TP09 : Méthodes numériques

L'objectif de ce TP est d'étudier diverses méthodes dites de calcul numérique, permettant de trouver des valeurs approchées de solutions d'équations que l'on ne sait pas résoudre de manière exacte algébriquement.

Nous étudierons quelques unes des méthodes les plus couramment utilisées, parmi lesquelles on peut citer :

- la recherche par balayage
- la recherche par dichotomie
- ou plus généralement l'utilisation de suites adjacentes
- la recherche de point fixe à l'aide de suites récurrentes
- dont le cas particulier de la méthode de Newton

Contexte général Dans tout ce TP, on suppose que l'on dispose d'une fonction $f : I \longrightarrow \mathbb{R}$ définie et **continue** sur un intervalle $I \subset \mathbb{R}$ et de $a, b \in I$ tels que $a < b$ et $f(a)f(b) < 0$.

Dans cette situation, on sait d'après le théorème des valeurs intermédiaires que l'équation $f(x) = 0$ admet au moins une solution $\alpha \in I$ et on souhaite développer des méthodes algorithmiques permettant d'obtenir une valeur approchée de α.

Si de plus la fonction f est strictement monotone sur I, on sait également que f est injective sur I et donc que la solution α de l'équation $f(x) = 0$ est nécessairement unique dans ce cas.

1 Recherche par balayage

La méthode de recherche par balayage est une méthode naïve consistant à choisir un pas $p > 0$ et à évaluer les valeurs prises par la fonction f en les points de la suite $(a + np)_{n\in\mathbb{N}^*}$, jusqu'à trouver une valeur de signe contraire à celui de $f(a)$. Avec cette méthode, les deux dernières valeurs calculées ne sont pas de même signe, donc les abscisses correspondantes donnent un encadrement de la valeur de α dans un intervalle de longueur p.

Plus précisément, en notant $N = \min\{n \in \mathbb{N}^*, f(a)f(a + np) \leq 0\}$, on a nécessairement l'encadrement :

$$a + (N - 1)p \leq \alpha \leq a + Np$$

Pour implémenter la méthode de balayage, on peut donc suivre la stratégie suivante :

- choisir un pas $p > 0$
- initialiser $n = 1$
- tant que $f(a + np)$ et $f(a)$ ont le même signe, incrémenter la valeur de n

En fin de boucle, les valeurs $a+(n-1)p$ et $a+np$ fournissent un encadrement de α et la moyenne $a+\left(n-\frac{1}{2}\right)p$ de ces deux valeurs fournit une approximation numérique de la valeur de α.

Exemple Compléter le code ci-dessous pour qu'il affiche une valeur approchée β de la solution $\alpha \in \mathbb{R}_+$ de l'équation $e^{-\frac{x}{100}} = \dfrac{x}{100}$ satisfaisant la condition $|\beta - \alpha| \leq 10^{-3}$.

```
from math import exp

def f(x):
    return ..................   # Choix de la fonction

a = 0                           # Valeur initiale
A = f(a)
p = ......                      # Choix du pas
n = 1
while .............:            # Condition d'arrêt
    n = ...                     # Incrémentation
print(a+(n-1/2)*p)
```

Bien entendu, le choix effectué à la première étape de la valeur du pas p est déterminant pour cette méthode, les avantages et inconvénients pouvant être résumés simplement dans le tableau ci-dessous :

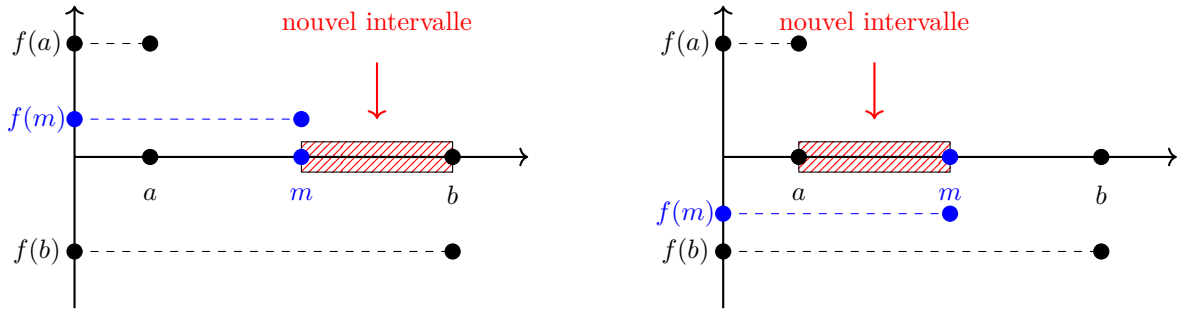
Valeur de p	Avantages	Inconvénients
Très proche de 0	Résultat précis	Temps de calcul élevé
Très éloignée de 0	Résultat imprécis	Temps de calcul faible

2 Recherche par dichotomie

La méthode de recherche dichotomique est une méthode plus évoluée dont le but est de diviser par deux la longueur de l'intervalle de recherche à chaque itération, en utilisant à chaque étape la valeur prise par la fonction au milieu de l'intervalle considéré.

