TP 24
Méthode de Newton

La *méthode de Newton* est une méthode numérique pour approcher la solution d'une équation $f(x) = 0$ qui est, nous allons le voir, plus efficace que la dichotomie.

I Principe

On se donne une fonction f dont on sait qu'il existe une valeur c pour laquelle $f(c) = 0$.
L'idée est la suivante : partant d'une valeur x_0, si possible à proximité de la solution, on calcule $f(x_0)$ puis on trace la tangente T_{x_0} à $f(x_0)$. C'est une droite, et on espère qu'en x_0 la tangente n'est pas horizontale. Elle va donc couper l'axe des abscisses en une valeur qu'on appelle x_1. Puis on répète le procédé à partir de x_1. Ceci définit une suite $(x_n)_{n\in\mathbb{N}}$. On peut espérer, comme sur le dessin, que la suite converge rapidement vers la solution c.

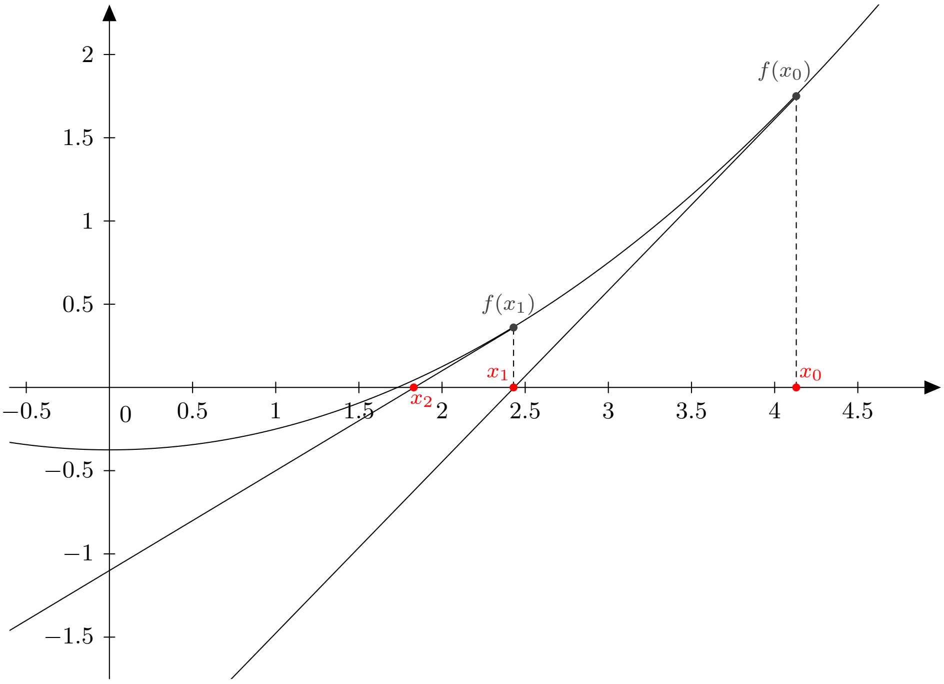

Figure 1 – La méthode de Newton

Exercice 1. [Mathématiques]

1. Supposant x_n construit, rappeler l'équation de la tangente à f en x_n puis montrer que cela conduit à définir la suite $(x_n)_{n\in\mathbb{N}}$ par la relation de récurrence
$$\forall n \in \mathbb{N}, \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1}$$
2. Montrer que pour calculer la racine carrée d'un nombre $\alpha > 0$, en utilisant la fonction $f : x \mapsto x^2 - \alpha$, cela conduit à définir la suite $(x_n)_{n\in\mathbb{N}}$ par la formule
$$\forall n \in \mathbb{N}, \quad x_{n+1} = \frac{1}{2}\left(x_n + \frac{\alpha}{x_n}\right) \tag{2}$$

Exercice 2. [Informatique]

1. Programmer la méthode de Newton pour calculer des racines carrées : écrire une fonction `racine_newton(alpha, n)` qui prend en argument un nombre réel $\alpha > 0$ et un entier n, et qui calcule et affiche les n premiers termes de la suite (2). On pourra prendre par exemple $x_0 = \alpha$.
2. Écrire une fonction similaire qui recherche la racine carrée de α par dichotomie : une fonction `racine_dichotomie(alpha, n)` qui cherche la racine par dichotomie (on pourra prendre comme intervalle initial $[a, b] = [0, \alpha]$, en remarquant que si $\alpha \geqslant 1$ alors $0 \leqslant \sqrt{\alpha} \leqslant \alpha$) et affiche les n premières approximations calculées (on entend par là, les milieux des intervalles).

3. Tester et comparer ces fonctions par exemple pour le calcul de $\sqrt{2}$ puis de $\sqrt{23}$ puis $\sqrt{1000}$, avec une valeur de n de l'ordre de 20.

