UNIVERSITÉ DE LORRAINE | IUT Saint-Dié-des-Vosges

DUT INFO2 M3202C **2018/2019**

Φ

Le Nombre d'Or

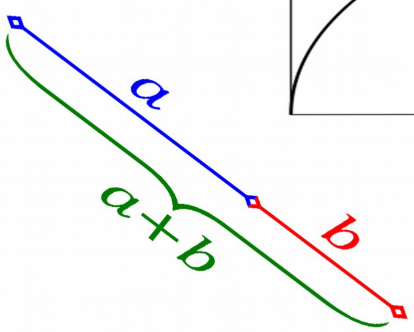
Modélisations Mathématiques

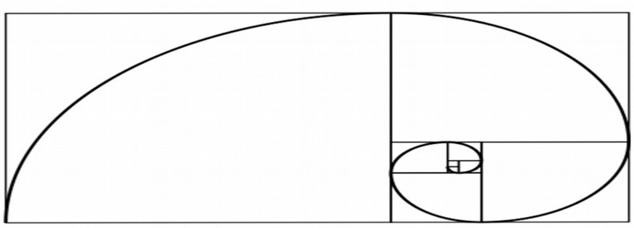
Mémoire réalisé par : AISSI Ayoub

DELANDHUY Mattéo

Supervisé par : Monsieur Patrick ADELBRECHT

Comment approcher le nombre d'or?





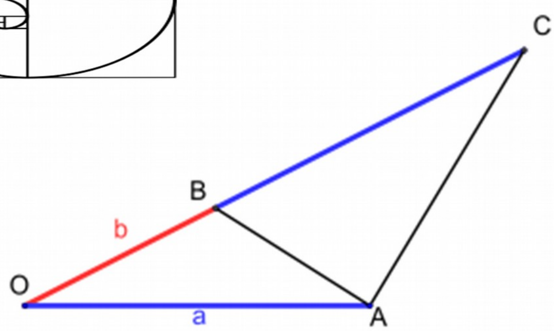


Table des matières

I. Introduction

Le nombre d'or, nommé Phi et noté ϕ est assez spécial. Ce nombre serait à la base de la vie, se retrouverait dans différents endroits, même inattendus.

La toute particularité de ce nombre est que $\phi^2 = 1 + \phi$.

Il faut savoir que ϕ un nombre irrationnel et on estime sa valeur à 1,618033...

Vous pouvez voir les 100 000 premières décimales en consultant le lien suivant : http://www.gecif.net/articles/mathematiques/nombre_d_or/nombre_d_or_decimales.html

Ce sujet est vraiment intéressant en terme de perspectives mathématiques, que ce soit théorique ou pratique, ce sujet a toujours attiré plusieurs mathématiciens mais aussi des artistes et des chercheurs.

Nous avions donc étudié ce sujet et nous avons dégagé une problématique basique qui était le centre des études menées sur ce nombre magique depuis plusieurs années. Notre problématique est donc la suivante :

Comment calculer et approcher la valeur exacte du nombre d'or ?

Pour procéder il a fallu tout d'abord revenir en arrière dans le temps pour savoir d'où nous vient ce nombre d'or et comment son apparition s'est produite. Après avoir eu une idée sur la source de ce nombre nous avons analysé ses aspects mathématiques, équations possibles, moyens de calcul… Ce qui nous a permis de faire des modélisation sur ce nombre de façon théorique (calculs) et informatique (programmes et algorithmes d'approche de ϕ). Par la suite, une étude sur la présence de ce nombre dans différents domaines dans notre vie a été menée à fin de visualiser à quel point c'est nombre unique et que l'on n'a pas nommé nombre d'or sans raison.

II. Le nombre d'or dans l'histoire :

L'histoire du nombre d'or n'est pas récente et ne date pas de l'ère de la technologie comme certaines personnes le pense, le nombre d'or est bien plus

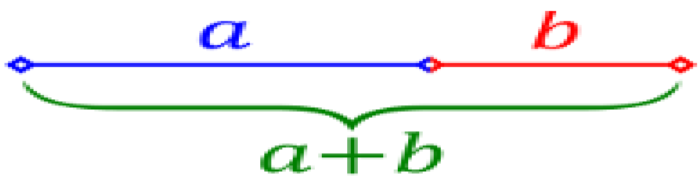
ancien que ce que l'on puisse imaginer.

Il y a 10 000 ans : Première manifestation humaine de la connaissance du nombre d'or (temple d'Andros découvert sous la mer des Bahamas).

2800 avant J-C : La pyramide de Khéops a des dimensions qui mettent en évidence l'importance que son architecte attachait au nombre d'or.

Vème siècle avant J-C : Théodore de Cyr montre l'irrationalité du nombre d'or.

***IIIè siècle avant J-C*. :** Euclide évoque le partage d'un segment en "extrême et moyenne raison" dans le livre VI des Éléments.



La proportion définie par *a* et *b* est dite d'« extrême et moyenne raison » lorsque $(a + b)/a = a/b$. Le rapport a/b est alors égal au nombre d'or.

<u>Vè siècle avant J-C. (447-432 av.JC)</u> **:** Le sculpteur grec Phidias utilise le nombre d'or pour décorer le Parthénon à Athènes, en particulier pour sculpter la statue d'*Athéna Parthénos* . Il utilise également la racine carrée de 5 comme rapport.

<u>1498</u> **:** Fra Luca Pacioli, un moine professeur de mathématiques, écrit *De divina proportione* ("La divine proportion").

<u>Au XIXème siècle</u> **:** Adolf Zeising (1810-1876), docteur en philosophie et professeur à Leipzig puis Munich, parle de "section d'or" (*der goldene Schnitt)* et s'y intéresse non plus à propos de géométrie mais en ce qui concerne l'esthétique et l'architecture. Il cherche ce rapport, et le trouve (on trouve facilement ce qu'on cherche ...) dans beaucoup de monuments classiques. C'est lui qui introduit le côté mythique et mystique du nombre d'or.

<u>Au cours du XXème siècle</u> **:** des peintres tels Dali et Picasso, ainsi que des architectes comme Le Corbusier, eurent recours au nombre d'or.

D'où vient le nom du nombre d'or et sa notation ?

On le désigne par la lettre grecque φ (phi) en hommage au sculpteur grec Phidias (né vers 490 et mort vers 430 avant J.C) qui décora le Parthénon à Athènes. C'est Théodore Cook qui introduisit cette notation en 1914. C'est Ghyka, un prince roumain, qui donnera le nom de nombre d'or. Lui aussi, tente de prouver le pouvoir esthétique de ce nombre.

III. Le nombre d'or en mathématiques :

1. Formules du nombre d'or :

$$\varphi = 1 + 1/\varphi$$

$$\varphi^{n+1} = \varphi^{n} + \varphi^{n-1}$$

$$\varphi^{2} - \varphi - 1 = 0$$

Le nombre d'or est la seule solution positive de l'équation suivante dans l'ensemble des réels : $x^2 = x+1$

$$\frac{1+\sqrt{5}}{2} \approx 1{,}6180339887$$

La solution de l'équation

2. L'irrationalité du nombre d'or :

Démonstration rapide de l'irrationalité du nombre d'or :

Supposons que le nombre d'or x soit rationnel, c'est-à-dire égal à une fraction p/q.

$$X = \frac{p}{q}$$

On suppose que les nombres entiers p et q sont choisis de manière que la fraction soit la plus simplifiée possible. Autrement dit, p et q n'ont pas de facteur commun. Ou encore: p de divise par q, pas plus que q ne divise p.

On peut aussi dire que leur plus grand commun diviseur est égal à 1 :

pgcd(p,q) = 1

Le nombre d'or est défini par l'équation : $x^2 - x - 1 = 0$

En y introduisant la valeur rationnelle de x **: $(p/q)^2 - (p/q) - 1 = 0$**

En multipliant par **q^2 :**

$$p^2 - pq - q^2 = 0$$

$$p(p - q) = q^2$$

En divisant par **p :**

$$p - q = q^2 / p$$

Le membre de gauche est un entier. En conséquence celui de droite doit l'être aussi. Mais est-ce bien vrai? $p - q$ = entier || q^2 / p = entier ?