Pour diviser par deux la longueur de l'intervalle de recherche, le principe est le suivant :

- à partir d'un intervalle de recherche $[a,b]$ pour lequel $f(a)f(b) \leq 0$
- on définit le milieu de l'intervalle $m = \dfrac{a+b}{2}$
- on compare le signe de $f(m)$ avec ceux de $f(a)$ et $f(b)$
- si $f(a)$ et $f(m)$ sont du même signe strict, alors f s'annule sur $[m,b]$
- si $f(a)$ et $f(m)$ ne sont pas du même signe strict, alors f s'annule sur $[a,m]$



Pour implémenter la méthode de dichotomie, on peut donc suivre la stratégie suivante :

- initialiser a et b
- choisir une distance $d > 0$
- tant que la distance entre a et b dépasse d
- définir $m = \dfrac{a+b}{2}$
- si $f(a)f(m) > 0$, remplacer a par m, sinon remplacer b par m

En fin de boucle, les valeurs a et b fournissent un encadrement de α satisfaisant la condition $|b-a| \leq d$ et la moyenne $\dfrac{a+b}{2}$ fournit une approximation numérique de α.

Exemple Compléter le code ci-dessous pour qu'il affiche une valeur approchée β de la solution $\alpha \in [0,100]$ de l'équation $e^{-\frac{x}{100}} = \dfrac{x}{100}$ satisfaisant la condition $|\beta - \alpha| \leq 10^{-3}$.

```
from math import exp

def f(x):
    return ..................   # Choix de la fonction

a, b = 0, 100                   # Intervalle de départ
d = ......                      # Choix de la précision
while .............:           # Condition d'arrêt
    m = .......                 # Définition du milieu
    if .............:           # Test
        . = m                   # Actualisation
    else:
        . = m                   # Actualisation
print((a+b)/2)
```

Remarque 2.1 La méthode de recherche dichotomique nécessite en pratique de savoir initialiser la recherche avec un intervalle $[a,b]$ pour lequel $f(a)f(b) \leq 0$. En pratique, il est courant, lorsque l'on n'a aucune idée de comment choisir une telle initialisation, de recourir à une méthode de balayage (avec un grand pas pour avoir un résultat rapidement) permettant de trouver un premier intervalle sur lequel la fonction considérée s'annule. Ensuite, on passe à la recherche dichotomique, qui est bien plus efficace que la recherche par balayage lorsque l'on cherche une valeur approchée de α avec une bonne précision.

3 Recherche dichotomique : approche théorique

La méthode pratique de recherche dichotomique présentée dans la section précédente peut être justifiée sur le plan théorique grâce au théorème de convergence des suites adjacentes. Observons que la méthode algorithmique revient à définir deux suites réelles $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ telle que $a_0 = a$ et $b_0 = b$ et pour tout $n \in \mathbb{N}$:

$$a_{n+1} = \begin{cases} \dfrac{a_n+b_n}{2} & \text{si } f(a_n)f\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ a_n & \text{sinon} \end{cases}$$

$$b_{n+1} = \begin{cases} b_n & \text{si } f(a_n)f\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ \dfrac{a_n+b_n}{2} & \text{sinon} \end{cases}$$

Observation fondamentale : Pour tout $n \in \mathbb{N}$, $\dfrac{a_n+b_n}{2} - a_n = b_n - \dfrac{a_n+b_n}{2} = \dfrac{b_n-a_n}{2}$.

Donc pour tout $n \in \mathbb{N}$, on a l'égalité : $b_{n+1} - a_{n+1} = \dfrac{b_n-a_n}{2}$.

Montrons par récurrence que pour tout $n \in \mathbb{N}$, $b_n - a_n = \dfrac{b-a}{2^n}$.

Tout d'abord, $b_0 - a_0 = b - a = \dfrac{b-a}{2^0}$.

