

GEO1302 – Modélisation et inversion en géophysique

6 - Inversion non linéaire

Bernard Giroux
(bernard.giroux@ete.inrs.ca)

Institut national de la recherche scientifique
Centre Eau Terre Environnement

Version 1.2.0
Hiver 2020

Régression non linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

Régression non linéaire

Régression non linéaire

Méthode de Newton

Méthode de

Gauss-Newton

Calcul de la jacobienne

Méthode de

Levenberg-Marquardt

Inversion non linéaire

- Les problèmes non linéaires n'obéissent pas aux principes de superposition et de mise à l'échelle;
- Le **modèle direct** $G(\mathbf{m})$ *n'est plus un système linéaire* d'équations algébriques $\mathbf{Gm} = \mathbf{d}$.
- Il n'existe pas de théorie générale donnant une solution pour les problèmes inverses non linéaires;
- Il existe néanmoins plusieurs approches, certaines basées sur des approximations linéaires, qui sont souvent applicables.

- L'idée de la méthode de Newton est d'utiliser une information sur la forme de l'erreur de prédiction $E(\mathbf{m})$ au voisinage d'une **solution d'essai** $\mathbf{m}^{(p)}$ pour trouver de façon itérative une meilleure solution $\mathbf{m}^{(p+1)}$.
- Les dérivées de $E(\mathbf{m})$ nous renseignent sur sa forme.
- À partir de l'expansion en série de Taylor de $E(\mathbf{m})$ au voisinage de $\mathbf{m}^{(p)}$ et en gardant les trois 1^{er} termes, on a

$$E(\mathbf{m}) \approx E(\mathbf{m}^{(p)}) + \sum_{i=0}^{M-1} b_i (m_i - m_i^{(p)}) + \frac{1}{2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} H_{ij} (m_i - m_i^{(p)}) (m_j - m_j^{(p)}) \quad (1)$$

où $b_i = \left. \frac{\partial E}{\partial m_i} \right|_{\mathbf{m}^{(p)}}$ et $H_{ij} = \left. \frac{\partial^2 E}{\partial m_i \partial m_j} \right|_{\mathbf{m}^{(p)}}$.

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Sous forme matricielle, l'équation précédente s'écrit

$$E(\mathbf{m}^{(p)}) + \nabla E(\mathbf{m}^{(p)})^T \Delta \mathbf{m} + \frac{1}{2} \Delta \mathbf{m}^T \mathbf{H}(E(\mathbf{m}^{(p)})) \Delta \mathbf{m} \quad (2)$$

où

$$\Delta \mathbf{m} = \mathbf{m} - \mathbf{m}^{(p)} \quad (3)$$

et $\nabla E \equiv \nabla E(\mathbf{m}^{(p)})$ est le **gradient**, et $\mathbf{H} \equiv \mathbf{H}(E(\mathbf{m}^{(p)}))$ est la **hessienne**, avec :

$$\nabla E = \left[\begin{array}{c} \frac{\partial E}{\partial m_0} \\ \vdots \\ \frac{\partial E}{\partial m_{M-1}} \end{array} \right]_{\mathbf{m}^{(p)}} \quad \mathbf{H} = \left[\begin{array}{ccc} \frac{\partial^2 E}{\partial m_0^2} & \cdots & \frac{\partial^2 E}{\partial m_0 \partial m_{M-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial m_{M-1} \partial m_0} & \cdots & \frac{\partial^2 E}{\partial m_{M-1}^2} \end{array} \right]_{\mathbf{m}^{(p)}} \quad (4)$$

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Le minimum de $E(\mathbf{m})$ peut maintenant être trouvé en dérivant à nouveau et en égalant à zéro :

$$\frac{\partial E(\mathbf{m})}{\partial m_q} = 0 = b_q + \sum_{j=0}^{M-1} H_{qj} (m_j - m_j^{(p)}). \quad (5)$$

- Sous forme matricielle, cela donne

$$\Delta \mathbf{m} = -\mathbf{H}^{-1} \nabla E \quad (6)$$

- Notez que pour le cas linéaire $\mathbf{Gm} = \mathbf{d}$, on peut trouver que $\nabla E = -2\mathbf{G}^T (\mathbf{d} - \mathbf{Gm}^{(p)})$ et que $\mathbf{H} = 2\mathbf{G}^T \mathbf{G}$, ce qui nous amène à $\mathbf{m} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d}$, soit la solution des moindres-carrés.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- L'algorithme de Newton est le suivant

Algorithm 1 Méthode de Newton

```

1:  $p \leftarrow 0$ 
2: while  $\nabla E \neq 0$ 
3:   Calculer  $\nabla E$  et  $\mathbf{H}$ 
4:   Résoudre  $\mathbf{H} \Delta \mathbf{m} = -\nabla E$ 
5:    $\mathbf{m}^{(p+1)} \leftarrow \mathbf{m}^{(p)} + \Delta \mathbf{m}$ 
6:    $p \leftarrow p + 1$ 

```

- Cet algorithme peut être très efficace, mais la convergence n'est pas garantie.
- L'algorithme peut aussi converger vers un minimum local si la solution initiale est trop loin du minimum global.

Régression non linéaire

Méthode de Newton

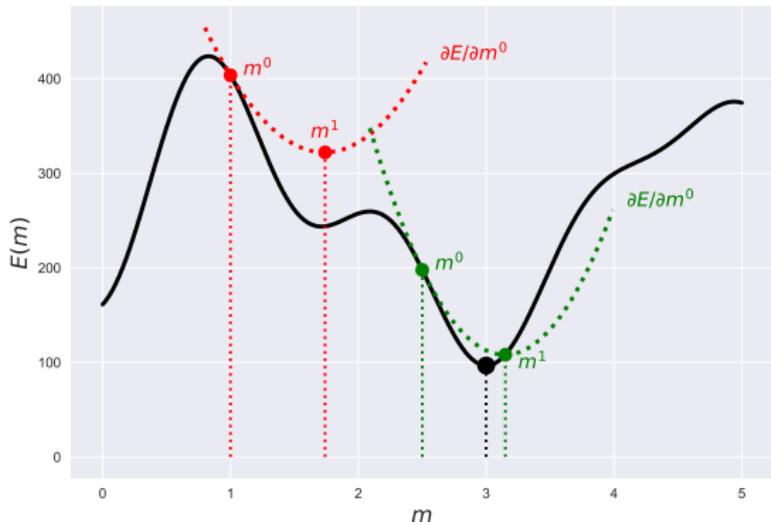
Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Exemple de cas où une solution initiale converge vers un minimum local (rouge), alors qu'une solution initiale plus proche du minimum global converge mieux (vert).



Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- En pratique, la convergence $\nabla E = 0$ n'est jamais atteinte exactement;
- En général, on considère qu'il y a convergence lorsqu'une des conditions suivante est atteinte :
 - $\max |\nabla E| < \epsilon_1$
 - $\max |\Delta m_i / m_i| < \epsilon_2$
 - $E / (N - M + 1) < \epsilon_3$
 - $|E^{(p)} - E^{(p-1)}| < \epsilon_4 E^{(p)}$ et $E^{(p)} < E^{(p-1)}$

sinon les itérations s'arrêtent après un nombre maximal prédéfini.

