Méthode de résolution numérique

▪ Objectifs

Les différents domaines des sciences de l'ingénieur visent à modéliser la réalité physique afin d'en prédire le comportement.

Dans cette activité, il s'agit de mettre en œuvre des méthodes numériques de résolution d'équations de la forme f(x) = 0 qu'il n'est pas possible de résoudre analytiquement.

▪ Travail demandé

La démarche suivie est la suivante :

- séparation des racines : opérer une étude graphique afin d'identifier un intervalle sur lequel la fonction ne s'annule qu'une fois
- construire une suite qui converge vers la solution x_0 de l'équation f(x) = 0 sur l'intervalle d'étude

Analyse d'une représentation graphique

Supposons l'étude de la puissance fournie par un système selon la loi :

$P(u)=4\sqrt{(u)}-3u-u^4$ où u est la tension d'alimentation du système et P la puissance fournie par le système.

Q.1 Elaborer un programme python afin de représenter graphiquement l'évolution de la puissance sur l'intervalle [0V,1V]

Aide : utiliser la fonction

- sqrt() du module numpy pour calculer $\sqrt{(x)}$
- linspace() du module numpy pour générer l'intervalle des abscisses utilisé pour la représentation graphique
- plot() et show() du module matplotlib.pyplot pour afficher le tracé de courbe représentative du mouvement

Q.2 Commenter chaque ligne de code pour en expliquer le rôle

On souhaite évaluer le point de puissance maximale. Pour cela, il suffit de trouver la valeur de tension u telle que P'(u)=0.

Q.3 Calculer P'(u)

Q.4 Elaborer un programme python afin de représenter graphiquement l'évolution de la fonction P' sur l'intervalle]0,1] et son intersection avec l'axe des abscisses

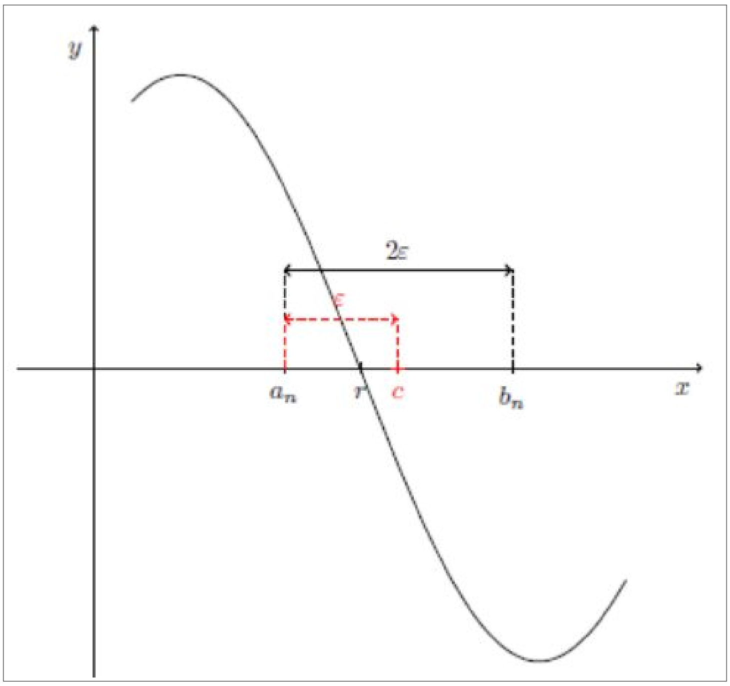
Q.5 Par lecture graphique, relever le point de fonctionnement du système à puissance maximale

Méthode de résolution numérique : la dichotomie

Hypothèse de travail : les lois physiques que nous étudions sont continues et monotones sur l'intervalle de l'étude.

Le principe est :

- de partir d'un encadrement $[a_0, b_0]$ de la valeur de x_0 (telle que $f(x_0)=0$) que l'on cherche à approcher,
- puis, par itérations successives, de réduire les bornes de l'intervalle de façon à encadrer toujours plus précisément x_0 selon la logique suivante :
  - calcul de $c_n=(a_n+b_n)/2$ (point milieu).
  - si $f(a_n)f(c_n) > 0$ alors $a_{n+1} = c_n$ et $b_{n+1} = b_n$
  - sinon $b_{n+1} = c_n$ et $a_{n+1} = a_n$
  - arrêt des itération lorque $b_{n+1} - a_{n+1} < \varepsilon$ où ε est une constante définissant la précision souhaitée.



