

# MODÉLISATION ET INVERSION EN GÉOPHYSIQUE

## 4 - Inversion non linéaire

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

Institut national de la recherche scientifique  
Centre Eau Terre Environnement

Version 1.0.0  
Hiver 2018

Régression non  
linéaire

Méthode de Newton

Méthode de  
Gauss-Newton

Calcul de la jacobienne

Méthode de  
Levenberg-Marquardt

# Régression non linéaire

## Régression non linéaire

Méthode de Newton

Méthode de  
Gauss-Newton

Calcul de la jacobienne

Méthode de  
Levenberg-Marquardt

- 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 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<sup>er</sup> termes, on a

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

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

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

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

où

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

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

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

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

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

- 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.

- L'algorithme de Newton est le suivant

---

## Algorithm 1 Méthode de Newton

---

```

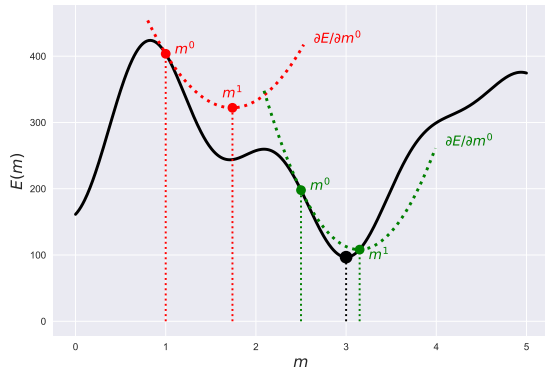
1:  $p \leftarrow 0$ 
2: while  $\nabla E \neq 0$ 
3:   Calculer  $\nabla E$  et  $\mathbf{H}$ 
4:   Résoudre  $\mathbf{H} \Delta \mathbf{m} = -\nabla E$ 
5:    $\mathbf{m}^{(p+1)} \leftarrow \mathbf{m}^{(p)} + \Delta \mathbf{m}$ 
6:    $p \leftarrow p + 1$ 

```

---

- 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.

- 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).





- En pratique, la convergence  $\nabla E = 0$  n'est jamais atteinte ;
- 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.

- La méthode de Newton n'est pas applicable quand il n'y a pas de solution exacte à  $G(\mathbf{m}) = \mathbf{d}$  ou que 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  $\sigma_i$  est l'écart-type de la  $i^e$  mesure.

- 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  $W_{ii} = 1/\sigma_i^2$ .

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

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

et les éléments du gradient sont

$$(\nabla E(\mathbf{m}))_j = \sum_{i=0}^{N-1} 2E_i(\mathbf{m}) (\nabla E_i(\mathbf{m}))_j \quad (12)$$

où l'indice  $j$  signifie  $\frac{\partial}{\partial m_j}$  et où  $j = 0, 1, \dots, M-1$ .

- 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 ( $N \times M$ ) :

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

- 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  $E_i(\mathbf{m})^2$ .

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

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

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

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

- 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

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

- 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  $\mathbf{J}^T \mathbf{W} \mathbf{J} \Delta \mathbf{m} = \mathbf{J}^T \mathbf{W} (\mathbf{d} - \hat{\mathbf{d}})$
- Comme pour la méthode de Newton, la méthode de Gauss-Newton peut converger vers un minimum local.



- Soit une fonction non linéaire

$$G_i(\mathbf{m}) = \sin(\omega_0 m_0 x_i) + m_0 m_1 \quad (23)$$

- 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
    
```

- 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 :

$$\frac{\partial \hat{d}_i}{\partial m_j} = \frac{\partial G_i(\mathbf{m})}{\partial m_j} \approx \frac{\partial G_i(\mathbf{m} + \Delta \mathbf{m}_j) - G_i(\mathbf{m})}{\Delta m_j} \quad (24)$$

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

Régression non  
linéaire

Méthode de Newton

Méthode de  
Gauss-Newton

Calcul de la jacobienne

Méthode de  
Levenberg-Marquardt

- Le choix de la perturbation  $\Delta m_j$  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.

- 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

$$\left( \mathbf{J}^T \mathbf{J} + \lambda \mathbf{I} \right) \Delta \mathbf{m} = \nabla E \quad (25)$$

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

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

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

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 (25), on trouve

$$\Delta \mathbf{m} = \left( \mathbf{V}\mathbf{S}^2\mathbf{V}^T + \beta^2\mathbf{I} \right)^{-1} \mathbf{V}\mathbf{S}\mathbf{U}^T \left( \mathbf{d} - \hat{\mathbf{d}} \right) \quad (27)$$

où  $\beta^2 = \lambda$ .

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

$$\left( \mathbf{V}\mathbf{S}^2\mathbf{V}^T + \beta^2\mathbf{I} \right) = \left( \mathbf{V}\text{diag}(\mathbf{s}^2)\mathbf{V}^T + \beta^2\mathbf{I} \right) \quad (28)$$

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

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

- L'inverse de cette expression est

$$\left( \mathbf{V}\text{diag}(\mathbf{s}^2 + \beta^2)\mathbf{V}^T \right)^{-1} = \mathbf{V}\text{diag} \left( \frac{1}{\mathbf{s}^2 + \beta^2} \right) \mathbf{V}^T \quad (30)$$

- En insérant cette dernière expression dans (27), on trouve

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

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

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

$$\beta = s_l \Delta E^{1/l} \quad (33)$$

où  $l = 1, \dots, M$  et

$$\Delta E^{(p)} = \frac{E^{(p-1)} - E^{(p)}}{E^{(p-1)}}. \quad (34)$$

- 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:     $p \leftarrow p + 1$ 
    
```

---



Régression non  
linéaire

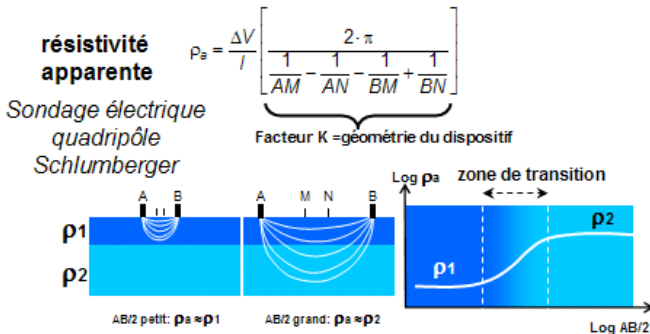
Méthode de Newton

Méthode de  
Gauss-Newton

Calcul de la jacobienne

Méthode de  
Levenberg-Marquardt

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



Source : Université de Lorraine

- Le fichier disponible à [https://github.com/bernard-giroux/geo1302/blob/master/ves\\_part.py](https://github.com/bernard-giroux/geo1302/blob/master/ves_part.py) contient une classe Sondage avec les méthodes
  - `mod` pour modéliser la courbe de resistivité 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  $\mathbf{d} - \hat{\mathbf{d}}$

$$\mathbf{m} = \exp(\log(\mathbf{m}) + \Delta \mathbf{m}) \quad (35)$$