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- La méthode de Newton n'est pas applicable quand
 - il n'y a pas de solution exacte à $G(\mathbf{m}) = \mathbf{d}$,
 - plusieurs solutions existent.
- La méthode de Gauss-Newton peut être vue comme une modification permettant de minimiser les moindres-carrés non linéaires.
- On cherche à minimiser la norme des résidus pondérés

$$E = \sum_{i=0}^{N-1} \left(\frac{d_i - \hat{d}_i}{\sigma_i} \right)^2, \quad (7)$$

où $\hat{d}_i = G_i(\mathbf{m})$ et σ_i est l'écart-type de la i^e mesure.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- On définit la fonction scalaire

$$E_i(\mathbf{m}) = \frac{d_i - \hat{d}_i}{\sigma_i} \quad i = 0, 1, \dots, N - 1 \quad (8)$$

et la fonction vecteur

$$\mathbf{E}(\mathbf{m}) = \begin{bmatrix} E_0(\mathbf{m}) \\ \vdots \\ E_{N-1}(\mathbf{m}) \end{bmatrix} \quad (9)$$

- Ainsi

$$E = \sum_{i=0}^{N-1} E_i(\mathbf{m})^2 = \|\mathbf{E}(\mathbf{m})\|_2^2 = (\mathbf{d} - \hat{\mathbf{d}})^T \mathbf{W} (\mathbf{d} - \hat{\mathbf{d}}) \quad (10)$$

où \mathbf{W} est une matrice diagonale avec $W_{ii} = 1/\sigma_i^2$.

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Le gradient de E est la somme des gradients des fonctions :

$$\nabla E = \sum_{i=0}^{N-1} \nabla (E_i(\mathbf{m})^2) \quad (11)$$

- Le gradient doit être calculé pour chacun des m_j paramètres, ∇E est donc un vecteur de M éléments.
- Les éléments du gradient sont

$$\nabla E_j = \sum_{i=0}^{N-1} 2E_i(\mathbf{m}) \frac{\partial E_i(\mathbf{m})}{\partial m_j} \quad (12)$$

où $j = 0, 1, \dots, M - 1$.

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Sous forme matricielle, on trouve

$$\nabla E = 2\mathbf{W}^{1/2}\mathbf{J}^T\mathbf{E}(\mathbf{m}) = 2\mathbf{J}^T\mathbf{W}(\mathbf{d} - \hat{\mathbf{d}}) \quad (13)$$

où \mathbf{J} est la matrice jacobienne ($N \times M$) :

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \hat{d}_0}{\partial m_0} & \cdots & \frac{\partial \hat{d}_0}{\partial m_{M-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{d}_{N-1}}{\partial m_0} & \cdots & \frac{\partial \hat{d}_{N-1}}{\partial m_{M-1}} \end{bmatrix} \quad (14)$$

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- D'une façon similaire, on peut exprimer la hessienne de $E(\mathbf{m})$ à partir de ses fonctions :

$$\mathbf{H}(E(\mathbf{m})) = \sum_{i=0}^{N-1} \mathbf{H}(E_i(\mathbf{m})^2) \quad (15)$$

$$= \sum_{i=0}^{N-1} \mathbf{H}^i(\mathbf{m}) \quad (16)$$

où $\mathbf{H}^i(\mathbf{m})$ est la hessienne de $E_i(\mathbf{m})^2$.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Les éléments j, k de $\mathbf{H}^i(\mathbf{m})$ sont

$$\mathbf{H}_{j,k}^i(\mathbf{m}) = \frac{\partial^2 (E_i(\mathbf{m}))}{\partial m_j \partial m_k} \quad (17)$$

$$= \frac{\partial}{\partial m_j} \left(2E_i(\mathbf{m}) \frac{\partial E_i(\mathbf{m})}{\partial m_k} \right) \quad (18)$$

$$= 2 \left(\frac{\partial E_i(\mathbf{m})}{\partial m_j} \frac{\partial E_i(\mathbf{m})}{\partial m_k} + E_i(\mathbf{m}) \frac{\partial^2 E_i(\mathbf{m})}{\partial m_j \partial m_k} \right) \quad (19)$$

- Sous forme matricielle, cela devient

$$\mathbf{H}(E(\mathbf{m})) \equiv \mathbf{H} = 2\mathbf{J}^T \mathbf{W} \mathbf{J} + \mathbf{Q}(\mathbf{m}) \quad (20)$$

où

$$\mathbf{Q}(\mathbf{m}) = 2 \sum_{i=0}^{N-1} E_i(\mathbf{m}) \mathbf{H}(E_i(\mathbf{m})). \quad (21)$$

Régression non linéaire

Méthode de Newton

Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Avec la méthode de Gauss-Newton, on ignore la matrice $\mathbf{Q}(\mathbf{m})$
 - Cela revient à assumer que la fonction est quasi linéaire *au voisinage* de $\mathbf{m}^{(p)}$.
- La matrice hessienne devient

$$\mathbf{H} \approx 2\mathbf{J}^T \mathbf{W} \mathbf{J} \quad (22)$$

- La mise à jour du modèle est obtenue en résolvant

$$\underbrace{\mathbf{J}^T \mathbf{W} \mathbf{J}}_{\mathbf{A}} \underbrace{\Delta \mathbf{m}}_{\mathbf{x}} = \underbrace{\mathbf{J}^T \mathbf{W} (\mathbf{d} - \hat{\mathbf{d}})}_{\mathbf{b}} \quad (23)$$

- Comme pour la méthode de Newton, la méthode de Gauss-Newton peut converger vers un minimum local.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Soit une fonction non linéaire

$$G_i(\mathbf{m}) = \sin(\omega_0 m_0 x_i) + m_0 m_1 \quad (24)$$

- Quel est le minimum de E après 20 itérations pour les données bruitées suivantes, en utilisant $\mathbf{m}^{(0)} = [1.0, 1.0]$ et en assumant que $G(\mathbf{m})$ est quasi-linéaire et que $\mathbf{W} = \mathbf{I}$:

```

N = 40
xmin = 0
xmax = 1.0
dx = (xmax-xmin)/(N-1)
x = dx*np.arange(N)

mt = [1.21, 1.54] # modèle vrai

w0 = 20
dtrue = np.sin(w0*mt[0]*x) + mt[0]*mt[1] # données propres
sd = 0.4
dobs = dtrue + sd*np.random.randn(N) # données bruitées
  
```

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- La méthode de Gauss-Newton requière le calcul de dérivées partielles pour construire la matrice jacobienne.
- Dans certains cas, les expressions analytiques de ces dérivées existent et il est possible de les calculer directement.
- Lorsque les expressions analytiques ne sont pas disponibles, on peut les calculer par différences finies :

$$\frac{\partial \hat{d}_i}{\partial m_j} = \frac{\partial G_i(\mathbf{m})}{\partial m_j} \approx \frac{G_i(\mathbf{m} + \Delta \mathbf{m}_j) - G_i(\mathbf{m})}{\Delta m_j} \quad (25)$$

où $\Delta \mathbf{m}_j$ est le vecteur \mathbf{m} perturbé de Δm_j uniquement à j ;

- cette façon de faire peut s'avérer coûteuse en calcul car $G(\mathbf{m})$ doit être évaluée à plusieurs reprises.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Le choix de la perturbation Δm_j peut s'avérer délicat lorsque les paramètres varient sur plusieurs ordres de grandeur ;
 - dans un tel cas, il est préférable que la perturbation soit une fraction de la valeur du paramètre plutôt qu'une valeur absolue.
- Lorsque les paramètres du modèle représentent des propriétés physiques différentes variant sur des ordres de grandeur différents, il peut être nécessaire de normaliser les termes de la jacobienne pour éviter de donner des poids trop différents.
- Il est également important que l'estimation des dérivées soit stable au voisinage de la perturbation.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Il peut arriver que la matrice $\mathbf{J}^T \mathbf{J}$ soit singulière ;
 - dans cette situation, la méthode de Gauss-Newton échoue.
- Avec la méthode de Levenberg-Marquardt, la mise à jour du modèle est

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta \mathbf{m} = \nabla E \quad (26)$$

- Le terme $\lambda \mathbf{I}$ permet de mieux conditionner le système.
- Avec cette méthode, le problème du choix de la valeur optimale de λ apparaît ;
 - si λ est trop élevé, la convergence est ralentie ;
 - si λ est trop faible, le problème de la singularité peut survenir à nouveau.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Pour des problèmes de taille modérée et lorsque $\mathbf{W} = \mathbf{I}$, une façon de déterminer λ passe par la SVD;
- La SVD de \mathbf{J} est

$$\mathbf{J} = \mathbf{USV}^T \quad (27)$$

où \mathbf{U} contient les vecteurs de base de l'espace des données, \mathbf{V} contient les vecteurs de base de l'espace des paramètres, et \mathbf{S} contient les valeurs singulières $[s_0, s_1, \dots, s_{M-1}]$ de \mathbf{J} .

