Table des matières


```python
import numpy as np
import matplotlib.pyplot as plt
```

I Résolution approchée de l'équation $f(x) = 0$

Données et hypothèses.

1. (Séparation) On connaît un segment $[a, b]$ ne contenant qu'une seule racine $x_\star$ de f. Remarque : dans les cas généraux, l'obtention de la séparation demande une vraie étude dont on ne peut faire l'économie.
2. f est continue sur le segment $[a, b]$.

On cherche à calculer une valeur approchée de $x_\star$ à une constante $\varepsilon > 0$ donnée près, appelée précision.

Dans toute la suite, on sera amené à considérer l'équation (E) suivante sur un certain intervalle $[a, b]$:

$$(E) \qquad \cos x - x = 0 \quad \text{sur } [a, b].$$

[Q1.] Proposer graphiquement un intervalle $[a, b]$ contenant la solution $x_\star$ de (E).

I 1) Rappel et principe de la méthode par balayage

1. *Principe :* Partant du point a, on évalue par boucle les valeurs de f aux points $x_k = a + k\varepsilon$, $k \in \mathbf{N}$.
2. *Critère d'arrêt :* dès que l'on a un changement de signe de $f(x_i)$ pour deux valeurs consécutives de x_k et x_{k+1}, on en déduit que $x_\star \in [x_k, x_{k+1}]$ (pourquoi ?). Comme $x_{k+1} - x_k = \varepsilon$, tout point de cet intervalle fournit une valeur approchée de $x_\star$ au moins à la précision ε.

[Q2.]

1. Écrire une fonction `balayage(f,a,eps)` prenant en entrée une fonction `f`, des flottants `a` et `eps` et renvoyant en sortie la solution de l'équation $f(x) = 0$ à la précision `eps` calculée par balayage, ainsi que le nombre `n` d'itérations effectuées pour l'obtenir.
2. Donner le résultat avec `eps`$= 10^{-7}$ pour l'équation (E).

I 2) Rappel et prinicipe de la méthode par dichotomie

1. *Principe :* Partant du premier segment $[a, b]$, on construit, à chaque passage dans la boucle, un nouveau segment $[a_n, b_n]$ de longueur deux fois plus petite que celle du précédent, contenant la solution localisée $x_\star$ de l'équation et dont une des extrémités est soit a_{n-1}, soit b_{n-1}, l'autre étant par construction $c_n = (a_{n-1} + b_{n-1})/2$.
2. *Critère d'arrêt :* dès que la longueur du segment $[a_n, b_n]$ est de longueur $< \varepsilon$, tout point de cet intervalle fournit une valeur approchée de $x_\star$ au moins à la précision ε.

[Q3.]

1. Écrire une fonction `dicho(f,a,b,eps)` prenant en entrée une fonction `f`, des flottants `a`, `b`, et `eps`, et renvoyant en sortie la solution de l'équation $f(x) = 0$ à la précision `eps` calculée par dichotomie, ainsi que le nombre `n` d'itérations effectuées pour l'obtenir.
2. Donner le résultat avec `eps`$= 10^{-7}$ pour l'équation (E).
3. Comparer avec la méthode par balayage.

I 3) Rappel et principe de la méthode de Newton

1. f est $\mathscr{C}^2([a, b])$.
2. f' ne s'annule pas sur $[a, b]$ (f est donc strictement monotone sur I).

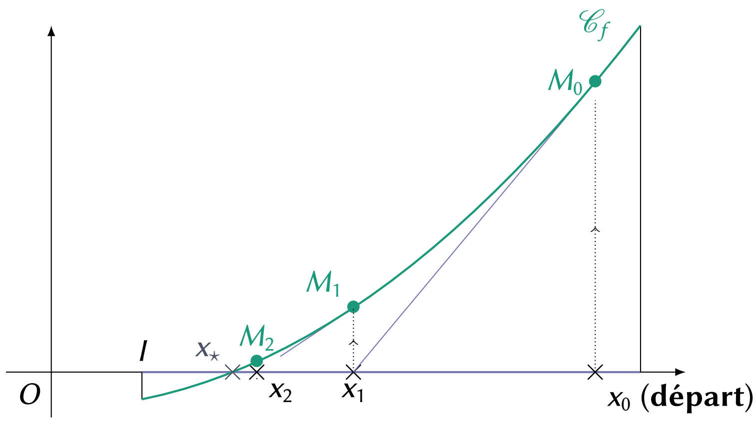
Principe : Partant d'un point $x_0 \in [a, b]$, on considère $M_0(x_0, f(x_0)) \in \mathscr{C}_f$, puis, de proche en proche, on construit des points M_n d'abscisse x_n tels que