La fonction de résolution par dichotomie respecte le prototype suivant :

```python
def dichotomie(f: Callable[[float], float], borne_inf: float, borne_sup: float, precision: float)-> tuple[float, int]:
    '''
    Calcule l'approximation x0 solution de f(x0) = 0

        Parameters:
            f (callable): la fonction réelle à variable réelle à résoudre
            borne_inf (float): borne inférieure de l'intervalle d'encadrement initial
            borne_sup (float): borne supérieure de l'intervalle d'encadrement initial
            precision (float): précision souhaitée sur la valeur recherchée

        Returns:
            milieu (float): valeur approchée de x0 solution de f(x0)=0
            nbIteration (int) : nombre d'itération dans le processus de dichotomie
    '''
```

Q.6 Ecrire un module python intégrant une fonction de résolution par dichotomie conformément au cahier des charges ci-dessus. La documentation du code intègre l'annotation des fonctions et les doctrings comme dans la capture ci-dessus.

Q.7 Vérifier le bon fonctionnement du module en créant un programme python qui évalue de point de puissance maximale du système décrit en Q.1.

Q.8 Observer les résultats obtenus pour des précisions successives de 0.1 à 10^{-10}. Proposer une justification à cette observation en cherchant à relier le nombre d'itérations à la précision souhaitée.

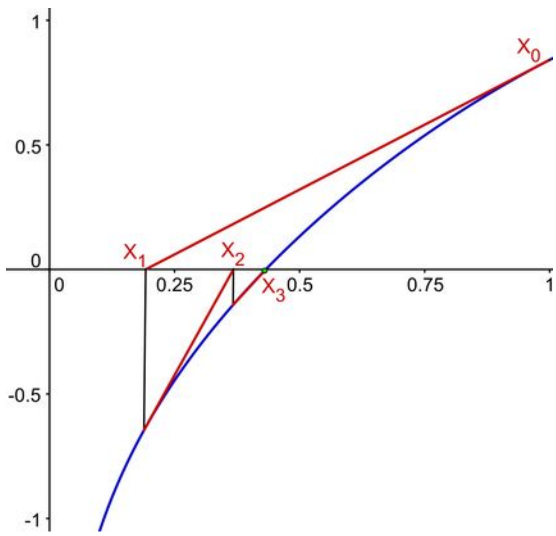
Méthode de résolution numérique : la méthode de Newton

Hypothèse de travail : les lois physiques que nous étudions sont continues et monotones sur l'intervalle de l'étude.

Une hypothèse supplémentaire est nécessaire pour la mise en œuvre de la méthode : la loi est C^1 sur l'intervalle d'étude et sa dérivée ne s'annule pas sur cet intervalle.

Le principe est :

- de choisir une première abscisse x_0 dans l'intervalle d'étude et de tracer la tangente à la courbe en x_0. Cette tangente coupe l'axe des absisses en x_1.
- À partir de l'abscisse x_1, tracer de nouveau la tangente à courbe en x_1. Cette tangente coupe l'axe des abscisses en x_2, plus proche encore du l'abscisse « r » recherchée (car f(r)=0)



- Itèrer le processus tant que $|x_{n+1}-x_n|$ > précision souhaitée

Q.9 A partir de la formule décrivant l'équation de la tangente à une courbe d'équation y=f(x), relier x_0, x_1, f et f'.

Q.10 En déduire, pour l'itération n, la relation entre x_n, x_{n+1}, f et f'

Q.11 Compléter le module python de la question Q.6 afin d'intégrer une fonction de résolution par méthode Newton conformément au cahier des charges ci-dessus. La documentation du code doit intégrer l'annotation des fonctions et les doctrings comme précédemment.

Q.12 Vérifier le bon fonctionnement du module en créant un programme python qui évalue de point de puissance maximale du système décrit en Q.1.

Q.13 Observer les résultats obtenus pour des précisions successives de 0.1 à 10^{-10}. Comparer à la résolution par dichotomie.