- En insérant cette expression dans l'équation (26), on trouve

$$\Delta \mathbf{m} = (\mathbf{VS}^2\mathbf{V}^T + \beta^2\mathbf{I})^{-1} \mathbf{VSU}^T (\mathbf{d} - \hat{\mathbf{d}}) \quad (28)$$

où $\beta^2 = \lambda$.

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- En ajoutant le facteur β^2 sur la diagonale de \mathbf{S}^2 , on trouve

$$(\mathbf{V}\mathbf{S}^2\mathbf{V}^T + \beta^2\mathbf{I}) = (\mathbf{V}\text{diag}(\mathbf{s}^2)\mathbf{V}^T + \beta^2\mathbf{I}) \quad (29)$$

$$= \mathbf{V}\text{diag}(\mathbf{s}^2 + \beta^2)\mathbf{V}^T \quad (30)$$

où \mathbf{s} est le vecteur des valeurs singulières.

- L'inverse de cette expression est

$$(\mathbf{V}\text{diag}(\mathbf{s}^2 + \beta^2)\mathbf{V}^T)^{-1} = \mathbf{V}\text{diag}\left(\frac{1}{\mathbf{s}^2 + \beta^2}\right)\mathbf{V}^T \quad (31)$$

- En insérant cette dernière expression dans (28), on trouve

$$\Delta\mathbf{m} = \mathbf{V}\text{diag}\left(\frac{1}{\mathbf{s}^2 + \beta^2}\right)\mathbf{V}^T \mathbf{V}\mathbf{S}\mathbf{U}^T (\mathbf{d} - \hat{\mathbf{d}}) \quad (32)$$

$$= \mathbf{V}\text{diag}\left(\frac{\mathbf{s}}{\mathbf{s}^2 + \beta^2}\right)\mathbf{U}^T (\mathbf{d} - \hat{\mathbf{d}}) \quad (33)$$

Régression non
linéaire

Méthode de Newton

Méthode de

Gauss-Newton

Calcul de la jacobienne

Méthode de

Levenberg-Marquardt

Inversion non
linéaire

- L'estimation de β est dynamique et se fait en fonction de la convergence;
- Une façon de déterminer β est d'utiliser

$$\beta = s_l \Delta E^{1/l} \quad (34)$$

où $l = 1, \dots, M$ et

$$\Delta E^{(p)} = \frac{E^{(p-1)} - E^{(p)}}{E^{(p-1)}}. \quad (35)$$

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- La procédure est la suivante :

```

1 :  $\mathbf{m} \leftarrow \mathbf{m}^{(0)}$ 
2 : while  $p < p_{max}$ 
3 :   Calculer  $\hat{\mathbf{d}}$  et  $E^{(p)}$ 
4 :   if convergence atteinte
5 :     return
6 :   Calculer  $\mathbf{J}$  et SVD( $\mathbf{J}$ )
7 :    $l \leftarrow 1, k \leftarrow 1$ 
8 :   while  $l < M$ 
9 :      $\beta \leftarrow s_l \Delta E^{1/l}$ 
10 :    Calculer  $\Delta \mathbf{m}, \hat{\mathbf{d}}$  et  $E^{(p)}$ 
11 :    if  $E^{(p)} > E^{(p-1)}$ 
12 :       $l \leftarrow l + 1, k \leftarrow k + 1$ 
13 :      if  $k == M$ 
14 :        return
15 :      else
16 :         $\mathbf{m} \leftarrow \mathbf{m} + \Delta \mathbf{m}$ 
17 :        Calculer  $\Delta E$ 
18 :         $l \leftarrow M$ 
19 :        if convergence atteinte
20 :          return
21 :         $E^{(p-1)} \leftarrow E^{(p)}$ 
22 :         $p \leftarrow p + 1$ 
  
```

Méth. de Levenberg-Marquardt – Exercice

Régression non linéaire

Méthode de Newton

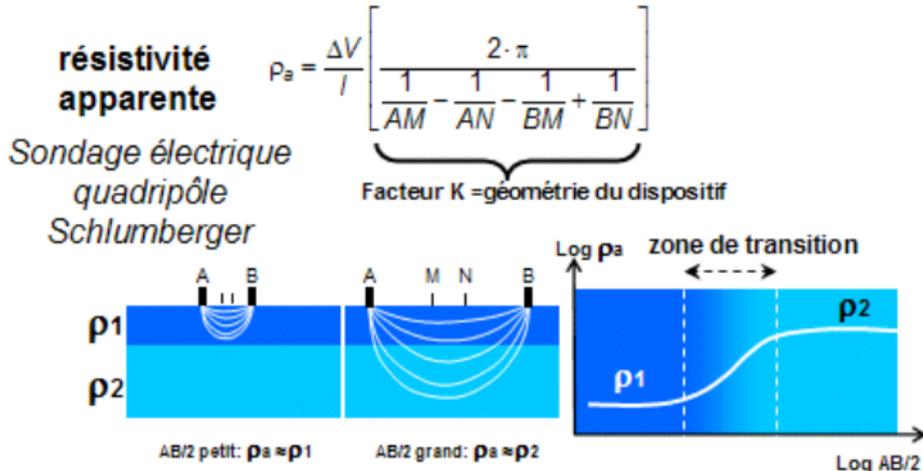
Méthode de Gauss-Newton

Calcul de la jacobienne

Méthode de Levenberg-Marquardt

Inversion non linéaire

- Utilisez la méthode de Levenberg-Marquardt pour estimer un modèle géoélectrique 1D à partir de mesures de sondages Schlumberger.



