

Projet 4 : Systèmes d'équations non-linéaires et méthode de Newton-Raphson

Groupe 2 - Equipe 4

Responsable : Baptiste Roseau

Secrétaire : Alexandre Heuillet

Codeurs : Lucien Casteres, Hadhami Ben Abdeljelil, Yann Julienne

Résumé : Ce projet consiste en la résolution de systèmes non linéaires.

La méthode ici mise en oeuvre est l'algorithme de Newton-Raphson dont les applications mettront en lumière ses atouts et ses défauts.

1 Méthode de Newton-Raphson

Il s'agit ici de mettre en oeuvre la méthode de Newton-Raphson, inspirée de la méthode de Newton dans l'optique de résoudre un système non linéaire. La méthode de Newton consiste à chercher des points de plus en plus proches de la solution en utilisant le point d'intersection de la tangente à la fonction au point actuel avec l'axe des abscisse. L'application à plusieurs dimensions de ce principe repose sur la relation suivante :

$$U_{k+1} = U_k - H(U_k)^{-1}f(U_k)$$

Où (U_k) est une suite de vecteurs colonne qui représente un point dans l'espace de définition de la fonction, f est un vecteur colonne de fonctions et H en est la matrice jacobienne.

1.1 Implémentation de Newton-Raphson générique

Dans le but d'écrire un algorithme stable numériquement et afin d'éviter le calcul de l'inverse de la matrice jacobienne, nous faisons la résolution du système suivant $J(U) * V = -F(U)$ ayant pour inconnue V .

Nous avons constaté que la vitesse de convergence change en fonction du solveur choisi. Dans l'exemple utilisé pour tester nos implémentations, en utilisant `numpy.linalg.lstsq` il a fallu 225 itérations pour converger contre 115 seulement en utilisant `numpy.solve`.

La méthode de Newton-Raphson n'aboutit pas toujours à la solution. En effet certains facteurs peuvent en causer la divergence :

- Le choix du vecteur de départ très éloigné de la solution ;
- Point d'intersection de la tangente hors du domaine de définition de la fonction voire aucun point d'intersection.

1.2 Protocole de test simple

On considère le système d'équations non linéaires suivant :

$$\begin{cases} 3x_1 - \cos(x_2 x_3) - \frac{3}{2} = 0 \\ 4x_1^2 - 625x_2^2 + 2x_3 - 1 = 0 \\ 20x_3 + e^{-x_1 x_2} + 9 = 0 \end{cases}$$

Les fonctions qui composent ce système ont la particularité d'être $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

L'algorithme permettant de fournir la solution repose sur la relation entre U_{k+1} et U_k énoncée plus haut. Deux versions ont été implémentées, l'une prend en paramètre la matrice jacobienne (*Newton_Raphson*) et l'autre la calcule en interne (*Newton_Raphson_Auto_Jacob*).

1.3 Implémentation avec Backtracking

Le backtracking est une méthode efficace permettant d'éviter la divergence de Newton-Raphson. Son principe est de revenir en arrière si le point calculé ne permet pas la convergence de l'algorithme. Pour ce faire on utilise un facteur de retour arrière λ . La condition de convergence s'exprime alors comme suit : $\|f(U + \lambda V)\| \leq \|f(U)\|$. On utilisera donc deux boucles imbriquées. L'une servira à vérifier la précision de la solution et la seconde sa convergence en divisant par 2 la valeur de λ à chaque tour de boucle.

2 Calcul des points de Lagrange

L'objectif de cette partie est d'implémenter différentes forces et de trouver leur point d'équilibre grâce à la méthode de Newton-Raphson. Ces forces sont :

