

Supporting Information for: Direct Measurement of Lighthill's Energetic Efficiency of a Minimal Magnetic Microswimmer

Carles Calero^{1,2}, José García-Torres^{1,2}, Antonio Ortiz-Ambriz¹,
Francesc Sagués^{2,3}, Ignacio Pagonabarraga^{1,4,5}, and Pietro Tierno^{1,2,5*}

¹*Departament de Física de la Matèria Condensada,
Universitat de Barcelona, 08028 Barcelona, Spain.*

²*Institut de Nanociència i Nanotecnologia, Universitat de Barcelona, 08028 Barcelona, Spain.*

³*Departament de Ciència de Materials i Química Física,
Universitat de Barcelona, 08028 Barcelona, Spain.*

⁴*CECAM, Centre Européen de Calcul Atomique et Moléculaire,
École Polytechnique Fédérale de Lausanne, Batochime, Avenue Forel 2, 1015 Lausanne, Switzerland.*

⁵*Universitat de Barcelona Institute of Complex Systems (UBICS),
Universitat de Barcelona, 08028 Barcelona, Spain.*

(Dated: September 17, 2019)

PARAMETERS FOR ANALYTICAL MODEL

The coefficients of the analytical model for the velocity V (b_1, a_0, a_2, a_4) and the efficiency e (n_2, d_2, d_4, d_6) are given in the attached python notebook 'Model.coefficients.ipynb' (and also in the following .pdf file).

Efficiency as a function of the parameter of the problem

In Fig. 1 we represent the variation of the efficiency with χ, δ and α_{ind} predicted by the theoretical model for a given actuating magnetic field $\mathbf{B}(t)$. The results are rescaled to the parameters and efficiency of the microswimmer used in our experiments. As shown in Fig. 1a, the efficiency strongly depends on the susceptibility of the paramagnetic particle. In contrast, the efficiency is not very sensitive to reasonable variations of α_{ind} and δ (see Fig. 1b-c), although for large variations of δ it can increase by a factor ≈ 7 (see inset in Fig. 1b).

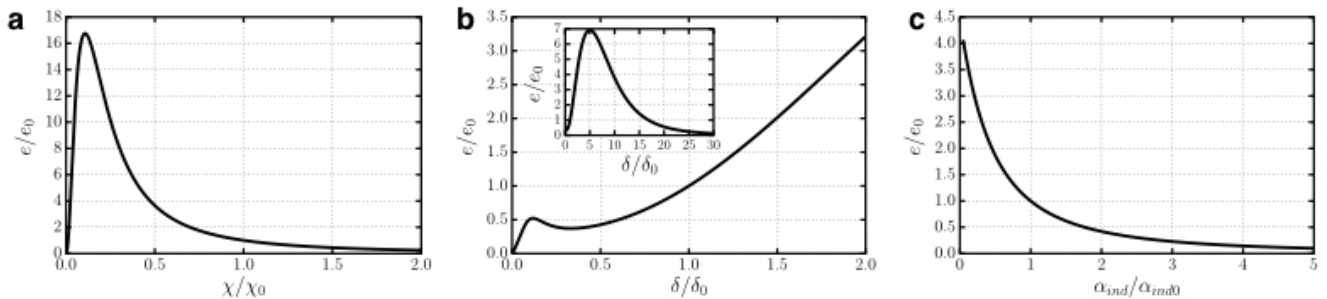


FIG. 1. Dependence of the efficiency of the microswimmer on the parameters of the problem (a) χ , (b) δ , and (c) α_{ind} for a given actuating field $\mathbf{B}(t)$. The results are rescaled to the parameters and efficiency of the microswimmer used in our experiments.

SUPPORTING VIDEOS

With the article are 2 videoclips as support of Fig.1.