II Étude d'un exemple

Dans cette partie on étudie la méthode de Newton pour le calcul de $\sqrt{2}$. On considère donc la suite définie par

$$\begin{aligned} x_0 &= 2 \\ \forall n \in \mathbb{N}, \quad x_{n+1} &= \frac{1}{2}\left(x_n + \frac{2}{x_n}\right) \end{aligned} \tag{3}$$

et on pose $F : x \mapsto \frac{1}{2}\left(x + \frac{2}{x}\right)$.
On rappelle que pour représenter graphiquement des fonctions, on peut utiliser les bibliothèques `numpy` et `matplotlib` :

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

On initialise ensuite un tableau des abscisses `X` avec la fonction `np.linspace()`, et on crée un tableau des ordonnées `Y` avec les opérations vectorielles. La fonction `plt.plot(X, Y)` permet de tracer le graphe qu'il faut afficher avec `plt.show()`. On peut en fait passer à `plt.plot` *plusieurs* listes d'abscisses et d'ordonnées, pour tracer plusieurs courbes sur un même graphique.
Pour tout ceci, on s'aide d'un aide-mémoire ou d'une documentation...
https://www.concours-agro-veto.net/IMG/pdf/polypython_nb_141019.pdf
ce qui permet aussi de soigner l'affichage et les paramètres de la fenêtre.

Exercice 3. [Informatique]
Représenter sur un même graphique la fonction F et la droite d'équation $y = x$. Que conjecture-t-on sur la suite $(x_n)_{n\in\mathbb{N}}$?

Exercice 4. [Mathématiques]

1. Démontrer par récurrence que $\forall n \in \mathbb{N}$, $\sqrt{2} \leqslant x_n \leqslant 2$.
2. En étudiant le signe de $F(x) - x$, montrer « comme d'habitude » que $(x_n)_{n\in\mathbb{N}}$ converge vers $\sqrt{2}$.

On propose maintenant une approche un peu nouvelle pour étudier cette suite.

Exercice 5. [Mathématiques]

1. Montrer que $\forall x \in [\sqrt{2}, 2]$, $|F'(x)| \leqslant \frac{1}{2}$.
2. En déduire que si $x, y \in [\sqrt{2}, 2]$ alors $|F(x) - F(y)| \leqslant \frac{1}{2}|x - y|$.
3. Conclure que
$$\forall n \in \mathbb{N}, \quad |x_{n+1} - \sqrt{2}| \leqslant \tfrac{1}{2}|x_n - \sqrt{2}| \tag{4}$$
puis
$$\forall n \in \mathbb{N}, \quad |x_n - \sqrt{2}| \leqslant \left(\tfrac{1}{2}\right)^n |x_0 - \sqrt{2}| \tag{5}$$

Cela semble correspondre à la même vitesse de convergence que pour la dichotomie : l'écart à la limite est divisé par deux à chaque étape ! Mais en fait, la convergence est plus rapide que cela...

Exercice 6. [Mathématiques]

1. Montrer que $\forall n \in \mathbb{N}$, $x_{n+1} - \sqrt{2} = \frac{(x_n - \sqrt{2})^2}{2x_n}$ puis que
$$\forall n \in \mathbb{N}, \quad 0 \leqslant x_{n+1} - \sqrt{2} \leqslant (x_n - \sqrt{2})^2. \tag{6}$$
2. En déduire $\forall n \in \mathbb{N}$, $0 \leqslant x_n - \sqrt{2} \leqslant (2 - \sqrt{2})^{2^n}$.
3. Montrer que $2 - \sqrt{2} \leqslant \frac{2}{3}$, et en déduire
$$\forall n \in \mathbb{N}, \quad 0 \leqslant x_n - \sqrt{2} \leqslant \left(\frac{2}{3}\right)^{2^n}. \tag{7}$$

Exercice 7. [Informatique] Grâce à l'équation(7) on sait exactement combien de termes de la suite il faut calculer pour s'assurer d'avoir une approximation $\varepsilon > 0$ près du nombre $\sqrt{2}$.

1. Écrire une fonction `nombre(epsilon)` qui prend comme argument un nombre réel $\varepsilon > 0$ et renvoie le plus petit entier $n \geqslant 0$ pour lequel $\left(\frac{2}{3}\right)^{2^n} < \varepsilon$.
2. En déduire une fonction `racine2_newton_seuil(epsilon)` qui renvoie une approximation de $\sqrt{2}$ par la méthode de Newton, garantie exacte à ε près.

III Étude générale

Pour formaliser un peu mieux, on fixe les hypothèses suivantes :

1. f est une fonction continue et dérivable sur un intervalle $I = [a, b]$,
2. l'équation $f(x) = 0$ admet une unique solution sur I,
3. f est strictement croissante sur I,
4. $\forall x \in I,\ f'(x) > 0$,
5. En posant $F : x \mapsto x - \frac{f(x)}{f'(x)}$ (bien définie sur I) alors $\forall x \in I,\ F(x) \in I$.

Par le théorème des valeurs intermédiaires, on pourrait aussi supposer directement $f(a) < 0$, $f(b) > 0$ et f strictement croissante sur I. Cependant la condition 4 est nécessaire pour que F soit bien définie et, même dans ce cas, la condition 5 n'est pas automatiquement vérifiée… (pouvez-vous faire un dessin ?)