Eh bien non! p ne peut pas diviser q. Ce serait contraire à notre hypothèse de départ. Il n'y a aucun facteur commun entre p et q.

Cette contradiction montre qu'il n'est pas possible d'écrire le nombre d'or sous forme de fraction. Le nombre d'or est donc irrationnel.

3. Les équations du nombre d'or :

EQUATIONS	
$x^2 - x - 1 = 0$ x2 – x1 – x0 = 0	_ Le nombre d'or est racine réelle de cette équation. _C'est donc un nombre : réel algébrique mais non transcendant
1 / Φ = Φ - 1 Φ = 1 + 1 / Φ Φ 2 = Φ + 1 Φ 3 = 2 Φ + 1 = (Φ + 1) / (Φ - 1)	_Relations entre phi et ses puissances. _Seul nombre qui, lorsqu'on lui soustrait l'unité, devient son propre inverse.
x4 – 2x3 + x2 – 1 = $(x^2 - x - 1)(x^2 - x + 1) = 0$	Le nombre d'or et son inverse sont solutions de cette équation.

4. Le nombre d'or et les puissances :

Φ et les puissances		
φ	$1+\frac{1}{\varphi}=\sqrt{(1+\varphi)}$	1.618...
φ^2	$= \varphi+1$	2.618...
φ^3	$\varphi^2+\varphi = 2\varphi+1 = (\varphi+1) / (\varphi-1)$	4.326...
φ^4	$3\varphi + 2$	6.854...
φ^5	$5\varphi + 3$	11.090...
φ^n	φ n-1 + φ n-2	Somme des deux précédents
φ^n	$F_n \varphi + F_{n-1}$	F_n= nombre de la suite de Fibonacci

5. Le nombre d'or et la suite de Fibonacci

Limite du rapport entre nombres consécutifs de Fibonacci ou de Lucas.

$$\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \Phi$$

La suite de Fibonacci a un rapport avec le nombre d'or. En effet, le rapport de deux nombres consécutifs de la suite est alternativement supérieur et inférieur au nombre d'or.

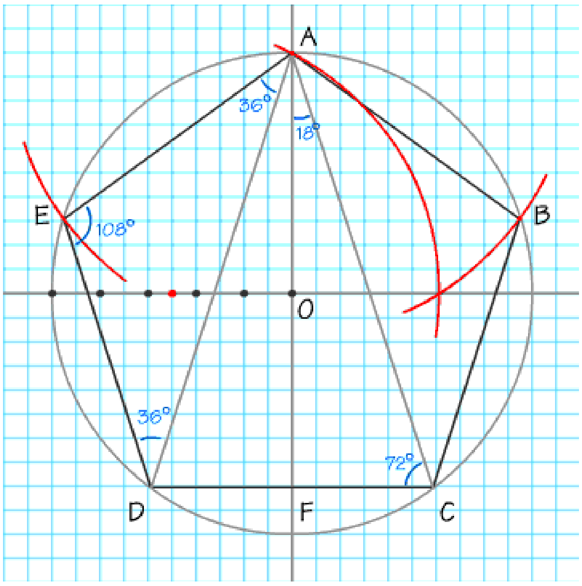
La suite de Fibonacci fonctionne comme suit : Il suffit de prendre deux nombres de départ. Les ajouter donne le troisième, puis le deuxième + le troisième donne le quatrième et ainsi de suite. Les termes de cette suite sont appelés nombres de Fibonacci.

Exemple :

13/8 = 1.625 ; 21/13 = 1.61538… ; 34/21 = 1.61904…et ainsi de suite…

Plus on avance dans la suite de Fibonacci, plus l'écart s'amenuise, et plus le rapport des deux nombres successifs, comme montré sur la formule au dessus, tend vers la valeur du nombre d'or qui est de 1,618..

6. Nombre d'or et le pentagone régulier :



Soit le triangle ADC et les triangles ADF et AFC. Les triangles AED et ADC sont des "triangles d'or" puisque leurs côtés sont dans le rapport du nombre d'or.

Calcul de FD :

$$\frac{FD}{AD} = \sin 18^\circ$$

$$FD = AD \times \sin 18^\circ$$

$$\frac{FC}{AC} = \sin 18^\circ$$

Soit **DC = AD sin 18° + AC sin 18°**

Soit **DC = AD x 2sin 18° => DC = AD x 0,618**

$$\frac{DC}{AC} = 0{,}618 \quad \& \quad \boxed{\frac{AD}{DC} = 1{,}618 = \Phi}$$

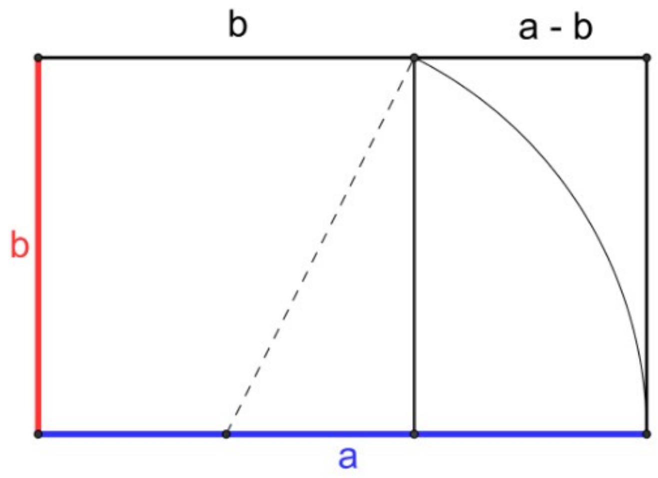
Le rapport entre une diagonale et un côté du pentagone est égal au nombre d'or

On a donc :

$$\cos 36^\circ = \frac{\Phi}{2} \quad \& \quad 2\sin 18^\circ = \frac{1}{\Phi}$$

7. Rectangle et spirale d’or :

Dans un rectangle d'or, le rapport de la longueur sur la largeur est égal au nombre d'or ϕ .



On a donc :

- a = φ
- b = 1

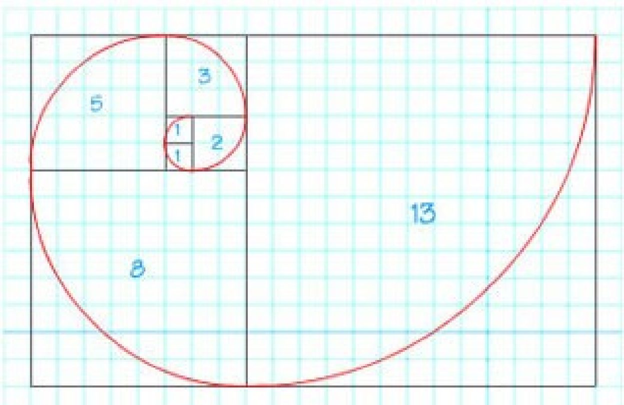
1. Comment construire un rectangle d’or ?

• Tracer un carré ABCD ayant comme côté la largeur souhaitée
• Prendre le milieu K de [AD]
• Rabattre le point C sur (AD) en traçant le cercle de centre K, passant par C. Ce cercle coupe [AD] en E
• Terminer la construction du rectangle d'or ABFE

On peut également utiliser cette manière ci dessous qui fait intervenir la suite de Fibonacci dans les aires des rectangles :

Les côtés des carrés sont une suite de Fibonnaci : 1-1-2-3-5-8-13-...

On peut donc tracer la spirale d’or avec un compas en prenant pour milieu le sommet du carré et tracer une diagonale en courbe.



8. La chaîne dorée :

Chaîne infinie non répétitive de 1 et 0. Sorte de fractale, car comporte une auto-similarité.

Sa construction :

Départ: S_0= 0 || S1 = 1

Itération: S_n = concaténation de S_n-1 et de S_n-2

Valeurs de la chaîne dorée :

01

10

101

10110

10110101

1011010110110

101101011011010110101

…

9. Fractions d'or et fractions continues :

Fractions d'approximation du nombre d'or appelées réduites du nombre d'or il s'agit du quotient de deux Fibonacci consécutifs & présentation de la fraction continue correspondante.

