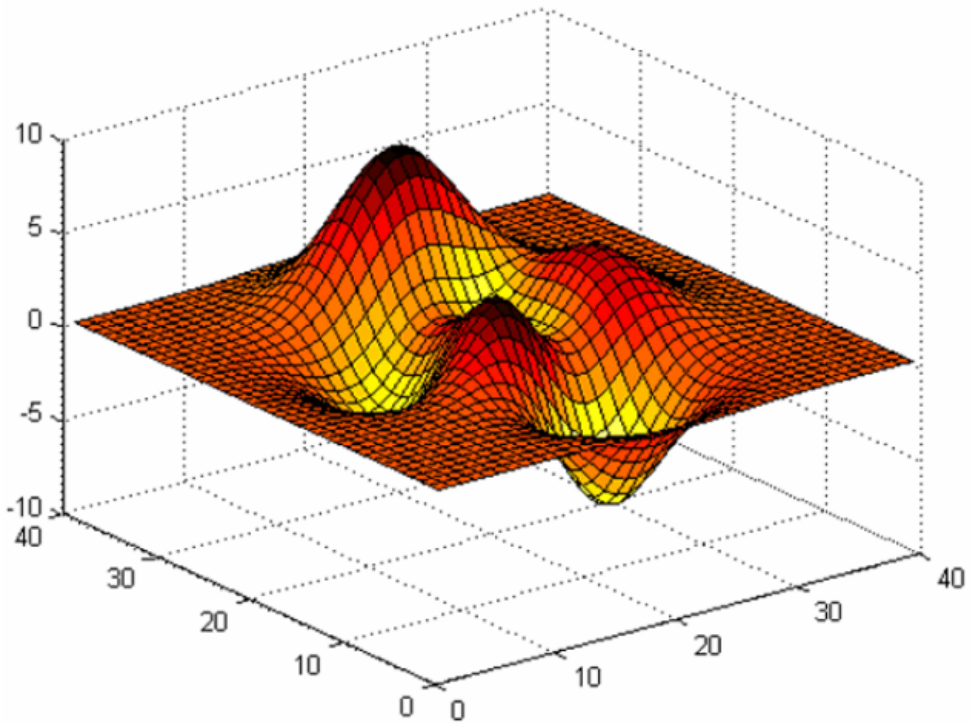
Guide Python
Guide pratique & Exercices corrigés



José OUIN - www.joseouin.fr

.

José OUIN

Ingénieur I.N.S.A Toulouse
Professeur Agrégé de Génie civil
Professeur Agrégé de Mathématiques

Site Internet de ressources pédagogiques : www.joseouin.fr

José OUIN - www.joseouin.fr

Document rédigé avec LaTeX
José OUIN - www.joseouin.fr

José OUIN - www.joseouin.fr

Python : Guide pratique & Exercices corrigés

Table des matières

José OUIN - www.joseouin.fr

José OUIN - www.joseouin.fr

José OUIN - www.joseouin.fr

José OUIN - www.joseouin.fr

2 Calculs et Opérateurs

2.1 Exemples de calculs avec Python

Voici quelques exemples commentés de calculs dans l'éditeur de texte. Les commentaires sont toujours précédés du symbole # afin de ne pas être interprétés par Python. Le symbole ↪ dans le code source indique que la ligne a été coupée pour les besoins du traitement de texte. Dans le code source, il ne s'agit que d'une seule et même ligne.

```
1  a = 5
2  # a prend la valeur 5 (5 est affecté à a)
3  # le symbole '=' est une affectation de la droite vers la gauche
4  b = a
5  # b prend la valeur a donc b vaut 5
6  c = a + b
7  # c prend la valeur de la somme a + b, donc c vaut 10
8  a = a + 4
9  # a prend la valeur 5 + 4 = 9. L'ancienne valeur 5 est perdue
10 d = a*5 + 3
11 # d prend la valeur 9*5 + 3 = 48
12 f = b + 3
13 # f prend la valeur 5 + 3 = 8
14 # b a gardé sa valeur 5 donnée par a à la ligne 3. Le fait de changer la
   ↪  valeur de a à la ligne 7 n'affecte pas la valeur de b.
```

