LES FONCTIONS

Note: Les questions des exercices ne constituent que la trame principale de ce que vous devez faire pour vous guider, rien n'interdit bien sûr que vous alliez plus loin, que vous testiez d'autres choses, bref que vous expérimentiez…

EXERCICE 1 : APPROXIMATION D'UNE FONTION PAR UNE SOMME DE SINUS

On considère la fonction constante par morceau suivante :

$$f(t)=\begin{cases}1 \text{ pour } 0<t<T/2\\ 0 \text{ pour } t=T/2\\ -1 \text{ pour } T/2<t<T\end{cases}$$

a) Tracer cette fonction sur un papier.

Il est possible d'approximer $f(t)$ par la somme suivante :

$$\boxed{S(t;n)=\frac{4}{\pi}\sum_{i=1}^{n}\frac{1}{(2i-1)}\sin\left(\frac{2(2i-1)\pi t}{T}\right)}$$

On peut montrer que $S(t;n)\to f(t)$ quand $n\to\infty$.

b) Ecrire sous Python une fonction $\mathrm{S}(t,n,T)$ qui retourne la valeur de $S(t;n)$.

c) Ecrire sous Python une fonction $f(t,T)$ qui retourne la valeur de $f(t)$.

d) Faire afficher des listes qui montrent comment l'erreur $f(t)-S(t;n)$ évolue avec n et t pour les cas $n=1;3;5;10;30;100$ et $t=\alpha T$ avec $T=2\pi$ et $\alpha=0{,}01;0{,}25;0{,}49$.

e) Commenter comment la qualité de l'approximation dépends de α et n.

EXERCICE 2 : CALCULER LA VITESSE ET L'ACCELERATION A PARTIR DE LA POSITION POUR UNE TRAJECTOIRE EN 2D

On considère un objet qui se déplace dans le plan xy tel que à l'instant t l'objet se trouve au point $(x(t),y(t))$. Le **vecteur vitesse** à l'instant t peut être approximé par :

$$v(t)\approx\left(\frac{x(t+\Delta t)-x(t-\Delta t)}{2\Delta t},\frac{y(t+\Delta t)-y(t-\Delta t)}{2\Delta t}\right).$$

Le **vecteur accélération** dans le plan à l'instant t peut être approximé par :

$$a(t)\approx\left(\frac{x(t+\Delta t)-2x(t)+x(t-\Delta t)}{\Delta t^2},\frac{y(t+\Delta t)-2y(t)+y(t-\Delta t)}{\Delta t^2}\right).$$

Ici, Δt est un petit intervalle de temps.

a) Vérifier que lorsque $\Delta t\to 0$, on retrouve bien $v(t)=(x'(t),y'(t))$ et $a(t)=(x''(t),y''(t))$.

b) Ecrire une fonction `Cinematique(x,y,t,dt)` (ici dt correspond à Δt) pour calculer la vitesse et l'accélération d'un objet à partir des relations précédentes. La fonction doit retourner des listes contenant la position, la vitesse et l'accélération pour chaque instant t.

c) De nombreuses courbes fascinantes sont générées par un point sur une roue en mouvement. Le mouvement de la valve d'une roue de vélo est nommé une **cycloïde** et est décrit par la courbe paramétrée suivante :

$$\left.\begin{aligned} x(t) &= c \times (t - sin(t)) \\ y(t) &= c \times (1 - \cos(t)) \end{aligned}\right\} \text{pour } t \geq 0 \text{ et } c = cste$$

La figure 12.11 représente le graphe d'une cycloïde pour $c = 1$ et $0 \leq t \leq 3\pi$. Lorsque la roue fait une révolution dans l'intervalle $0 \leq t \leq 2\pi$, on obtient une arche de la cycloïde.

