1. On détermine le flux de $\overrightarrow{B}$ à travers les N spires, en orientant la surface d'après la règle du tire bouchon à partir du sens proposé de l'intensité :
$\Phi = N.\oiint_{spire} B_0\overrightarrow{u_x} \cdot dS.\overrightarrow{u_y}' = N.\oiint_{spire} B_0.cos\theta.dS = B_0.cos\theta.4.a^2$
On en déduit par la loi de Faraday la fem d'induction $e_{ind} = -\dfrac{d\Phi}{dt} = +N.\dot{\theta}.B_0.4.a^2.sin\theta$
Ce qui donne, en modélisant les phénomènes d'induction par un générateur de fem e_{ind} :
$i_{ind} = \dfrac{N.\dot{\theta}.B_0.4.a^2}{R}$

2. Pour un élément $\overrightarrow{dl}$ du contour : $\overrightarrow{df} = i_{ind}\overrightarrow{dl} \wedge \overrightarrow{B}$
Le moment de cette force élémentaire, projeté selon Oz s'écrit alors :
$d\overrightarrow{\mathcal{M}} \cdot \overrightarrow{u_z} = \left[\overrightarrow{OP} \wedge (i_{ind}\overrightarrow{dl} \wedge \overrightarrow{B})\right].\overrightarrow{u_z}$
On doit séparer le calcul pour les 4 segments, mais on peut rapidement s'apercevoir que les forces de Laplace exercées sur les segments CD et EA n'ont pas d'effet sur la rotation du cadre. Leur moment projeté sur l'axe de rotation est donc nul.
On s'intéresse au segment AC :
$d\overrightarrow{\mathcal{M}} \cdot \overrightarrow{u_z} = \left[\overrightarrow{OP} \wedge (i_{ind}dz\overrightarrow{e_z} \wedge B_0\overrightarrow{e_y})\right].\overrightarrow{u_z}\left[\left(-a.\overrightarrow{e_x}' + z.\overrightarrow{e_z}\right) \wedge (-i_{ind}.B_0.dz.\overrightarrow{e_x}\right].\overrightarrow{u_z}$
Comme $\overrightarrow{e_x}' \wedge \overrightarrow{e_x} = -sin\theta\overrightarrow{e_z}$, on obtient $\mathcal{M}_{\Delta(AC)} = -N.a.i.B_0.sin\theta$
La force de Laplace s'exerçant sur le segment DE est opposée à celle s'exerçant sur le segment AC. Il s'agit donc d'un couple de forces. On montre facilement que $\mathcal{M}_{\Delta(DE)} = \mathcal{M}_{\Delta(AC)}$
Le moment des actions de Laplace est donc : $\mathcal{M} = -2.\dfrac{N^2.\dot{\theta}.sin^2\theta.B_0^2.4.a^4}{R}$

3. On applique alors le TMC au cadre. Les actions de moment non nul sont celles des forces de Laplace ainsi que celles du couple de torsion : $J_\Delta\dot{\omega} = -C.\theta - 2.\dfrac{N^2.\dot{\theta}.sin^2\theta.B_0^2.4.a^4}{R}$
On obtient donc $\ddot{\theta} + \dfrac{N^2.\dot{\theta}.sin^2\theta.B_0^2.8.a^4}{R.J_\delta} + \dfrac{C}{J_\delta}.\theta = 0$

4.
```python
import numpy as np
import matplotlib.pyplot as plt

#Variables globales
tau=.1
w_0=6.14

####### Premiere methode ##########
def systeme(S):
    T=np.zeros((2))
    T[0]=S[1]                          # d(theta)/dt
    T[1]=-(S[1]*(np.sin(S[0]))**2)/tau-w_0**2*S[0]       #d^2(theta)/dt^2 deduit de l'ED.
    return T

def Euler_1(S_0,ti,tf,N):
    t=np.linspace(ti,tf,N)           # On cree le tableau des instants
    theta=np.zeros((N))              # On cree le tableau permettant d'enregistrer les valeurs de theta
    theta[0]=S_0[0]                  # Condition initiale pour theta
    pas=t[1]-t[0]                    # On pourrait egalement ecrire pas=(tf-ti)/(N-1)
    S=S_0                        # Tableau S(theta,dtheta/dt) a l'instant initial
    for n in range(1,N):             # On a deja S_0
        S=S+pas*systeme(S)           # Relation de recurrence entre S_i et S_(i+1)
        theta[n]=S[0]                # on place dans le tableau la valeur theta_i
    return (t,theta)
####### Fin de la premiere methode ##########



####### Seconde methode ##########
def Euler_2(S_0,ti,tf,N):
    t=np.linspace(ti,tf,N)           # On cree le tableau des instants
    theta=np.zeros((N))     # On cree le tableau permettant d'enregistrer les valeurs de theta
    theta[0]=S_0[0]                # Condition initiale pour theta
    pas=t[1]-t[0]                    # On pourrait egalement ecrire pas=(tf-ti)/(N-1)
    theta[1]=theta[0]+pas*S_0[1]     # car theta_(i+1)=theta_i+pas.(dtheta/dt)
    for i in range(0,N-2):           # On a deja les deux premieres valeurs de theta
        theta[i+2]=2*theta[i+1]-theta[i]-pas**2*(w_0**2*theta[i]+((np.sin(theta[i]))**2*(theta[i+1]-theta[i])/pas)/tau)
    return (t,theta)
####### Fin de la seconde methode ##########


#Execution des deux methodes
S_0=np.array([.2,0])
t_1,theta_1=Euler_1(S_0,0,10,100000)
t_2,theta_2=Euler_2(S_0,0,10,100000)
plt.plot(t_1,theta_1)
plt.plot(t_2,theta_2)
```