



## Jupyter Notebook : Un outil interactif pour l'apprentissage des mathématiques

Mercredi 21 août 2024

# Présentation

- Cécile Lombart : conseillère aux études pour les *Sciences Informatiques*
  
- Nicolas Leblanc : responsable des relations avec le secondaire, partie « *mathématiques* »



# Facilement compréhensible ?

```
1 import numpy as np
2 from scipy.stats import binom
3 import matplotlib.pyplot as plt
4
5 # Paramètres de l'expérience
6 n = 50 # nombre d'essais
7 p = 0.56 # probabilité de succès
8 X = 31 # nombre minimum de succès
9
10 # Calcul de la probabilité d'avoir au moins X succès
11 prob_at_least_X = 1 - binom.cdf(X-1, n, p)
12
13 print(f"La probabilité d'avoir au moins {X} succès sur {n} essais est de {prob_at_least_X:.5f}")
14
15 # Nettoyage de la figure précédente
16 plt.clf()
17 plt.close()
18
19 # Visualisation de la distribution binomiale
20 x = np.arange(0, n+1)
21 pmf_values = binom.pmf(x, n, p)
22
23 plt.bar(x, pmf_values, color='lightblue')
24 plt.xlabel('Nombre de succès (k)')
25 plt.ylabel('Probabilité P(X=k)')
26 plt.title('Distribution Binomiale: n=%d, p=%.2f' % (n, p))
27 plt.axvline(X, color='red', linestyle='--', label=f'X = {X}')
28
29
30 # Surligner la probabilité d'avoir au moins X succès
31 plt.bar(x[X:], pmf_values[X:], color='orange', label=f'P(X >= {X}) = {prob_at_least_X:.3f}')
32 plt.legend()
33
34 plt.show()
```



Jupyter Notebook (JPN)



Python

**Plan**

1. Introduction et exemples
2. Apprendre les bases en Python
3. JPN comme élève : Régression linéaire
4. JPN comme prof : Loi binomiale



- Jupyter Notebook à l'université, exemples :
  - *LEPL1102 : Analyse I*
  - *LINMA1702 : Méthodes et modèles d'optimisation*
  - *LGBIO2050 : Medical Imaging*
  - ***LSINC1112 : Algèbre linéaire***





- Python en secondaire, exemples Dédra-MATH-isons :



## Les cycloïdes

GELOT Oscar, JANDRI Kevin, GHAFOURI Adrian, BUZZO Federico



Collège Saint-Michel d'Etterbeek

Dédra-MATH-isons  
11 Avril 2024



- Python en secondaire, exemples Dédra-MATH-isons :

