Projet n°4
Systèmes non linéaires d'équations / Méthode de Newton-Raphson

Groupe n°3 - Équipe n°6
Responsable : paubreton
Secrétaire : dchapotte
Codeurs : lhing, lpfleger, ndehaiesdit

Introduction :

Le but de ce projet est de développer une manière capable de résoudre des systèmes non linéaires d'équations. La méthode de Newton-Raphson permet de trouver les racines d'un de ces systèmes linéaires particuliers. Nous allons ici nous servir de cette méthode pour résoudre deux problèmes distincts : la recherche des points de lagrange et la recherche des points d'équilibre électrostatique.

1 Méthode de Newton-Raphson

Le but ici donc est d'implémenter de la manière la plus convenable possible la méthode de Newton-Raphson, qui sera ensuite utilisée tout au long de ce projet.

1.1 Analyse

La méthode de Newton-Raphson permet de trouver numériquement une approximation précise d'une racine d'une fonction réelle d'une variable réelle. Soit une fonction $f : \mathbb{R}^n \to \mathbb{R}^m$ dérivable de telle sorte que

$$f : \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \to \begin{bmatrix} f_1(x_1, ..., x_n) \\ \vdots \\ f_m(x_1, ..., x_n) \end{bmatrix}$$

On lui attribue la Jacobienne H définie ci-dessous

$$H : \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \to \begin{bmatrix} \frac{\partial f_i(x_1, ..., x_n)}{\partial x_j} \end{bmatrix}$$

Le principe de la méthode de Newton-Raphson est que, mettons que nous ayons une valeur U, une position. L'éventuelle meilleure direction pour tenter de s'approcher de la racine est de considérer celle de la tangente en ce point U. Nous calculons alors une nouvelle position nommée ici V qui vérifie $f(U + V) = 0$, où $f(U + V)$ est approximmée par $f(U) + \underbrace{H(u)}_{matrice} \times \underbrace{V}_{vecteur}$.

Puis à l'aide de python et notamment de la méthode *numpy.linalg.lstsq*, nous pouvons aisément retrouver la valeur de V qui est liée à H et U *via* l'équation $H(U) \times V = -f(U)$. Pour la manière de fonctionner, on réitère ceci jusqu'à ce que la valeur de U converge vers un *minima* (ou que l'on ait fait beaucoup d'itérations), en remplaçant à chaque itération U par $U + V$.

Nous pouvons en déduire un algorithme :

Données : f : Fonction, H : matrice Jacobienne, U_0 : *vecteur*, N : *entier*,ϵ : reel
Résultat : U : vecteur

début
- U $\leftarrow U_0$
- **pour** $i \in [0, N]$ **faire**
  - $V \leftarrow (f(U) + H(U) \times V = 0)$
  - $U \leftarrow U + V$
  - **si** $||f(U)|| \leq \epsilon$ **alors**
    - ***Retourner*** U
- ***Retourner*** U

Algorithme 1 : Algorithme de Newton-Raphson : Newton_Raphson(f, J, U0, N, epsilon)

Nous avons donc implémenté cet algorithme en modifiant légèrement la boucle. Nous l'avons remplacée par une boucle $Tantque$ avec plusieurs conditions.

1.2 Méthode avec Backtracking

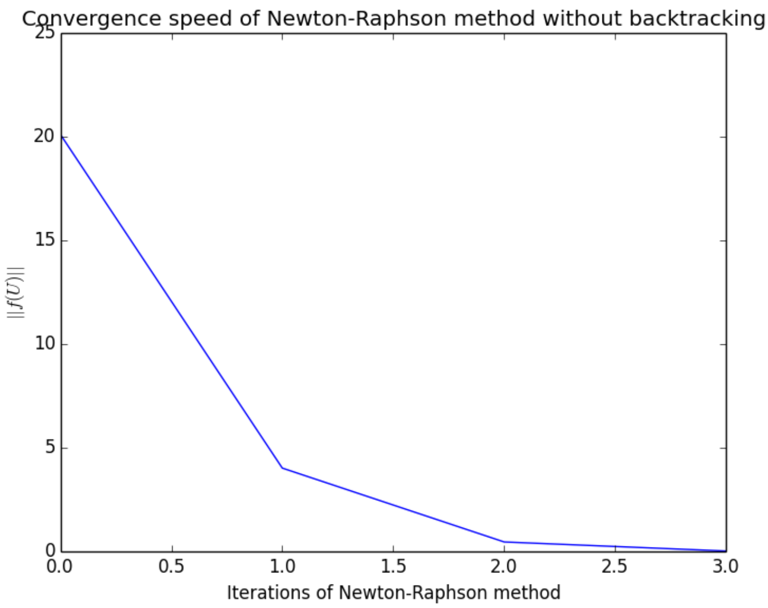
Le fait est que la méthode de Newton-Raphson ne converge pas dans tous les cas. Nous avons donc eu la possibilité d'améliorer cette méthode en utilisant le *Backtracking*. Ceci consiste en la véirification à chaque tour de boucle du bon résultat du calcul du vecteur V, de telle manière que $||f(U+V)|| \leq f||(U)||$. Ainsi notre nouvel algorithme avec backtracking contient simplement quelques lignes supplémentaires permettant d'assurer les bonnes valeurs de V.