- Une force élastique
- Une force gravitationnelle
- Une force centrifuge

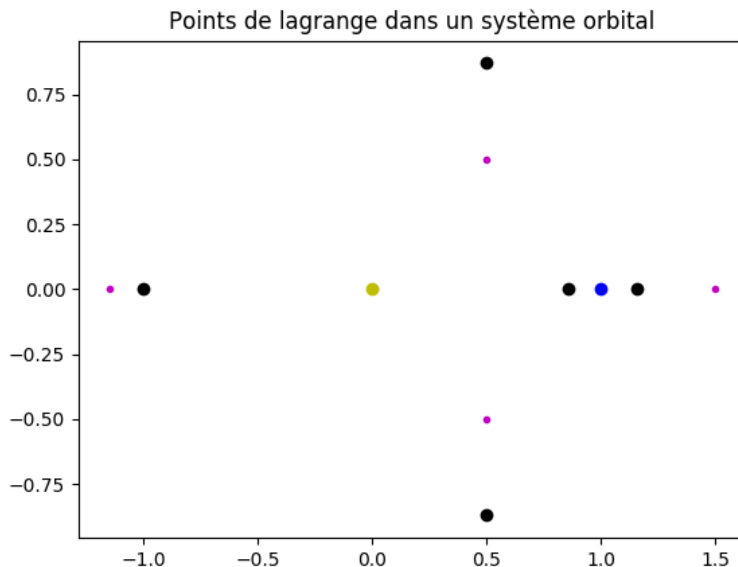
Il faut ensuite rechercher les points de Lagrange. Il s'agit de positions de l'espace où les champs de gravité de deux corps en mouvement orbital fournissent exactement la force centripète requise pour que ce point de l'espace accompagne simultanément le mouvement orbital des deux corps¹.

Il existe 5 points de Lagrange qui avec m_1 point de la masse dominante et m_2 point de la seconde masse :

- **L₁** : Appartenant $[m_1, m_2]$.
- **L₂ et L₃** : Appartenant respectivement à $[m_1, m_2)$ et $(m_2, m_1]$ et exclus de $[m_1, m_2]$.
- **L₄ et L₅** : Tel que $L_4 m_1 m_2$ soit un triangle équilatéral et L_5 soit l'image de L_4 par la symétrie d'axe (m_1, m_2) .

1. cf. Wikipedia

En conséquence, pour calculer les points de Lagrange à l'aide de la méthode de Newton_Raphson, nous avons appliqué cette dernière avec 5 points proches des points théoriques de Lagrange. Après avoir appliqué ce principe, nous avons obtenu la représentation suivante dans laquelle les points roses représentent les points de départ et les verts les solutions trouvées. Le jaune correspond à la masse m_1 et le bleu à la masse m_2 .



3 Méthode de Bairstow pour la factorisation polynomiale

Inventée par Leonard Bairstow à l'aube du XXème siècle, la méthode de Bairstow a pour objectif l'obtention des racines de polynômes réels de degrés arbitraires. Elle repose sur une itération de divisions polynomiales réalisées par l'algorithme de Newton-Raphson.

3.1 Fonction principale

Cet algorithme se base sur une fonction principale F , dont on va chercher à annuler les composantes en dimension 2. Cette dernière est celle qui, considérant le polynôme P qui a pour ensemble de définition \mathbb{R} (polynômes réels), va calculer la division euclidienne de P par un polynôme quadratique de manière à écrire le polynôme $P(x)$ sous la forme :

$$P(x) = (x^2 + Bx + C) * Q(x) + Rx + S \quad (1)$$

où B et C sont les coefficients d'un polynôme quadratique par lequel on divise P pour obtenir le polynôme quotient Q . R et S sont quant à eux les coefficients du polynôme reste d'ordre 1, ce sont donc des fonctions de B et C .

Cette fonction s'annulera lorsque R et S seront simultanément nuls ($x^2 + Bx + C$ est alors un diviseur entier de P). Cette fonction a pour domaine de définition \mathbb{R} (puisque P est un polynôme réel) et renvoie également des valeurs appartenant à \mathbb{R} , avec $|R| < |B|$. Ainsi, le but est donc de trouver des coefficients B et C tels que R et S soient simultanément nuls et donc tels que le polynôme quadratique $x^2 + Bx + C$ possède les mêmes racines que P .