Soit $n \in \mathbb{N}$. Supposons que $b_n - a_n = \dfrac{b-a}{2^n}$. D'après l'observation fondamentale $b_{n+1} - a_{n+1} = \dfrac{b_n-a_n}{2}$.

Donc par hypothèse de récurrence, $b_{n+1} - a_{n+1} = \dfrac{\frac{b-a}{2^n}}{2} = \dfrac{b-a}{2^{n+1}}$.

Donc d'après le principe de récurrence, pour tout $n \in \mathbb{N}$, $b_n - a_n = \dfrac{b-a}{2^n}$.

En particulier, pour tout $n \in \mathbb{N}$, $b_n - a_n > 0$ car $b - a > 0$.

Déduisons-en que les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ sont adjacentes.

Pour tout $n \in \mathbb{N}$, $a_{n+1} - a_n = \begin{cases} \dfrac{b_n-a_n}{2} & \text{si } f(a_n)f\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ 0 & \text{sinon} \end{cases}$ donc pour tout $n \in \mathbb{N}$, $a_{n+1} - a_n \geq 0$.

Pour tout $n \in \mathbb{N}$, $b_{n+1} - b_n = \begin{cases} 0 & \text{si } f(a_n)f\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ \dfrac{a_n-b_n}{2} & \text{sinon} \end{cases}$ donc pour tout $n \in \mathbb{N}$, $b_{n+1} - b_n \leq 0$.

Donc la suite $(a_n)_{n\in\mathbb{N}}$ est croissante et la suite $(b_n)_{n\in\mathbb{N}}$ est décroissante.

Enfin, pour tout $n \in \mathbb{N}$, $b_n - a_n = \dfrac{b-a}{2^n}$. Or $\lim\limits_{n\to+\infty} \dfrac{b-a}{2^n} = 0$, donc $\lim\limits_{n\to+\infty} b_n - a_n = 0$.

Donc les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ sont adjacentes. Donc par théorème de convergence des suites adjacentes, les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ convergent vers une limite commune $\ell \in \mathbb{R}$ et pour tout $n \in \mathbb{N}$, $a_n \leq \ell \leq b_n$.

Montrons alors par récurrence que pour tout $n \in \mathbb{N}$, $f(a_n)f(b_n) \leq 0$.

Par hypothèse, $f(a_0)f(b_0) \leq 0$ car $a_0 = a$ et $b_0 = b$ et on a supposé que $f(a)f(b) \leq 0$.

Soit $n \in \mathbb{N}$. Supposons que $f(a_n)f(b_n) \leq 0$ et posons $m = \dfrac{a_n+b_n}{2}$.

Si $f(a_n)f(m) > 0$, alors $f(a_{n+1})f(b_{n+1}) = f(m)f(b_n) = \dfrac{(f(a_n)f(m))(f(a_n)f(b_n))}{(f(a_n))^2} \leq 0$.

Si $f(a_n)f(m) \leq 0$, alors $f(a_{n+1})f(b_{n+1}) = f(m)f(a_n) \leq 0$.

Donc dans tous les cas, $f(a_{n+1})f(b_{n+1}) \leq 0$.

Donc d'après le principe de récurrence, pour tout $n \in \mathbb{N}$, $f(a_n)f(b_n) \leq 0$.

Supposons enfin que l'équation $f(x) = 0$ admet pour **unique** solution α dans l'intervalle $[a, b]$.

Alors d'après le théorème des valeurs intermédiaires, pour tout $n \in \mathbb{N}$, $a_n \leq \alpha \leq b_n$.

Donc par théorème de passage à la limite des inégalités, $\ell \leq \alpha \leq \ell$, donc $\ell = \alpha$.

Donc les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ convergent bien vers la solution α de l'équation $f(x) = 0$.

4 Démonstration du théorème des valeurs intermédiaires

La démarche dichotomique présentée ci-dessus est très utile pour des résolutions approchées pratiques d'équations qui résistent à des approches algébriques exactes, mais son utilité théorique va bien au-delà : ce que dit cette méthode sur la nature profonde des nombres réels peut servir à démontrer des énoncés d'analyse tout à fait non triviaux, comme par exemple le théorème des valeurs intermédiaires, dont on rappelle l'énoncé ci-dessous.

Proposition 4.1 : Théorème des valeurs intermédiaires

Soient $a, b \in \mathbb{R}$ tels que $a < b$ et $f : [a,b] \longrightarrow \mathbb{R}$ une fonction définie et **continue** sur le segment $[a,b]$.
Si $f(a) \leq f(b)$, alors $[f(a), f(b)] \subset f([a,b])$ et si $f(a) \geq f(b)$, alors $[f(b), f(a)] \subset f([a,b])$.