Source : Université de Lorraine

Méth. de Levenberg-Marquardt – Exercice

Régression non
linéaire

Méthode de Newton

Méthode de
Gauss-Newton

Calcul de la jacobienne

Méthode de
Levenberg-Marquardt

Inversion non
linéaire

- Le fichier disponible à https://github.com/bernard-giroux/geo1302/blob/master/ves_part.py contient une classe Sondage avec les méthodes
 - `mod` pour modéliser la courbe de résistivité apparente;
 - `_jacobian` pour calculer la jacobienne;
 - `inv` pour faire l'inversion.
- La fonction `inv` est à compléter.
- Important :
 - la résistivité peut varier sur plusieurs ordres de grandeur;
 - il est préférable de calculer l'erreur de prédiction avec le log de la résistivité (i.e. $\mathbf{d} = \log(\rho_a^{\text{obs}})$ et $\hat{\mathbf{d}} = \log(\rho_a^{\text{pre}})$) pour éviter de donner un poids relatif trop faible aux faibles résistivités;
 - il faut alors en tenir compte lors de la mise à jour car $\Delta \mathbf{m}$ est obtenu à partir de $\mathbf{d} - \hat{\mathbf{d}}$

$$\mathbf{m} = \exp(\log(\mathbf{m}) + \Delta \mathbf{m}) \quad (36)$$

Régression non
linéaire

**Inversion non
linéaire**

Régularisation

Inversion d'Occam

Résolution

Inversion non linéaire

Régression non
linéaire

Inversion non
linéaire

Régularisation
Inversion d'Occam
Résolution

- Les méthodes vues précédemment fonctionnent bien lorsqu'il y a peu de paramètres à estimer.
- Les modèles comportant un plus grand nombre de paramètres sont susceptibles d'être partiellement indéterminés;
 - comme pour le cas linéaire, il faut régulariser le problème.
- Les approches visant à minimiser la norme $\|\mathbf{Dm}\|_2$ (\mathbf{D} est une matrice de lissage) peuvent être adaptées au cas non linéaire;
- Le problème consiste à minimiser le résidu, sous une contrainte sur $\|\mathbf{Dm}\|_2$

$$\begin{aligned} \min \|G(\mathbf{m}) - \mathbf{d}\|_2 \\ \|\mathbf{Dm}\|_2 \leq \varepsilon \end{aligned}$$

ou en posant la forme amortie

$$\min \|G(\mathbf{m}) - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{Dm}\|_2^2 \quad (37)$$

- La méthode de Gauss-Newton peut être utilisée si on récrit (37) tel que

$$\min \left\| \begin{array}{c} G(\mathbf{m}) - \mathbf{d} \\ \alpha \mathbf{D}\mathbf{m} \end{array} \right\|_2^2 \quad (38)$$

- La jacobienne de (38) à l'itération p est

$$\mathbf{K}^{(p)} = \begin{bmatrix} \mathbf{J}^{(p)} \\ \alpha \mathbf{D} \end{bmatrix} \quad (39)$$

- En insérant (39) dans (23), on obtient

$$\mathbf{K}^{(p)T} \mathbf{K}^{(p)} \Delta \mathbf{m} = -\mathbf{K}^{(p)T} \begin{bmatrix} G(\mathbf{m}^{(p)}) - \mathbf{d} \\ \alpha \mathbf{D}\mathbf{m}^{(p)} \end{bmatrix} \quad (40)$$

Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- En combinant (39) et (40), on trouve finalement

$$(\mathbf{J}^T \mathbf{J} + \alpha^2 \mathbf{D}^T \mathbf{D}) \Delta \mathbf{m} = -\mathbf{J}^T (\mathbf{G}(\mathbf{m}) - \mathbf{d}) - \alpha^2 \mathbf{D}^T \mathbf{D} \mathbf{m} \quad (41)$$

où l'indice (p) indiquant l'itération a été omis pour plus de clarté.

- Il est important de noter que *la jacobienne doit être recalculée à chaque itération.*
- Le terme de régularisation α stabilise le système;
 - en général il n'est pas nécessaire d'utiliser le terme $\lambda \mathbf{I}$ de Levenberg-Marquardt.

Régression non linéaire

Inversion non linéaire

Régularisation

Inversion d'Occam

Résolution

- Pour certaines applications, il peut être intéressant d'appliquer un lissage différent selon la direction;
 - dans un milieu sédimentaire, un lissage horizontal plus marqué est souvent préférable car les paramètres varient moins horizontalement que verticalement.
- Cela se fait en définissant des matrices de lissage pour chaque direction et en utilisant un poids différent pour chacune d'elle :

$$\begin{aligned}
 & (\mathbf{J}^T \mathbf{J} + \alpha^2 \mathbf{D}_x^T \mathbf{D}_x + \beta^2 \mathbf{D}_z^T \mathbf{D}_z) \Delta \mathbf{m} = \\
 & - \mathbf{J}^T (G(\mathbf{m}) - \mathbf{d}) - \alpha^2 \mathbf{D}_x^T \mathbf{D}_x \mathbf{m} - \beta^2 \mathbf{D}_z^T \mathbf{D}_z \mathbf{m} \quad (42)
 \end{aligned}$$

- Un poids α plus élevé que β génère un modèle où les couches horizontales sont accentuées.

Régression non
linéaire

Inversion non
linéaire

Régularisation

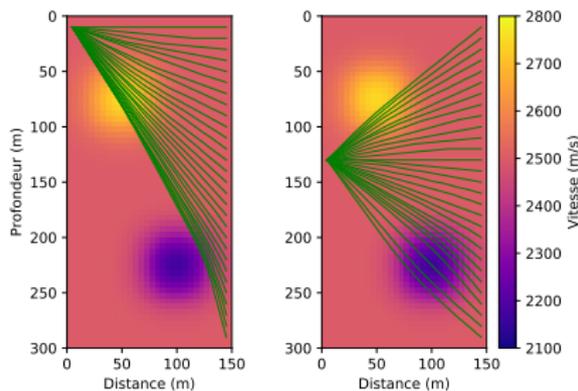
Inversion d'Occam

Résolution

- Considérons le cas de la tomographie entre forages pour lequel les rais sont infléchis en raison des contrastes de vitesse.
- On cherche à estimer la vitesse à partir des temps de parcours.
- Le temps de parcours de la source T_x au récepteur R_x est

$$t = \int_{T_x}^{R_x} s(l) dl \quad (43)$$

où s est la lenteur et l est la trajectoire.



Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Le modèle est discrétisé en cellules à l'intérieur desquelles la vitesse/lenteur est constante, i.e.

$$\mathbf{m} = [s_0, s_1, s_2, \dots, s_{M-1}]^T \quad (44)$$

- Pour un tel modèle, le temps de parcours pour une paire $Tx - Rx$ devient

$$t = \sum_i^{nseg} s_i l_i \quad (45)$$

où i correspond aux cellules traversées par un segment de rai et l_i est la longueur du segment.

Régularisation – Exemple

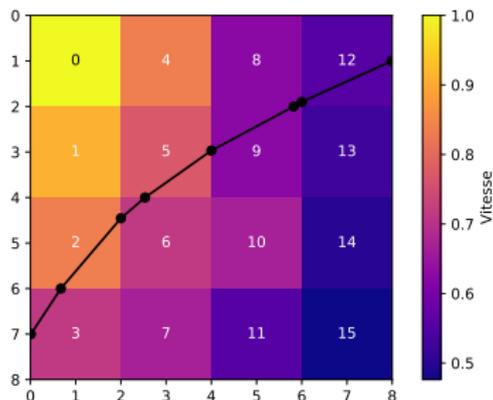
Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution



- Pour le modèle ci-dessus, le temps est

$$\begin{aligned}
 t &= s_2 l_2 + s_3 l_3 + s_5 l_5 + s_6 l_6 + s_8 l_8 + s_9 l_9 + s_{12} l_{12} \\
 &= 2.0s_2 + 1.2s_3 + 1.8s_5 + 0.7s_6 + 0.2s_8 + 2.1s_9 + 2.2s_{12} \\
 &= \mathbf{lm}
 \end{aligned}$$

où $\mathbf{l} = [0, 0, 2.0, 1.2, 0, 1.8, 0.7, 0, 0.2, 2.1, 0, 0, 2.2, 0, 0, 0]$

- Pour plusieurs mesures à des paires $Tx - Rx$ différentes, on aura

$$\mathbf{d} = \mathbf{Lm} \quad (46)$$

où

$$\mathbf{d} = [t_0, t_1, t_2, \dots, t_{N-1}]^T \quad (47)$$

et \mathbf{L} est une matrice creuse $N \times M$ contenant les longueurs de segments.