2.2 Les opérateurs avec Python

Le tableau ci-dessous dresse la liste des opérateurs mathématiques et logiques :

Opérateur	Définition	Logique	Définition
=	affectation	==	égalité
+	addition	< ; <=	inférieur ; inférieur ou égal
−	soustraction	> ; >=	supérieur ; supérieur ou égal
*	multiplication	!=	différent
/	division	and	ET logique
**	puissance	or	OU logique
//	quotient d'une division	not	négation
%	reste d'une division		

José OUIN - www.joseouin.fr

Remarque : le signe '+' permet d'effectuer une somme mais aussi de concaténer des chaînes de caractères. Exemple :

```
a = "Bonjour "
# a est une chaîne de caractère (str en Python (string))
b = "Monsieur"
# b est également une chaîne de caractères
c = a + b
# c est la chaîne : "Bonjour Monsieur"
```

3 Saisie et affichage des variables

3.1 Saisie des variables : la fonction input()

Les noms de variables sont des noms que vous choisissez. Ce sont des suites de lettres (non accentuées) et/ou de chiffres. Le premier caractère est obligatoirement une lettre (le caractère _ est considéré comme une lettre). Python respecte la casse (il distingue les minuscules des majuscules).
Les cinq principaux types de variables sont les suivants :

- bool : variable booléenne True ou False (bool, abréviation de boolean) ;
- int : les entiers (int, abréviation de integer) ;
- float : les flottants ou réel (float, signifie flottant) ;
- str : les chaînes de caractères (str, abréviation de string)
- complex : nombres complexes de la forme $a+jb$ (j est le symbole par défaut) avec $j^2 = -1$

A partir de la version 3 de python, la fonction input() renvoie une chaîne de caractères (str). Si l'on souhaite obtenir un entier ou un réel, on doit transtyper la valeur lue au clavier par la fonction input().
Les lignes de codes ci-après montre les différents types rencontrés. La fonction type() renvoie le type de variable. La fonction eval() permet d'évaluer la chaîne de caractère renvoyée par la fonction input() soit par un entier (int) ou soit par un réel (float).

José OUIN - www.joseouin.fr

```
>>> a = True
>>> type(a)
<class bool> # a est une variable booléenne (bool)
>>> a= 45
>>> type(a)
<class int> # a est un entier (int)
>>> a = 45.124
>>> type(a)
<class float> # a est un réel (float)
>>> a = "Bonjour Monsieur"
>>> type(a)
```

Exemple :

```
from cmath import *
angle = pi/2

print("Valeur de l'angle : ", round(angle,3), " radians")
# Affichage du résultat avec 3 décimales.

Console :
Valeur de l'angle :  1.571  radians
```

La fonction `int()` permet d'effectuer la troncature à l'unité.
Exemple :

```
from cmath import *
angle = pi/2

print("Valeur de l'angle : ", int(angle), " radians")
# Affichage de la troncature à l'unité.

Console :
Valeur de l'angle :  1  radians

>>> int(45.4589)
45
```

3.9 Chargement des bibliothèques

Les bibliothèques sont chargées dans un programme Python par la commande : **import**.
Voici quelques exemples :

www.joseouin.fr

```
from cmath import *
from math import *
import numpy as np
from random import *

z = 2 +3j
# z est un nombre complexe

y = cos(pi)
# y vaut -1

>>> e
2.718281828459045
```

```
>>> pi
3.141592653589793
# constantes mathématiques

>>> a = cos(pi/4)
>>> a
0.7071067811865476
>>> alpha = acos(a)
>>> alpha
0.7853981633974483
# alpha est en radians
>>>degrees(alpha)
45.0
# On a converti les radians en degrés
>>> radians(45)
0.7853981633974483
# On a converti les degrés en radians

k = randint(1,6)
# randint(1,6) retourne en entier compris entre 1 et 6 inclus
# Ceci permet d'effectuer la simulation du lancer d'un dé équilibré à 6 faces.
p = uniform(0,1)
# uniform(0,1) renvoie un nombre réel aléatoire compris strictement entre 0 et 1.
```