1 1

2 2

3/2 1+1/`2`

5/3 1+1/(1+1/`2`)

8/5 1+1/(1+1/(1+1/`2`))

13/8 1+1/(1+1/(1+1/(1+1/`2`)))

21/13 1+1/(1+1/(1+1/(1+1/(1+1/`2`))))

34/21 1+1/(1+1/(1+1/(1+1/(1+1/(1+1/`2`)))))

55/34 1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/`2`))))))

...

10. Angle d'or et trigonométrie :

Le nombre d'or a une relation avec la trigonométrie aussi, d'une part il a des formes cachées dans le tableau des angles ainsi que des formules en relation avec le fameux nombre **Pi** en plus d'autres formules de calculs permettant d'approcher sa vrai valeur.

Quelques formes cachées du nombre d'or :

sin 18° = 1/(2Φ)

cos 72° = 1/(2Φ)

sin 54° = Φ/2

cos 36° = Φ/2

Il existe deux valeurs qui sont la base de construction des triangles d'or :

Φ = 2 sin 3/10 = 2 cos /5 = 2 cos 36°

= 2 x 0,809 … = 1, 618…

****

1/Φ = 2 sin /10 = 2 cos 2/5 = 2 sin 18°

= 2 x 0,309 … = 0, 618 …

Dont celui à 36° représente l'angle des étoiles à cinq branches, c'est aussi le tiers de l'angle au sommet d'un pentagone.

1. L'angle d'or :

137,5° = 306 / φ Angle d'or, célèbre en phyllotaxie.

L'angle d'or est le résultat du ratio de l'angle a / b .

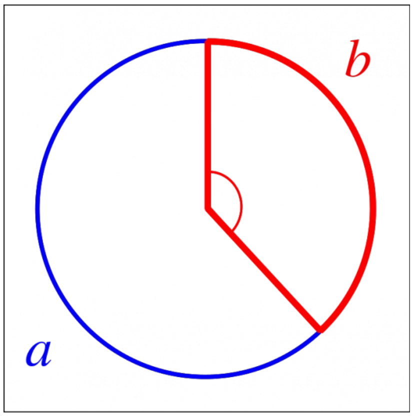
D'ailleurs on a :

- 360 / φ = 222,5 = a
- b = 360- a = 137,5

La division de l'angle a sur b donne le résultat suivant :

b/a = φ = 1.618...

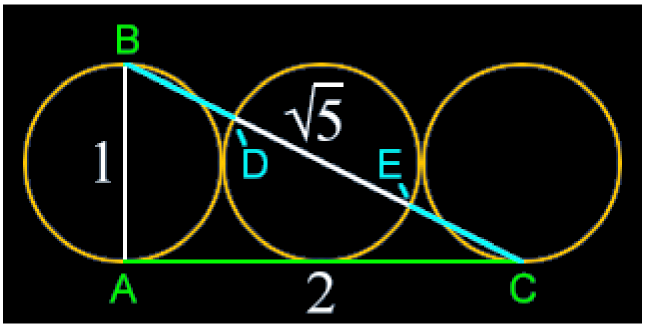
On a donc : **a/b = c/a = ϕ**



2. Approcher Pi avec φ :

$$\pi = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\left(\varphi^{-2k-1}+\varphi^{-6k-3}\right) = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\left((\varphi-1)^{2k+1}+(2\varphi-3)^{2k+1}\right)$$

On constate donc que le nombre d'or est lié au calcul du nombre Pi, de plus il existe d'autres raison pour que ça soit le cas :



Ce schéma nous fait bien rendre compte qu'il y a un rapport entre les cercles donc Pi ainsi que le nombre d'or avec exactement ce qui représente le nombre d'or, c'est à dire le nombre 1, $\sqrt{5}$ et 2.

On verra l'algorithme d'approche du nombre d'or en utilisant Pi dans la partie qui suit.

IV. Modélisation du nombre d'or :

Pour cette partie, nous allons procéder au calcul du nombre d'or, autrement dit, la manière d'approcher la vraie valeur du nombre d'or.

Il existe plusieurs manières et plusieurs astuces pour approcher le nombre d'or, nous avons choisis les manières les plus pertinentes en les implémentant sous forme d'algorithmes en langage Python, mais ils seront présentés en pseudo code compréhensible par tout le monde même les personnes n'ayant aucune expérience dans le domaine d'informatique. Nous avons aussi profité de l'occasion pour créer une interface graphique à fin de visualiser le développement de la création d'une spirale d'or à partir de la suite de Fibonacci, son code sera fourni en annexe.

1. Trouver le nombre d'or avec la suite de Fibonacci :

Tout d'abord on a choisis de commencer par cet algorithme vu qu'il est connu plus que les autres et plus facile à implémenter, cet algorithme sera présenté de deux façons : itératif et récursif.

Comme vous avez pu le voir plus haut, la limite du nombre d'or est coincé entre 2 rangs de la suite de Fibonacci lorsque l'on tend vers l'infini et que l'on divise le rang supérieur par le rang inférieur.

$$\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \Phi$$

Nous calculons alors la suite de Fibonacci jusqu'au rang désiré et nous diviserons le rang désiré par le rang désire -1, c'est à dire le terme qui vient juste avant le terme souhaité.

Algorithme itératif

```
Afficher ("Saisir  rang souhaité de la suite de fibonnaci")
Demander (nterms)
        n1 = 0
        n2 = 1
        count = 0
        SI nterms <= 0 OU nterms = 1:
             Afficher("Rentrer un nombre au dessus de 1!")
        FIN SI
        SINON
             TANT QUE count < nterms:
                        temp = n1 + n2
                        n1 = n2
                        n2 = temp
                        count += 1
           FIN TANT QUE
        FIN SINON
resultat = n2/n1
```

- Complexité linéaire : n

Algorithme récursif

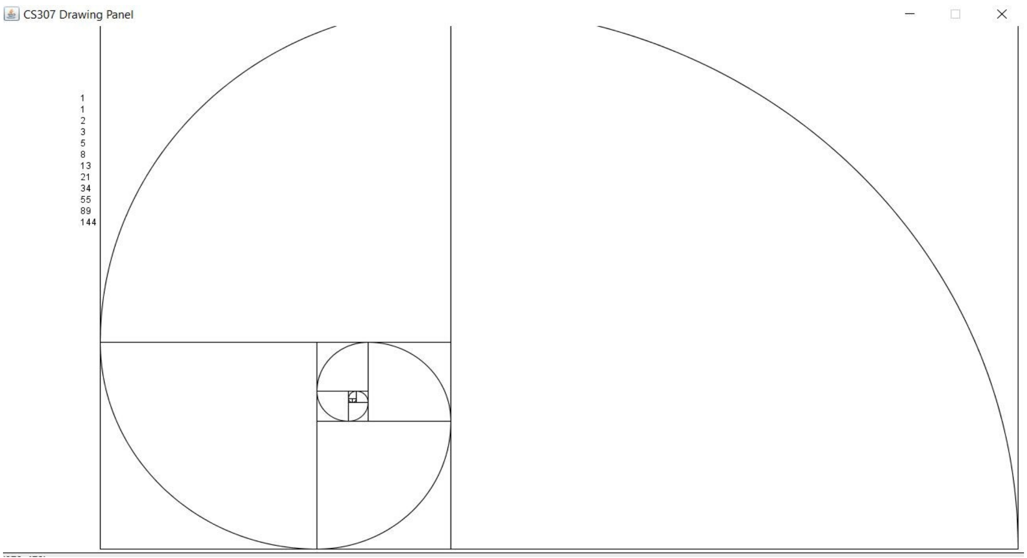
```
fonction fibonacci(n):
      Afficher (“Saisir  rang souhaité de la suite de fibonnaci“)
      Demander (n)

            SI n < 2:
                  RETOURNER n
            FIN SI
            SINON:
                  RETOURNER fibonacci(n-1)+fibonacci(n-2)
            FIN SINON
```

- Le problème est que cet algorithme est bien plus long que l’algorithme itératif
- Sa complexité est e^n

2. Représentation graphique de la spirale d’or

La suite de Fibonacci génère des rectangles, puis la spirale d’or est construite à partir des aires de ces rectangles. L’utilisateur peut choisir l’échelle de son dessin ainsi que le nombre de termes souhaité, ces paramètres sont entrées sur la console lors de l’exécution du programme. Le programme affiche également le nombre du rang sur le coté . La génération de la spirale d’or se fait devant l’utilisateur il pourra visionner le traçage des courbes en même temps que le déroulement de l’algorithme.