1.3 Tests et résultats

Pour vérifier le bon fonctionnement de cette méthode nous allons l'appliquer à un système comme par exemple :

$$\begin{cases} x^2 - 2 = 0 \\ y^2 - 5 = 0 \end{cases}$$

Après avoir réalisé quelques ajustements sur les fonctions dans le but de pouvoir traiter des fonctions avec plusieurs paramètres, nous en arrivons à des résultats concluants, et ce pour les méthodes avec et sans backtracking. De plus grâce au graphique qui suit, nous pouvons observer la vitesse de convergence des calculs de la méthode de Newton-Raphson.



2 Calcul des points de Lagrange

2.1 Analyse du Problème

Dans cette partie, nous devons déterminer les points d'équilibre d'un système dans un espace à deux dimensions. Nous considérerons ici trois forces :

— force élastique f_e : $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow -k_0 * \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$

— force centrifuge f_c : $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow k_1 * \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$

— force gravitationnelle f_g : $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow -k_2 * \begin{bmatrix} (x - x_0)/((x - x_0)^2 + (y - y_0)^2)^{\frac{3}{2}} \\ (y - y_0)/((x - x_0)^2 + (y - y_0)^2)^{\frac{3}{2}} \end{bmatrix}$

avec k_0, k_1, k_2 trois constantes.

Pour étudier les positions d'équilibre, on doit donc chercher les positions telles que $\sum \vec{F} = 0$. Or les forces sont représentées dans deux dimensions, cela revient donc à résoudre un système d'équations. Et comme la force gravitationnelle n'est pas linéaire ($\vec{f_g}$ est proportionnelle à $1/position^2$) par rapport à la position, ce système est donc non-linéaire.

2.2 Résolution du problème

Nous avons tout d'abord implémenté des fonctions qui permettent de représenter les forces élastiques, centrifuges, et gravitationnelles (Toutes ces fonctions sont de complexité constante car il ne s'agit que de simples calculs à faire).
Ensuite on applique la méthode de Newton-Raphson avec les fonctions qui à un vecteur associent chaque force. Pour cela on calcule d'abord la somme des forces selon x, puis selon y qui s'appliquent au solide, puis on calcule la jacobiennne de ces résultantes de force. Nous avons ainsi crée trois fonctions qui calculent la jacobienne d'une force élastique, centrifuge, et gravitationnelle. Et finalement, grâce la méthode de Newton-Raphson, on a directement les positions d'équilibre du système.

2.3 Exemple

Nous avons testé cette fonction avec un système composé de deux solides qui sont tous les deux soumis à leur propre force gravitationnelle (le premier avec un coefficient de 1, et le deuxième un coefficient de 0.001), le système entier est aussi soumis à une force centrifuge appliquée à l'isobarycentre du système. Grâce à cet exemple, nous avons pu vérifier que nos fonctions précédentes étaient correctes car nous avions les réponses attendues pour la résultante des forces appliquées en U[1.5, 0] et sa jacobienne.

Nous n'avons malheureusement pas eu le temps de terminer cette partie et de conclure sur les points de Lagrange. Ironiquement, on dira que oui, il est probablement possible d'obtenir les points de Lagrange...

3 Équilibre électrostatique

Dans cette partie, on étudie un système constitué de 2 charges éléctrostatiques fixes et N mobiles situées entre les deux fixes. Le but est de trouver les positions d'équilibre de ce système. Pour cela, il nous faut résoudre le système linéaire $\bigtriangledown E(x_1, x_2, .., x_N) = 0$ avec

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

3.1 Analyse et description de la jacobienne

Nous allons donc tenter de trouver ces points d'équilibre. Ces derniers ne seront atteints que si l'énergie est minimale ou maximale, c'est à dire si la dérivée de E (la force électrostatique) relative à chaque composante soit zéro.

$$\bigtriangledown E(x_1, x_2, ..., x_N) = \begin{bmatrix} \frac{\partial E(x_1,...,x_N)}{\partial x_1} \\ \vdots \\ \frac{\partial E(x_1,...,x_N)}{\partial x_N} \end{bmatrix} = 0$$

Nous pouvons partir de cette expression et la simplifier pour pouvoir après calculer la Jacobienne. Ainsi, nous pouvons observer que pour chaque entrée du vecteur $\bigtriangledown E$ nous avons