- **VideoS1**(.WMV) This videoclip shows the dynamics of nanorod-colloid micropropeller driven by a swinging magnetic field with frequency $\nu = 20\text{Hz}$, and amplitudes $B_x = 2.15\text{mT}$ and $B_x = 2.74\text{mT}$. The video has been recorded at 504 frame per second (fps) and here it is played at 20 fps in order to visualize the relative movement of the ferromagnetic nanorod and the paramagnetic microsphere. The propeller moves from left to right at an average speed of $3\mu\text{s}^{-1}$. The video corresponds to Fig.1(b) (PRL) of the article.
- **VideoS2**(.WMV) This video is the same as VideoS1 but it is played in real time, here at 504 frame per second.

* ptierno@ub.edu

In [1]:

```
import os, sys
import matplotlib.pyplot as plt
#import seaborn
%matplotlib inline
import pandas as pd
import warnings
import numpy as np
warnings.filterwarnings("ignore", category=DeprecationWarning)
from scipy.signal import savgol_filter
from matplotlib.colors import LightSource, Normalize
#reload(seaborn)
#global constants
Homedir = '.'
from matplotlib.font_manager import FontProperties
rc={'axes.labelsize': 20, 'font.size': 14, 'xtick.labelsize' : 16, 'ytick.labelsize' : 16, 'legend.fontsize': 16.0, 'axes.titlesize': 18, 'legend.numpoints': 1, 'axes.linewidth': 2, 'xtick.major.size':5, 'xtick.major.width':2, 'xtick.minor.size':3, 'xtick.minor.width':2, 'ytick.major.size':5, 'ytick.major.width':2, 'ytick.minor.size':3, 'ytick.minor.width':2}
plt.rc('text',usetex=True)
plt.rc('font', family='serif',serif='cm10',weight='bold',size=25)
clrs = ['black', 'blue', 'green', 'red', 'magenta', 'orange', 'black', 'blue', 'green', 'cyan', 'black', 'blue', 'green', 'gray', 'red', 'magenta', 'orange', 'cyan']
sim = ['o', 's', '^', 'v', '<', '>', 'p', 'o', 's', '^', 'v', '<', '>', 'p', 'o', 's', '^', 'o', 's', '^', 'v', '<', '>', 'p', 'o', 's', '^', 'v', '<', '>', 'p']
plt.rcParams.update(**rc)
```

Coefficients for velocity and efficiency of microswimmer, Eqs. (8) and (16)