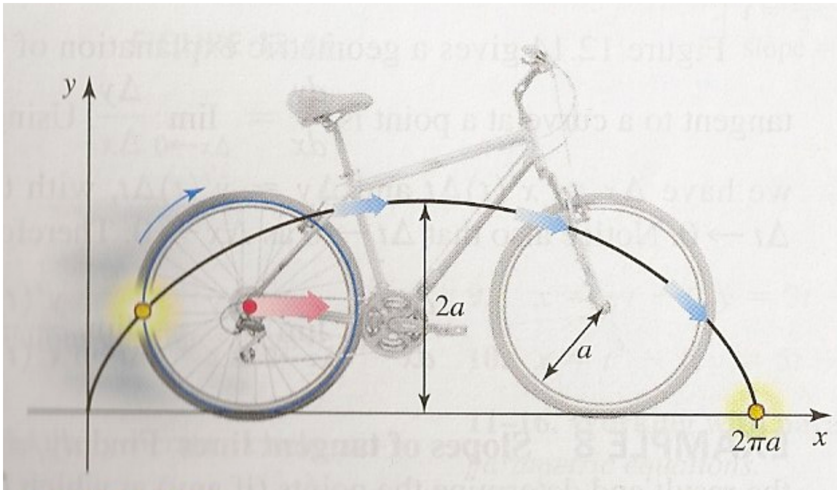


FIGURE 12.10

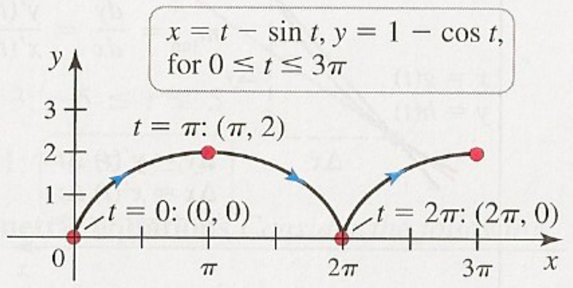


FIGURE 12.11

d) Utiliser votre fonction `Cinematique` dans un programme avec la courbe paramétrique de la cycloïde pour $c = 1$. Tester le programme en calculant, pour $t = 1$ par exemple, la vitesse et l'accélération avec $\Delta t = 10^{-5}$. Vérifier et comparer vos résultats avec un calcul à la main.

e) Vérifier que pour $t = 2\pi n$ avec $n = 0; 1; 2$ ... la vitesse du point s'annule. Que vaut son accélération ?

EXERCICE 3 : DETERMINATION NUMERIQUE DU ZERO D'UNE FONCTION REELLE

Il est très rare de pouvoir déterminer le zéro d'une fonction réelle (d'une variable réelle) d'une façon analytique (ou du moins de façon relativement simple). Il est donc nécessaire d'opter pour une résolution numérique. Une procédure simple et très utilisée est ***la méthode de Newton-Raphson*** qui est illustrée sur la figure ci-dessous qui montre le graphe d'une fonction $f(x)$.

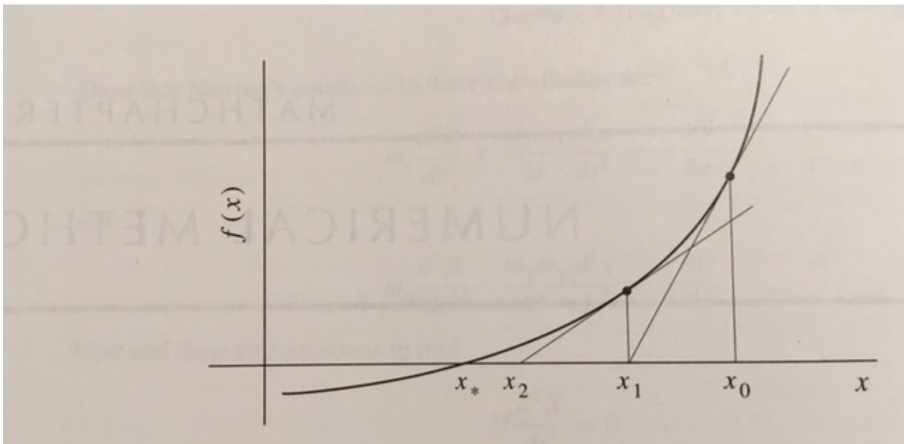


FIGURE A.1
A graphical illustration of the Newton-Raphson method.