3. Trouver le nombre d’or seulement avec des fractions et des 1 :

Les fractions infini de 1 ci-dessous tendent vers le nombre d’or. Nous pouvons donc l’approcher aussi loin que nous irons dans dans le nombre de répétition du schéma : 1/(1+ ...

Nous retirerons le (+) final une fois la boucle finie, et nous ajouterons autant de parenthèses qu’il y a eu de répétitions dans la boucle.

$$\varphi = \sqrt{1+\varphi} = \sqrt{1+\sqrt{1+\varphi}} \quad \text{et} \quad \varphi = \sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\cdots}}}}.$$

Le code

```
AFFICHER(“Rentrer le nombre fois de répétition de l’algorithme“)
DEMANDER(nterm)
        count = 0
        count2= 0
        resultat=""
        SI nterms <= 0:
                AFFICHER("Rentrer un nombre positif!")
        FIN SI
        SI nterms == 1:
                AFFICHER(“Le résultat au rang 1 est 1“)
        SINON:
                TANT QUE count < nterms:
                        resultat= Concaténation(resultat,"1/(1+")
                        count += 1
     FIN TANT QUE
                resultat=resultat[:-1] // retirer le dernier caractère
                TANT QUE count2 < nterms:
                        resultat= Concaténation(resultat,")"
                        count2 = count2 + 1
           FIN TANT QUE
```

4. Trouver le nombre d’or avec uniquement des racines carrées et des 1 :

Nous pouvons approcher le nombre d’or avec des racines carré pour cette raison :

Comme $\Phi + 1 = \Phi^2$, on a $\Phi = \sqrt{1+\Phi}$, et par récurrence on obtient une succession de ce schéma : sqrt(1+

Il faudra là aussi penser à retirer le + final une fois la boucle finie, et nous ajouterons autant de parenthèses qu’il y a eu de répétitions dans la boucle.

Le code

```
def RacineOr():
AFFICHER(“Rentrer le nombre fois de répétition de l’algorithme“)
DEMANDER(nterm)
        count = 0
        count2= 0
        resultat=""
        SI nterms <= 0:
                AFFICHER("Rentrer un nombre positif!")
        FIN SI
       SI nterms == 1:
                AFFICHER(“Le résultat au rang 1 est 1“)
                print(n1)
        SINON:
                TANT QUE count < nterms:
                resultat= Concaténation(resultat,"sqrt(1+")
                count += 1
           FIN TANT QUE
                 resultat=resultat[:-1] // retirer le dernier caractère
                TANT QUE count2 < nterms:
              resultat= Concaténation(resultat,")"
                          count2 = count2 + 1
          FIN TANT QUE
     AFFICHER (resultat)
```

- La seule chose qui change entre la fonction de la racine carré et la fraction est la concaténation qui est différente

5. Trouver le nombre d’or en approchant la racine carré de 5

Le nombre d’or étant précisément égale à (1+sqrt(5))/2

Sqrt(5) étant irrationnel, si l’on approche sqrt(5) on pourra donc approcher le nombre d’or.

Cette méthode nécessitant de connaître la racine carré de 5, racine carré de 5 possédant un nombre infini de décimal, nous allons utiliser la méthode de Héron caractérisé par :

$$\forall n \in \mathbb{N} \quad x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

a représente le nombre dont lequel on cherche sa racine, $\mathbf{x_n}$ représente le nombre au rang n, on lui attribue ici la partie entière de racine(5) qui est 2 car 2*2 = 4 et 3*3 = 9 donc 2*2<racine(5)<3*3

<u>Le code</u>

```
fonction racine5():
        nterms = Afficher("Nombre de répétition de l'algorithme")
        count = 0
        n1 = '2'
        SI nterms <= 0:
                AFFICHER("Rentrer un nombre positif!")
       FIN SI
        SI nterms == 1:
                AFFICHER("Le résultat est : ",n1)
     FIN SI
        SINON:
                SI (count < 1):
                          n2 = Inverse (n1)
                FIN SI
                SI (count >= 1):
                          n2 = "2/(n3 +5*n4)"
                FIN SI
                n3 = n1
                n4 = n2
                n5= (n3+5*n4)/2
                n1 = "(n3+5*n4)/2"
                count = count+1
     FIN SINON
     RETOURNER n5
     AFFICHER((1+ racine5())/2))
```

On peut calculer le racine de 5 avec des fractions continues de cette forme :

$$\sqrt{5} = 2 + \cfrac{1}{4 + \cfrac{1}{4 + \cfrac{1}{4 + 1 \ldots}}}$$

Pour cela, il suffit de mettre en place l'algorithme similaire du calcul de Phi mais en changeant les nombres, cela nous donne :

Le code

```
def FractionRacine5():


AFFICHER(“Rentrer le nombre fois de répétition de l’algorithme“)
DEMANDER(nterm)
        count = 0
        count2= 0
        resultat=2
        SI nterms <= 0:
                AFFICHER("Rentrer un nombre positif!")
        FIN SI
        SI nterms == 1:
                AFFICHER(“Le résultat au rang 1 est 1“)
        SINON:
                TANT QUE count < nterms:
                        resultat= Concaténation(resultat,"1/(4+")
                        count += 1
                FIN TANT QUE
                resultat=resultat[:-1] // retirer le dernier caractère
                TANT QUE count2 < nterms:
                        resultat= Concaténation(resultat,")")
                        count2 = count2 + 1
                FIN TANT QUE
        AFFICHER(resultat)
```

6. Trouver le nombre d’or avec Pi :

On sait que φ = 2*cos(Pi / 5)

On peut trouver le nombre d’or, pour cela in nous faut approcher PI :

Le code

```
def Pi():
AFFICHER( Saisir le nombre de répétition de l’algorithme“)
   SAISIR( P)
    C =4 / 1
    I = 1
    POUR(P, P<0, P-1):
             z =  4 / (I+2)
          SI i%2 == 0:
             z = z*(-1)
          FIN SI
         C = C+ z
         I += 2
 RETOURNER C
```

Maintenant on utilise cette fonction d’approche de Pi pour approcher le nombre d’or comme suit :

Le code

```
def approchePhiAvecPi():
    Phi = 2*cos(Pi()/5)
    AFFICHER(Phi)
```

Ici on aurait aussi pu utiliser la formule suivante aussi :

φ = 1 - 2*cos(3*Pi()/5)

7. Résultats :

Tous les algorithmes après leur déroulement donnent une valeur approchée à la vraie valeur du nombre d’or soit 1,618033….. et plusieurs décimales…

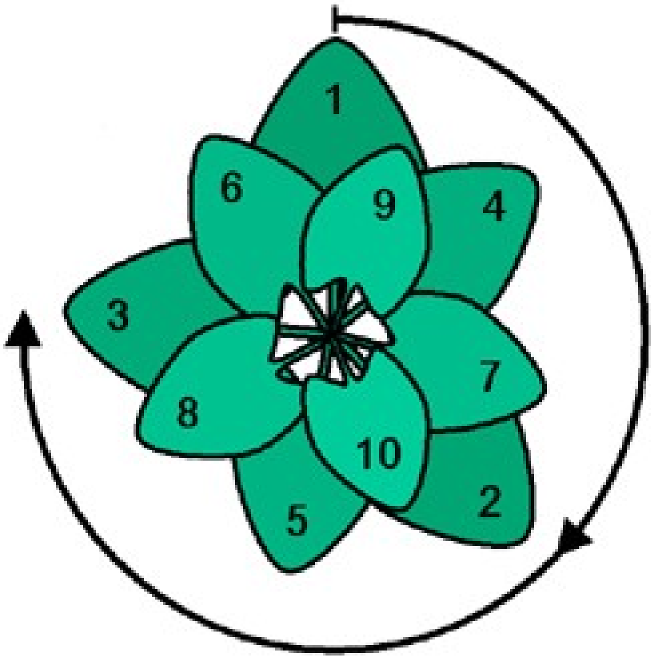
La différence entre les algorithmes est la précision ainsi que la rapidité et la complexité. Les résultats étant trop grands et difficiles à comprendre seront présentés sur le support multimédia, ainsi pendant la soutenance.