In [2]:

```
pi = np.pi
# d = L/R is delta in the main article
# aind = mu0*m_n/(R^3*B) is alpha_ind in the main article
# adip = 4*pi*chi/3, where chi is the susceptibility in the main article
# g = L/D is gamma in the main article

# VELOCITY COEFFICIENTS:
def b1(d, g, adip, aind):
    res = 9*(2*pi)**2*adip*d**3*g**2*pi*(d + 2)**4*(pi*d**3 + 6*pi*d**2 + 12*pi*d + aind + 8*pi)*(48*pi*g - 36*adip*g - 2*adip*d**3 + 96*pi*d*g - 36*adip*d*g + 72*pi*d**2*g + 24*pi*d**3*g + 3*pi*d**4*g - 12*adip*d**2*g)
    return res
def a0(d, g, adip, aind):
    res = 8*adip**2*d**4*g**2*(d + 3*g)*(2*d + 3*g)**2*(pi*d**3 + 6*pi*d**2 + 12*pi*d + aind + 8*pi)**2
    return res
def a2(d, g, adip, aind):
    res = 2*(2*pi)**2*(d + 3*g)*(144*adip**2*aind**2*d**6 + 1728*adip**2*aind**2*d**5*g + 5184*adip**2*aind**2*d**4*g**2 + 5184*adip**2*aind**2*d**4*g + 31104*adip**2*aind**2*d**3*g**2 + 5184*adip**2*aind**2*d**3*g + 77760*adip**2*aind**2*d**2*g**2 + 93312*adip**2*aind**2*d*g**2 + 46656*adip**2*aind**2*g**2 + 288*pi*adip**2*aind*d**9 + 2808*pi*adip**2*aind*d**8*g + 1728*pi*adip**2*aind*d**8 + 6480*pi*adip**2*aind*d**7*g**2 + 24624*pi*adip**2*aind*d**7*g + 3456*pi*adip**2*aind*d**7 + 73872*pi*adip**2*aind*d**6*g**2 + 88128*pi*adip**2*aind*d**6*g + 2304*pi*adip**2*aind*d**6 + 369360*pi*adip**2*aind*d**5*g**2 + 162432*pi*adip**2*aind*d**5*g + 1054944*pi*adip**2*aind*d**4*g**2 + 155520*pi*adip**2*aind*d**4*g + 1866240*pi*adip**2*aind*d**3*g**2 + 62208*pi*adip**2*aind*d**3*g + 2052864*pi*adip**2*aind*d**2*g**2 + 1306368*pi*adip**2*aind*d*g**2 + 373248*pi*adip**2*aind*g**2 + 144*pi**2*adip**2*d**12 + 1080*pi**2*adip**2*d**11*g + 1728*pi**2*adip**2*d**11 + 2025*pi**2*adip**2*d**10*g**2 + 15552*pi**2*adip**2*d**10*g + 8640*pi**2*adip**2*d**10 + 34020*pi**2*adip**2*d**9*g**2 + 98496*pi**2*adip**2*d**9*g + 23040*pi**2*adip**2*d**9 + 259524*pi**2*adip**2*d**8*g**2 + 359424*pi**2*adip**2*d**8*g + 34560*pi**2*adip**2*d**8 + 1187136*pi**2*adip**2*d**7*g**2 + 829440*pi**2*adip**2*d**7*g + 27648*pi**2*adip**2*d**7 + 3615840*pi**2*adip**2*d**6*g**2 + 1244160*pi**2*adip**2*d**6*g + 9216*pi**2*adip**2*d**6 + 7682688*pi**2*adip**2*d**5*g**2 + 1188864*pi**2*adip**2*d**5*g + 11560320*pi**2*adip**2*d**4*g**2 + 663552*pi**2*adip**2*d**4*g + 12192768*pi**2*adip**2*d**3*g**2 + 165888*pi**2*adip**2*d**3*g + 8646912*pi**2*adip**2*d**2*g**2 + 3732480*pi**2*adip**2*d*g**2 + 746496*pi**2*adip**2*g**2 + 972*pi**2*adip*aind*d**9*g**2 + 17496*pi**2*adip*aind*d**8*g**2 + 139968*pi**2*adip*aind*d**7*g**2 + 653184*pi**2*adip*aind*d**6*g**2 + 1959552*pi**2*adip*aind*d**5*g**2 + 