Démonstration : Il suffit de traiter le cas $f(a) \leq f(b)$ (l'autre cas s'en déduisant en remplaçant f par $-f$).

Soit $y \in [f(a), f(b)]$. On souhaite montrer qu'il existe $x \in [a,b]$ tel que $f(x) = y$.

Posons $\begin{array}{rccc} g : & [a,b] & \longrightarrow & \mathbb{R} \\ & x & \longmapsto & f(x) - y \end{array}$.

On souhaite démontrer qu'il existe $x \in [a,b]$ tel que $g(x) = 0$.

Observons que $g(a) = f(a) - y \leq 0$ et $g(b) = f(b) - y \geq 0$ car $y \in [f(a), f(b)]$.

Donc $g(a)g(b) \leq 0$.

Considérons les deux suites réelles $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ telles que $a_0 = a$, $b_0 = b$ et pour tout $n \in \mathbb{N}$:

$$a_{n+1} = \begin{cases} \dfrac{a_n + b_n}{2} & \text{si } g(a_n)g\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ a_n & \text{sinon} \end{cases}$$

$$b_{n+1} = \begin{cases} b_n & \text{si } g(a_n)g\left(\dfrac{a_n+b_n}{2}\right) > 0 \\ \dfrac{a_n + b_n}{2} & \text{sinon} \end{cases}$$

On peut alors démontrer **exactement** comme dans la section précédente que :

- pour tout $n \in \mathbb{N}$, $b_{n+1} - a_{n+1} = \dfrac{b_n - a_n}{2}$
- pour tout $n \in \mathbb{N}$, $b_n - a_n = \dfrac{b-a}{2^n}$
- la suite $(a_n)_{n\in\mathbb{N}}$ est croissante
- la suite $(b_n)_{n\in\mathbb{N}}$ est décroissante
- $\lim\limits_{n\to+\infty} b_n - a_n = 0$
- les suites adjacentes $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ convergent vers une limite commune $\ell \in \mathbb{R}$
- pour tout $n \in \mathbb{N}$, $a_n \leq \ell \leq b_n$
- pour tout $n \in \mathbb{N}$, $g(a_n)g(b_n) \leq 0$

En particulier, $a_0 \leq \ell \leq b_0$, donc $\ell \in [a,b]$.

Or par hypothèse, f est continue sur $[a,b]$.

Donc g est continue sur $[a,b]$, comme somme de deux fonctions continues sur $[a,b]$ (l'une étant constante).

En particulier, g est continue en ℓ.

Donc par composition de limites, $\lim\limits_{n\to+\infty} g(a_n) = \lim\limits_{n\to+\infty} g(b_n) = g(\ell)$.

Donc par produit de limites, $\lim\limits_{n\to+\infty} g(a_n)g(b_n) = (g(\ell))^2$.

Or pour tout $n \in \mathbb{N}$, $g(a_n)g(b_n) \leq 0$.

Donc par théorème de passage à la limite des inégalités, $\lim\limits_{n\to+\infty} g(a_n)g(b_n) \leq 0$.

Donc nécessairement, $(g(\ell))^2 \leq 0$. Donc $g(\ell) = 0$.

D'où le résultat.

5 Méthode de point fixe

Toute équation de la forme $f(x) = 0$ peut-être mise sous une forme alternative $g(x) = x$, dite "équation de point fixe". L'intérêt d'une telle remarque réside dans le fait que dans de nombreux cas, une solution d'une équation de point fixe peut être obtenue comme limite d'une suite récurrente. On présente ci-dessous la démarche générale ainsi que les conditions à réunir pour qu'elle puisse être mise en œuvre.

Posons, pour tout $x \in [a, b]$, $g(x) = f(x) + x$, de sorte à ce que pour tout $x \in [a, b]$, $f(x) = 0 \iff g(x) = x$.

Observons que comme f est continue sur $[a, b]$, g est également continue sur $[a, b]$.

Supposons que l'intervalle $[a, b]$ est stable par g, c'est-à-dire que pour tout $x \in [a, b]$, $g(x) \in [a, b]$.

Considérons alors une suite réelle $(u_n)_{n\in\mathbb{N}}$ telle que $u_0 \in [a, b]$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = g(u_n)$.

L'hypothèse ci-dessus garantit que $(u_n)_{n\in\mathbb{N}}$ est correctement définie et que pour tout $n \in \mathbb{N}$, $u_n \in [a, b]$.