3.10 Opérations avec les nombres complexes

Les lignes de codes ci-après donnent des exemples d'utilisation des fonctions relatives aux nombres complexes :

www.joseouin.fr

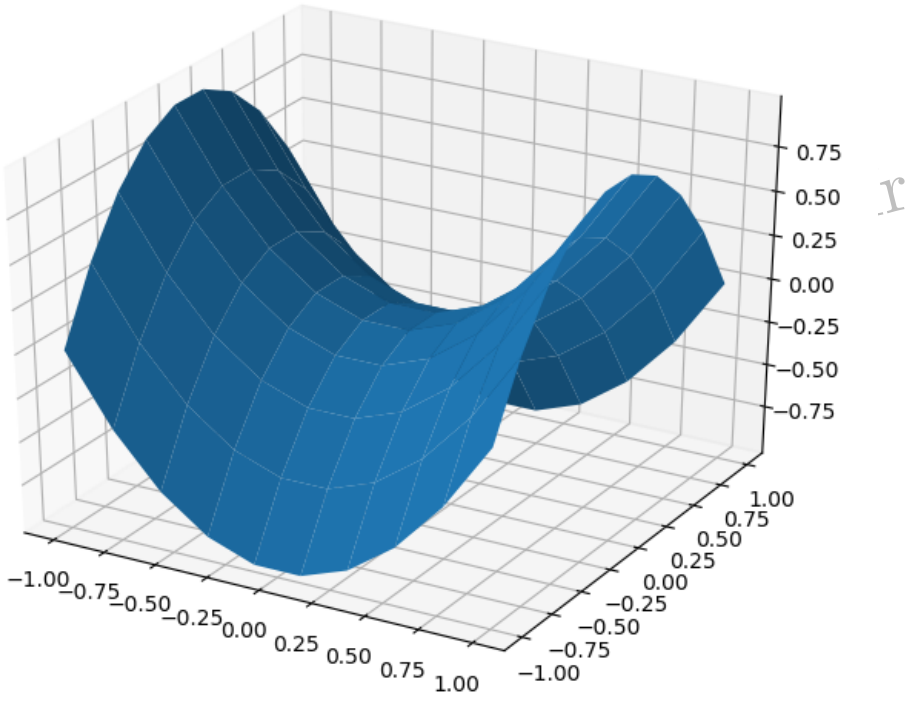
```
>>> from cmath import *
# cmath est la bibliothèque relative aux nombres complexes

>>> z = 3 + 4j
# Saisie d'un nombre complexe (ici i est noté j en langage Python)

>>> z.real
3.0
# calcul de la partie réelle
>>> z.imag
4.0
# calcul de la partie imaginaire

```

7.2.4 Représenter une surface de l'espace $z = f(x,y)$

On considère la fonction de 2 variables : $f(x,y) = x^2 - y^2$

```
from math import *
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def f(x,y):
    return x**2 - y**2

ax = Axes3D(plt.figure())

X = np.linspace(-1,1,10)
Y = np.linspace(-1,1,10)

X, Y = np.meshgrid(X, Y)
Z = f(X, Y)
ax.plot_surface(X, Y, Z)
plt.show()
```

On obtient le graphique suivant :



L'instruction `np.linspace(-1,1,10)` permet de définir un tableau (array) de 10 valeurs allant de -1 à $+1$.