3919104*pi**2*adip*aind*d**4*g**2 + 5225472*pi**2*adip*aind*d**3*g**2 + 4478976*pi**2*adip*aind*d**2*g**2 + 2239488*pi**2*adip*aind*d*g**2 + 497664*pi**2*adip*aind*g**2 + 486*pi**3*adip*d**12*g**2 + 11664*pi**3*adip*d**11*g**2 + 128304*pi**3*adip*d**10*g**2 + 855360*pi**3*adip*d**9*g**2 + 3849120*pi**3*adip*d**8*g**2 + 12317184*pi**3*adip*d**7*g**2 + 28740096*pi**3*adip*d**6*g**2 + 49268736*pi**3*adip*d**5*g**2 + 61585920*pi**3*adip*d**4*g**2 + 54743040*pi**3*adip*d**3*g**2 + 32845824*pi**3*adip*d**2*g**2 + 11943936*pi**3*adip*d*g**2 + 1990656*pi**3*adip*g**2 + 81*pi**4*d**14*g**2 + 2268*pi**4*d**13*g**2 + 29484*pi**4*d**12*g**2 + 235872*pi**4*d**11*g**2 + 1297296*pi**4*d**10*g**2 + 5189184*pi**4*d**9*g**2 + 15567552*pi**4*d**8*g**2 + 35582976*pi**4*d**7*g**2 + 62270208*pi**4*d**6*g**2 + 83026944*pi**4*d**5*g**2 + 83026944*pi**4*d**4*g**2 + 60383232*pi**4*d**3*g**2 + 30191616*pi**4*d**2*g**2 + 9289728*pi**4*d*g**2 + 1327104*pi**4*g**2)
    return res
def a4(d, g, adip, aind):
    res = 1458*(2*pi)**4*pi**4*(d + 3*g)*(d + 2)**14
    return res
```

```
# EFFICIENCY COEFFICIENTS:
```

```
def n2(d, g, adip, aind):
```

```
    res = 243*(2*pi)**2*adip**2*d**3*g**3*pi*(d + 2)**8*(pi*d**3 + 6*pi*d**2 + 12*pi*d  
+ aind + 8*pi)**2*(48*pi*g - 36*adip*g - 2*adip*d**3 + 96*pi*d*g - 36*adip*d*g + 72*pi*  
d**2*g + 24*pi*d**3*g + 3*pi*d**4*g - 12*adip*d**2*g)**2
```

```
    return res
```

```
def d0(d, g, adip, aind):
```

```
    res = 2*(d + 3*g)**2*(16*adip**2*aind**2*d**6*g**2 + 48*adip**2*aind**2*d**5*g**3 +  
36*adip**2*aind**2*d**4*g**4 + 32*pi*adip**2*aind*d**9*g**2 + 96*pi*adip**2*aind*d**8*g  
**3 + 192*pi*adip**2*aind*d**8*g**2 + 72*pi*adip**2*aind*d**7*g**4 + 576*pi*adip**2*ain  
d*d**7*g**3 + 384*pi*adip**2*aind*d**7*g**2 + 432*pi*adip**2*aind*d**6*g**4 + 1152*pi*a  
dip**2*aind*d**6*g**3 + 256*pi*adip**2*aind*d**6*g**2 + 864*pi*adip**2*aind*d**5*g**4 +  
768*pi*adip**2*aind*d**5*g**3 + 576*pi*adip**2*aind*d**4*g**4 + 16*pi**2*adip**2*d**12*  
g**2 + 48*pi**2*adip**2*d**11*g**3 + 192*pi**2*adip**2*d**11*g**2 + 36*pi**2*adip**2*d*  
*10*g**4 + 576*pi**2*adip**2*d**10*g**3 + 960*pi**2*adip**2*d**10*g**2 + 432*pi**2*adip  
**2*d**9*g**4 + 2880*pi**2*adip**2*d**9*g**3 + 2560*pi**2*adip**2*d**9*g**2 + 2160*pi**  
2*adip**2*d**8*g**4 + 7680*pi**2*adip**2*d**8*g**3 + 3840*pi**2*adip**2*d**8*g**2 + 576  
0*pi**2*adip**2*d**7*g**4 + 11520*pi**2*adip**2*d**7*g**3 + 3072*pi**2*adip**2*d**7*g**  
2 + 8640*pi**2*adip**2*d**6*g**4 + 9216*pi**2*adip**2*d**6*g**3 + 1024*pi**2*adip**2*d*  
*6*g**2 + 6912*pi**2*adip**2*d**5*g**4 + 3072*pi**2*adip**2*d**5*g**3 + 2304*pi**2*adip  