Supposons que la suite $(u_n)_{n\in\mathbb{N}}$ converge vers une limite $\ell \in \mathbb{R}$.

Alors nécessairement $\lim\limits_{n\to+\infty} u_{n+1} = \ell$.

Or pour tout $n \in \mathbb{N}$, $a \leq u_n \leq b$, donc par théorème de passage à la limite des inégalités, $a \leq \ell \leq b$.

Or g est continue sur $[a, b]$, donc en particulier, g est continue en ℓ.

Donc par composition de limites, $\lim\limits_{n\to+\infty} u_{n+1} = \lim\limits_{n\to+\infty} g(u_n) = g(\ell)$.

Donc par théorème d'unicité de la limite, $\ell = g(\ell)$. Autrement dit, ℓ est un point fixe de g.

Ainsi, sous ces conditions, la suite $(u_n)_{n\in\mathbb{N}}$ converge vers une solution dans $[a, b]$ de l'équation $f(x) = 0$.

En supposant que l'équation $f(x) = 0$ admet une **unique** solution $\alpha \in [a, b]$, on a donc $\lim\limits_{n\to+\infty} u_n = \alpha$.

Remarque Si la suite $(u_n)_{n\in\mathbb{N}}$ est monotone, alors sachant qu'elle est bornée (à valeurs dans $[a, b]$), le théorème de la limite monotone garantit que l'hypothèse de convergence de la suite $(u_n)_{n\in\mathbb{N}}$ ci-dessus sera vérifiée. Cependant, cette condition suffisante de monotonie n'est pas nécessaire, il existe des exemples de suites récurrentes non-monotones qui sont convergentes.

Pour implémenter la méthode de point fixe, on peut donc suivre la stratégie suivante :

- Trouver un intervalle $[a, b]$ stable par g
- Initialiser $u_0 \in [a, b]$
- Montrer que la suite $(u_n)_{n\in\mathbb{N}}$ converge
- Choisir un entier naturel N, a priori grand
- Calculer la valeur de u_N à l'aide d'une boucle

La valeur obtenue en fin de boucle sera une valeur approchée de la limite α qui est aussi solution de l'équation $f(x) = 0$. Un défaut apparent de cette méthode est qu'en l'absence d'informations plus précises, le choix de l'entier N est tout à fait arbitraire et il est envisageable, si la valeur choisie n'est pas assez grande, que la valeur approchée obtenue soit très mauvaise.

Exemple Compléter le code ci-dessous pour qu'il affiche une valeur approchée β de la solution $\alpha \in [2, 3]$ de l'équation $1 - x + \sqrt{x} = 0$ (NE PAS tourner la page avant d'avoir terminé de traiter cet exemple).

On justifiera que toutes les hypothèses de la démarche de point fixe sont satisfaites.

```
from math import sqrt

def g(x):
    return ..................   # Choix de la fonction

u = ...                         # Choix de la valeur initiale
N = ...                         # Choix du nombre d'itérations
for k in range(1,N+1):
    u = ....                    # Actualisation
print(u)
```

Déterminer ensuite la valeur exacte de la solution $\alpha \in [2, 3]$ de l'équation $1 - x + \sqrt{x} = 0$, puis comparer avec la valeur approchée affichée par le code ci-dessus.

6 Utilisation d'encadrements

La méthode de point fixe présentée dans la section précédente peut être raffinée dans certains cas par l'utilisation d'encadrements (obtenus à la main ou par théorème des suites adjacentes) permettant à la fois de garantir la convergence de la suite $(u_n)_{n\in\mathbb{N}}$ étudiée (ce qui est fort pratique quand elle n'est pas monotone) et également de choisir un peu moins au hasard le nombre d'itérations N en fonction de la précision souhaitée.

Exemple Reprenons l'exemple de l'équation $1-x+\sqrt{x}=0$ étudiée dans la section précédente.

Cette équation admet une unique solution $\alpha \in [2,3]$ et une résolution algébrique directe donne $\alpha = \dfrac{3+\sqrt{5}}{2}$.

L'équivalence $1-x+\sqrt{x}=0 \iff 1+\sqrt{x}=x$ conduit à poser, pour tout $x\in[2,3]$, $g(x)=1+\sqrt{x}$ pour utiliser la méthode de point fixe. On peut alors vérifier facilement que pour tout $x\in[2,3]$, $g(x)\in[2,3]$.

