

```

## Import des Librairies

from math import *
import matplotlib.pyplot as plt
plt.close('all')
import numpy as np
pi=3.14

## Données pour tout Le programme

R10 = 1 # rd/s
L1 = 50 * 0.001 # m

## Programme

# Données

r = 2
O10_d = 45 # °

# Création de variables intermédiaires

O10_r = O10_d * pi / 180
L2 = r * L1
c = -(L1/L2)*cos(O10_r)
O21_r = - O10_r - acos(c)

# Création de La matrice cinématique K

K = np.zeros([3,3])

K[0,0] = 1
K[0,1] = 1
K[2,2] = 1
K[1,0] = L2 * sin(O21_r + O10_r)
K[2,0] = - L2 * cos(O21_r + O10_r)

# Création du vecteur F

F0 = - R10
F1 = L1 * sin(O10_r) * R10
F2 = - L1 * cos(O10_r) * R10
F = np.array([F0,F1,F2])

# Résolution

Sol = np.linalg.solve(K,F)

## Mise en place d'une fonction de résolution

def Resolution(O10_d,r):
    O10_r = O10_d * pi / 180
    L2 = r * L1
    c = -(L1/L2)*cos(O10_r)
    O21_r = - O10_r - acos(c)
    K = np.zeros([3,3])
    K[0,0] = 1
    K[0,1] = 1
    K[2,2] = 1
    K[1,0] = L2 * sin(O21_r + O10_r)
    K[2,0] = - L2 * cos(O21_r + O10_r)

```

```

F0 = - R10
F1 = L1 * sin(010_r) * R10
F2 = - L1 * cos(010_r) * R10
F = np.array([F0,F1,F2])
Sol = np.linalg.solve(K,F)
return Sol

def f_Affiche_liste(fig_i,Liste_X,Liste_Y,Legende):
    fig = plt.figure(fig_i)
    plt.plot(Liste_X,Liste_Y,label=Legende)
    plt.ylabel('Données')
    plt.legend()
    plt.show()

# Résolution

Tps = []
R32 = []
R21 = []
V30 = []
for i in range(720):
    Temps = i / (R10 * 180 / pi)
    [r32,r21,v03] = Resolution(i,r)
    v30 = - v03
    Tps.append(Temps)
    R32.append(r32)
    R21.append(r21)
    V30.append(v30)

# Affichage

Fig = 0
f_Affiche_liste(Fig,Tps,R32,"R32")
f_Affiche_liste(Fig,Tps,R21,"R21")
Fig += 1
f_Affiche_liste(Fig,Tps,V30,"V30")

def f_Derivee(Liste_x,Liste_y):
    Liste_Derivee = []
    N = len(Liste_x) - 1
    for i in range(N):
        dt = Liste_x[i+1] - Liste_x[i]
        dy = Liste_y[i+1] - Liste_y[i]
        Acc = dy / dt
        Liste_Derivee.append(Acc)
    Liste_X = Liste_x[0:N]
    return Liste_X,Liste_Derivee

# Calcul accélération 3/0

T30,A30 = f_Derivee(Tps,V30)

Fig += 1
f_Affiche_liste(Fig,T30,A30,"A30")

# Programme pour r variable
#Mise en place d'une fonction

def f_Etude_r(r):
    Tps = []

```

```

R32 = []
R21 = []
V30 = []
for i in range(720):
    Temps = i / (R10 * 180 / pi)
    [r32,r21,v03] = Resolution(i,r)
    v30 = - v03
    Tps.append(Temps)
    R32.append(r32)
    R21.append(r21)
    V30.append(v30)
T30,A30 = f_Derivee(Tps,V30)
return T30,A30

```

Programme

```

Fig += 1
Plage_r = [i/10 for i in range(11,50,5)]
for i in range(len(Plage_r)):
    r = Plage_r[i]
    T30,A30 = f_Etude_r(r)
    f_Affiche_liste(Fig,T30,A30,str(r))

```