**2*d**4*g**4)*(16*adip**2*aind**2*d**6 + 120*adip**2*aind**2*d**5*g + 144*adip**2*ain  
d**2*d**4*g**2 + 288*adip**2*aind**2*d**4*g + 432*adip**2*aind**2*d**3*g**2 + 288*adip**  
2*aind**2*d**3*g + 432*adip**2*aind**2*d**2*g**2 + 32*pi*adip**2*aind*d**9 + 240*pi*adi  
p**2*aind*d**8*g + 192*pi*adip**2*aind*d**8 + 288*pi*adip**2*aind*d**7*g**2 + 2016*pi*a  
dip**2*aind*d**7*g + 384*pi*adip**2*aind*d**7 + 2592*pi*adip**2*aind*d**6*g**2 + 6912*pi  
i*adip**2*aind*d**6*g + 256*pi*adip**2*aind*d**6 + 9504*pi*adip**2*aind*d**5*g**2 + 122  
88*pi*adip**2*aind*d**5*g + 17856*pi*adip**2*aind*d**4*g**2 + 11520*pi*adip**2*aind*d**  
4*g + 17280*pi*adip**2*aind*d**3*g**2 + 4608*pi*adip**2*aind*d**3*g + 6912*pi*adip**2*a  
ind*d**2*g**2 + 16*pi**2*adip**2*d**12 + 120*pi**2*adip**2*d**11*g + 192*pi**2*adip**2*  
d**11 + 144*pi**2*adip**2*d**10*g**2 + 1728*pi**2*adip**2*d**10*g + 960*pi**2*adip**2*d  
**10 + 2160*pi**2*adip**2*d**9*g**2 + 10944*pi**2*adip**2*d**9*g + 2560*pi**2*adip**2*d  
**9 + 14256*pi**2*adip**2*d**8*g**2 + 39936*pi**2*adip**2*d**8*g + 3840*pi**2*adip**2*d  
**8 + 54144*pi**2*adip**2*d**7*g**2 + 92160*pi**2*adip**2*d**7*g + 3072*pi**2*adip**2*d  
**7 + 129600*pi**2*adip**2*d**6*g**2 + 138240*pi**2*adip**2*d**6*g + 1024*pi**2*adip**2  
*d**6 + 200448*pi**2*adip**2*d**5*g**2 + 132096*pi**2*adip**2*d**5*g + 195840*pi**2*adi  
p**2*d**4*g**2 + 73728*pi**2*adip**2*d**4*g + 110592*pi**2*adip**2*d**3*g**2 + 18432*pi  
**2*adip**2*d**3*g + 27648*pi**2*adip**2*d**2*g**2)
```

```
    return res
```

```
def d2(d, g, adip, aind):
```

```
    res = 2*(2*pi)**2*(d + 3*g)**2*(16*adip**2*aind**2*d**6 + 120*adip**2*aind**2*d**5*  
g + 144*adip**2*aind**2*d**4*g**2 + 288*adip**2*aind**2*d**4*g + 432*adip**2*aind**2*d*  
*3*g**2 + 288*adip**2*aind**2*d**3*g + 432*adip**2*aind**2*d**2*g**2 + 32*pi*adip**2*ai  
nd*d**9 + 240*pi*adip**2*aind*d**8*g + 192*pi*adip**2*aind*d**8 + 288*pi*adip**2*aind*d  
**7*g**2 + 2016*pi*adip**2*aind*d**7*g + 384*pi*adip**2*aind*d**7 + 2592*pi*adip**2*ain  
d*d**6*g**2 + 6912*pi*adip**2*aind*d**6*g + 256*pi*adip**2*aind*d**6 + 9504*pi*adip**2*  
aind*d**5*g**2 + 12288*pi*adip**2*aind*d**5*g + 17856*pi*adip**2*aind*d**4*g**2 + 11520  
*pi*adip**2*aind*d**4*g + 17280*pi*adip**2*aind*d**3*g**2 + 4608*pi*adip**2*aind*d**3*g  
+ 6912*pi*adip**2*aind*d**2*g**2 + 16*pi**2*adip**2*d**12 + 120*pi**2*adip**2*d**11*g +  
192*pi**2*adip**2*d**11 + 144*pi**2*adip**2*d**10*g**2 + 1728*pi**2*adip**2*d**10*g + 9  
60*pi**2*adip**2*d**10 + 2160*pi**2*adip**2*d**9*g**2 + 10944*pi**2*adip**2*d**9*g + 25  