On considère alors une suite $(u_n)_{n\in\mathbb{N}}$ telle que $u_0\in[2,3]$ et pour tout $n\in\mathbb{N}$, $u_{n+1}=g(u_n)$.

Par définition, $g(\alpha)=\alpha$, donc on peut observer que pour tout $n\in\mathbb{N}$:

$$|u_{n+1}-\alpha| = |g(u_n)-g(\alpha)| = |(1+\sqrt{u_n})-(1+\sqrt{\alpha})| = |\sqrt{u_n}-\sqrt{\alpha}| = \frac{|u_n-\alpha|}{\sqrt{u_n}+\sqrt{\alpha}}$$

De plus, on sait que $\alpha\geq 2$, donc $\sqrt{\alpha}\geq\sqrt{2}$ et pour tout $n\in\mathbb{N}$, $u_n\geq 2$ donc $\sqrt{u_n}\geq\sqrt{2}$.

Donc pour tout $n\in\mathbb{N}$, $\sqrt{u_n}+\sqrt{\alpha}\geq 2\sqrt{2}$. Donc pour tout $n\in\mathbb{N}$, $|u_{n+1}-\alpha| = \dfrac{|u_n-\alpha|}{\sqrt{u_n}+\sqrt{\alpha}} \leq \dfrac{|u_n-\alpha|}{2\sqrt{2}}$.

On en déduit par récurrence que pour tout $n\in\mathbb{N}$, $|u_n-\alpha| \leq \dfrac{|u_0-\alpha|}{(2\sqrt{2})^n} \leq \left(\dfrac{1}{2\sqrt{2}}\right)^n$ car $|u_0-\alpha|\leq 1$.

Une telle inégalité fournit d'une part un encadrement permettant de prouver que la suite $(u_n)_{n\in\mathbb{N}}$ converge vers α et d'autre part de contrôler la vitesse de convergence : la distance entre les termes de la suite $(u_n)_{n\in\mathbb{N}}$ et sa limite est majorée par les termes de la suite géométrique $(q^n)_{n\in\mathbb{N}}$ avec $q=\dfrac{1}{2\sqrt{2}}$, qui converge très vite vers 0 (à vitesse exponentielle), ce qui permet de savoir qu'il ne sera sans doute pas nécessaire de calculer beaucoup de termes de la suite $(u_n)_{n\in\mathbb{N}}$ pour obtenir une bonne approximation de α.

Compléter le code ci-dessous pour qu'il affiche une valeur approchée β de la solution $\alpha\in[2,3]$ de l'équation $1-x+\sqrt{x}=0$ satisfaisant la condition $|\beta-\alpha|\leq 10^{-10}$.

```
from math import sqrt

def g(x):
    return ..................    # Choix de la fonction

p = 1/10**10                     # Choix de la précision
u = ...                          # Choix de la valeur initiale
n = 0
while ..................:        # Condition d'arrêt
    u = ....
    n = ....                     # Actualisation
print(u)
```

Exercice 1 (*a*) Montrer que l'équation $\sqrt{1+x}-x=0$ admet une unique solution dans $[1,2]$, notée φ et déterminer la valeur exacte de φ.

(*b*) Montrer que pour tout $x\in[1,2]$, $|\sqrt{1+x}-\varphi| \leq \dfrac{|x-\varphi|}{2}$.

(*c*) En déduire une mise en œuvre de la méthode de point fixe permettant d'écrire un code Python affichant une valeur approchée ψ de la solution $\varphi\in[1,2]$ de l'équation $\sqrt{1+x}-x=0$ satisfaisant la condition $|\psi-\varphi|\leq 10^{-10}$.

Exercice 2 (*a*) Montrer que l'équation $e^{-x}-x=0$ admet une unique solution $\alpha\in[0,1]$.

(*b*) On considère une suite réelle $(u_n)_{n\in\mathbb{N}}$ telle que $u_0=0$ et pour tout $n\in\mathbb{N}$, $u_{n+1}=e^{-u_n}$.
Montrer que la suite $(u_n)_{n\in\mathbb{N}}$ est correctement définie et que pour tout $n\in\mathbb{N}$, $u_n\in[0,1]$.

(*c*) Montrer que les suites $(u_{2n})_{n\in\mathbb{N}}$ et $(u_{2n+1})_{n\in\mathbb{N}}$ convergent et en déduire que $\lim\limits_{n\to+\infty} u_n=\alpha$.