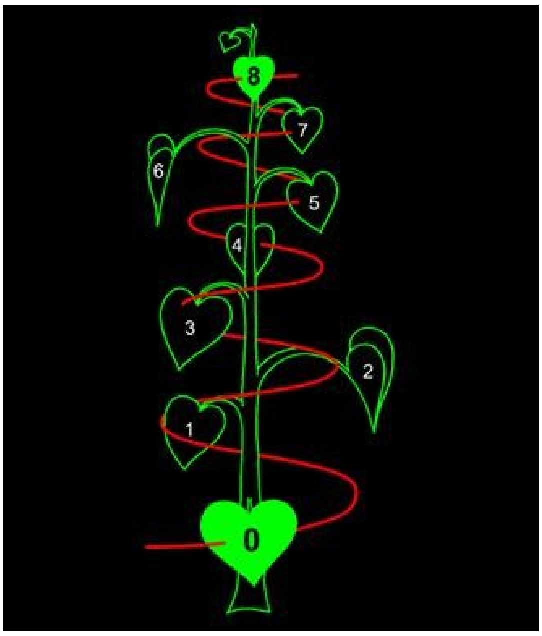
V. Nombre d'or dans la nature :

On peut observer dans la nature le nombre d'or que cela soit à l'état naturel avec le corps humain, les plantes mais aussi dans les constructions de l'homme.

Nous verrons ci-dessous quelques exemples.

1. Les plantes :

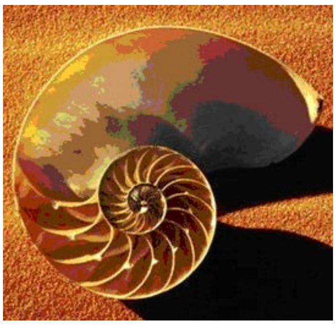




Les feuilles se positionnent très souvent de cette manière : tout les 137° une nouvelle feuille pousse… et ceci correspond à la valeur de l'angle d'or qu'on a vu dans les chapitres précédents. Il semble que ça marche avec 80% des plantes, Selon p.74 de: Jean-François Morot-Gaudry,Biologie végétale: Croissance et développement, Dunod, 2017.

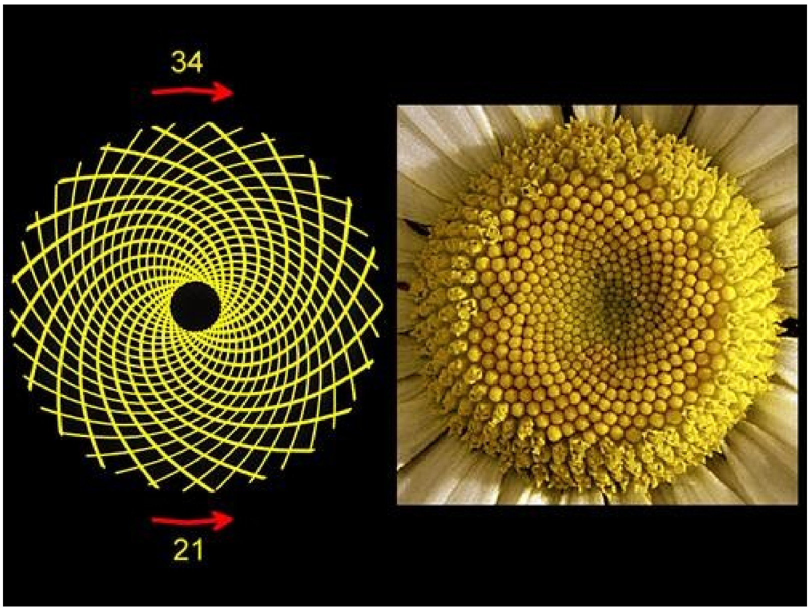
2. Le coquillage du Nautile :

Il grandit en spirale, en suivant la proportion divine. Il se trouve en effet que le rapport entre le diamètre de chaque spirale formant sa coque, et le diamètre de la suivante est égale à Phi.



3. Les pâquerettes :

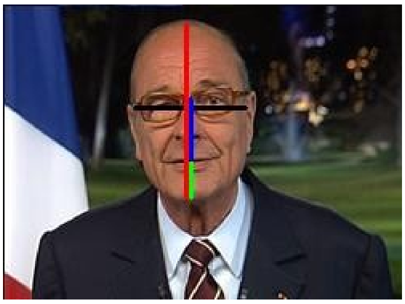
Comme le montre schéma suivant, on trouve deux types de spirales superposées comme chez le tournesol : on a 34 spirales vers la droite et 21 spirales vers la gauche. 34 est le résultat du nombre suivant du rang de Fibonacci qui donne le résultat 21. Or 34/21 = à peu près à 1,61905 qui est égale environ au nombre d'or au millième près.



On retrouve le même concept dans le chou-fleur, la pomme de pin et bien d'autres…

(En 1968 le mathématicien Alfred Brosseau a observé 4290 de pommes de pin et conclu que c'était vrai à 98.2%. Le botaniste Roger V. Jean a refait une telle étude en 1992 avec plus de 12750 pommes de pins de différents genres et il est arrivé à 92% de pives arrangées ainsi. Source: p132 du livre: Le nombre d'or: le langage mathématique de la beauté)

4. L'Homme :



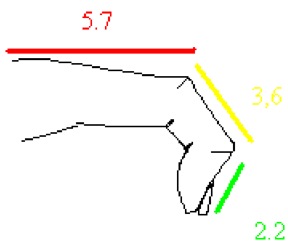
En prenant pour modèle le célèbre portrait de Jacques Chirac, peut on dire que les mesures de sa tête ont un rapport avec le nombre d'or.

hauteur de la tête : 110 px
largeur de la tête : 70 px
hauteur du nez à la bouche : 45 px
hauteur du menton à la bouche : 27 px

*les précision ci-dessus sont en couleur en fonction des traits sur la photo, sur la version papier cela ne sera pas visible si le rapport est en noir et blanc.

En faisant le rapport de ces mesures, l'on trouve :

110/70 = 1.57 cm
70/45 = 1.55 cm
45 / 27 = 1.66 cm

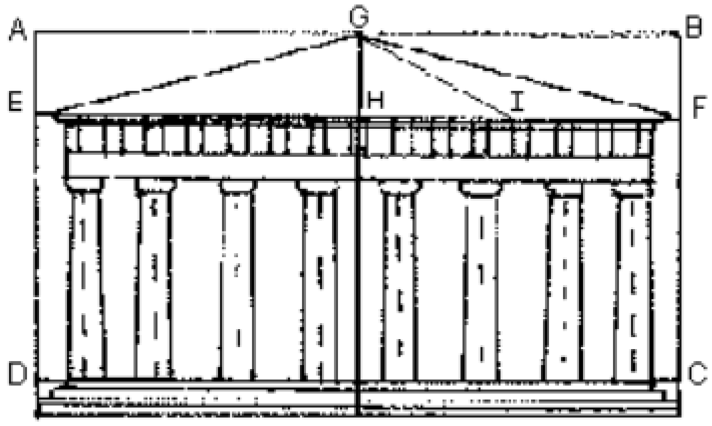


=> 5.7/3.6 = 1.58 ≈ φ
=> 3.6/2.2 = 1.63 ≈ φ

5. Le Parthénon d'Athènes :

Le Parthénon s'inscrit dans un rectangle doré, c'est-à-dire tel que le rapport de la longueur à la hauteur est égal au nombre d'or.
Sur la figure : DC/DE = φ

Sur la toiture du temple, GF/GI = phi.



6. Astronomie :

Nous ne pouvons pas donner d'explication à ce jour. N'empêche que c'est une coïncidence qui a peu de probabilité d'exister juste par du hasard et aussi proche de nous.

7. Carte de crédit :

Enfin si vous voulez visualiser tout de suite un rectangle d’or , vous pouvez prendre votre carte bancaire et la regarder.