60*pi**2*adip**2*d**9 + 14256*pi**2*adip**2*d**8*g**2 + 39936*pi**2*adip**2*d**8*g + 38  
40*pi**2*adip**2*d**8 + 54144*pi**2*adip**2*d**7*g**2 + 92160*pi**2*adip**2*d**7*g + 30  
72*pi**2*adip**2*d**7 + 129600*pi**2*adip**2*d**6*g**2 + 138240*pi**2*adip**2*d**6*g +  
1024*pi**2*adip**2*d**6 + 200448*pi**2*adip**2*d**5*g**2 + 132096*pi**2*adip**2*d**5*g  
+ 195840*pi**2*adip**2*d**4*g**2 + 73728*pi**2*adip**2*d**4*g + 110592*pi**2*adip**2*d*  
*3*g**2 + 18432*pi**2*adip**2*d**3*g + 27648*pi**2*adip**2*d**2*g**2)*(144*adip**2*aind  
**2*d**6 + 1728*adip**2*aind**2*d**5*g + 5184*adip**2*aind**2*d**4*g**2 + 5184*adip**2*  
aind**2*d**4*g + 31104*adip**2*aind**2*d**3*g**2 + 5184*adip**2*aind**2*d**3*g + 77760*  
adip**2*aind**2*d**2*g**2 + 93312*adip**2*aind**2*d*g**2 + 46656*adip**2*aind**2*g**2 +  
288*pi*adip**2*aind*d**9 + 2808*pi*adip**2*aind*d**8*g + 1728*pi*adip**2*aind*d**8 + 64  
80*pi*adip**2*aind*d**7*g**2 + 24624*pi*adip**2*aind*d**7*g + 3456*pi*adip**2*aind*d**7
```

```

+ 73872*pi*adip**2*aind*d**6*g**2 + 88128*pi*adip**2*aind*d**6*g + 2304*pi*adip**2*aind
*d**6 + 369360*pi*adip**2*aind*d**5*g**2 + 162432*pi*adip**2*aind*d**5*g + 1054944*pi*a
dip**2*aind*d**4*g**2 + 155520*pi*adip**2*aind*d**4*g + 1866240*pi*adip**2*aind*d**3*g
**2 + 62208*pi*adip**2*aind*d**3*g + 2052864*pi*adip**2*aind*d**2*g**2 + 1306368*pi*adip
**2*aind*d*g**2 + 373248*pi*adip**2*aind*g**2 + 144*pi**2*adip**2*d**12 + 1080*pi**2*ad
ip**2*d**11*g + 1728*pi**2*adip**2*d**11 + 2025*pi**2*adip**2*d**10*g**2 + 15552*pi**2*
adip**2*d**10*g + 8640*pi**2*adip**2*d**10 + 34020*pi**2*adip**2*d**9*g**2 + 98496*pi**
2*adip**2*d**9*g + 23040*pi**2*adip**2*d**9 + 259524*pi**2*adip**2*d**8*g**2 + 359424*p
i**2*adip**2*d**8*g + 34560*pi**2*adip**2*d**8 + 1187136*pi**2*adip**2*d**7*g**2 + 8294
40*pi**2*adip**2*d**7*g + 27648*pi**2*adip**2*d**7 + 3615840*pi**2*adip**2*d**6*g**2 +
1244160*pi**2*adip**2*d**6*g + 9216*pi**2*adip**2*d**6 + 7682688*pi**2*adip**2*d**5*g**
2 + 1188864*pi**2*adip**2*d**5*g + 11560320*pi**2*adip**2*d**4*g**2 + 663552*pi**2*adip
**2*d**4*g + 12192768*pi**2*adip**2*d**3*g**2 + 165888*pi**2*adip**2*d**3*g + 8646912*p
i**2*adip**2*d**2*g**2 + 3732480*pi**2*adip**2*d*g**2 + 746496*pi**2*adip**2*g**2 + 972
*pi**2*adip*aind*d**9*g**2 + 17496*pi**2*adip*aind*d**8*g**2 + 139968*pi**2*adip*aind*d
**7*g**2 + 653184*pi**2*adip*aind*d**6*g**2 + 1959552*pi**2*adip*aind*d**5*g**2 + 39191
04*pi**2*adip*aind*d**4*g**2 + 5225472*pi**2*adip*aind*d**3*g**2 + 4478976*pi**2*adip*a
ind*d**2*g**2 + 2239488*pi**2*adip*aind*d*g**2 + 497664*pi**2*adip*aind*g**2 + 486*pi**