(*d*) En déduire une mise en œuvre de la méthode de point fixe permettant d'écrire un code Python affichant une valeur approchée β de la solution $\alpha\in[0,1]$ de l'équation $e^{-x}-x=0$ satisfaisant la condition $|\beta-\alpha|\leq 10^{-10}$.

7 Méthode de Newton

La méthode de Newton est une méthode plus évoluée utilisant les propriétés de dérivabilité de la fonction étudiée. On suppose dans cette partie que $f : I \longrightarrow \mathbb{R}$ est dérivable sur I et que pour tout $x \in I$, $f'(x) \neq 0$.

L'idée de la méthode de Newton consiste à remplacer, dans le but de la résolution approchée de l'équation $f(x) = 0$, la fonction f par une fonction affine dont la représentation graphique est une tangente à la courbe représentative de f, construite en un point dont l'abscisse n'est pas trop éloignée de la solution cherchée.

Soit $c \in [a, b]$. Alors l'équation de la tangente T_c à la courbe représentative de f au point d'abscisse c est :

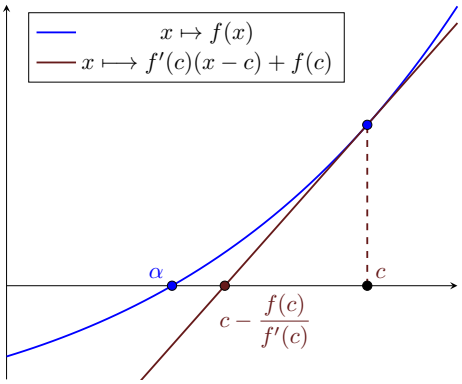
$$y = f'(c)(x - c) + f(c)$$

L'équation $f(x) = 0$ peut être approchée par l'équation $f'(c)(x - c) + f(c) = 0$ en considérant que les courbes représentatives de f et de sa tangente au point d'abscisse c sont, au moins localement, assez proches.

Or l'équation $f'(c)(x - c) + f(c) = 0$ est élémentaire à résoudre et a pour unique solution $x = c - \dfrac{f(c)}{f'(c)}$.

Ainsi, on peut proposer, comme valeur approchée de la solution $\alpha \in [a, b]$ de l'équation $f(x) = 0$, la valeur :

$$\beta = c - \frac{f(c)}{f'(c)}$$



Il apparaît au moins graphiquement qu'une telle méthode d'approximation est très grossière et donne un résultat numérique qui peut être assez éloigné de la valeur cherchée. L'idée de Newton est alors d'itérer ce procédé en recommençant la même construction à partir de la tangente au point d'abscisse β, et ainsi de suite.

La méthode de Newton consiste donc à considérer une suite récurrente telle que $u_0 \in [a, b]$ et :

$$\forall n \in \mathbb{N}, u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}$$

puis à choisir une valeur $N \in \mathbb{N}$ (grand) et à calculer la valeur de u_N pour obtenir une bonne valeur approchée de α. En pratique, cette méthode est souvent redoutablement efficace, et sous des hypothèses qui ne seront pas détaillées ici, on peut s'attendre à obtenir une excellente approximation de α pour $N = 100$.

Exemple Compléter le code ci-dessous pour qu'il affiche une valeur approchée de la solution $\alpha \in [0, 1]$ de l'équation $e^{-x} = x$ (on choisira la valeur $N = 10$, qui est dans ce cas très largement suffisante).

```
from math import exp

def f(x):
    return ..........          # Définition de f
def df(x):
    return ..............      # Définition de f'

u = ..                         # Choix de la valeur initiale
N = ..                         # Choix du nombre d'itérations
for k in range(1,N+1):
    u = ....                   # Actualisation
print(u)
```

8 Exercices

Exercice 3 Écrire le code d'une fonction Python, nommée `balayage`, prenant en paramètres une fonction `f` satisfaisant le contexte général défini en page 1, la valeur de `a` et un pas `p` strictement positif et renvoyant en sortie une valeur approchée β de la solution α de l'équation $f(x) = 0$ satisfaisant la condition $|\beta - \alpha| \leq p$.

Tester cette fonction pour la fonction $f : x \longmapsto xe^x - 10^{20}$, puis pour $f : x \longmapsto x^3 + x^2 - 5x - 10000$, à chaque fois avec $a = 0$ et $p = 10^{-3}$.

Exercice 4 Écrire le code d'une fonction Python, nommée `dichotomie`, prenant en paramètres une fonction `f` satisfaisant le contexte général défini en page 1, les valeurs de `a` et `b` et une précision `p` strictement positive et renvoyant en sortie une valeur approchée β de la solution α de l'équation $f(x) = 0$ satisfaisant la condition $|\beta - \alpha| \leq p$.