La solution de l'équation $f(x)=0$ est notée x^*. L'idée derrière la méthode de Newton–Raphson est de « deviner » une valeur initiale de x notée x_0 « suffisamment proche » de x^* et de dessiner la tangente à la courbe $f(x)$ en x_0, comme illustré sur la figure. Très souvent, le prolongement de la tangente coupe l'axe horizontal en une valeur plus proche de x^* que de x_0. On note x_1 cette valeur de x et on répète la procédure en utilisant x_1 pour produire une nouvelle valeur x_2 qui sera encore plus proche de x^*. En répétant cette procédure, on fait ce que l'on appelle ***une itération***, on peut approcher, avec le degré de précision souhaité, la valeur de x^*.

En utilisant la figure, on peut obtenir une relation commode pour les itérations de x. La pente de $f(x)$ en x_n, $f'(x_n)$, est donnée par :

$$f'(x_n)=\frac{f(x_n)-0}{x_n-x_{n+1}}$$

ce qui donne en résolvant pour x_{n+1} :

$$\boxed{x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}}$$

qui est la relation d'itération pour la méthode de Newton–Raphson.

Comme application, nous pouvons considérer l'équation de réaction suivante :

$$2NOCl_{(g)} \rightleftarrows 2NO_{(g)}+Cl_{2(g)}$$

dont la constante d'équilibre à une température donnée vaut 2,18. Si l'on introduit 1 bar de $NOCl_{(g)}$ dans une enceinte, on obtient à l'équilibre, $P_{NOCl}=1{,}00-2x$, $P_{NOl}=2x$ et $P_{Cl}=x$. Ces pressions doivent satisfaire l'expression de la constante d'équilibre :

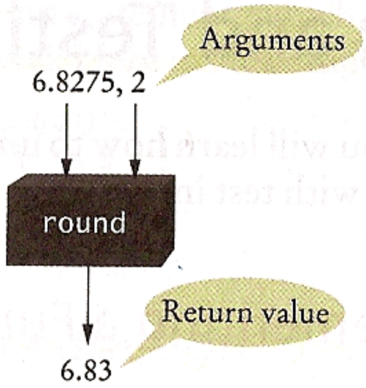
$$\frac{P_{NO}^2P_{Cl_2}}{P_{NOCl}^2}=\frac{(2x)^2x}{((1{,}00-2x)^2}=2{,}18$$

qui peut s'écrire (vérifiez le) : $f(x)=4x^3-8{,}72x^2+8{,}72x-2{,}18=0$.
A cause de la stoechiométrie de l'équation de réaction, la valeur de x recherchée doit se trouver entre 0 et 0,5. Ainsi on peut choisir 0,250 comme notre valeur « devinée » initiale x_0.

Ecrire une fonction $\mathrm{Newton\text{-}Raphson}(x_0,n)$ qui aura pour paramètre d'entrée la valeur « devinée » initiale x_0 et le nombre d'itérations de la procédure n. Cette fonction devra vous retourner la valeur des différents x_n.
La fonction Newton-Raphson devra elle-même faire appel à deux fonctions : $\mathrm{f\text{-}x}(x_n)$ retournant la valeur de $f(x_n)$ et $\mathrm{f\text{-}derive\text{-}x}(x_n)$ retournant la valeur de $f'(x)$.

RAPPELS SUR LES FONCTIONS SOUS PYTHON

Un fonction est une « boite noire » qui nécessite des paramètres (ou arguments) d'entrée de type quelconque (string, list, float, integer etc.…) et qui vous retourne en sortie une ou plusieurs valeurs de type quelconque, ci-dessous l'exemple de la fonction arrondie (round) à deux chiffres après la virgule.



Function Definition

Syntax def *functionName*(*parameterName*$_1$, *parameterName*$_2$, . . .) :
statements

Name of function

Name of parameter variable

Function header

```
def cubeVolume(sideLength) :
    volume = sideLength ** 3
    return volume
```

Function body, executed when function is called.

return statement exits function and returns result.

Program with Functions

By convention, main is the starting point of the program.

The cubeVolume function is defined below.

```
def main() :
    result = cubeVolume(2)
    print("A cube with side length 2 has volume", result)

def cubeVolume(sideLength) :
    volume = sideLength ** 3
    return volume

main()
```

This statement is outside any function definitions.