En effet ses dimensions sont de - 85,60 × 53,98 mm

Le rapport entre ces dimension nous approche beaucoup du nombre d’or :

$$85.6/53.98 = 1.5858\ldots \text{ cm}$$

VI. Bilan :

Pour commencer, il y a fallu constituer un groupe, notre choix a été naturellement de se mettre ensemble l'un avec l'autre du fait que nous avons déjà réalisé dans le passé des projets ensemble qui s'étaient bien déroulés.

Nous avons dû choisir un sujet, ce n'était pas une mince affaire, finalement nous avons tous les deux choisi quelque chose dont nous avions tous deux un intérêt commun : le nombre d'Or. Étant tous deux curieux de nature, nous connaissions cette curiosité mathématique qu'est le nombre d'or.

Nous avons donc plonger dans la présence du nombre d'Or que cela soit dans les mathématiques ou dans la nature afin d'obtenir une vue globale satisfaisante du sujet. Nous avons cherché sur internet ainsi que dans des livres empruntés à la bibliothèque universitaire de Lorraine. Tout au long de notre parcours dans les recherches nous avons tenu un site web qui est celui-ci : https://matteodelandhuypro.wixsite.com/nombre-d-or

On le mettait à jour de séance en séance tel un carnet de bord. Nous avons réalisé une affiche de format A2 et une pour le web. Dans cette affiche on peut y voir l'intrigue du sujet, un début de réponse, une trace de notre démarche et de nos résultats. Nous avons écris également le rapport que vous êtes en train de lire. Nous avons essayé dans celui-ci de présenter le contexte, une problématique, montrer l'intérêt du nombre d'Or dans les mathématiques et en quoi l'informatique peut aider dans la résolution d'approcher Phi.

Nous sommes très satisfaits de notre travail. Il nous a permis d'approfondir un sujet qu'on ne connaissait que vaguement et nous sommes heureux de pouvoir partager nos recherches avec notre entourage. Pendant les recherches, nous étions comme subjugués de comment s'organise le nombre d'Or dans la nature et les multiples possibilités de calcul pour s'en approcher.

VII. Sources :

1. Bibliographie :

- Le nombre d'or, Claude_jacques Willard
- Le nombre d'or, rites et rythmes pythagoriciens dans le développement de la civilisation occidentale

2. Webographie ;

http://trucsmaths.free.fr/nombre_d_or.htm#historique

https://martouf.ch/2018/07/le-nombre-dor-base-systeme-unite-mesure/

https://www.goldennumber.net/pi-phi-fibonacci/

http://villemin.gerard.free.fr/Wwwgvmm/Nombre/Rac5.htm

http://www.gecif.net/articles/mathematiques/nombre_d_or/

http://debart.pagesperso-orange.fr/histoire/nombre_d_or.html#rect_or

http://tecfaetu.unige.ch/etu-maltt/nestor/gogniac0/stic/Stic_ex16_7/eLML/tecfa/
Le_Nombre_d_or/fr/html/ProprietesGeom_SpiraleOr.html

https://fr.wikipedia.org/wiki/Nombre_d%27or

https://www.podcastscience.fm

3. Avis sur la fiabilité des sources :

Concernant les deux livres, les informations extraites ont été justifiées à chaque fois ainsi qu'elles étaient conformes et cohérentes avec ce que l'on trouvait sur internet. Pour les sites web, c'était souvent des blogs ou des sites conçus par des professeurs ou des élèves à l'issu d'un projet sauf quelques uns qui étaient des sites web d'une organisation ou autre, les informations présentes sur ces sites web étaient à chaque fois les mêmes et aucune information ne nous a paru différente ou pas bien justifiée.

VIII. Annexes :

1. Glossaire :

- Angle d’or
- Approche du nombre d’or
- Architecture et nombre d’or
- L’art et le nombre d’or
- Chaîne dorée
- Équation d’or
- Fraction continues
- Formules de calcul
- Histoire du nombre d’or
- Nombre d’or
- Mathématiques
- Monuments historiques
- Phi
- Pi
- Rectangle d’or
- Spirale d’or
- Suite de Fibonacci
- Le Triangle d’or
- Trigonométrie et nombre d’or

2. Proposition détaillée du sujet :

Pour notre sujet il a fallu tout d’abord comprendre le nombre d’or et ses aspects mathématiques ainsi que dans d’autres domaines. Après vient l’étape des calculs où il a fallu penser à tout les calculs possibles et réussir à les implémenter soit dans une application informatique soit dans une démonstration en plus de la compréhension des formules existantes et essayer d’effectuer des expériences de calcul avec. Ensuite on devait modéliser ces résultats et les analyser pour en tirer une conclusion ainsi que effectuer des recherches sur la présence de ce fameux nombre d’or dans notre vie et dans notre entourage. Par ailleurs, on devait réaliser deux posters pour présenter au mieux notre sujet au lecteur, ainsi qu’un site web disponible dans le bilan pour regrouper tout notre travail et nos comptes rendus. Les tâches ont été séparées et chacun de nous avait à faire ce qu’il aimait et étais fort à faire comme vous allez le voir dans les parties qui suivent.

3. Planning prévisionnel et final :

1. Planning prévisionnel :

Séances TD

- TD 3 : Trouver la problématique principale, trouver les liens avec l'informatique, établir un plan
- TD 4 : Commencer la réalisation de l'affiche sous Photoshop, finaliser l'application informatique
- TD 5 : Mise en page du mémoire, Finalisation du support multimédia
- TD 6 : Rendre les fichiers attendus
- TD 7 & 8 : Soutenance orale

Séances TP

- TP 1 : Construire le blog, et chercher des livres intéressants sur le sujet
- TP 2 : Enrichir le blog (design contenu ...) et chercher les différentes pistes de sujet à aborder dans le nombre d'or
- TP 3 : Développer les parties du plan, et commencer l'application informatique
- TP 4 : Développer les parties du plan ainsi que l'application informatique.
- TP 5 : Développer l'affiche, commencer la rédaction du mémoire
- TP 6: Finaliser l'affiche, Développer le mémoire
- TP 7 : Développer le mémoire, commencer à placer le plan du support multimédia (diapo).

2. Planning Final :

Séances TD

- TD 3 : Trouver la problématique principale, trouver les liens avec l'informatique, établir un plan
- TD 4 : Continuer les recherches et les différentes pistes possible
- TD 5 : Finaliser les applications informatiques
- TD 6 : Réaliser l'affiche, la mise en page du mémoire et commencer le support multimédia
- TD 7 & 8 : Soutenance orale

Séances TP

- TP 1 : Construire le blog, et chercher des livres intéressants sur le sujet
- TP 2 : Enrichir le blog (design contenu ...) et chercher les différentes pistes de sujet à aborder dans le nombre d'or
- TP 3 : Développer les parties du plan, et commencer l'application informatique
- TP 4 : Développer les parties du plan ainsi que l'application informatique
- TP 5 : Commencer la rédaction du mémoire
- TP 6: Continuer sur le mémoire ainsi que les applications informatiques
- TP 7 : Enrichir le mémoire et cherche encore plus d'informations

Comme vous pouvez le constater nous n'avons pas respecté le planning qu'on avait posé car le temps n'était pas suffisant, ainsi que des parties étaient plus longues à réaliser et ont pris plus de temps que prévu, ce qui fait que la moitié du travail été faite chez nous ce qui nous a laissé encore moins de temps pour le reste de travail qu'on avait à faire pour les autres matières.

4. Fonctions et contributions :

Dans notre groupe on s'est séparé les tâches intelligemment, chacun devait faire ce qu'il savait faire au mieux, et ceci était notre plan pour la séparation des tâches et nos contributions :

Mattéo DELANDHUY :

- ✔ Recherches sur le sujet
- ✔ Recherches sur les différentes pistes possibles
- ✔ Réalisation des algorithmes d'approche du nombre d'or
- ✔ Rédaction du mémoire & mise en page
- ✔ Contribution à la réalisation des 2 posters web & papier (informations, contenu,..)
- ✔ Contribution au contenu du site web
- ✔ Réalisation du support multimédia

Ayoub AISSI :

- ✔ Recherches sur le sujet
- ✔ Dégagement des idées possibles à traiter
- ✔ Réalisation d'algorithme de visualisation de la spirale d'or
- ✔ Rédaction du mémoire & mise en page
- ✔ Réalisation des posters (sous Photoshop, Informations, illustrations..)
- ✔ Maintenance du site web et du contenu
- ✔ Réalisation du support multimédia

On peut dire qu'on a tout les deux fais les mêmes tâches mais avec quelques différences sur la quantité de travail sur chaque tâche, mais au final nos rendus se complétaient et étaient bien cohérents.