3*adip*d**12*g**2 + 11664*pi**3*adip*d**11*g**2 + 128304*pi**3*adip*d**10*g**2 + 855360
*pi**3*adip*d**9*g**2 + 3849120*pi**3*adip*d**8*g**2 + 12317184*pi**3*adip*d**7*g**2 +
28740096*pi**3*adip*d**6*g**2 + 49268736*pi**3*adip*d**5*g**2 + 61585920*pi**3*adip*d**
4*g**2 + 54743040*pi**3*adip*d**3*g**2 + 32845824*pi**3*adip*d**2*g**2 + 11943936*pi**3
*adip*d*g**2 + 1990656*pi**3*adip*g**2 + 81*pi**4*d**14*g**2 + 2268*pi**4*d**13*g**2 +
29484*pi**4*d**12*g**2 + 235872*pi**4*d**11*g**2 + 1297296*pi**4*d**10*g**2 + 5189184*p
i**4*d**9*g**2 + 15567552*pi**4*d**8*g**2 + 35582976*pi**4*d**7*g**2 + 62270208*pi**4*d
**6*g**2 + 83026944*pi**4*d**5*g**2 + 83026944*pi**4*d**4*g**2 + 60383232*pi**4*d**3*g
**2 + 30191616*pi**4*d**2*g**2 + 9289728*pi**4*d*g**2 + 1327104*pi**4*g**2) + 2*(d + 3*g
)**2*(9289728*d*pi**4 - 995328*adip*pi**3 + 1327104*pi**4 + 30191616*d**2*pi**4 + 60383
232*d**3*pi**4 + 83026944*d**4*pi**4 + 83026944*d**5*pi**4 + 62270208*d**6*pi**4 + 3558
2976*d**7*pi**4 + 15567552*d**8*pi**4 + 5189184*d**9*pi**4 + 1297296*d**10*pi**4 + 2358
72*d**11*pi**4 + 29484*d**12*pi**4 + 2268*d**13*pi**4 + 81*d**14*pi**4 - 16422912*adip*
d**2*pi**3 - 27371520*adip*d**3*pi**3 - 30792960*adip*d**4*pi**3 - 24634368*adip*d**5*p
i**3 - 14370048*adip*d**6*pi**3 - 6158592*adip*d**7*pi**3 - 1924560*adip*d**8*pi**3 - 4
27680*adip*d**9*pi**3 - 64152*adip*d**10*pi**3 - 5832*adip*d**11*pi**3 - 243*adip*d**12
*pi**3 - 5971968*adip*d*pi**3)*(16*adip**2*aind**2*d**6*g**2 + 48*adip**2*aind**2*d**5*
g**3 + 36*adip**2*aind**2*d**4*g**4 + 32*pi*adip**2*aind*d**9*g**2 + 96*pi*adip**2*aind
*d**8*g**3 + 192*pi*adip**2*aind*d**8*g**2 + 72*pi*adip**2*aind*d**7*g**4 + 576*pi*adip
**2*aind*d**7*g**3 + 384*pi*adip**2*aind*d**7*g**2 + 432*pi*adip**2*aind*d**6*g**4 + 11
52*pi*adip**2*aind*d**6*g**3 + 256*pi*adip**2*aind*d**6*g**2 + 864*pi*adip**2*aind*d**5
*g**4 + 768*pi*adip**2*aind*d**5*g**3 + 576*pi*adip**2*aind*d**4*g**4 + 16*pi**2*adip**
2*d**12*g**2 + 48*pi**2*adip**2*d**11*g**3 + 192*pi**2*adip**2*d**11*g**2 + 36*pi**2*ad
ip**2*d**10*g**4 + 576*pi**2*adip**2*d**10*g**3 + 960*pi**2*adip**2*d**10*g**2 + 432*pi
**2*adip**2*d**9*g**4 + 2880*pi**2*adip**2*d**9*g**3 + 2560*pi**2*adip**2*d**9*g**2 + 2
160*pi**2*adip**2*d**8*g**4 + 7680*pi**2*adip**2*d**8*g**3 + 3840*pi**2*adip**2*d**8*g
**2 + 5760*pi**2*adip**2*d**7*g**4 + 11520*pi**2*adip**2*d**7*g**3 + 3072*pi**2*adip**2*
d**7*g**2 + 8640*pi**2*adip**2*d**6*g**4 + 9216*pi**2*adip**2*d**6*g**3 + 1024*pi**2*ad
ip**2*d**6*g**2 + 6912*pi**2*adip**2*d**5*g**4 + 3072*pi**2*adip**2*d**5*g**3 + 2304*pi
**2*adip**2*d**4*g**4)

