmes de descente de gradient, dont le schéma est le suivant :

Algorithme de descente	
1.	Initialisation : c_0 une première approximation de la solution recherchée, $\epsilon > 0$ précision demandée.
2.	Étape courante : c_k connu.
3.	Direction de descente : $d_k = -g_k$.
4.	Recherche linéaire : Choisir un pas s_k à faire dans cette direction, tel que : $E(c_{k+1}) \leq E(c_k)$.
5.	Mise à jour : $c_{k+1} = c_k + s_k d_k$.
6.	Incrémenter l'indice k.
7.	Test d'arrêt.

Dans un premier temps, on choisit la taille de pas $s_k = 10^2/\|g_k\|$.

Question 2 *En utilisant l'algorithme de descente, résoudre le problème de minimisation* $\min_{c\in\mathbb{R}^N} E(c)$. *Commenter les résultats obtenus pour différents points de départ* c_0. *Que se passe-t-il si l'on change la taille de pas* s_k *? Quelles sont les nouvelles limitations de la méthode ? Proposer une solution.*

Question 3 *Inverser différents profils de vitesse. Le problème est-il bien posé dans ce cas ? Que peut-on en conclure ?*

Question 4 *On perturbe maintenant* p_{obs} *par un bruit aléatoire Gaussien de variance* η *et on reprend les mêmes questions qu'avant. Commenter les résultats obtenus.*

Question 5 *On suppose qu'on possède également d'un a priori* $c_{apriori}$ *du modèle de référence* c_{true}. *Dans ce cadre, Que peut-on proposer ? Implémenter cette proposition.*

Indication : Pour obtenir $c_{apriori}$ *on pourra prendre* c_{true} *qu'on lisse (voir la fonction* `smooth` *de MatLab).*

Disposer d'un apriori proche du profil exact de la vitesse n'est pas réaliste. En pratique, on construit l'apriori en suivant une approche multi-échelle. L'idée est de démarrer la procédure d'inversion par une fréquence de pic faible à partir d'un profil de vitesse quelconque (en pratique homogène). La solution obtenue à ce stade servira d'apriori pour une nouvelle procédure d'inversion (mais avec une fréquence de pic plus élevée). On répète la même procédure pendant plusieurs itérations. On obtient donc le schéma suivant :

Algorithme de descente multi-échelle	
1.	Initialisation : k=0, c_0^{opt} une première approximation de la solution recherchée, f_0 une première fréquence de pic (faible) et N_f le nombre de niveaux.
2.	Étape courante : f_k et c_k^{opt} connus.
3.	Calculer c_{k+1}^{opt} : appeler "Algorithme de descente" avec $c_0 = c_k^{\text{opt}}$ et $f = f_k$.
4.	Choisir $f_{k+1} > f_k$.
5.	Incrémenter l'indice k.
6.	Test d'arrêt ($k = N_f$).

Question 6 *Tester l'approche multi-échelle. Commenter. On pourra prendre* $N_f = 3$ *avec une fréquence de pic maximale de* $10Hz$.