5. Auto-évaluation et critères :

Sur quels critères pouvons-nous nous baser à fin d'avoir une auto-évaluation correcte ?

Au niveau de l'auto-évaluation, nous avons posé quelques critères et pour chaque critères on a mis une note sur 10, la note final sera la moyenne de toutes les notes :

Critères / Étudiant	Mattéo Delandhuy	AISSI Ayoub
Qualité et pertinence des résultats :	10	10
Régularité de travail :	8	7
Durée de temps pour réaliser la tâche :	8	7
La concentration sur l'objectif :	8	7
Les Recherches effectués :	9	8.5
La capacité à surmonter les problèmes rencontrés :	9	9.5
Répartition intelligente des tâches :	10	10
MOYENNE :	8.85 / 10	8.42/ 10

6. Comptes rendus de séances :

Les comptes rendus des séances TD & TP ne contiennent pas beaucoup d'informations vu que la plus part du temps on passait la séance à chercher des informations ou à résoudre les problèmes de nos programmes vu qu'on était réuni en ces moments là cela nous a permis de vite résoudre nos problèmes et surmonter nos difficultés et une grande partie de progression a été faite chez nous vu qu'on était plus à l'aise. Ce qui fait que les comptes rendus seront combinés et feront donc deux parties, une pour les séances TD et une pour les séances TP.

1. Séances TD :

Les deux premières séances étaient réservées à la recherche des sujets ainsi que la prise d'une décision, on devait se mettre d'accord sur un sujet car on avait une grande liste devant nous et tous les sujets étaient intéressants pour nous deux. Après avoir choisis notre sujet, les séances TD3 & TD 4 ont connu une grande progression, on avait trouvé une problématique principale, les liens avec l'informatique, et on a établi un plan de travail. On également avancé sur les recherches pour notre rendu final. La séances TD5 était consacrée à la finalisation des applications informatiques, il a fallu tester tout les programmes et apporter les dernières retouches pour être sûrs que tout fonctionne comme on le souhaite. La séance TD6 a été une dernière ligne droite, on a commencé à réaliser les deux posters, format papier et web, en plus de la mise en page du mémoire et rajouter des informations dedans. Le support multimédia a été commencé le même jour aussi et a été fini plus tard en dehors des plages horaires du module des maths.

2. Séances TP :

Pour les deux premières séances de TP on a continué notre travail après avoir choisis le sujet et avoir établi le plan et la problématique. On a construit notre site web et on a mis les informations qui étaient nécessaires pour ce jour sur un blog. On a cherché sur la bibliothèque en ligne de l'université de Lorraine des livres intéressants et on les a reservé pour la semaine d'après. On a également enrichi le blog niveau design et contenu et continué notre recherche sur les différentes pistes possibles. Lors du TP3 et 4 on a développé les parties du plan et les différents algorithmes et applications informatiques. Le TP5 a été le début de la rédaction du mémoire, et on a continué sur sa rédaction pendant le 6ème TP ainsi que le développement des application informatiques qui ont pris du temps à cause des différents bugs rencontré lors de la conception. Le TP7 a été un complément pour le mémoire seulement, on a recherché d'autres informations, formules, et calculs pour pouvoir les mettre dans notre rapport.

Les comptes rendus sont également disponibles sur notre site :

https://matteodelandhuypro.wixsite.com/nombre-d-or

7. Algorithmes et programmes :

1. Approche avec Pi

```
from decimal import *

from math import *

def Pi():

    P = 10000

    C = Decimal(4) / 1

    I = 1

    a = 10000

    getcontext().prec = a

    for i in range(P, 0, -1):

        z = Decimal(4) / (I+2)

        if i%2 == 0:

            z = z*(-1)

        C += z

        I += 2


    return C

def approchePhiAvecPi():


    a = 10000

    getcontext().prec = a


    Phi = Decimal(1 - 2*cos(3*Pi()/5))

    print(Decimal(Phi))
```

2. Fibonacci Itératif

```
from decimal import *

import time

def FibonacciIteratif():

        nterms = int(input("Approximation du nombre d'or avec la suite de Fibonacci au rang:"))

        a = int(input("Nombre de chiffre significatif:"))

        n1 = 0

        n2 = 1

        count = 0

        x = time.clock()

        if nterms <= 0:

                print("Rentrez un nombre positif!")


        elif nterms == 1:

                print("rang 1:",nterms,":")

                print(n1)

        else:

                while count < nterms:

                      getcontext().prec = a

                        temp = n1 + n2


                        n1 = n2

                        n2 = temp

                        count += 1

                        y=time.time

        print("rang: ",count,":",Decimal(n2)/Decimal(n1))

        print("temps mis: ",time.clock()-x)
```

3. Fibonacci Récursif

```
import sys


def fibonacci(n):
    print(n)


    if n < 2:
        return n
    else:
        return fibonacci(n-1)+fibonacci(n-2)
def main():
    print(fibonacci(20))


main()
```

4. Fraction d’or

```
from decimal import *

import time

def FractionOr():

        nterms = int(input("Approximation du nombre d'or avec la suite infernale de fraction avec des 1:"))

        a = int(input("Nombre de chiffre significatif:"))

        count = 0

        count2= 0

        resultat=""

        x = time.clock()

        if nterms <= 0:

                print("Rentrez un nombre positif!")

        elif nterms == 1:

                print("rang 1:",nterm,":")

                print(n1)

        else:

                while count < nterms:

                        getcontext().prec = a

                        resultat=resultat+"1/(1+"

                        count += 1

                resultat=resultat[:-1]

                while count2 < nterms:

                        resultat=resultat+")"

                        count2 = count2 + 1

                print("rang: ",count,":",Decimal(eval(resultat)))

                print("temps mis: ",time.clock()-x)
```

5. Par approche racine de 5

```
from decimal import *

import time

x = time.clock()

def lrac2():

        nterms = int(input("Approximation de la racine carré de 5"))

        a = int(input("Nombre de chiffre significatif:"))

        count = 0

        n1 = '9/4'

        getcontext().prec = a

        if nterms <= 0:

                print("Rentrez un nombre positif!")

        elif nterms == 1:

                print("rang 1:",nterms,":")

               print(n1)

        else:

                while count < nterms:

                        ##print("bonjour")

                        getcontext().prec = a

                        if (count < 1):

                                ##print("n1: ",n1)

                                n2 = n1[::-1]

                                ##print("n2: ",n2)

                        if (count >= 1):

                                ##print("n1: ",n1)

                                n2 = "2/(n3 +5*n4)"

                                ##print("n2: ",n2)

                        n3 = Decimal(eval((n1)))

                        ##print("n3: ",n3)

                        n4 = Decimal(eval((n2)))

                        ##print("n4: ",n4)
```

```
                    n5= (n3+5*n4)/2
                    ##print("n5: ",n5)
                    n1 = "(n3+5*n4)/2"
                    count = count+1
        ##print("rang: ",count,":",n5)
        return n5
z = lrac2()
print(Decimal((1+z)/2))
print("temps mis: ",time.clock()-x)
```

6. Affichage graphique de la spirale d'or

(Il faut avoir les deux codes)

