main

September 19, 2022

0.0.1 Measurement of the earth's magnetic field

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import sympy as smp
```

The oscillation period of the rod

[ ]:
```
vibQ:int = 5 #
time = np.array([36,36,40]) #      5
timeError = 0.5 #                    (           +    )
period: float = np.mean(time/(vibQ)) #          -
periodMSE: float = np.sqrt(np.sum(np.power((time/vibQ)-period,2))/len(time))␣
↪#
periodError2: float = timeError/vibQ #   ,                   +
periodError: float = np.sqrt(periodMSE**2+periodError2**2) #               ␣
↪
print("Bunny flutter period: " + str(period)+" +/- "+str( periodError))
```

```
Bunny flutter period: 7.466666666666666 +/- 0.3901566636906541
```

Other experimental data

[ ]:
```
L_n,LError_n = 0.86,0.01 #                  ,        -         (      ␣
↪   (    )+                (    5   ))
l_n,lError_n = 0.04,0.0005 #                       +    (                )
m_n,mError_n = 5.9*np.power(0.1,3),0.1*np.power(0.1,3) #        +    (      ␣
↪  )
R_n,RError_n = 0.25,0.005 #          (          )
x1_n,x1Error_n = 0.04,0.005 #
r_n,rError_n = 0.005,0.0005 #         +    (                )
mu_n = 4 * np.pi * np.power(0.1,7) #
```

Calculation of the horizontal component of the earth's magnetic field

[ ]:
```
T,R,mu,J,L,x1 = smp.symbols("T R \mu_{0} J L x_{1}") #        ,
dT,dR,dJ,dL,dx1 = smp.symbols(r"\Delta_{T} \Delta_{R} \Delta_{J} \Delta_{L}␣
↪\Delta_{x_{1}}") #
m,l,r  = smp.symbols(r"m l r")
```

```
dm,dl,dr = smp.symbols(r"\Delta_{m} \Delta_{l} \Delta_{r}")
#
J = m*(((l**2)/12)+((r**2)/(4)))
B0 = 2*smp.pi/(T*R)
B0*=smp.sqrt(mu*J*L/(2*smp.pi*R*x1))
B0variables = [m,l,r,R,T,L,x1] #     ,             B0
B0ErrorVariablesA = [dm,dl,dr,dR,dT,dL,dx1]
B0ErrorVariables = smp.Matrix(B0ErrorVariablesA) #    ,               B0
B0gradient = smp.Matrix([B0.diff(i) for i in B0variables])
B0ErrorsVector = smp.Matrix([B0gradient[i]*B0ErrorVariables[i] for i in range(len(B0ErrorVariables))]) #   ,                B0
dB0 = B0ErrorsVector.norm() #        -
dB0 #                    B0
```

[ ]:

$$\sqrt{\frac{9\pi\left|\frac{\Delta_R\sqrt{\frac{L\mu_0 m\left(\frac{l^2}{12}+\frac{r^2}{4}\right)}{Rx_1}}}{R^2T}\right|^2}{2}+2\pi\left|\frac{\Delta_T\sqrt{\frac{L\mu_0 m\left(\frac{l^2}{12}+\frac{r^2}{4}\right)}{Rx_1}}}{RT^2}\right|^2+\frac{\pi\left|\frac{\Delta_L\sqrt{\frac{L\mu_0 m\left(\frac{l^2}{12}+\frac{r^2}{4}\right)}{Rx_1}}}{LRT}\right|^2}{2}+\frac{\pi\left|\frac{\Delta_m\sqrt{\frac{L\mu_0 m\left(\frac{l^2}{12}+\frac{r^2}{4}\right)}{Rx_1}}}{RTm}\right|^2}{2}+\frac{\pi\left|\frac{\Delta_{x_1}\sqrt{\frac{L\mu_0 \cdots}{\cdots}}}{RT\cdots}\right.}{2}\cdots}$$

[ ]:
```
B0_n = smp.lambdify(B0variables+[mu],B0)(m_n,l_n,r_n,R_n,period,L_n,x1_n,mu_n) #      B0
B0Error_n = smp.lambdify(B0variables+B0ErrorVariablesA+[mu],dB0)(m_n,l_n,r_n,R_n,period,L_n,x1_n,mError_n,l
#              B0
print("The horizontal component of the earth's magnetic field: "+str(B0_n*np.power(10,9))+" +/- "+str(B0Error_n*np.power(10,9))+" nTl ")
```

The horizontal component of the earth's magnetic field: 12668.351589637483 +/- 1119.1884063813147 nTl

0.0.2 Electrodynamic constant measurement

Experiment data

[ ]:
```
N_n: int = 34
C_n = 9*np.power(10,5)
CError_n = 0.02*C_n
U_n,UError_n = 95/300,95*0.01/300 #
x_n,xError_n = np.mean(np.array([11.5,12]))/200,0.001
nu_n = 50
```

Current calculation in si units

[ ]:
```
I, B0, R, x, L, N = smp.symbols(r"I B_{0} R x L N")
dI,dB0,dR,dx,dL = smp.symbols(r"\Delta_{I} \Delta_{B_{0}} \Delta_{R} \Delta_{x} \Delta_{L}")
```