Tester cette fonction pour la fonction $f : x \longmapsto e^x + x$, puis pour $f : x \longmapsto x + \ln(x)$, à chaque fois avec $p = 10^{-6}$ (on réfléchira à une initialisation pertinente pour les valeurs de `a` et `b`).

Exercice 5 Écrire le code d'une fonction Python, nommée `Newton`, prenant en paramètres une fonction `f` satisfaisant les hypothèses définies en page 7, sa dérivée `df`, un nombre réel `a` et un nombre entier naturel `N` et renvoyant en sortie une valeur approchée β de la solution α de l'équation $f(x) = 0$ obtenue grâce à la méthode de Newton en N itérations à partir de la valeur initiale a.

Tester cette fonction pour la fonction $f : x \longmapsto x^3 - 2x - 5$, puis pour $f : x \longmapsto x^5 + x - 3$, à chaque fois avec $N = 100$ (on réfléchira à une initialisation pertinente pour la valeur de `a`).

Exercice 6 Soit $N \in \mathbb{N}^*$ un nombre entier naturel qui n'est pas un carré parfait.

On souhaite déterminer une valeur approchée de $\alpha = \sqrt{N}$.

Le nombre α vérifie la relation $\alpha^2 = N$, donc est solution dans $\mathbb{R}$ de l'équation $N - x^2 = 0$.

De plus, α est non-nul, donc est également solution dans $\mathbb{R}^*$ de l'équation $\dfrac{N - x^2}{2x} = 0$.

Pour tout $x \in \mathbb{R}^*$, posons $f(x) = \dfrac{N - x^2}{2x}$ et $g(x) = \dfrac{N + x^2}{2x}$.

Soit $(u_n)_{n\in\mathbb{N}}$ une suite réelle telle que $u_0 = \lfloor \alpha \rfloor$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = g(u_n)$.

(a) Montrer que l'intervalle $[1, +\infty[$ est stable par g.

(b) Montrer que pour tout $x \in \mathbb{R}^*$, $g(x) - \alpha = \dfrac{(x - \alpha)^2}{2x}$.

(c) En déduire que pour tout $n \in \mathbb{N}$, $u_n \in [1, +\infty[$ et $|u_{n+1} - \alpha| \leq \dfrac{|u_n - \alpha|^2}{2}$.

(d) En déduire que pour tout $n \in \mathbb{N}$, $|u_n - \alpha| \leq \left(\dfrac{1}{2}\right)^{2^n - 1}$, puis que $\lim\limits_{n\to+\infty} u_n = \alpha$.

(e) Écrire le code d'une fonction Python, nommée `approx_racine`, prenant en paramètres un entier naturel non-nul `N` qui n'est pas un carré parfait et un réel `p` strictement positif et renvoie en sortie une valeur approchée β de $\alpha = \sqrt{N}$ telle que $|\beta - \alpha| \leq p$ (l'usage de la fonction `sqrt` étant interdit).

(f) En quoi la démarche présentée ci-dessus est-elle une démarche de point fixe ?

(g) En quoi la démarche présentée ci-dessus est-elle reliée à la méthode de Newton ?

Exercice 7 Pour tout $n \in \mathbb{N}^*$, on pose $u_n = \sum\limits_{k=1}^{n} \dfrac{1}{k^2}$ et $v_n = u_n + \dfrac{1}{n}$.

Montrer que les suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ sont adjacentes.

Proposer un code Python permettant d'obtenir une valeur approchée β de la limite commune α des suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ telle que $|\beta - \alpha| \leq 10^{-6}$.

Culture générale : la valeur exacte de α est $\dfrac{\pi^2}{6}$.

Exercice 8 Pour tout $n \in \mathbb{N}^*$, on pose $u_n = \sum\limits_{k=1}^{n} \dfrac{1}{k} - \ln(n)$ et $v_n = \sum\limits_{k=1}^{n} \dfrac{1}{k} - \ln(n+1)$.

Montrer que les suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ sont adjacentes.

Proposer un code Python permettant d'obtenir une valeur approchée δ de la limite commune γ des suites $(u_n)_{n\in\mathbb{N}^*}$ et $(v_n)_{n\in\mathbb{N}^*}$ telle que $|\delta - \gamma| \leq 10^{-6}$.

Culture générale : le nombre γ (c'est sa notation officielle) s'appelle constante d'Euler-Mascheroni.