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

De ceci nous terminons par en déduire les valeurs de la jacobienne suivant leur position dans la matrice. Les coefficients diagonaux de cette dernière auront donc pour valeurs celles données par l'équation (1) ci-dessous, et les coefficients extra-diagonaux auront pour valeurs celles de l'équation (2).

$$\frac{\partial^2 E}{\partial x_i^2} = \frac{1}{(x_i + 1)^2} + \frac{1}{(x_i - 1)^2} + \sum_{j=1, j\neq i}^{N} \frac{-1}{(x_i - x_j)^2} \tag{1}$$

$$\frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{1}{(x_i - x_j)^2} \tag{2}$$

Nous avons donc implémenté plusieurs fonctions dans le but de résoudre ce problème. Tout d'abord il a fallu calculer la Jacobienne (une fonction qui réalise le calcul pur de la Jacobienne ainsi qu'une sous-fonction calculant les termes diagonaux). Puis deux fonctions calculant l'une la somme dérivée de E, et l'autre la somme des logarithmes dans le calcul de E. Enfin une fonction donnant le résultat de l'évaluation complète de E. Nous avons également réalisé pour nos tests une fonction calculant le Laplacien E.

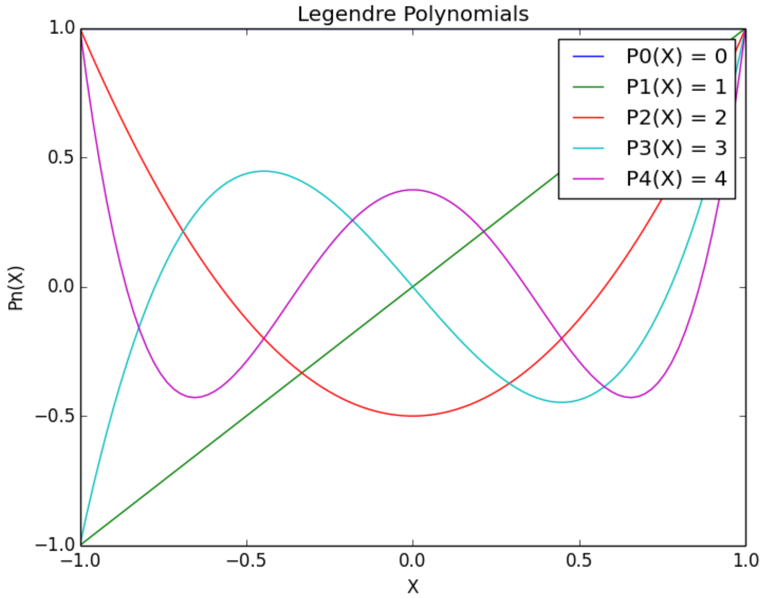
3.2 Application de Newton-Raphson et comparaison avec les polynômes de Legendre

Le but de cette partie est de résoudre le problème de l'équilibre électrostatique *via* la méthode de NewtonRaphson, et de comparer les résultats avec les racines des dérivées des polynômes de Legendre. Nous avons donc implémenté une fonction permettant de créer un graphique à partir des résultats de la résolution par le méthode de Newton-Raphson, ainsi qu'avec les données des dérivées des polynômes de Legendre.

Le fait est que malheureusement ici aussi, par manque de temps nous n'avons pas pu terminer totalement. Nous avons une fonction qui "fonctionne" en effet, mais des bugs que nous n'avons pas pu corriger à temps font que nous ne pouvons obtenir de figure concluante.

Nous avons cependant des tests avec des valeurs arbitraires qui eux semblent relativement concluants, et qui nous permettent dans une certaine mesure de "valider" les fonctions que nous avons implémentées jusqu'à présent.

L'équilibre électrostatique repose soit sur un maximum soit sur un minimum. Les résultats que nous avons pu avoir grâce à nos tests semblent montrer que les positions de l'équilibre reposent sur un maximum.



Ci-dessus le début de notre figure concernant la comparaison de nos résultats avec les polynômes de Legendre. Cependant comme dit au dessus, il manque la présence de nos résultats ...

4 Conclusion

Nous avons pu durant ce projet résoudre plusieurs problèmes qui nécessitaient la résolution de système d'équations non-linéaires en utilisant à chaque fois la méthode de Newton Raphson. Le fait est que cette méthode permet de trouver des racines, comme beaucoup d'autres. Son attrait que nou avons pu voir, résident dans sa vitesse de convergence quadratique (alors que d'autres, comme la dichotomie par exemple, convergent de manière linéaire).

Nous avons réussi tant bien que mal à passer les tests et à réaliser ces problèmes. Nous avons ainsi pu appliquer et voir quel était l'intérêt de la méthode de Newton-Raphson lors de diverses résolutions d'équations. Une certaine importance se dégage de ce genre de méthodes approchées dans la résolution de problèmes relativement complexes, du fait qu'elles sont "solutions" d'un grand éventail de problèmes.