- \mathbf{L} doit être recalculée chaque fois que \mathbf{m} est mis à jour.
- Il est intéressant de noter que la jacobienne est égale à \mathbf{L} , par exemple pour le 1^{er} terme on a

$$\frac{\partial d_0}{\partial m_0} = \frac{\partial}{\partial m_0} (s_0 l_0 + s_1 l_1 + s_2 l_2 + \dots) = l_0 \quad (48)$$

Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Les données sont contenues dans le fichier `model1_tt.dat`

```
data = np.loadtxt('model1_tt.dat')
Tx = data[:, :2].copy()
Rx = data[:, 2:4].copy()
dobs = data[:, -1]
```

- Le modèle initial peut être défini à partir de la lenteur apparente moyenne :
 - la lenteur apparente est définie par le temps de parcours entre $Tx - Rx$ divisé par la distance en ligne droite entre $Tx - Rx$.
- La matrice L peut être obtenue à partir de la classe `Grid2d` du module `ttcrpy.rgrid`

Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

```

# Paramètres de la grille
dx = 5.0
dz = 5.0
xmin = 0.0
zmin = 0.0
xmax = 150.0
zmax = 300.0
nx = int((xmax-xmin)/dx+0.001)
nz = int((zmax-xmin)/dz+0.001)
xg = np.linspace(xmin, xmax, nx+1)
zg = np.linspace(zmin, zmax, nz+1)

g = rg.Grid2d(xg, zg, method='SPM', nsnx=10, nsnz=10)

# Modèle initial
Ldroit = rg.Grid2d.data_kernel_straight_rays(Tx, Rx, xg, zg)
l_rai = Ldroit.sum(axis=1)
lent_app = dobs/l_rai
m0 = np.mean(lent_app) + np.zeros((nx*nz,))
  
```

Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Pour les raiis courbes, le calcul se fait en appelant
`d, L = g.raytrace(Tx, Rx, slowness=m, compute_L=True)`

- Examinez les résultats pour 4 itérations pour les valeurs suivantes de α

```
alpha_val = [10.0, 15.0, 25.0, 40.0, 80.0, \
             160.0, 300.0, 500.0, 1000.0, 10000.]
```

- La fonction `compute_K` peut être utilisée pour calculer la matrice de lissage :

```
Dx, Dz = g.compute_K(order=2)
D = Dx + Dz
```

Régularisation – Exemple

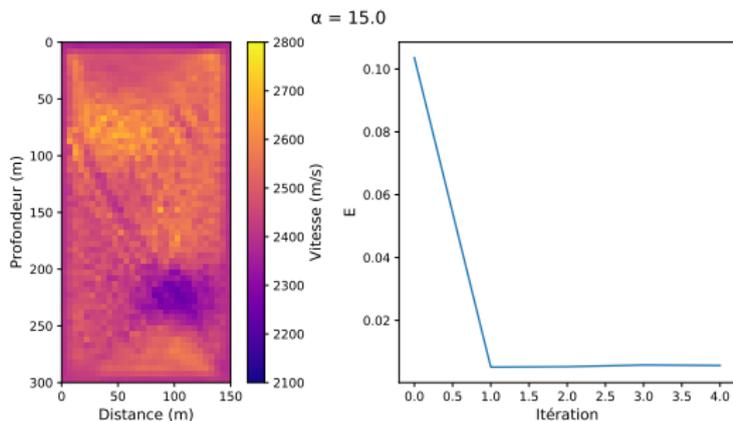
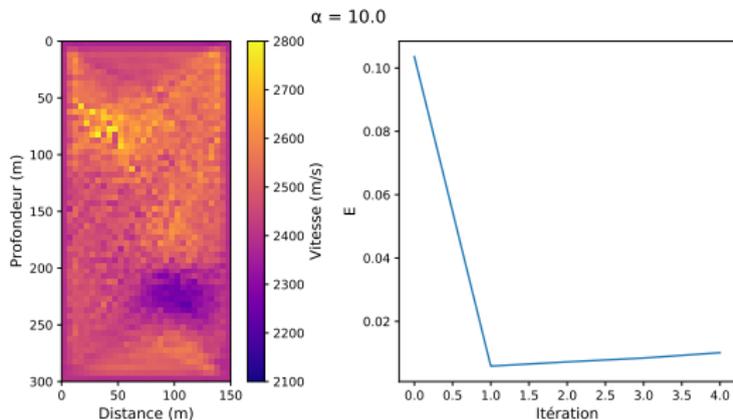
Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution



Régularisation – Exemple

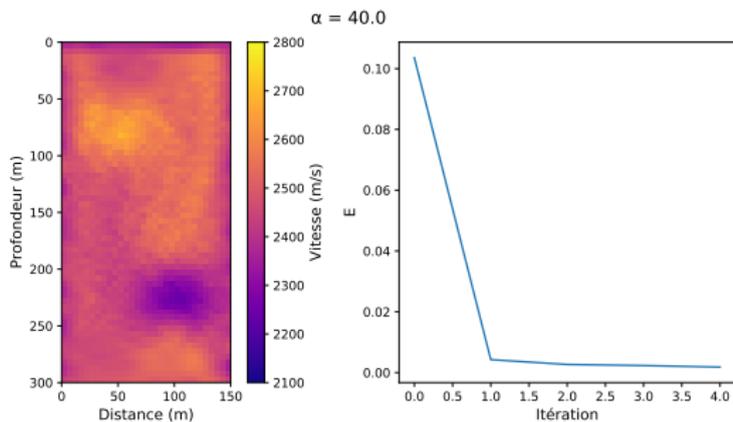
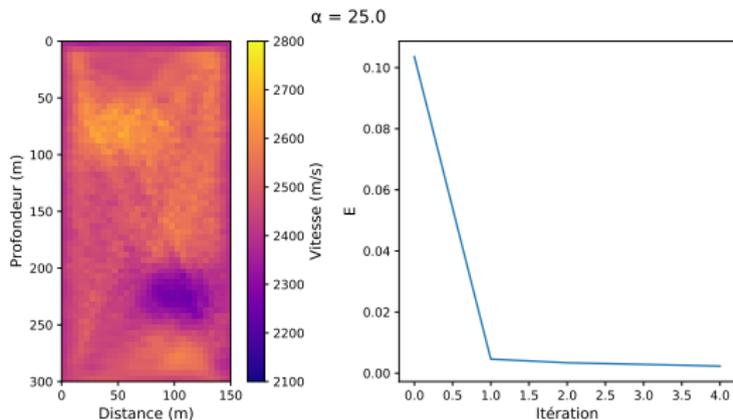
Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution



Régularisation – Exemple

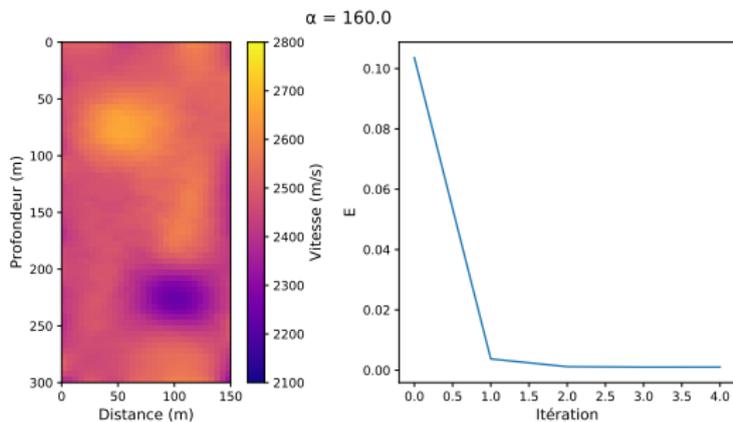
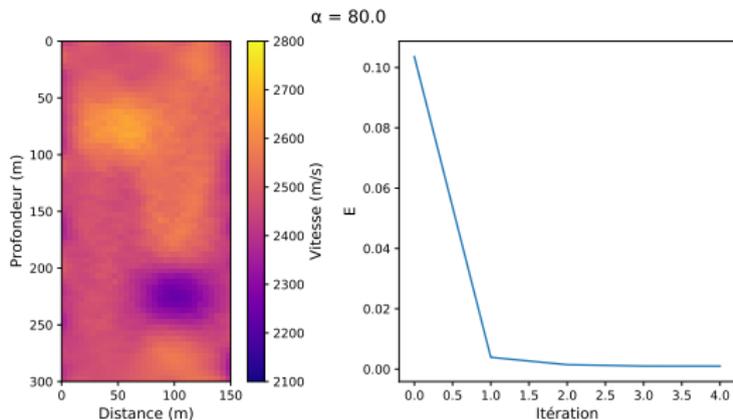
Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution



Régularisation – Exemple

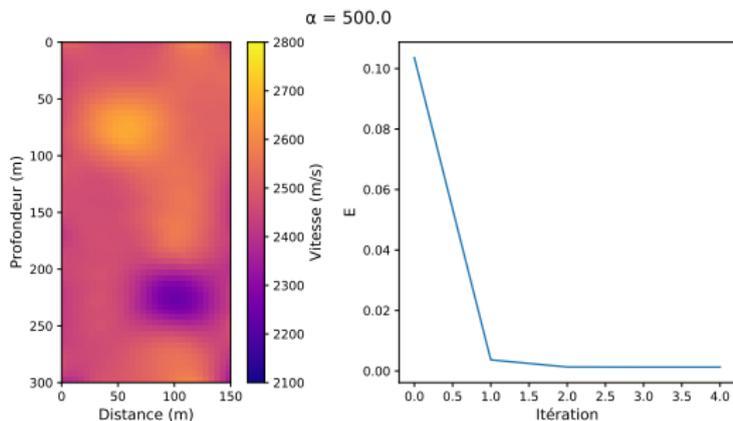
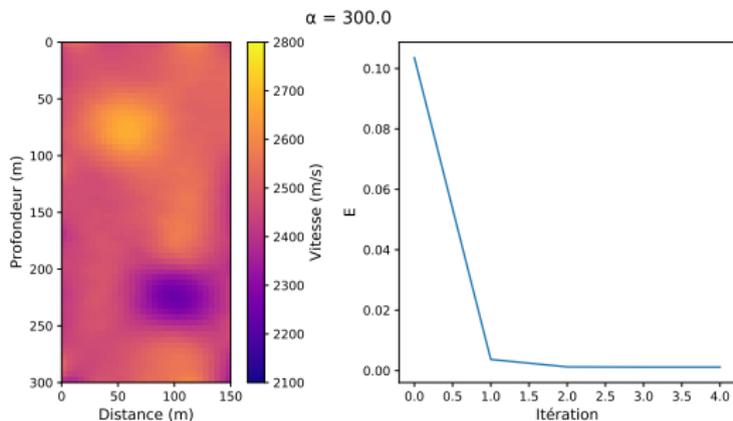
Régression non linéaire

Inversion non linéaire

Régularisation

Inversion d'Occam

Résolution



Régularisation – Exemple

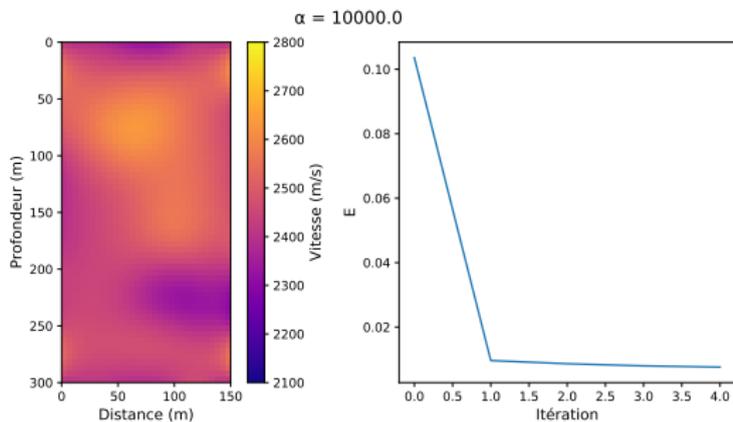
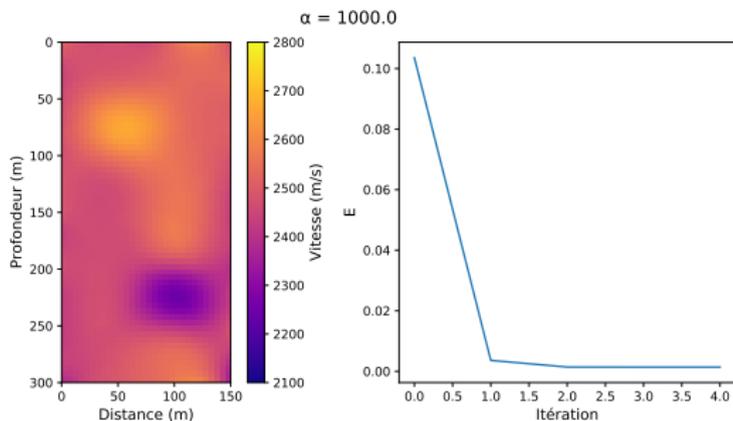
Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution



Régularisation – Exemple

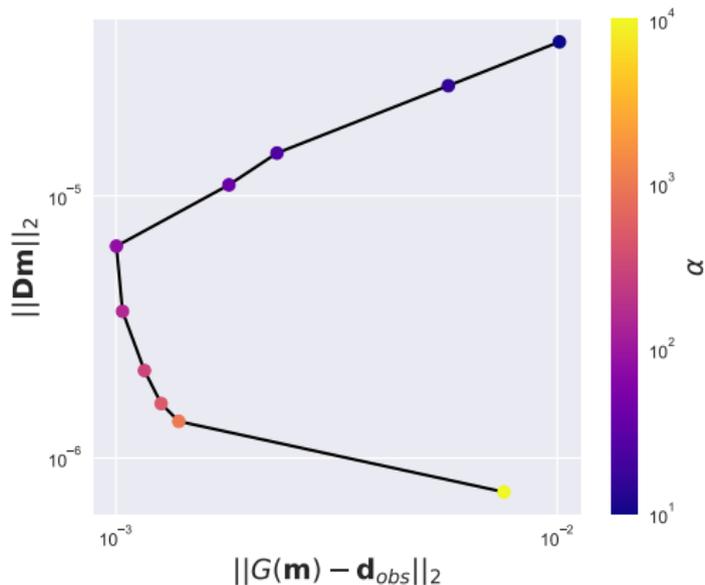
Régression non linéaire

Inversion non linéaire

Régularisation

Inversion d'Occam

Résolution



- Un algorithme fréquemment rencontré dans la littérature consiste à minimiser $\|\mathbf{D}\mathbf{m}\|_2$ sous la contrainte $\|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2 \leq \delta$
- La méthode repose sur le théorème de Taylor et l'utilisation d'un modèle de départ \mathbf{m}^k , i.e.

$$\mathbf{G}(\mathbf{m}^k + \Delta\mathbf{m}) \approx \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m}$$

où $\mathbf{J}(\mathbf{m}^k)$ est la jacobienne.

