TP - MÉTHODES NUMÉRIQUES
Analyse et méthodes numériques - Période 3

I. Etude de suites numériques

Exercice 1

1. La suite (a_n) est définie à l'aide du code ci-dessous. Calculer (à la main) les quatre premiers termes de cette suite et donnez-en une définition plus "mathématique".

```
def a(n):
  res=0.25
  for i in range(n):
    res=2*res+1
  return res
```

2. En utilisant Python :

a) Calculer les 10 premières valeurs de cette suite.

b) Placer dans un repère les points correspondants, c'est-à-dire les points de coordonnées (n, a_n) pour $0 \leq n \leq 9$.

3. La suite (a_n) est-elle minorée ? majorée ? bornée ? (précisez)

4. Conjecturer (à l'aide du graphe 2.b) la limite de la suite (a_n) que l'on notera L.

5. Déterminer (à l'aide d'un algorithme) le rang à partir duquel $a_n \approx L$ à 10^{-4} près. Autrement dit, on cherche la plus petite valeur de n pour laquelle $|a_n - L| \leq 10^{-4}$.

Exercice 2

Mêmes questions avec les suites ci-dessous après avoir écrit le code python correspondant :

1. $\begin{cases} b_0 = 0.2 \\ b_{n+1} = \dfrac{1}{2\,b_n + 1} \end{cases}$

2. $c_n = \dfrac{1}{n+1}$

3. $\begin{cases} f_0 = 0,7 \\ f_{n+1} = 3,6\,f_n(1 - f_n) \end{cases}$

4. $\begin{cases} g_0 = 3 \\ g_{n+1} = \dfrac{3}{g_n + 1} \end{cases}$

II. Complexité

En informatique, on s'intéresse à la complexité des algorithmes lorsque l'on veut :

- Evaluer l'efficacité d'un algo
- Comparer deux algo

indépendamment de l'environnement (machine, système, compilateur, . . .). On évalue alors le nombre d'opérations élémentaires en fonction de la taille des données, de la nature des données.

<u>Notations :</u>

- n : taille des données,
- $T(n)$: nombre d'opérations élémentaires **dans le pire des cas**

Si on a très rarement l'expression de $T(n)$ en fonction de n, on arrive à comparer $T(n)$ à des suites « *de référence.* »

Exercice 3 ***Premier exemple : un algorithme de calcul***

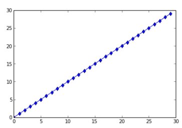
On veut évaluer la complexité $T(n)$ du premier algorithme de l'exercice 1.

```python
def a(n):
  res=0.25
  for i in range(n):
    res=2*res+1
  return res
```

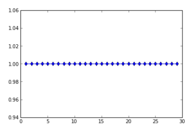
Expliquez chacune des 3 étapes ci-dessous et concluez.

```python
def a(n):
  nbiter=0
  res=0.25
  for i in range(n):
    res=2*res+1
    nbiter+=1
  return (res,nbiter)
def T(n):
  return a(n)[1]
```

```python
xn=range(30)
tn=[T(n) for n in xn]
plot(xn,tn,'db-')
```



```python
xn=range(1,30)
qn=[T(n)*1.0/n for n in xn]
plot(xn,qn,'db-')
```



Exercice 4 ***Un autre algorithme de calcul***

On veut évaluer la complexité $T(n)$ de l'algorithme suivant :

```
def A(n):
  res=0
  for k in range(n):
    if (n%2==0):
      for i in range(n):
        res=res+i
    else:
      res=3*res+1
  return res
```

1. Ajouter un compteur permettant de compteur le nombre d'opérations élémentaires.
2. Construire la liste des $T(n)$ pour n variant de 0 à 50.
3. Représenter la suite $T(n)$ et conjecturer son comportement
4. Validez votre conjecture.

Exercice 5 ***Recherche séquentielle dans une liste***

On veut évaluer la complexité $T(n)$ de l'algorithme de recherche séquentielle dans un tableau (où n est la taille du tableau).

```
from random import *
def tab(n):
    return [randint(1,1000) for i in range(n)]


def Recherche(x,tab):
# renvoie True si x est dans tab, False sinon


print(Recherche(randint(1,1000),tab(500)))
```

1. Ajouter un compteur permettant de compteur le nombre d'opérations élémentaires.
2. Construire la liste des $T(n)$ pour n variant de 0 à 100000 avec un pas de 1000. (On recherchera un nombre aléatoire choisi entre 1 et 1000)
3. Représenter la suite $T(n)$ et conjecturer son comportement
4. Validez votre conjecture.