Exercice 8. [Mathématiques]
Sous ces hypothèses, on définit une suite $(x_n)_{n\in\mathbb{N}}$ par un choix de x_0 quelconque dans I et la relation de récurrence $\forall n \in \mathbb{N},\ x_{n+1} = F(x_n)$.

1. Montrer que si la suite $(x_n)_{n\in\mathbb{N}}$ converge, alors sa limite ℓ vérifie bien $f(\ell) = 0$.
2. Montrer que si f est deux fois dérivable sur I, alors F est dérivable et de plus $F'(\ell) = 0$.
3. On suppose maintenant que f est $\mathcal{C}^2$ sur I. Montrer que F est continue puis que pour tout nombre $k \in \left]0, 1\right[$ il existe un intervalle ouvert J contenant ℓ tel que $\forall x \in J,\ |F'(x)| \leqslant k$.
4. Montrer qu'alors pour tous $x, y \in J$, $|F(x) - F(y)| \leqslant k|x - y|$.
5. En déduire que pour tout $n \in \mathbb{N}$, $|x_n - \ell| \leqslant k^n|x_0 - \ell|$.

Quelques mises en application :

Exercice 9. [Informatique]
Montrer que l'équation $(E) : \cos(x) = x$ a une unique solution dans $[0, \frac{\pi}{2}]$ (on pourra s'aider de représentations graphiques avec Python) puis écrire une fonction `E_newton(n)` qui affiche les n premières approximations de la solution obtenues par la méthode de Newton.

Mais allons plus loin…

Exercice 10. [Mathématiques]
On suppose que la fonction f est $\mathcal{C}^2$ sur l'intervalle $I = [a, b]$.

1. Soit $A \in \mathbb{R}$. Soit $g : t \mapsto f(b) - f(t) - (b - t)f'(t) - A\frac{(b-t)^2}{2}$.
   (a) Calculer g'.
   (b) Montrer qu'on peut choisir A tel que $g(a) = g(b) = 0$.
   (c) En déduire la *formule de Taylor à l'ordre* 2 : il existe un nombre $c \in \left]a, b\right[$ tel que
   $$f(b) = f(a) + (b - a)f'(a) + \frac{(b - a)^2}{2} f''(c) \tag{8}$$
2. Justifier l'existence de $m > 0$ et de $M \geqslant 0$ tels $\forall x \in I,\ f'(x) \geqslant m$ et $|f''(x)| \leqslant M$.
3. En appliquant la formule de Taylor entre ℓ et x, en déduire que $|F(x) - \ell| \leqslant \frac{M}{2m}|x - \ell|^2$.
4. On pose $K = \frac{M}{2m}$. Montrer que $\forall n \in \mathbb{N},\ K|x_{n+1} - \ell| \leqslant K^2|x_n - \ell|^2$ puis
   $$\forall n \in \mathbb{N}, \quad K|x_n - \ell| \leqslant \big(K|x_n - \ell|\big)^{2^n} \tag{9}$$

On parle de **convergence quadratique**. Concrètement, quand la méthode converge, le nombre de chiffres obtenus est multiplié par 2 à chaque étape. Cela est bien mieux que pour la dichotomie, pour laquelle l'intervalle est divisé par 2 et donc le nombre de chiffres gagnés à chaque étape est constant.

IV Je veux plus de bits

On aimerait observer ce qui se passe avec plus de chiffres. Or les nombres réels avec lesquels Python calcule, de base, sont des nombres à virgule flottante représentés par une certaine notation scientifique sur un nombre fixe de bits, et dont la précision est limitée à environ 15 chiffres significatifs. Pour avoir avoir plus de chiffres, il faut charger une bibliothèque qui fournit une autre représentation des nombres réels, avec des opérations. Le plus simple est d'utiliser la bibliothèque `decimal` qu'on chargera d'un coup avec `from decimal import *`. Il faut ensuite fixer une fois pour toute le nombre de chiffres avec lesquels on travaille. Les nombres sont alors de type `Decimal`, et on peut leur appliquer les opérations arithmétiques usuelles et la fonction `print()`. Tester :

```
from decimal import *
# fixer une fois pour toute 20 décimales
getcontext().prec = 20
# x et y sont des nombres de type Decimal, enregistrés avec leurs 20 chiffres
x = Decimal(1)
y = Decimal(3)
# résultat : il y a bien 20 chiffres
print(x/y)
```

Exercice 11. [Informatique]

1. Reprendre le calcul de $\sqrt{2}$, par la méthode de Newton et par dichotomie, dans ce contexte : on pourra fixer le nombre de chiffres à 100 et écrire des fonctions `racine2_newton_decimal(n)` et `racine2_dichotomie_decimal(n)` qui affichent pendant n étapes les approximations obtenues de $\sqrt{2}$. Observer le nombre de chiffres exacts qui double à chaque étapes.
2. Donner 1000 décimales de $\sqrt{2}$, garanties exactes, en calculant d'abord combien d'itérations de la méthode seront nécessaires.