

DS3
CORRECTION

Exercice 1 :

```
#Exercice1: Résolution du système d'équation linéaire
from numpy import *
from numpy.linalg import *
A=array([[ -1,1,3],[2,-1,2],[4,1,-4]])
B=array([12,-8,15])
D=det(A)

if D==0:
    print("Le système admet une infinité de solution ou bien n'admet aucune solution")
else:
    #1ère Méthode
    X=dot(inv(A),B)
    print(X)

    #2ème Méthode: système de cramer
    Y=solve(A,B)
    print(Y)

    #3ème Méthode: calcul matricielle
    a=matrix(A)
    BB=array([[12],[-8],[15]])
    b=matrix(BB)
    Z=(a**(-1))*b
    print(Z)
```

Exercice 2 : La résolution d'une équation non linéaire

1)

```
from numpy import *
from matplotlib.pyplot import *
X=linspace(0,1,100)
I=ones(100)
def f(X):
    return(log((X**2)+I))

def g(X):
    return(sin(I/(I+X)))

title("Solution graphique de l'équation  $\text{Log}(x^2+1)=\sin(1/(1+x))$ ")
xlabel("$x$")
ylabel(" $\text{Log}(x^2+1)$  et  $\sin(1/(1+x))$ ")
plot(X,f(X))
plot(X,g(X))
grid(True)
savefig("Figds.pdf")
```

2)

#Exercice2 : La résolution d'une équation non linéaire: Méthode de la dichotomie

```
from scipy.optimize import bisect
from math import *
def f(x):
    return(log((x**2)+1)-sin(1/(1+x)))

s=bisect(f,0.8,1.)
print("La solution de l'équation est:",s)
```

#Exercice2 : La résolution d'une équation non linéaire: Méthode de Newton

```
from scipy.optimize import newton
from math import *
def f(x):
    return(log((x**2)+1)-sin(1/(1+x)))

#def der_f(x):
#    h=0.01
#    return((((log(((x+h)**2)+1)-sin(1/(1+(x+h)))))-(log((x**2)+1)-sin(1/(1+x)))))/h)

def der_f(x):
    return((2*x/(x**2+1))+(cos(1/(1+x))/((1+x)**2)))
s=newton(f,0.8,der_f)
print("La solution de l'équation est:",s)
```

Exercice 3 :

#Exercice3: produit scalaire & produit vectorielle

```
from numpy import *
from numpy.linalg import norm

U=array([1,2,3])
W=array([1,2,0])

Alpha=arccos(vdot(U,W)/(norm(U)*norm(W)))
print("L'angle en radian est :",Alpha)
Alphadeg=rad2deg(Alpha)
print("L'angle en degré est :",Alphadeg)
```

```
#V=cross(U,W)
```

```
#Alpha=arcsin(norm(V)/(norm(U)*norm(W)))
#print("L'angle en radian est :",Alpha)
#Alphadeg=rad2deg(Alpha)
#print("L'angle en degré est :",Alphadeg)
```

Exercice 4 : Evolution d'une température en chimie

```
#Exercice4: Evolution d'une température en chimie
```

```
from numpy import *
from scipy.integrate import odeint
t=arange(0,24.5,0.5)
phi=lambda y,t:(exp(-0.25*t))-y
s=odeint(phi,20,t)
print(s)
```

Exercice 5 : Le parachute

```
#Exercice5: Le parachute
```

```
from numpy import arange
from scipy.integrate import odeint
t=arange(0.,121.,2)
X0=0.,3.2
m=8
g=10
k=25
def phi(X,t):
    x,x_prime=X
    return([x_prime,g-((k*x_prime)/m)])

s=odeint(phi,X0,t)
print(s[-1,0])
```