$$\forall n \in \mathbf{N} \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

[Q4.] Mettons que l'on ait calculé le terme de rang k, x_k, de la suite de Newton $(x_n)_{n\geq 0}$. Rappeler l'équation de la tangente à $\mathscr{C}_f$ au point d'abscisse x_k.

Ainsi x_{k+1} s'obtient comme l'abscisse du point d'intersection de la tangente à $\mathscr{C}_f$ au point $M_k(x_k, f(x_k))$ avec l'axe des abscisses.

Lycée Chateaubriand, Rennes
Classe de Biospé 2023-2024
MYP



Critère d'arrêt : on adopte comme critère d'arrêt le suivant : si deux termes consécutifs x_n et x_{n+1} de la suite diffèrent de moins de ε, alors on considère que x_{n+1} est une valeur approchée de $x_\star$ à ε près.

[Q5.]

1. Écrire une fonction prenant en entrée un flottant `eps` et renvoyant en sortie une valeur approchée à `eps` près calculée par la méthode de Newton de l'équation (E) définie en **[Q.13]** , ainsi que le nombre n d'itérations pour l'obtenir.
2. Comparer le résultat pour `eps` $= 10^{-7}$ à celui obtenu par la méthode de dichotomie.

[Q6.] Soit $g : x \mapsto x^3 - x - 1$ sur $I = [0, 3]$.

1. Appliquer 3 fois la méthode de Newton à g en prenant pour point initial $x_0 = 1$, puis $x_0 = 0,6$, et enfin $x_0 = 0,57$ pour obtenir une valeur approchée à 10^{-6} près de la solution dans I à l'équation $g(x) = 0$ en précisant le nombre d'itérations réalisées dans chaque cas (penser à programmer une boucle).
2. Comment expliquer que la méthode soit si sensible au choix de la condition initiale x_0 ?

II Un peu de mathématiques

Soit x un nombre entier, et $\mathrm{ND}(x)$ le nombre de chiffres dans l'écriture (en base 10) de x. Par exemple : $\mathrm{ND}(1) = 1$, $\mathrm{ND}(87) = 2$, *etc.*

[Q7.] Rappeler la définition de la partie entière d'un réel x, notée $\lfloor x \rfloor$.

[Q8.]

1. Soit $n \in \mathbf{N}^\star$ fixé. Donner deux réels α_n et β_n tels que l'on ait :

$$\mathrm{ND}(x) = n \Leftrightarrow \alpha_n \leq x < \beta_n$$

2. On rappelle que le logarithme décimal d'un réel u, noté $\log u$, est, par définition : $\log u = \dfrac{\ln u}{\ln 10}$. Déduire de ce qui précède une expression à l'aide des fonctions usuelles de la fonction ND. Dorénavant, à l'aide de cette expression, on pourra considérer la fonction ND définie sur $\mathbf{R}$.

[Q9.] Programmer la fonction `ndigit(x)` qui prend en entrée un flottant x supposé strictement positif et qui renvoie en sortie l'entier $\mathrm{ND}(x)$.

[Q10.] Expliquer à l'aide de la fonction ND en quoi le logarithme est une fonction à croissance lente.

III Croissances comparées, équivalents

III 1) Échelle en $+\infty$

On considère $x \in \mathbf{R}$, x destiné à tendre vers $+\infty$. On rappelle que :

1. Toute fonction f telle que $\lim\limits_{x \to +\infty} |f(x)| = +\infty$ s'appelle *infiniment grand* au voisinage de $+\infty$.
2. Toute fonction f telle que $\lim\limits_{x \to +\infty} |f(x)| = 0$ s'appelle *infiniment petit* au voisinage de $+\infty$.