Exercice 6 ***Recherche dichotomique dans un tableau TRIÉ***

On veut évaluer la complexité $T(n)$ de l'algorithme de recherche dichotomique dans un tableau (où n est la taille du tableau).

```python
from random import *
def tab(n):
    temp=[randint(1,1000) for i in range(n)]
    temp.sort()
    return temp


def RechercheDicho(x,tab):
# renvoie l'indice de x dans tab
# ou -1 si x n'est pas dans tab


print(RechercheDicho(randint(1,1000),tab(1000)))
```

1. Ajouter un compteur permettant de compteur le nombre d'opérations élémentaires.
2. Construire la liste des $T(n)$ pour n variant de 0 à 1000. (On recherchera un nombre aléatoire choisi entre 1 et 1000)
3. Représenter la suite $T(n)$ et conjecturer son comportement
4. Validez votre conjecture.

Exercice 7 ***Un algorithme de tri ?***

Evaluer la complexité $T(n)$ des algorithmes de tri que vous avez vus (où n est la taille du tableau).

- Tri par sélection
- Tri à bulles
- Tri par insertion

III. Résolutions d'équations

A. Méthode de dichotomie

La méthode repose uniquement sur un cas particulier du théorème des valeurs intermédiaires.

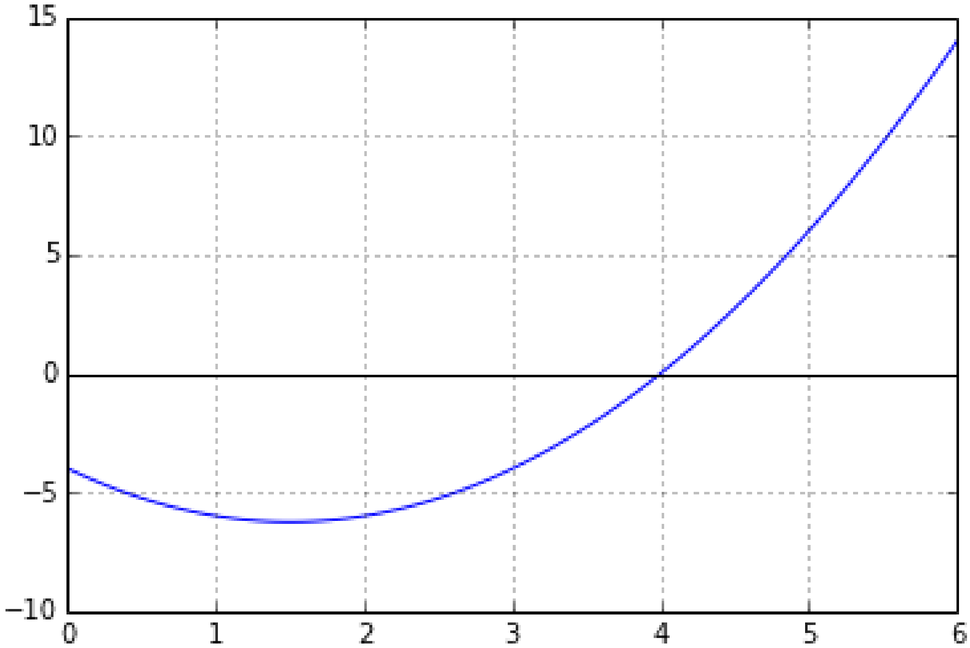
> **Théorème :** *Soit $f : [a,b] \to \mathbb{R}$ une fonction continue. Si $f(a)f(b) < 0$, alors il existe $\alpha \in]a,b[$ tel que $f(\alpha) = 0$.*

Dans la suite on considère $f : [a,b] \to \mathbb{R}$ une fonction continue et on suppose qu'il existe un unique $\alpha \in]a,b[$ tel que $f(\alpha) = 0$.

Exercice 8 ***Implémentation de l'algorithme de dichotomie***

1. Ecrire l'algorithme de dichotomie.

2. Placer les quatre premiers termes des suites $(a_n), (b_n)$ et (c_n) sur le graphe ci-dessous avec $a_0 = 0$ et $b_0 = 6$.



3. Ecrire une fonction `dicho_zero(f,a,b,n)` prenant en paramètre f, a, b et un entier n, et qui retourne une valeur approchée à 10^{-n} près de α par la méthode de dichotomie.

Exercice 9 ***Application à la résolution de*** $f(x) = 0$

Soit la fonction $f(x) = x^2 - 3x + 2$.

1. Déterminer à la main les solutions de l'équation $f(x) = 0$.

2. Que renvoie le code ci-dessous :

```python
print(dicho_zero(f,0,1.5,3))
print(dicho_zero(f,1.5,3,3))
print(dicho_zero(f,0,3,3))
```

3. Que peut-on en conclure ?

Exercice 10 ***Application à la résolution de $f(x) = 0$***

Soit la fonction $f(x) = x^3 - 3x^2 + 1$.