3.2 Dérivées partielles

La fonction précédente peut être ainsi considérée comme une fonction à deux composantes, $R(B, C) : \mathbb{R} \rightarrow \mathbb{R}$ et $S(B, C) : \mathbb{R} \rightarrow \mathbb{R}$. Pour utiliser la méthode de Newton-Raphson, on va avoir besoin de construire la matrice jacobienne de F et donc de calculer les dérivées partielles de R et S :

$$\begin{aligned}\frac{\partial R}{\partial C} &= -R_1 \\ \frac{\partial S}{\partial C} &= -S_1 \\ \frac{\partial S}{\partial B} &= -C * \frac{\partial R}{\partial C} \\ \frac{\partial R}{\partial B} &= -B * \frac{\partial R}{\partial C} + \frac{\partial R}{\partial C}\end{aligned}$$

Où R_1 et S_1 sont les coefficients du polynôme reste issu de la division de Q par le polynôme quadratique $x^2 + Bx + C$.

3.3 Algorithme

Ainsi, on peut implémenter cet algorithme en passant la fonction décrite précédemment, de même que sa jacobienne, en arguments dans la fonction *Newton_Raphson* réalisée dans la première. Cette dernière va alors se charger d'itérer l'algorithme jusqu'à obtenir les racines avec la précision ε . Cependant, cette méthode ne permet d'obtenir que les deux premières racines des polynômes de degré $\deg > 2$.

Dans ce cas, on a pu obtenir la totalité des racines d'un polynôme P en appliquant successivement la méthode de Bairstow aux polynômes de la suite $P_{i+1} = P_i/Q_i$, pour i allant de 1 à $\deg(P)/2$ et $P_0 = P$, où Q_i représente le polynôme quadratique dont les coefficients ont été obtenus en appliquant la méthode de Bairstow à P_i . Les racines de Q_i , correspondant à un couple de racines de P , sont sauvegardées à chaque tour de boucle et sont renvoyées dans leur totalité à la fin du programme.

4 Équilibre électrostatique

Le but de cette partie est de déterminer l'équilibre électrostatique d'un système composé de deux charge fixes et de N charges mobiles entre les deux charges fixes dont l'énergie est égale à :

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

Pour trouver les points d'équilibre il faut que l'énergie du système soit maximale ou minimale. Cela revient à résoudre :

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

On pose d'abord :

$$F = \nabla E$$

On doit, pour résoudre le système, établir la matrice jacobienne de F qui est définie comme suit :

$$\begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_N} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_N}{\partial x_1} & \frac{\partial F_N}{\partial x_2} & \dots & \frac{\partial F_N}{\partial x_N} \end{pmatrix}$$

Avec :

$$\frac{\partial F_i}{\partial x_k} = \begin{cases} -\left(\frac{1}{(x_i-1)^2} + \frac{1}{(x_i+1)^2}\right) + \sum_{i \neq j} \frac{1}{(x_i-x_j)^2} s_i & i = k \\ \frac{1}{(x_i-x_k)^2} & \text{sinon} \end{cases}$$

Les fonctions *generate_F* et *generate_jacobienne* ont été implémentées puis passées en paramètre à la fonction *newton_Raphson* qui fournit ainsi la position d'équilibre.

Pour savoir si la position d'équilibre correspond à un minimum ou maximum d'énergie, on fait l'analogie avec la dimension 1 (Vérification du signe de la dérivée seconde) en évaluant la matrice jacobienne de F en la solution obtenue. Si la matrice obtenue est symétrique définie positive, il s'agit d'un minimum, sinon il s'agit d'un maximum.

5 Analyse et conclusion

En conclusion, nous avons pu observer que l'algorithme de Newton-Raphson permettait de grandement simplifier l'écriture de méthodes de calcul itératives basées sur la convergence des valeurs de fonctions réelles multidimensionnelles. Le problème se simplifie en effet au calcul de ces fonctions et à la construction de leurs jacobienes. Dans le cas de la méthode de Bairstow et des points de Lagrange notamment, il aura ainsi simplement fallu formaliser mathématiquement le problème, puis écrire un algorithme "cycle" correspondant au calcul de la fonction, des dérivées partielles de celle-ci et faire appel à Newton-Raphson pour les itérations. Cependant, malgré ses nombreuses qualités, cette méthode possède tout de même des défauts majeurs pouvant empêcher la convergence, liés au choix du vecteur de départ qui est particulièrement critique.

Finalement, Newton-Raphson apparaît comme une solution efficace et parfaitement exploitable en algorithme numérique pour résoudre des systèmes d'équations non linéaires.