8 Algèbre linéaire avec Python

La bibliothèque Numpy est chargée à l'aide de la commande `import numpy as Nom`

8.1 Dimension d'une matrice

On souhaite déterminer la dimension de la matrice suivante, c'est-à-dire le nombre de lignes et de colonnes de cette matrice :

$$A = \begin{pmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \end{pmatrix}$$

```
import numpy as np
A = np.array([[1, 3, 3],[1, 4, 3]])
print("Dimension de A : ", A.shape)

# On obtient :
Dimension de A :  (2, 3)
# A comporte 2 lignes et 3 colonnes
```

8.2 Multiplication de deux matrices

On souhaite effectuer la multiplication des matrices A et B :

$$A = \begin{pmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{pmatrix} \text{ et } B = \begin{pmatrix} 0 & 2 & 1 \\ 7 & -3 & 2 \\ 2 & -1 & 1 \end{pmatrix}$$

On rappelle que la multiplication des matrices n'est pas commutative :

$$A * B \neq B * A$$

José OUIN - www.joseouin.fr

```
import numpy as np
A = np.array([[1, 3, 3],[1, 4, 3],[1, 3, 4]])
B = np.array([[0, 2, 1],[7, -3, 2],[2, -1, 1]])

C = A.dot(B)
D = B.dot(A)

print("A*B = ",C)
print("B*A = ",D)

```

9.1 Saisir et afficher des variables

9.1.1 EXB-1

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir l'année de naissance d'un utilisateur ;
2- Afficher l'âge correspondant. Par exemple : "Vous avez 21 ans".

Solution

```
a = eval(input("Entrer l'année de votre naissance : "))
annee = 2020
print("Vous avez : ", 2020 - a, " ans.")
```

9.1.2 EXB-2

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir un entier compris entre 0 et 500 ;
2- Afficher sa parité. Par exemple : " Le nombre 452 est pair".
Aide :

```
>>> 14%2
0
# Le reste de la division de 14 par 2 est égal à 0
>>> 15%2
1
# Le reste de la division de 15 par 2 est égal à 1
```

OUIN - www.joseouin.fr

Solution

```
a = int(input("Entrer un entier compris entre 0 et 500 : "))
if a%2 == 0 :
    print("Le nombre ", a, " est pair.")
else :
    print("Le nombre ", a, " est impair.")
```

9.1.3 EXB-3

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir le nom de l'utilisateur (exemple : Bidule) ;
2- Saisir le sexe de l'utilisateur (M ou F) ;
3- Afficher le texte de bienvenue "Bonjour Monsieur/Madame Bidule" selon le cas.

Solution

```
nom = input("Entrer votre nom : ")
s = input("Sexe M/F : ")

if s == "M" :
    print("Bonjour Monsieur ", nom)
else :
    print("Bonjour Madame ", nom)
```

9.1.4 EXB-4

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir le prénom de l'utilisateur (exemple : Joseph) ;
2- Saisir l'âge de l'utilisateur ;
3- Afficher le texte "Joseph, vous avez moins/plus de 25 ans" selon le cas.

Solution

```
prenom = input("Entrer votre prénom : ")
age = eval(input("Quel est votre âge : "))

if age == 25 :
    print(prenom, ", vous avez 25 ans.")
elif age > 25 :
    print(prenom, ", vous avez plus de 25 ans.")
else :
    print(prenom, ", vous avez moins de 25 ans.")
```

José OUIN - www.joseouin

9.1.5 EXB-5

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir un nombre réel ;
2- Afficher le texte "Ce nombre est inférieur/supérieur à 500" selon le cas.

Solution

```
a = eval(input("Entrer un nombre réel : "))
if a >= 500 :
    print("Le nombre ", a, " est supérieur ou égal à 500.")
else :
    print("Le nombre ", a, " est strictement inférieur à 500.")
```

9.1.6 EXB-6

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir un nombre réel ;
2- Afficher la partie entière et la partie décimale. Exemple pour 45.478 ; afficher "Partie entière 45 et partie décimale 0.478".

Solution

```
a = float(input("Entrer un nombre réel : "))

print("Partie réelle : ", int(a))
print("Partie décimale ", a - int(a))
# les erreurs d'arrondis sont normales. Elles sont liées à la variable de type float.
```

9.2 Effectuer des opérations avec les vecteurs

9.2.1 EXB-7

On se place dans le plan muni d'un repère orthonormé. Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir deux vecteurs $\overrightarrow{u}$ et $\overrightarrow{v}$;
2- Calculer la somme : $\overrightarrow{w} = \overrightarrow{u} + \overrightarrow{v}$;
3- Afficher le résultat.