1. Montrer (à la main !) que l'équation $f(x) = 0$ admet une unique solution α dans $[0, 2]$ et conjecturer sa valeur.

2. Déterminer une valeur approchée à 10^{-6} près de α par la méthode de dichotomie.

Exercice 11 ***On cherche à déterminer une valeur approchée du réel $\sqrt{3}$.***

1. Déterminer une fonction f telle que $\sqrt{3}$ soit solution de l'équation $f(x) = 0$.

2. Sur quel(s) intervalle(s) peut-on appliquer la méthode de dichotomie ? Déterminer une valeur approchée à 10^{-6} près de $\sqrt{3}$ via cette méthode et comparer ce résultat à la valeur obtenue avec Python.

B. Méthode de Newton

Théorème : Méthode de Newton.

Soient I un intervalle de $\mathbb{R}$ et $f : I \to \mathbb{R}$ une fonction de classe C^1 ayant un unique zéro α dans I. Si x_0 est suffisamment proche de α et si $f'(\alpha) \neq 0$, alors la suite (x_n) définie pour tout $n \in \mathbb{N}$ par $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ converge vers le zéro α.

Soit I un intervalle de $\mathbb{R}$. Dans la suite on considère $f : I \to \mathbb{R}$ une fonction de classe C^1 ayant un unique zéro α dans I.

Exercice 12 ***Rappels***

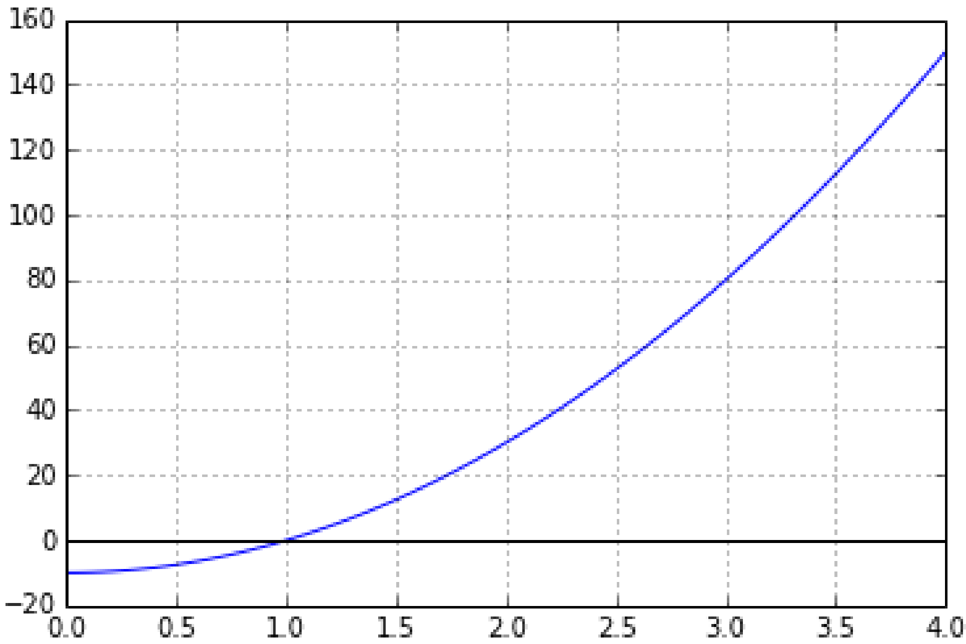
1. Donner la définition du nombre dérivé en $a \in I$ de f.

2. Donner l'équation de la tangente au graphe de f au point de coordonnées $(a, f(a))$.

3. Donner une approximation affine de f au voisinage de a.

Exercice 13 ***Interprétation graphique de la méthode de Newton.***

1. Ecrire (à la main) les trois premiers termes de la suite (x_n).
2. Ecrire l'équation de la tangente au graphe de f au point de coordonnées $(x_n, f(x_n))$. En quel point cette tangente intersecte-t-elle la droite des abscisses ?
3. En déduire une interprétation graphique de la suite (x_n).
4. Placer les quatre premiers termes de la suite (x_n) sur le graphe ci-dessous avec $x_0 = 3.75$.



Exercice 14 ***Implémentation de la méthode de Newton.***

1. Ecrire une fonction `newton(f,f',x0,n)` qui retourne une valeur approchée x_n du zéro de f par la méthode de Newton.
2. Ecrire une fonction `newton_graphe(f,f',x0,n)` qui renvoie le graphe d'une fonction f ainsi que ses tangentes aux points de coordonnées $(x_n, f(x_n))$.
3. Retrouver le résultat de l'exercice 10 en partant du point $x_0 = 1$. Que se passe-t-il si l'on prend $x_0 = 0, x_0 = 2$ ou $x_0 = 1.9$? Que peut-on en déduire ?
4. Appliquer cette méthode pour le calcul approché de $\sqrt{3}$ et comparer le résultat à celui de l'exercice 11.