[Q11.] Rappeler la définition de $f(x) = o(g(x)) \quad x \to +\infty$ pour deux fonctions f et g à valeurs réelles définies au voisinage de $+\infty$, ainsi que l'échelle de croissance en $+\infty$.

Dans ce qui suit, on travaillera avec des infiniment grands pour se fixer les idées, mais les résultats mis en évidence seraient également valables en analysant des infiniment petits.

[Q12.] Définir en python les fonctions `id`, `ln`, `sq` renvoyant (pour un flottant x en entrée donné) respectivement les flottants x, x^2, ln x.

[Q13.] Recopier et expliquer ce que fait le script suivant :

```python
 print("{0:5}|{1:22}|{2:22}|{3:22}|{4:22}".format("x=",
                                              "ln x",
                                               "sqrt x",
                                               "x**2",
                                               "exp x"))
for x in range(1,50):
    print("{0:5}|{1:22}|{2:22}|{3:22}|{4:22}".format(id(x),
                                                    int(ln(x)),
                                                    int(np.sqrt(x)),
                                                    int(sq(x)),
                                                    int(np.exp(x))))
```

III 2) Équivalents

[Q14.] Soit f et g deux infiniments grands. Rappeler la définition de $f(x) \underset{x \to +\infty}{\sim} g(x)$.

La notion d'équivalent permet de donner un ordre de grandeur. Intuitivement, on peut penser que f et g sont équivalents signifie que les infiniment grands f et g ont, pour x assez grand [1], le même premier chiffre significatif (et le même ordre de grandeur bien entendu, c'est-à-dire le nombre de chiffres).

[Q15.] Écrire une fonction `first_digit(x)` qui prend en entrée un flottant x et qui renvoie en sortie l'entier égal à la valeur du premier chiffre de l'écriture (en base 10) de x. Par exemple : `first_digit(87431.1943)` devrait renvoyer 8. Il sera commode de convertir d'abord x en chaîne de caractères.

[Q16.] Écrire une fonction `synthese(x)` qui prend en entrée un flottant x et renvoie en sortie le tuple $\mathtt{n}, \mathtt{p} = (\mathtt{first\_digit}(\mathtt{x}), \mathtt{ND}(\mathtt{x}))$

Intuitivement, si $\mathtt{synthese}(\mathtt{x})$ renvoie $(\mathtt{n}, \mathtt{p})$, cela signifie que x est de l'ordre de $\mathtt{n} \times 10^{\mathtt{p}-1}$

[Q17.] Recopier et dire ce que fait le script suivant :

```python
def equivalents(f,g,N=13):
    for p in range(1,N):
        x = 10**p
        print('x =',x)
        print(synthese(f(x)))
        print(synthese(g(x)))
        print("diff")
        print(f(x)-g(x))
        print('\n')
```

Puis dans la console :

```
console
In [1]: equivalents(f1,g1)
```

où vous aurez au préalable défini en python $f_1(x) = x^3 + 2x^2$ et $g_1(x) = x^3$.

[Q18.] Est-ce que $f(x) \underset{+\infty}{\sim} g(x) \Rightarrow f(x) - g(x) \to 0$?

1. C'est ce que signifie asymptotiquement

[Q19.] Utiliser la fonction `equivalents` pour les couples de d'infiniment grands $(f(x), g(x))$ qui suivent afin de dire si ces dernières sont équivalentes ou non en $+\infty$:

1. $(\ln x, \sqrt{x})$.
2. $(x^3 + 2x, x^3 + x)$.
3. $(x, 2x)$. . Peut-on sommer les équivalents ?
4. $\left(\exp\left(x + \sqrt{x}\right), \exp x\right)$ (appliquer la fonction `equivalents` avec N=3) .Peut-on composer les équivalents ?
5. $(\sqrt{x^4 + x^3}, x^2)$ (appliquer la fonction `equivalents` avec N=5). Calculer mathématiquement l'équivalent.
6. $(\ln(1 + e^x), x)$.