Solution

uin.fr

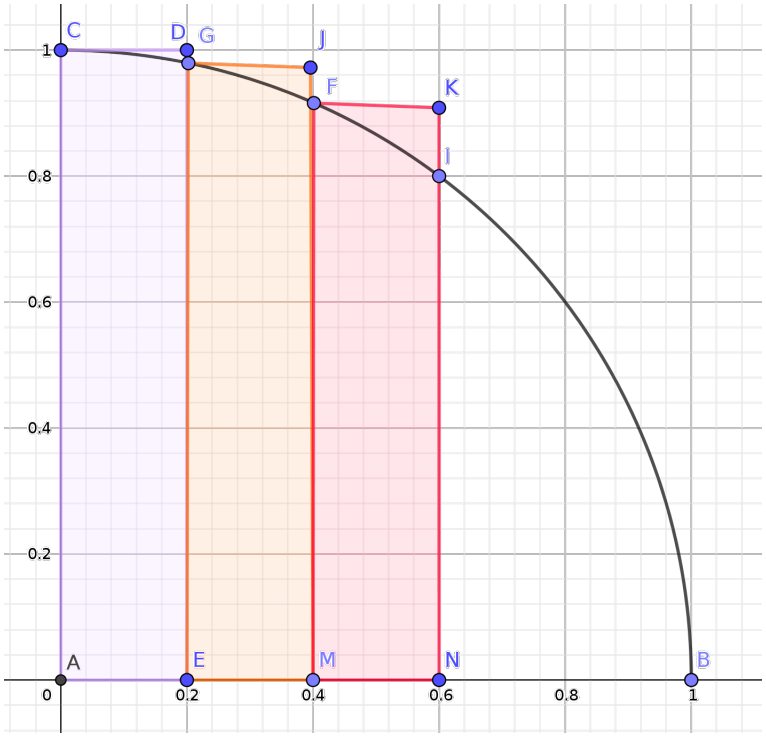
```
import numpy as np

u = eval(input("Entrer le vecteur u : [xu , yu] : "))
v = eval(input("Entrer le vecteur v : [xv , yv] : "))

npu = np.array(u)
npv = np.array(v)
w = npu + npv

print("Vecteur w = u + v : ", w)
```

9.2.2 EXB-8

Écrire un algorithme puis un programme Python qui effectue les opérations suivantes :
1- Saisir un vecteur $\overrightarrow{u}$ dans le plan muni d'un repère orthonormé ;

12.1 TP- [1]-1 - Intégration : Méthode des rectangles

On considère le quart de disque de rayon 1. Il s'agit de déterminer l'aire, S, de ce quart de disque en effectuant la somme des aires des rectangles comme indiqué par la figure ci-dessous.



En considérant un nombre N de rectangle suffisamment grand, la somme des aires de ces rectangles, S, est une approximation de l'aire du quart du disque de rayon 1, c'est-à-dire $\frac{\pi}{4}$.
On a donc : $\lim\limits_{N\to+\infty} S = \frac{\pi}{4}$, c'est-à-dire $4 \times \lim\limits_{N\to+\infty} S = \pi$.

On rappelle qu'un point $M(x;y)$ appartient au cercle si et seulement si : $x^2 + y^2 = 1$.

Travail demandé :
1- Écrire un programme qui permet d'afficher une approximation du nombre π . L'utilisateur doit pouvoir choisir le nombre N de rectangles.
2- Combien de rectangles faut-il considérer pour obtenir 5 décimales exactes de π ?
On donne les 14 premières décimales de π ; $\pi = 3{,}14159265358979...$

José OUIN - www.joseouin.fr

Solution

```
from math import *

N = int(input("Entrer le nombre de rectangles : N = "))

h = 1/N
x = 0
S = 0
for i in range(1,N+1) :
    y = sqrt(1 - x**2)
    S = S + h*y
    x = x + h


print("Approximation de Pi pour ",N," rectangles : ",4*S)
```

José OUIN - www.joseouin.fr