ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

TRAVAUX PRATIQUES 8

L'objectif de ce TP est de manipuler les fonctions scientifiques des modules : `Numpy, Scipy, Matplotlib, Sympy`.

Vous créerez votre compte rendu à l'aide d'un document ipython notebook (.ipynb).
Ce TP va vous donner l'occasion d'approfondir vos connaissances sur les `nombres de Fibonacci` et le très fameux `nombre d'or`, que l'on retrouve par exemple dans la nature ou l'architecture.
Cette histoire, entre la suite de Fibonacci et le nombre d'or, fait partie de l'une des plus belles des mathématiques. Je vous invite donc à l'aide de Python d'en parcourir quelques lignes .
En suivant le lien ci-dessous, je vous propose quelques minutes de rêve mathématique .

Nature by numbers
`https://www.youtube.com/embed/kkGeOWYOFoA`

En fait ce TP est une compréhension de cette video.

Les exercices sont, dans une large mesure, indépendants.

Ce TP est très long. Néanmoins il vous donnera, je l'espère, l'envie de poursuivre par vous même, cette introduction à l'informatique avec le langage Python. Vous pourrez alors tenter de le finir.

Commençons tout d'abord par le premier thème où nous allons comparer "l'efficacité" en temps de différents programmes permettant le calcul des nombres de Fibonacci .
Ensuite, vous pourrez aborder les thèmes qui vous attirent le plus.

C'est partie pour le dernier TP de l'année...

1 Comparaison de 3 méthodes de calcul

Voici ci-dessous, 3 codes en `Python` donnant le $n^{ième}$ nombre de Fibonacci. Vous les retrouvez sur la plateforme dans un fichier `TP5fibonacci.py`.
méthode récursive

```
def fibo1(n):
    if (n==1 or n==2):
        return 1
    else:
        return fibo1(n-1)+fibo1(n-2)
```

```
    # méthode itérative

 def fibo2(n) :
    a,b,c=1,1,1
    if (n==1) :
       return a
    elif (n==2) :
       return b
    else :
        while (c<n-2) :
            a,b,c=b,a+b,c+1
        return (a+b)

    # méthode "python"

def fibo3(n):
    x=[1,1]
    for i in range(n-2):
        x.append(x[-1]+x[-2])
    return x[-1]
```

Exercice 1. Pour cet exercice, vous allez vous inspirer du code ci-dessous qui permet d'évaluer le temps machine lors de l'exécution de la fonction fibo.

```
import time            # module des fonctions de temps
tps1 = time.clock()    # déclenchement du chronomètre par clock() du module time
fibo(1000)             # lancement de la fonction fibo  pour n=1000
tps2 = time.clock()    # arrêt du chronomètre
print(tps2 - tps1)     # temps écoulé
```

1. Essayer la fonction `fibo1` avec des petites valeurs de n (inférieur à 15) . Que se passe-t-il à votre avis pour des grandes valeurs? pourquoi?
2. Ecrire une fonction `timefibo2` qui renvoie le temps d'exécution de la fonction `fibo2` pour des valeurs de n variant de 100 à 10000 par pas de 100...
3. Ecrire une fonction `ComparFibo2Fibo3` qui donne un tableau des valeurs obtenues pour des valeurs n variant comme dans la question précédente pour les 2 fonctions `fibo2( ) et fibo3( )` .
4. En utilisant `Matplotlib` tracer les 2 courbes obtenues. Petites recommandations, si vous utilisez ipython copier les instructions suivantes :

   ```
   %pylab
   import numpy as np
   import matplotlib.pyplot as plt
   ipython qtconsole --pylab=inline.
   fig = plt.figure()
   ax = fig.add_subplot(111) # changer des 1 en 2 pour voir...
   plt.plot( ...
   ```

5. Quelle est votre conclusion?

Pour la suite du TP, vous utiliserez la fonction `fibo?` la plus rapide...

2 Le nombre d'or

Si 2 longueurs a et b sont dans l'égalité de rapport $\frac{a+b}{a} = \frac{a}{b}$. Elles sont alors dans ce qu'on appelle la divine proportion .
Le cas où a=x et b=1, nous amène à résoudre l'équation $x^2 - x - 1 = 0$
(Le vérifier : C'est-à-dire qu'elle est équivalente à l'égalité suivante $\frac{x+1}{x} = \frac{x}{1}$*...).*

Exercice 2. Des égalités remarquables

1. A l'aide du module `sympy` retrouver les valeurs exactes de cette équation nous appellerons ϕ la valeur positive (le nombre d'or et ψ sa valeur négative .
2. Vérifier toujours à l'aide du module `sympy` que :
   (a) $\phi^2 = \phi + 1$
   (b) $\phi^{-1} = \phi - 1$
   (c) $\phi^{-1} = \psi$
   (d) Calculer pour n variant de 2 à 10 les valeurs successives de ϕ^n. *(Utiliser une liste en compréhension...)*
3. Donner une valeur approchée à 10^{-9} de ϕ en utilisant sa valeur exacte déterminée dans la question 1.!

Exercice 3. Valeur "approchée" du nombre d'or par résolution de calcul numérique d'une équation

— Dans la mesure où on ne sait pas résoudre de manière exacte toutes les équations numériques que l'on peut être amené à rencontrer, il est légitime de mettre au point des démarches permettant d'obtenir une valeur approchée d'une solution d'équation.

