INSA Département GM : TP Résolution de systèmes non-linéaires

A. Tonnoir (2018 - 2019)

Dans ce TP, on s'intéressera à l'étude de méthodes numériques de résolution d'équation non linéaire. Le TP se décompose en 3 parties indépendantes, mais je vous invite à les faire dans l'ordre (difficulté croissante). Je vous invite également à faire preuve d'esprit critique sur vos résultats et de curiosité. N'hésitez pas à tester des nouvelles choses, même non demandées dans dans le TP !

Remarque : **Pour le VENDREDI** après la séance de TP (donc le **21/12**), vous rendrez **vos CODES COMPLÉTÉS dans un dossier .ZIP et vos rapports de TP au format PDF**. Dans vos rapports, n'incluez pas vos codes !

Ordre de convergence

Dans cette première partie, nous allons nous intéresser à l'analyse des *méthodes de Newton* et de la *fausse position*. Pour ce faire, nous allons considérer le cas modèle du calcul de la racine cubique $\sqrt[3]{a}$.

1. **(Théorique)**

   a) Formuler le problème du calcul de la racine sous la forme $f(x) = 0$ en précisant f (bien sur sans utiliser la racine cubique).

   b) Rappeler l'ordre théorique pour chaque de ces méthodes

2. **(Code)**

   a) Dans le fichier *algo.py*, compléter les méthodes *newton* et *fp* (intelligemment !) implémentant respectivement les méthodes de Newton et de la fausse position.

   b) Dans le fichier *data.py*, compléter les fonctions f et *df*.

   c) Dans le fichier *main1.py*, choisissez une valeur a pour déterminer facilement la solution exacte x^*. Compléter également les ordres de théoriques des méthodes.

   d) Tester vos codes avec le programme *main1.py* et interpréter les courbes de convergence obtenues.

Le λ-mécanisme de Chebychev

On s'intéresse dans cette partie au λ-mécanisme représenté sur la figure 1. Ce mécanisme a été inventé par le mathématicien russe Pafnouti Chebychev et a été présenté à l'exposition universelle de 1878 à Paris avec la machine plantigrade. L'objectif initiale de ce mécanisme est de convertir un *mouvement de rotation* en un *mouvement rectiligne*, ce qui était particulièrement important à l'époque de la machine à vapeur. Notre objectif ici va être, connaissant les longueurs a, b, c et d et l'angle θ, déduire la position (x, y) du point $\underline{A}$.

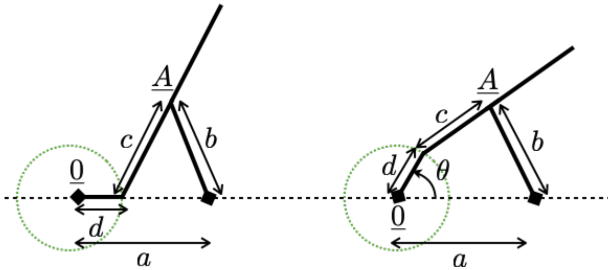

FIGURE 1 – Position du λ-mécanisme pour $\theta = 0$ (à gauche) et $\theta > 0$ (à droite)

1. **(Théorique)**

   a) Montrer que les coordonnées du point (x, y) doivent satisfaire les systèmes d'équations

   $$\begin{cases} (x - d\cos(\theta))^2 + (y - d\sin(\theta))^2 - c^2 = 0 \\ (x - a)^2 + y^2 - b^2 = 0 \end{cases}$$

   Déduire une formulation du problème sous la forme $\underline{F}(x, y) = 0$ en précisant $\underline{F}$.

   b) Ce système détermine t'il de manière unique la position (x, y) ? Dans le code *main2.py*, expliquer l'initialisation de la méthode de Newton.

2. **(Code)**

a) Dans le fichier *data.py*, compléter les fonctions $F2D$ et $dF2D$.

b) Tester votre code à l'aide du fichier *main2.py*. Le mouvement circulaire est-il transformé en un mouvement rectiligne ?

La fractale de Newton

Dans cette dernière partie, on va s'intéresser à la recherche des racines d'un polynôme $P(z) = z^n - 1$ où $z \in \mathbb{C}$ et $n \in \{1, \cdots, 10\}$. On rappelle que pour représenter informatiquement un polynôme de degré n, on utilise un tableau de taille $(n + 1)$ dont le coefficient i correspond au coefficient devant x^i dans l'écriture du polynôme.

1. **(Théorique)**

   a) Donner explicitement les racines de P.

   b) Comment initialiser la méthode de Newton pour obtenir une racine complexe ?

2. **(Code)**

   a) Dans le fichier *algo.py*, compléter la fonction *Horner* qui implemente la factorisation de Hörner et la fonction *NewtonHorner*.

   b) Tester votre code à l'aide du programme *main3.py*. Expliquer l'image obtenue et commenter.

   c) D'après vous, obtient on le même type de résultat avec la méthode de la fausse position ? Tester à l'aide d'un fichier *main4.py* que vous écrirez vous-même.

Pour conclure le TP...

Dans la seconde partie de ce TP, nous avons présenté un mécanisme permettant de convertir (approximativement) un mouvement circulaire en mouvement rectiligne. Il est intéressant de noter que dans l'histoire, de nombreux autres mécanismes ont été proposés comme le dispositif de Peaucellier - Lipkin ou encore le parallélogramme de Watt. Suivant la même démarche que pour le λ-mécanisme, vous pouvez essayer de simuler ces différents mécanismes. Aujourd'hui, on retrouve ce type de mécanisme dans diverses applications en mécanique

La dernière partie du TP porte sur la fractale de Newton. Si vous êtes curieux d'en apprendre davantage sur cette fractale, je vous invite à lire l'article de image maths (on y apprend aussi pourquoi les feuilles A4 sont au format 21×29.7). Pour approfondir le sujet et voir le lien avec la théorie de Chaos, je vous recommande le très bon livre *Théorie du Chaos* de J. Gleick.