
TP n° 3 – Méthodes de résolution approchée d'équations

1 Code à compléter

Vous pouvez récupérer le code à compléter sur l'espace claroline de l'UE.

Vous pouvez enregistrer l'un ou l'autre des fichiers, puis les lancer avec l'application correspondante. Vous pourrez alors sauvegarder vos modifications.

2 Méthode de Newton

On rappelle que la méthode de Newton est une méthode de calcul approché des zéros d'une fonction $f: I \rightarrow \mathbb{R}$ dérivable, avec $I \subset \mathbb{R}$, et qu'elle est définie par l'algorithme suivant

$$\begin{cases} x_0 \in \mathbb{R} \text{ donné} \\ x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \text{ pour tout } k \in \mathbb{N} \end{cases}$$

Une implémentation de cette méthode est donnée ci-dessous :

```
def newton(f, df, x0, tol, nmax):
    valeurs = [x0]
    x = x0
    err = tol + 1

    niter = 0
    while err > tol and niter < nmax:
        x = x - f(x) / df(x)
        valeurs.append(x)
        err = np.abs(valeurs[niter+1] - valeurs[niter])
        niter = niter + 1

    if niter == nmax and err > tol:
        print("Arrêt, nombre maximum d'itérations atteint")

    return np.array(valeurs), niter
```

La fonction `newton` prend en arguments d'entrée une fonction `f`, sa dérivée `df`, un point initial `x0`, la tolérance souhaitée `tol` et le nombre maximal d'itérations souhaitées `nmax`. En sortie, cette fonction renvoie la suite des approximations $(x_k)_{0 \leq k \leq niter}$ dans la variable `valeurs` et le nombre d'itérations pris par l'algorithme pour atteindre la tolérance `tol` dans la variable `niter`.

2.1 Un premier exemple

On considère la fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, $x \mapsto x + e^x + \frac{10}{1+x^2} - 5$. Dans la suite, on prendra `tol=1e-8`.

1. Tracer le graphe de la fonction f sur l'intervalle $[-2, 2]$ à l'aide de la commande `plot`.
2. Calculer algébriquement la dérivée de f , et définir la fonction Python correspondante, que l'on appellera `df`.

RÉPONSE : pour tout $x \in \mathbb{R}$,

$$f'(x) =$$

3. Donner une valeur approchée du zéro $\bar{x}$ de f à l'aide de la commande `fsolve` du paquet `scipy.optimize` (qui utilise une autre méthode que la méthode de Newton) pour $x_0 = -0.1$.

RÉPONSE :

$$\bar{x} =$$

Cette valeur sera utilisée comme valeur de référence pour étudier la convergence de la méthode de Newton.

4. Écrire la fonction `newton`.
5. Tracer la suite des itérés $(x_k)_{0 \leq k \leq niter}$ calculés par la méthode de Newton.
6. On cherche à évaluer numériquement l'ordre de la méthode de Newton. À partir des quantités calculées par la fonction `newton` avec les valeurs choisies plus haut, définir un vecteur `err` contenant les erreurs $e_n = |x_n - \bar{x}|$.

Quel graphe doit-on tracer pour obtenir l'ordre de la méthode comme la pente d'une droite? Justifier.

RÉPONSE :

7. Tracer la figure mettant en évidence l'ordre de convergence de la méthode de Newton.
8. En utilisant la commande `polyfit` (cf TP2), déterminer l'ordre de convergence.

RÉPONSE :

9. Que donnent les méthodes `fsolve` et `newton` si l'on prend $x_0 = 1$? Proposer une explication.

RÉPONSE :

2.2 Un deuxième exemple

On considère la fonction $g :]0, +\infty[\rightarrow \mathbb{R}, x \mapsto (x - 1) \ln x$.

Dans la suite, on prendra `tol=1e-8`.

1. Calculer algébriquement la dérivée de g , et définir la fonction Python correspondante, que l'on appellera `dg`.

RÉPONSE : pour tout $x \in \mathbb{R}$,

$$g'(x) =$$

2. Calculer $g(1)$ et $g'(1)$.

RÉPONSE :

$$g(1) =$$

$$g'(1) =$$

3. Pour trois valeurs différentes de x_0 , tracer la suite des itérés calculés par la méthode de Newton. On tracera les trois courbes sur une même figure, avec trois couleurs différentes, et on fera apparaître une légende sur la figure.
4. On cherche à évaluer numériquement l'ordre de la méthode de Newton dans ce cas. Fixer x_0 pour lequel la méthode converge, puis, à partir des quantités calculées par la fonction `newton` dans ce cas, définir un vecteur `err` contenant les erreurs $e_n = |x_n - 1|$.
5. Tracer la figure mettant en évidence l'ordre de convergence de la méthode de Newton.
6. Déterminer l'ordre de convergence. Proposer une explication.

RÉPONSE :

3 Méthode de la sécante

On rappelle que la méthode de la sécante est une méthode de calcul approché des zéros d'une fonction $f : I \rightarrow \mathbb{R}$, avec $I \subset \mathbb{R}$, et qu'elle est définie par l'algorithme suivant

$$\begin{cases} x_0, x_1 \in \mathbb{R} \text{ donnés;} \\ \text{pour tout } k \in \mathbb{N}^*, & x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}. \end{cases}$$

Une implémentation de cette méthode est donnée ci-dessous :

```
def secante(f, x0, x1, tol, nmax):
    valeurs = [x1]
    x_prev = x0
    x = x1
    err = tol + 1

    niter = 0
    while err > tol and niter < nmax:
        x_next = x - f(x) * (x - x_prev) / (f(x) - f(x_prev))
        x_prev = x
        x = x_next
        valeurs.append(x)
        err = np.abs(valeurs[niter + 1] - valeurs[niter])
        niter = niter + 1

    if niter == nmax and err > tol:
        print("Arrêt, nombre maximum d'itérations atteint")

    return np.array(valeurs), niter
```

On considère la fonction

$$h : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto (\cos(2x))^2 - x^2.$$

On fixera dans la suite $\text{tol}=1\text{e-}8$ et $\text{nmax}=40$.

1. Définir la fonction h , puis représenter h restreinte à l'intervalle $[0, 2]$ à l'aide de la commande `plot`.
2. En utilisant la commande `fsolve` du paquet `scipy.optimize`, donner une valeur approchée du zéro $\bar{x}$ de h sur $[0, 2]$. Cette valeur sera utilisée comme valeur de référence pour étudier la convergence de la méthode de la sécante.

RÉPONSE :

$$\bar{x} =$$

3. Pour trois valeurs différentes du couple (x_0, x_1) , tracer la suite des itérés calculés par la méthode de la sécante. On tracera les trois courbes sur une même figure, avec trois couleurs différentes, et on fera apparaître une légende sur la figure.
4. On cherche à évaluer numériquement l'ordre de la méthode de la sécante. Fixer (x_0, x_1) pour lequel la méthode converge, puis, à partir des quantités calculées par la fonction `secante`, définir un vecteur `err` contenant les erreurs $e_n = |x_n - \bar{x}|$.

Quel graphe doit-on tracer pour obtenir l'ordre de la méthode comme la pente d'une droite ?

RÉPONSE :

Tracer la figure mettant en évidence l'ordre de convergence de la méthode de la sécante.

5. En utilisant la commande `polyfit` et la réponse à la question précédente, donner une valeur approchée de l'ordre de la méthode de la sécante. Cette valeur correspond-elle à la valeur théorique donnée dans le cours ?

RÉPONSE :

Ordre obtenu numériquement :

Ordre théorique :