— La méthode de la dichotomie et la méthode de Newton sont deux techniques permettant, de manière algorithmique, de calculer une approximation d'une solution de l'équation $f(x) = 0$, où f est une fonction définie sur un intervalle et à valeurs réelles.

Il faudra faire attention aux conditions d'arrêt

Résoudre par valeur approchée le nombre d'or ϕ solution positive de l'équation $x^2 - x - 1 = 0$ en utilisant :

1. d'une part en utilisant le principe dichotomique
2. d'autre part en utilisant l'algorithme de Newton

Exercice 4. Graphique de $f_2(x) = x^2 - x - 1$

1. Tracer le graphe de la fonction f_2.
2. Verifier en "zoomant" la valeur approchée de ϕ trouvée précédemment

Exercice 5. Convergence du rapport de nombres consécutifs de Fibonacci

Ecrire une fonction `EcartfiboNbOr` qui prend en argument une valeur n et renvoie u_{n+1}, u_n , $\dfrac{u_{n+1}}{u_n}$, ϕ

La tester avec de grandes valeurs de n : 10, 100, 1000 10000 .

Que constatez-vous?

3 Formule de Binet

Exercice 6. Valeur "exacte" : La formule de Binet

Par définition la suite de Fibonacci est récursive pour l'instant nous ne pouvons calculer directement le 1000^{me} terme sans avoir déjà calculé les 999 précédents .

La formule de Binet ci-dessous permet de calculer les termes de la suite de Fibonacci sans utiliser la récursivité ,et donc, il est inutile de connaître les deux termes précédents pour trouver celui qui nous intéresse.

Voici cette formule :

$$\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right) = \frac{1}{\sqrt{5}}\left((\phi)^n - (\psi)^n\right)$$

Essayer quelques valeurs de n avec :`Numpy`, `Sympy`. Quelles sont vos conclusions?

Exercice 7. Retrouver la formule de Binet à l'aide des matrices

Pour cette partie regarder les fonctions (valeurs et vecteur propres ...) liées aux matrices dans numpy...

Appellons $(F_n)_{n\geq 0}$ la suite définie par la relation de récurrence $F_{n+1} = F_n + F_{n-1}$ pour $n \geq 1$, avec $F_0 = 0$ et $F_1 = 1$.

1. Déterminer une matrice $A \in M_2(\mathbb{R})$ telle que, pour tout $n \geq 1$,

$$\begin{pmatrix} F_2 \\ F_1 \end{pmatrix} = A \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}.$$

2. Montrer que A admet deux valeurs propres réelles distinctes que l'on note λ_1 et λ_2 avec $\lambda_1 < \lambda_2$. Que constatez-vous par rapport à ϕ et ψ
3. Trouver des vecteurs propres ε_1 et ε_2 associés aux valeurs propres λ_1 et λ_2, sous la forme $\begin{pmatrix}\alpha\\1\end{pmatrix}$, avec $\alpha \in \mathbb{R}$.
4. Déterminer les coordonnées du vecteur $\begin{pmatrix}F_1\\F_0\end{pmatrix}$ dans la base $(\varepsilon_1, \varepsilon_2)$, on les note x_1 et x_2.
5. Montrer que $\begin{pmatrix}F_{n+1}\\F_n\end{pmatrix} = \lambda_1^n x_1 \varepsilon_1 + \lambda_2^n x_2 \varepsilon_2$. En déduire que

$$F_n = \frac{\lambda_1^n}{\lambda_1 - \lambda_2} - \frac{\lambda_2^n}{\lambda_1 - \lambda_2}\,.$$

6. Retrouvez la formule de Binet de l'exercice précédent.

4 Dessin avec matplolib

Exercice 8. Rectangle d'or

Construire et enregistrer dans un fichier un rectangle d'or de largeur 5 cm. C'est-à-dire que la longueur est égale à ϕ fois la largeur.

Exercice 9. Spiral d'or

Nous allons utiliser `Matplotlib` pour reproduire le dessin de la spirale vue dans la video *nature by numbers*.

En utilisant le code disponible sur la plateforme moodle spiral.py . Essayer de reproduire la spirale d'or.

Il vous faudra tout d'abord comprendre le sens des paramètres des objets de pyplot rectangle arc ...

5 Fraction continue de ϕ

Pour vous autoformer avec le module `fractions` de Python.

`https://fr.wikibooks.org/wiki/Math%C3%A9matiques_avec_Python_et_Ruby/Fractions_en_Python`

Allez sur wikipedia afin de comprendre ce que sont les fractions continues.

Nous allons montrer que : oh surprise ! $\phi = \cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}}}}$

Exercice 10. fraction continue

1. Montrer que $\phi = \frac{1}{1+\phi}$
2. Puis que $\phi = \frac{1}{1+\frac{1}{1+\phi}}$
3. Ecrire une fonction `frc(n,d)` qui prend n le numérateur et d le dénominateur d'une fraction et renvoie un tableau de fraction continue
4. Ecrire la fonction réciproque qui d'un tableau de fraction continue renvoie la fraction $\dfrac{n}{d}$
5. Appliquer la fonction `frc(n,d)` au rapport de 2 nombres de Fibonacci consécutifs de plus en plus grand. Conclusion.