## 4.2 Python

### 4) Code

#### 4.1 Geogebra

$O = \begin{pmatrix} \theta \\ 0 \end{pmatrix}$   
 $= (0, 0^\circ)$

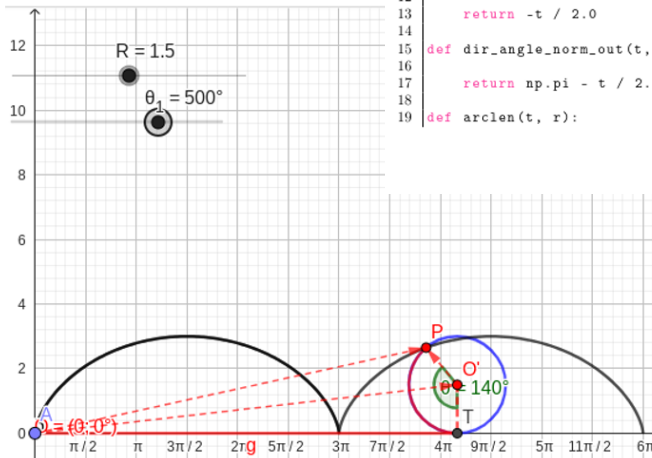
$R = 1.5$   
 $1 \text{ --- } 2$

$\theta_1 = 500^\circ$   
 $0^\circ \text{ --- } 720^\circ$

$O' = (R \theta_1, R)$   
 $= (13.09, 1.5)$

$T = (R \theta_1, 0)$   
 $= (13.09, 0)$

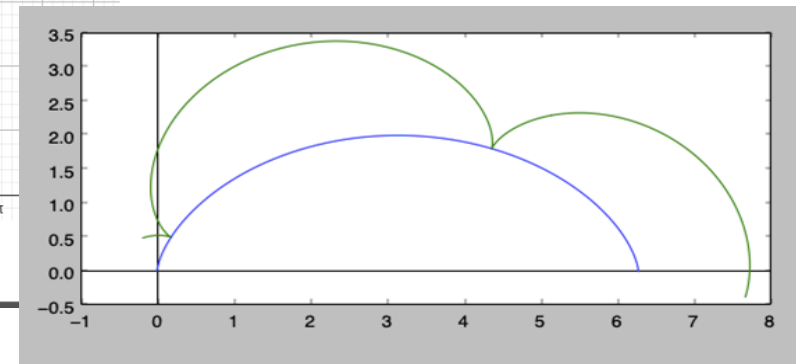
$a = \text{Courbe}(R(\theta - \sin(\theta)), R(1 - \cos(\theta)), \theta, 0, 2\pi)$   
 $\left. \begin{matrix} x = 1.5(\theta - \sin(\theta)) \\ y = 1.5(1 - \cos(\theta)) \end{matrix} \right\} 0 \leq \theta \leq 6.28$