- À partir de cette expression, le problème des moindres-carrés régularisé devient

$$\min \|\mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} - \mathbf{d}\|_2^2 + \alpha^2\|\mathbf{D}(\mathbf{m}^k + \Delta\mathbf{m})\|_2^2,$$

où \mathbf{m}^k est constant et $\Delta\mathbf{m}$ est la variable à estimer ;

- On suppose ici que $\mathbf{G}(\mathbf{m}) = \mathbf{d}$ a été mis à l'échelle pour que les écarts-type σ_i soient égaux.

- Le problème est reformulé en définissant une nouvelle variable $\mathbf{m}^{k+1} = \mathbf{m}^k + \Delta\mathbf{m}$
- La norme des résidus devient

$$\min \|\mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} + \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k - \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k - \mathbf{d}\|_2^2$$

ou

$$\min \|\mathbf{J}(\mathbf{m}^k)(\mathbf{m}^k + \Delta\mathbf{m}) - (\mathbf{d} - \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k)\|_2^2$$

- En posant

$$\hat{\mathbf{d}}(\mathbf{m}^k) = \mathbf{d} - \mathbf{G}(\mathbf{m}^k) + \mathbf{J}(\mathbf{m}^k)\mathbf{m}^k,$$

on obtient l'expression suivante pour le problème des moindres-carrés régularisé :

$$\min \|\mathbf{J}(\mathbf{m}^k)\mathbf{m}^{k+1} - \hat{\mathbf{d}}(\mathbf{m}^k)\|_2^2 + \alpha^2 \|\mathbf{D}(\mathbf{m}^k + \Delta\mathbf{m})\|_2^2 \quad (49)$$

- La solution est

$$\mathbf{m}^{k+1} = (\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \alpha^2 \mathbf{D}^T \mathbf{D})^{-1} \mathbf{J}(\mathbf{m}^k)^T \hat{\mathbf{d}}(\mathbf{m}^k). \quad (50)$$

- Cette solution est similaire à la méthode de Gauss-Newton;
- *La particularité de l'inversion d'Occam est que le poids α est ajusté de façon dynamique*
- À chaque itération, on choisit la valeur de α la plus élevée pour laquelle

$$\chi^2 = \sum_{i=0}^{N-1} \frac{(d_i - (\mathbf{G}(\mathbf{m}))_i)^2}{\sigma_i^2}$$

est inférieur à δ ;

- si ce n'est pas possible, on choisit α qui donne la valeur de χ^2 la plus faible.

Régression non
linéaire

Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Un exemple illustrant l'avantage de l'inversion d'Occam est donné à https://github.com/bernard-giroux/geo1302/blob/master/inv_occam.ipynb

- On a vu en inversion linéaire que la matrice de résolution des paramètres est donnée par

$$\mathbf{m} = \mathbf{G}^{\dagger} \mathbf{G} \mathbf{m}_{\text{vrai}} = \mathbf{R}_m \mathbf{m}_{\text{vrai}} \quad (51)$$

- Pour le cas non linéaire, nous aurions

$$\mathbf{m} = G^{-1} (G (\mathbf{m}_{\text{vrai}})) \quad (52)$$

où G^{-1} est l'opérateur inverse.

- Cependant, l'équation (52) ne peut pas être représenté par des matrices
 - \mathbf{R}_m ne peut pas être calculé.
- Par ailleurs, la résolution dépend du choix du modèle initial.
- Pour ces raisons, on procède plutôt avec des tests de résolution.

Régression non
linéaire

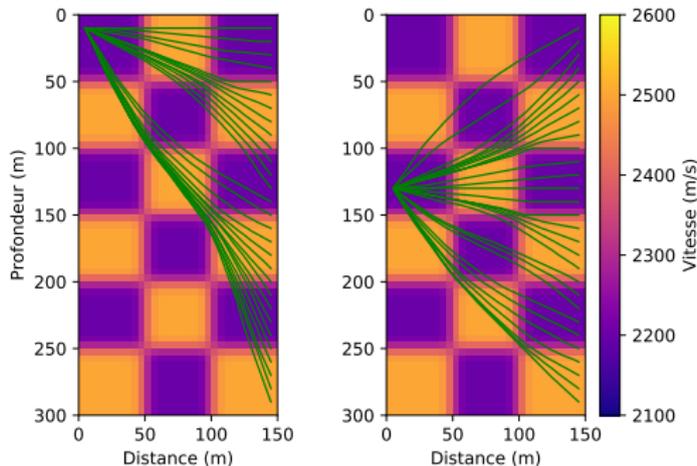
Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Le modèle en damier est souvent employé pour les tests de résolution.
- La configuration de mesure modélisée est celle qui est prévue sur le terrain.



Régression non
linéaire

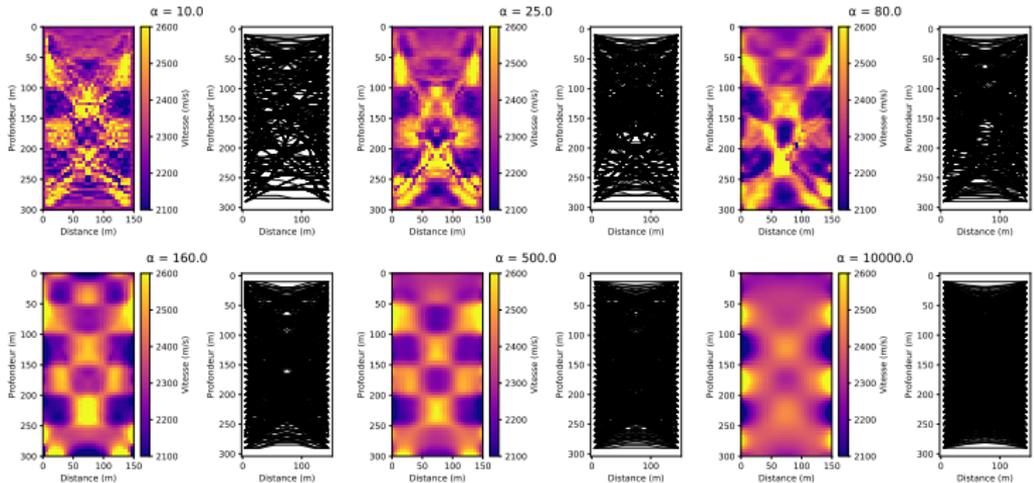
Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Résultats pour des données non bruitées.