CODE 1 : fibonacci.java

```
package testFibo;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import java.awt.Color;
import java.awt.Graphics;
import java.*;

public class fibonacci {

	public static int SCALE = 4;
	public static void main(String args[]) throws InterruptedException{

		//adding comments because need to make changes
		Scanner keyboard = new Scanner(System.in);
		System.out.print("Enter number upto which Fibonacci series to print: ");
		int number = keyboard.nextInt() + 1;
		System.out.print("Enter the scale you would like to use: ");
		SCALE = keyboard.nextInt();
		System.out.println();
		System.out.println();
```

```
        //System.out.print("Fibonacci series upto " + number +" numbers : ");
        int[] fibNums = getNumbers(number);


        drawSpiral(fibNums);

}
public static int[] getNumbers(int num) throws InterruptedException{

        int[] nums = new int[num];

        for (int i = 1; i<=num; i++){
                int currentNum = fibonacciRecursion(i);
                //System.out.print(currentNum + " ");
                nums[i-1] = currentNum;
        }

        return nums;
}

public static int fibonacciRecursion(int num) throws InterruptedException{

        if(num == 1 || num == 2)
                return 1;

        else
                return fibonacciRecursion(num-1)+ fibonacciRecursion(num-2);

}

public static void drawSpiral(int[] nums) throws InterruptedException{

        DrawingPanel panel = new DrawingPanel(1300, 700);
        Graphics g = panel.getGraphics();
```

```
        int xPos = panel.width/2-200;
        int yPos = panel.height/2+150;

        for(int i = 0; i<nums.length-1; i++){
              int seqNum = (i+1)%4;

              drawFromCorner(xPos,yPos,nums[i],seqNum,g);
              g.drawString(nums[i] + "", 100 , 100 + 15*i);

              try {
                  Thread.sleep(3000);
              } catch(InterruptedException ex) {
                  Thread.currentThread().interrupt();
              }

              if(i!=nums.length-1){
              if(seqNum == 1)
                    xPos-=nums[i]*SCALE + nums[i+1]*SCALE;
              else if (seqNum == 2)
                    yPos-=nums[i]*SCALE + nums[i+1]*SCALE;
              else if (seqNum == 3)
                    xPos+=nums[i]*SCALE + nums[i+1]*SCALE;
              else if (seqNum == 0)
                    yPos+=nums[i]*SCALE + nums[i+1]*SCALE;
        }
        }

  }

  public static void drawFromCorner(int xPos, int yPos,int size, int corner,
Graphics g){
        Color c = new Color(228,228,35);
        c= Color.BLACK;

        if(corner == 3){
```

```
                g.drawRect(xPos, yPos, size*SCALE, size*SCALE);
                g.setColor(c);
                g.drawArc(xPos, yPos, size*SCALE*2, size*SCALE*2, -180, -90);
            }
            else if (corner == 0){
                xPos-=size*SCALE;
                g.drawRect(xPos, yPos, size*SCALE, size*SCALE);
                g.setColor(c);
                g.drawArc(xPos-size*SCALE, yPos, size*SCALE*2, size*SCALE*2, -270, -
90);
            }
            else if (corner == 1){
                xPos-=size*SCALE;
                yPos-=size*SCALE;
                g.drawRect(xPos, yPos, size*SCALE, size*SCALE);
                g.setColor(c);
                g.drawArc(xPos-size*SCALE, yPos-size*SCALE, size*SCALE*2,
size*SCALE*2, 0, -90);

            }
            else if (corner == 2){
                yPos-=size*SCALE;
                g.drawRect(xPos, yPos, size*SCALE, size*SCALE);
                g.setColor(c);
                g.drawArc(xPos, yPos-size*SCALE, size*SCALE*2, size*SCALE*2, -90, -
90);
            }

            g.setColor(Color.BLACK);
        }



}
```

CODE 2 : DrawingPanel.java

```
/*
```

```
Stuart Reges and Marty Stepp
07/01/2005

The DrawingPanel class provides a simple interface for drawing persistent
images using a Graphics object.  An internal BufferedImage object is used
to keep track of what has been drawn.  A client of the class simply
constructs a DrawingPanel of a particular size and then draws on it with
the Graphics object, setting the background color if they so choose.

To ensure that the image is always displayed, a timer calls repaint at
regular intervals.
*/
package testFibo;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.swing.*;
import javax.swing.event.*;

public class DrawingPanel implements ActionListener {
    public static final int DELAY = 250;  // delay between repaints in millis

    private static final String DUMP_IMAGE_PROPERTY_NAME = "drawingpanel.save";
    private static String TARGET_IMAGE_FILE_NAME = null;
    private static final boolean PRETTY = true;  // true to anti-alias
    private static boolean DUMP_IMAGE = true;  // true to write DrawingPanel to file

    int width;    // dimensions of window frame

       int height;
    private JFrame frame;         // overall window frame
    private JPanel panel;         // overall drawing surface
    private BufferedImage image;  // remembers drawing commands
    private Graphics2D g2;        // graphics context for painting
```

```
    private JLabel statusBar;     // status bar showing mouse position
    private long createTime;

    static {
        TARGET_IMAGE_FILE_NAME = System.getProperty(DUMP_IMAGE_PROPERTY_NAME);
        DUMP_IMAGE = (TARGET_IMAGE_FILE_NAME != null);
    }

    // construct a drawing panel of given width and height enclosed in a window
    public DrawingPanel(int width, int height) {
        this.width = width;
        this.height = height;
        this.image = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);

        this.statusBar = new JLabel(" ");
        this.statusBar.setBorder(BorderFactory.createLineBorder(Color.BLACK));

        this.panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 0, 0));
        this.panel.setBackground(Color.WHITE);
        this.panel.setPreferredSize(new Dimension(width, height));
        this.panel.add(new JLabel(new ImageIcon(image)));

        // listen to mouse movement
        MouseInputAdapter listener = new MouseInputAdapter() {
            public void mouseMoved(MouseEvent e) {
                DrawingPanel.this.statusBar.setText("(" + e.getX() + ", " + e.getY() +
")");
            }

            public void mouseExited(MouseEvent e) {
                DrawingPanel.this.statusBar.setText(" ");
            }
        };
        this.panel.addMouseListener(listener);
        this.panel.addMouseMotionListener(listener);
```

```
        this.g2 = (Graphics2D)image.getGraphics();
        this.g2.setColor(Color.BLACK);
        if (PRETTY) {
            this.g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
            this.g2.setStroke(new BasicStroke(1.1f));
        }

        this.frame = new JFrame("CS307 Drawing Panel");
        this.frame.setResizable(false);
        this.frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if (DUMP_IMAGE) {
                    DrawingPanel.this.save(TARGET_IMAGE_FILE_NAME);
                }
                System.exit(0);
            }
        });
        this.frame.getContentPane().add(panel);
        this.frame.getContentPane().add(statusBar, "South");
        this.frame.pack();
        this.frame.setVisible(true);
        if (DUMP_IMAGE) {
            createTime = System.currentTimeMillis();
            this.frame.toBack();
        } else {
            this.toFront();
        }

        // repaint timer so that the screen will update
        new Timer(DELAY, this).start();
    }

    // used for an internal timer that keeps repainting
```

```
public void actionPerformed(ActionEvent e) {
    this.panel.repaint();
    if (DUMP_IMAGE && System.currentTimeMillis() > createTime + 4 * DELAY) {
        this.frame.setVisible(false);
        this.frame.dispose();
        this.save(TARGET_IMAGE_FILE_NAME);
        System.exit(0);
    }
}

// obtain the Graphics object to draw on the panel
public Graphics2D getGraphics() {
    return this.g2;
}

// set the background color of the drawing panel
public void setBackground(Color c) {
    this.panel.setBackground(c);
}

// show or hide the drawing panel on the screen
public void setVisible(boolean visible) {
    this.frame.setVisible(visible);
}

// makes the program pause for the given amount of time,
// allowing for animation
public void sleep(int millis) {
    try {
        Thread.sleep(millis);
    } catch (InterruptedException e) {}
}

// take the current contents of the panel and write them to a file
public void save(String filename) {
```

```
        String extension = filename.substring(filename.lastIndexOf(".") + 1);

        // create second image so we get the background color
        BufferedImage image2 = new BufferedImage(this.width, this.height,
BufferedImage.TYPE_INT_RGB);
        Graphics g = image2.getGraphics();
        g.setColor(panel.getBackground());
        g.fillRect(0, 0, this.width, this.height);
        g.drawImage(this.image, 0, 0, panel);

        // write file
        try {
            ImageIO.write(image2, extension, new java.io.File(filename));
        } catch (java.io.IOException e) {
            System.err.println("Unable to save image:\n" + e);
        }
    }

    // makes drawing panel become the frontmost window on the screen
    public void toFront() {
        this.frame.toFront();
    }
}
```