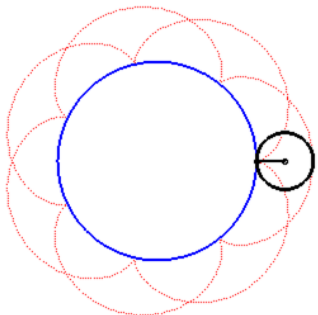
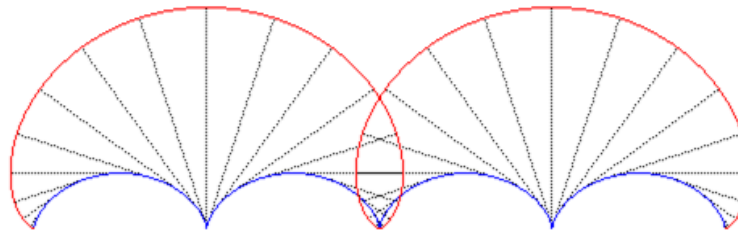
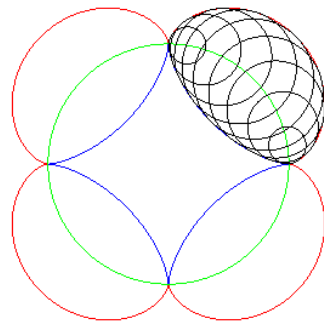
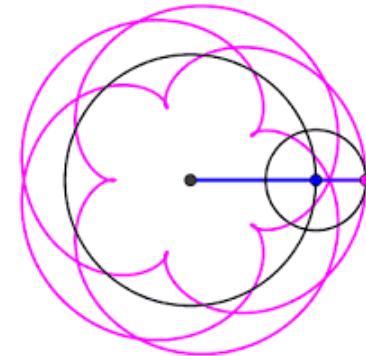
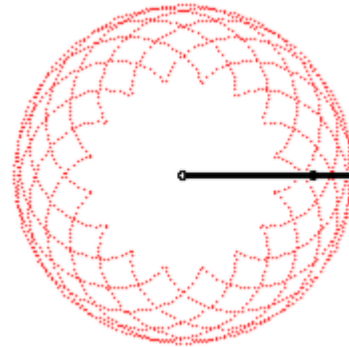
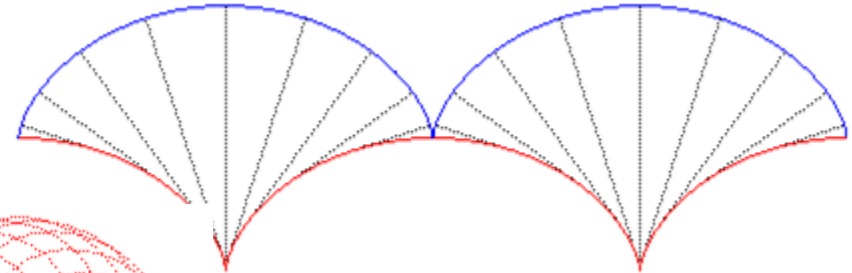
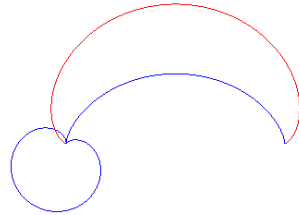
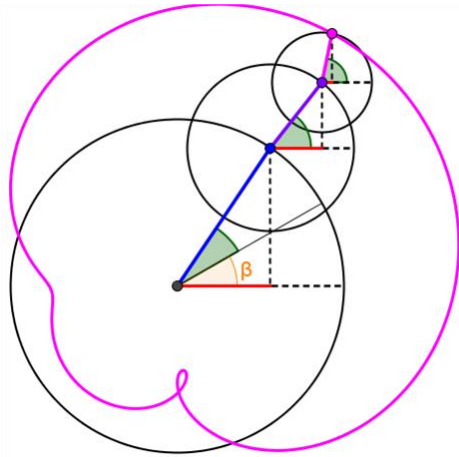


```

1 import numpy as np
2
3 def x(t, r):
4     return r * (t - np.sin(t))
5
6 def y(t, r):
7     return r * (1.0 - np.cos(t))
8
9 def dir_angle_norm_in(t, r):
10
11     return -t / 2.0
12
13 def dir_angle_norm_out(t, r):
14
15     return np.pi - t / 2.0
16
17 def arclen(t, r):
18
19
20     return 4.0 * r * (1.0 - np.cos(t / 2.0))
21
22 def xy_roulette(t, r, T, R):
23
24
25     px, py = x(t, r), y(t, r)
26
27     a1 = dir_angle_norm_out(t, r)
28
29     cx, cy = px + R * np.cos(a1), py + R * np.sin(a1)
30
31     d = arclen(t, r) - arclen(T, r)
32
33
34     a2 = dir_angle_norm_in(t, r) - phi
35
36     fx, fy = cx + R * np.cos(a2), cy + R * np.sin(a2)
37
38     return fx, fy
39
40
41 import matplotlib.pyplot as plt
42
43 r = 1
44 R = 0.75
45 T = np.pi / 3
46

```





- Python en secondaire, exemples Dédra-MATH-isons :

# UN PREMIER PAS DANS LE JARDIN DES NOMBRES...

Arnaud Stienon, Martin Theunissen, Gareth Colin, Alexia Peeters & Marie Colantoni

MATH 6H

Notre-Dame des Anges (Genval)



- Python en secondaire, exemples Dédra-MATH-isons :

## 2.1. Notre crible d'Eratostène

```

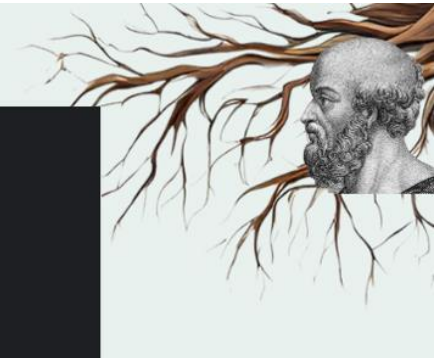
1 import math
2
3
4 usage
5
6 def crible_eratosthene(n):
7
8     L = [i for i in range(2, n + 1)] # Liste contenant les nombres de 2 à n+1
9     P = [] # Liste pour stocker les nombres premiers
10
11     while len(L) != 0:
12         # Ajoute le premier élément de L à la liste des nombres premiers
13         P.append(L[0])
14         i = L[0]
15
16         # Boucle qui passe sur chaque élément de la liste totale et qui supprime tous les multiples de l'élément
17         for k in L:
18             if k % i == 0:
19                 del L[L.index(k)]
20
21     return P
22
23 total = int(input("En dessous de quel nombre veux tu trouver les nombres premiers?: "))
24 resultat = crible_eratosthene(total)
25
26 print(resultat)