Régression non
linéaire

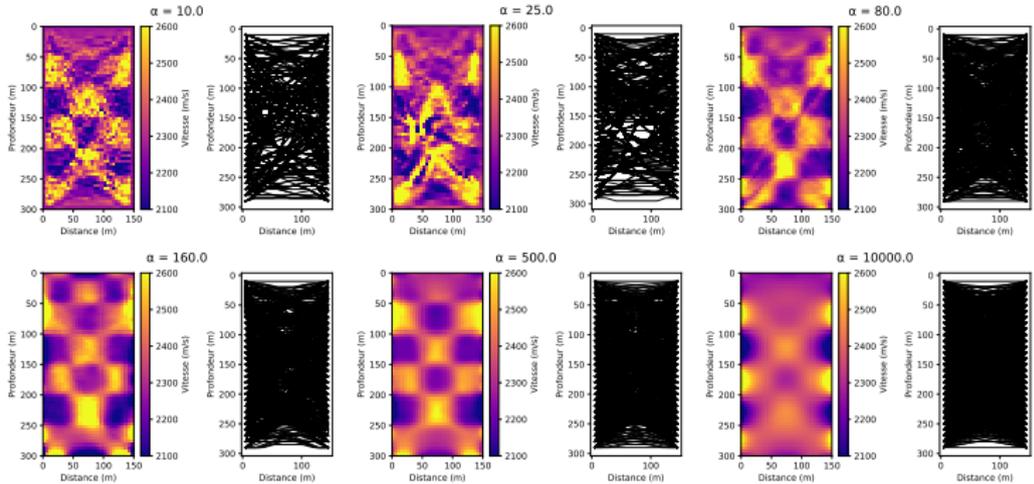
Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

● Résultats pour des données avec un bruit gaussien



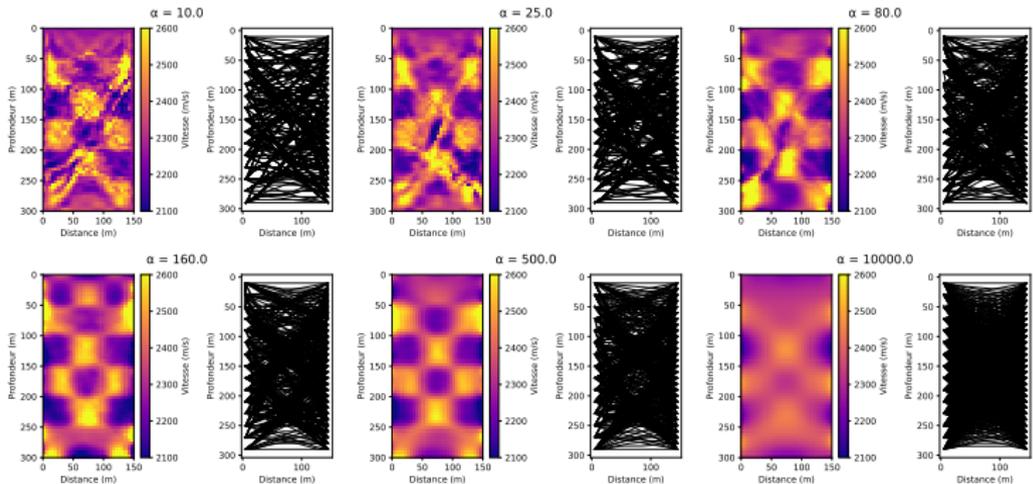
Régression non
linéaire

Inversion non
linéaire

Régularisation
Inversion d'Occam

Résolution

- Résultats pour des données avec un bruit gaussien et un point de tir sur deux



Régression non
linéaire

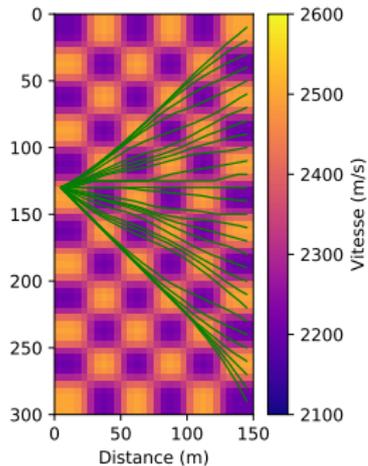
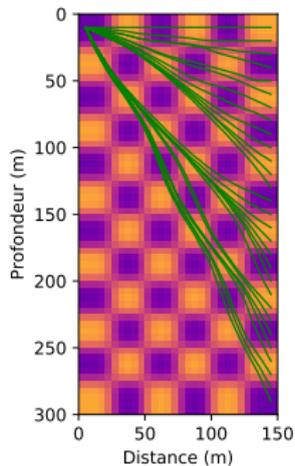
Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

- Des objets plus petits sont-ils résolus ?



Régression non linéaire

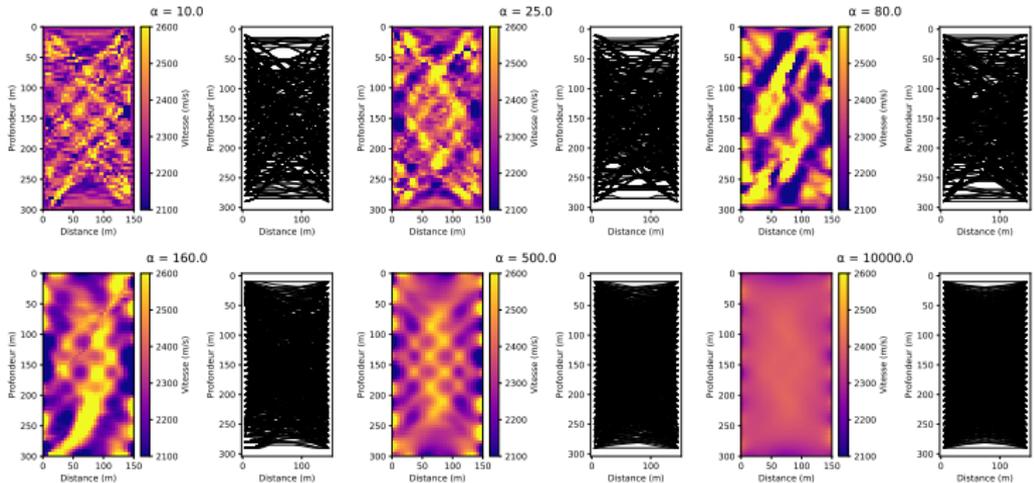
Inversion non linéaire

Régularisation

Inversion d'Occam

Résolution

- Résultats pour des données non bruitées.



Régression non
linéaire

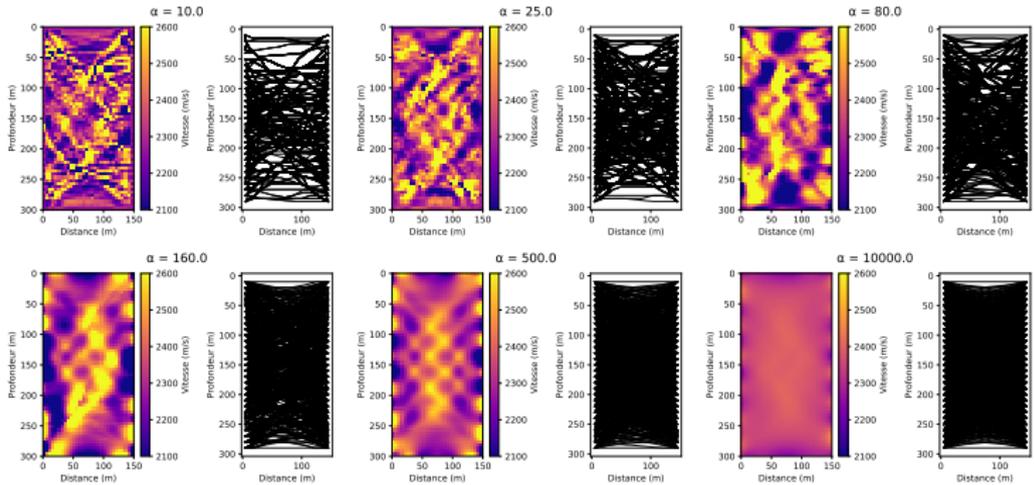
Inversion non
linéaire

Régularisation

Inversion d'Occam

Résolution

● Résultats pour des données avec un bruit gaussien



Régression non
linéaire

Inversion non
linéaire

Régularisation
Inversion d'Occam

Résolution

- Résultats pour des données avec un bruit gaussien et un point de tir sur deux