```
I = x*(2*B0*R)/(2*L*mu*N)
IVariables = [B0,R,x,L]
IErrorVariables = [dB0, dR ,dx, dL]
Igradient = smp.Matrix([I.diff(IVariables[i]) for i in range(len(IVariables))])
IErrorVector = smp.Matrix([Igradient[i]*IErrorVariables[i] for i in␣
↪range(len(IErrorVariables))])
dI = IErrorVector.norm()
dI
#
```

[ ]:

$$\sqrt{\left|\frac{B_0 R \Delta_x}{L N \mu_0}\right|^2 + \left|\frac{B_0 \Delta_R x}{L N \mu_0}\right|^2 + \left|\frac{R \Delta_{B_0} x}{L N \mu_0}\right|^2 + \left|\frac{B_0 R \Delta_L x}{L^2 N \mu_0}\right|^2}$$

[ ]:
```
I_n = smp.lambdify(IVariables+[N,mu],I)(B0_n,R_n,x_n,L_n,N_n,mu_n)
IError_n = smp.
↪lambdify(IVariables+IErrorVariables+[N,mu],dI)(B0_n,R_n,x_n,L_n,B0Error_n,RError_n,xError_n
print("Current strength in si units: "+str(I_n)+" +/- "+str(IError_n)+" A ")
```

```
Current strength in si units: 0.005063849098350563 +/- 0.00047041526498938104 A
```

Calculation of current strength in CGS units

[ ]:
```
C,U,nu,I2 = smp.symbols(r"C U \Omega I_{2}")
I2 = C*U*nu
dC,dU,dI2 = smp.symbols("\Delta_{C} \Delta_{U} \Delta_{I_{2}}")
I2Variables = [C,U]
I2ErrorVariables = [dC,dU]
IErrorVector = smp.Matrix([I2.diff(I2Variables[i])*I2ErrorVariables[i] for i in␣
↪range(len(I2Variables))])
dI2 = IErrorVector.norm()
dI2
```

[ ]:

$$\sqrt{\left|C \Delta_U \Omega\right|^2 + \left|U \Delta_C \Omega\right|^2}$$

[ ]:
```
I2_n = smp.lambdify(I2Variables+[nu],I2)(C_n,U_n,nu_n)
I2Error_n = smp.
↪lambdify(I2Variables+I2ErrorVariables+[nu],dI2)(C_n,U_n,CError_n,UError_n,nu_n)
print("Current strength in CGS units: "+str(I2_n)+" +/- "+str(I2Error_n)+" cgs␣
↪units for current measurement")
```

```
Current strength in CGS units: 14250000.0 +/- 318639.68679372006 cgs units for
current measurement
```

Constant calculation

[ ]:
```
c_n = I2_n/(10*I_n) #                          (     )
cError_n = np.sqrt(np.power(I2Error_n/(10*I_n),2)+np.power(IError_n*I2_n/(10*np.
↪power(I_n,2)),2))
```

```
print("Electrodynamic constant: "+str(c_n)+" +/- "+str(cError_n)+" m/s ")
```

```
Electrodynamic constant: 281406489.87035614 +/- 26888403.129702337 m/s
```