```

```

En dessous de quel nombre veux tu trouver les nombres premiers?: 100
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89]

```



×	2	3	×	5	×	7	×	×	×
11	×	13	×	×	×	17	×	19	×
×	×	23	×	×	×	×	×	29	×
31	×	×	×	×	×	37	×	×	×
41	×	43	×	×	×	47	×	×	×
×	×	53	×	×	×	×	×	59	×
61	×	×	×	×	×	67	×	×	×
71	×	73	×	×	×	×	×	79	×
×	×	83	×	×	×	×	×	89	×
×	×	×	×	×	×	97	×	×	×

- Python en secondaire, exemples Dédra-MATH-isons :



## 2.2. Wilson

Wilson a mis au point un théorème, qui peut nous servir à faire un test de primalité, qui dit que :

**Un entier  $p$  strictement plus grand que 1 est un nombre premier si et seulement s'il divise  $(p - 1)! + 1$**

```

1  def theoreme_de_wilson(n):
2
3      while True:
4
5          # Initialisation de la variable "factorielle"
6          factorielle = 1
7
8          # Boucle qui va multiplier les nombres entre 1 et n-1, pour calculer (p-1)!
9          for i in range(1, n):
10             factorielle = factorielle * i
11
12             # Test si (p-1)! + 1 est divisible par n et imprime le résultat
13             if (factorielle + 1) % n == 0:
14                 print(str(n) + " est premier")
15
16             else:
17                 print(str(n) + " n'est pas premier")
18
19
20  while True:
21      ans = int(input("Quel nombre voulez-vous tester?: ")) # Demande le nombre qui va être testé
22      theoreme_de_wilson(ans)

```

```

Quel nombre voulez-vous tester?: 18747
18747 n'est pas premier
Quel nombre voulez-vous tester?: 10453
10453 est premier
Quel nombre voulez-vous tester?: 101
101 est premier
Quel nombre voulez-vous tester?: 100
100 n'est pas premier

```



- Python en secondaire, exemples Dédra-MATH-isons :



- Python en secondaire, exemples Dédra-MATH-isons :

## Programme python: Codage

Texte codé

```
def decodage(t, ia, o):  
  
    ref=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r',  
        's','t','u','v','w','x','y','z',' ','à','_']  
    for i in range(len(t)):  
        for j in range(len(ref)):  
            if t[i]==ref[j] :  
                k=(ia*(j-b))%29  
                t[i]=ref[k]  
                break  
    return ''.join(t)
```

Texte clair





## Pourquoi utiliser les Jupyter Notebook ?

- Introduction à l'informatique et à Python
  - Présent dans une multitude de domaines
- Format accessible et pédagogique
- Facilité de prise en main



# Introduction à Python

## Jupyter Notebook comme introduction à la programmation

- <https://tinyurl.com/jpn-epl>
- <https://notebook.basthon.fr/>
- ***.JPN Intro***



**JPN comme élève**

## Jupyter Notebook comme utilisateur / élève

- *Régression linéaire et corrélation*
- ***.JPN Student***



**JPN comme prof**

## Jupyter Notebook comme prof

➤ *Probabilités : Loi Binomiale*

➤ *.JPN Prof*



## ***Intéressé·es d'utiliser un JPN dans (une de) vos classes ?***

- Votre expérience et vos retours nous sont précieux !
- Nous mettons à votre disposition les JPN et vous proposons un suivi pendant l'année afin d'améliorer et d'adapter au mieux le dispositif !
- [cecile.lombart@uclouvain.be](mailto:cecile.lombart@uclouvain.be) et [nicolas.leblanc@uclouvain.be](mailto:nicolas.leblanc@uclouvain.be)





**Merci !**