

TP1 : Résolution numérique des systèmes d'équations non linéaires

I.1. Le but du TP

Le but de ces Travaux Pratique est d'apprendre aux étudiants en MASTER la programmation des algorithmes sous environnement MATLAB permettant la résolution des équations non linéaires $f(x) = 0$, ainsi que l'utilisation des fonctions prédéterminés existante dans l'outil MATLAB. Les méthodes de résolution à programmées dans ce TP sont la méthode de dichotomie et la méthode de Newton Raphson.

I.2. Application (Programmation avec MATLAB) :

I.2.1. Calcul des racines d'un polynôme invariant avec MATLAB

Les polynômes sont représentés en MATLAB par des vecteurs contenant leurs coefficients dans l'ordre décroissant des puissances.

✓ Exemple :

Le polynôme cubique $p(x) = x^3 - 7x + 6$ est représenté par le vecteur $[1 \ -7 \ 6]$. Sous MATLAB, on peut le définir comme un vecteur ligne ou colonne :

`p = [1 -7 6] ; % Vecteur ligne`

`p = [1;-7;6] ; % Vecteur colonne`

✓ Application :

- ☐ Ecrire un programme qui permet de calculer les racines de $p(x) = 4x^3 + 4x^2 - 2$ en utilisant la fonction **roots**.
- ☐ Vérifier l'exactitude du résultat obtenu.

I.2.2. Recherche des zéros d'une fonction réelle quelconque

La fonction MATLAB **fzero** calcule un zéro de la fonction **f** en supposant une valeur initiale de ce zéro. Pour calculer un zéro d'une fonction **f** on doit premièrement la définir sous forme d'une fonction MATLAB de la forme suivante :

function y = fl(x)
y = ... ; % Calcul de y
en fonction de x
Ensuite, on applique la
fonction **fzero** comme
suit :
r = **fzero**(f1, x0)
Où x0 est une valeur
initiale supposée d'un
zéro.

✓ **Application :**

- ☐ Calculer les racines du polynôme p de l'application précédente en utilisant **fzero**. Utiliser $x_0=0.5$ puis $x_0=1.5$.
- ☐ Conclure sur l'efficacité de cette fonction.

I.3. Méthode de Dichotomie

- ☐ En utilisant la méthode de Dichotomie, trouvez, à la main, la valeur de x pour laquelle

$$f(x) = 4e^{\frac{x}{4}} - 6 \quad \text{s'annule avec une précision à } 10^{-1} \text{ en prenant comme intervalle } [1,5, 2].$$

- ☐ Ecrire un programme de la méthode de Dichotomie pour le calcul de zéro de **f** avec une précision de 10^{-3} puis 10^{-5} en précisant le nombre d'itérations effectuées. Utiliser pour cela les instructions **while** et **if**.

I.4. Méthode de Newton-Raphson

I.4.1 Application 1

- ☐ Refaire le même travail demandé avec la méthode de dichotomie en utilisant la méthode de Newton-Raphson et en prenant comme approximation initiale de x les valeurs 0 et 1,5.
- ☐ Comparer l'efficacité des deux méthodes.

I.4.2 Application 2

Soit le système d'équations non linéaires **S** suivant:

$$S \rightarrow \begin{cases} 4x_1^2 - 6x_2 - 6 \\ 3x_2^2 + 8x_1x_2 - 8 \end{cases}$$

On définit deux fonctions **fun** et **Jac** qui calculent les valeurs du système d'équations et sa matrice Jacobienne, respectivement.

- ☐ Écrire les programmes des fonctions **fun** et **Jac** pour ce système.
 - ☐ Écrire le programme principal de la méthode de Newton-Raphson qui permet de trouver les zéros de ce système en prenant comme approximation initiale le vecteur $[1, 2]^T$ avec une précision de 10^{-3}

□ **Remarque**

:

Pour déterminer la convergence de la méthode de Newton-Raphson, on utilise ici l'erreur en termes de la norme infinie, i.e. lorsque le maximum des valeurs absolues des erreurs partielles f_k devient inférieur à une valeur prédéfinie :

$$\max |f_k| <$$