

Projet n4

*Systèmes non linéaires d'équations - Méthode
Newton-Raphson*

Groupe n1 - Equipe n2

Responsable : cel-ha910e

Secrétaire : cplages

Codeurs : aguichard002, jcalandra001

Résumé : Le but de ce projet est de programmer des algorithmes dédiés à la recherche des racines de systèmes d'équations non linéaires. La méthode utilisée ici est l'algorithme de Newton-Raphson dont nous en évaluerons les avantages et inconvénients. Pour cela, nous testerons cette méthode dans différents cas : la détermination des points de Lagrange et la recherche de points d'équilibre électrostatique.

Méthode de Newton-Raphson

La méthode de Newton-Raphson représente la première partie de notre travail, permettant d'approximer les racines d'un système d'équation non linéaire, elle est requise dans les deux applications que nous étudierons par la suite.

Nous avons programmé la méthode de Newton-Raphson dans n'importe quelle dimension. L'idée de l'algorithme est de considérer qu'en une position U donnée, la meilleure direction pour s'approcher de la racine de la fonction f est donnée par la tangente de f en U . Considérant U comme le vecteur position, nous obtenons le vecteur V suivant correspondant à l'équation $f(U + V) = 0$ où $f(U + V)$ est approximé par $f(U) + H(U) * V$ avec f la fonction f et H son Jacobien.

Notre but est donc de résoudre l'équation suivante : $H(U) * V = -F(U)$
Par la suite nous remplaçons U par $U + V$ jusqu'à ce que U converge ou que le critère d'arrêt (nombre d'itérations) soit atteint. C'est à dire, pour tester la convergence de U , nous prenons une fonction f et son jacobien J , récupérons le vecteur U de la fonction Newton-Raphson et vérifions que $f(U)$ est égal à 0 à une certaine précision près.

```
def Newton_Raphson(f, J, U0, N, epsilon):
    U = np.copy(U0)
    for i in range(N):
        fu = f(U)
        na = np.linalg.norm(fu)
        if (na < epsilon):
            return U
        ju = J(U)
        V = np.linalg.lstsq(ju, -fu)[0]
        U = U + V
    print("ERREUR : _precision_non_atteinte")
return U
```

FIGURE 1 – Implémentation de la méthode `Newton_Raphson`

L'algorithme mis en place pour cette méthode est disponible dans la figure ci-dessus. Il prend en argument la fonction f et son Jacobien J ainsi que le vecteur position initial $U0$, le nombre maximal d'itération N et la précision ϵ .

Enfin nous avons amélioré notre programme en rajoutant un *backtracking* dans notre code, méthode qui permet de revenir légèrement en arrière sur des décisions prises afin de sortir d'un blocage. Cette méthode nous est très utile dans le cas où notre vecteur position se met à diverger, cas tout à fait plausible étant donné que notre algorithme se base sur une position de départ donnée par l'utilisateur.

Détermination des points de Lagrange

Recherche de points d'équilibre électrostatique

Le problème posé ici est celui de deux charges électrostatiques positionnées aux extrémités de l'axe $[-1, 1]$. Nous supposons qu'il existe N charges x_1, x_2, \dots, x_N en mouvement sur cet intervalle.

L'énergie totale du système vaut :

$$E(x_1, x_2, \dots, x_N) = \sum_{i=1}^N \log |x_i + 1| + \log |x_i - 1| + \frac{1}{2} \sum_{j=1, j \neq i}^N \log |x_i - x_j|$$

Afin de trouver les positions d'équilibre, nous devons résoudre le système d'équations non linéaire suivant :

$$\nabla E(x_1, x_2, \dots, x_N) = \left[\frac{\partial E(x_1, \dots, x_N)}{\partial x_i} \right] = 0$$

Nous avons utilisé la méthode de Newton Raphson implémentée précédemment. Il nous a donc fallu dans un premier temps calculer la matrice Jacobienne H de $\nabla E(x_1, x_2, \dots, x_N)$ telle que :

$$H = \left[\frac{\partial^2 E(x_1, \dots, x_N)}{\partial x_i \partial x_j} \right]_{1 \leq i, j \leq N}$$

TODO : question -> est ce que je détaille plus les calculs de la Jacobienne ou pas ?

Une fois la matrice Jacobienne implémentée, il ne nous reste plus qu'à utiliser notre fonction Newton Raphson afin de résoudre le système, puis placer les points correspondants sur l'axe $[-1, 1]$. Nous obtenons ainsi un graphique montrant les points d'équilibre pour 1,2,3 et 4 charges électrostatiques.

Une autre interrogation à présent est de comparer nos points aux racines des dérivées des polynômes de Legendre, nous avons donc superposé celles-ci à notre graphique existant.

Nous pouvons observer que les polynômes de Legendre s'annulent aux mêmes positions que les points d'équilibre.

TODO : inclure graphique !

Enfin notre dernière interrogation était de savoir si ces points correspondaient à des maximums ou des minimums d'énergie...

TODO : explications ? (quand la dérivée de l'énergie potentielle est nulle, on a un équilibre soit instable $E_p'' < 0$ ou stable $E_p'' > 0$ équilibre stable \Leftrightarrow minimum d'énergie potentielle équilibre instable \Leftrightarrow maximum d'énergie potentielle ???)

Conclusions