```

```

return res

```

```

def d4(d, g, adip, aind):

```

```

    res = 2*(2*pi)**4*(d + 3*g)**2*(9289728*d*pi**4 - 995328*adip*pi**3 + 1327104*pi**4
+ 30191616*d**2*pi**4 + 60383232*d**3*pi**4 + 83026944*d**4*pi**4 + 83026944*d**5*pi**4
+ 62270208*d**6*pi**4 + 35582976*d**7*pi**4 + 15567552*d**8*pi**4 + 5189184*d**9*pi**4
+ 1297296*d**10*pi**4 + 235872*d**11*pi**4 + 29484*d**12*pi**4 + 2268*d**13*pi**4 + 81*
d**14*pi**4 - 16422912*adip*d**2*pi**3 - 27371520*adip*d**3*pi**3 - 30792960*adip*d**4*
pi**3 - 24634368*adip*d**5*pi**3 - 14370048*adip*d**6*pi**3 - 6158592*adip*d**7*pi**3 -
1924560*adip*d**8*pi**3 - 427680*adip*d**9*pi**3 - 64152*adip*d**10*pi**3 - 5832*adip*d
**11*pi**3 - 243*adip*d**12*pi**3 - 5971968*adip*d*pi**3)*(144*adip**2*aind**2*d**6 + 1
728*adip**2*aind**2*d**5*g + 5184*adip**2*aind**2*d**4*g**2 + 5184*adip**2*aind**2*d**4
*g + 31104*adip**2*aind**2*d**3*g**2 + 5184*adip**2*aind**2*d**3*g + 77760*adip**2*aind
**2*d**2*g**2 + 93312*adip**2*aind**2*d*g**2 + 46656*adip**2*aind**2*g**2 + 288*pi*adip

```

```
**2*aind*d**9 + 2808*pi*adip**2*aind*d**8*g + 1728*pi*adip**2*aind*d**8 + 6480*pi*adip*
*2*aind*d**7*g**2 + 24624*pi*adip**2*aind*d**7*g + 3456*pi*adip**2*aind*d**7 + 73872*pi
*adip**2*aind*d**6*g**2 + 88128*pi*adip**2*aind*d**6*g + 2304*pi*adip**2*aind*d**6 + 36
9360*pi*adip**2*aind*d**5*g**2 + 162432*pi*adip**2*aind*d**5*g + 1054944*pi*adip**2*ain
d*d**4*g**2 + 155520*pi*adip**2*aind*d**4*g + 1866240*pi*adip**2*aind*d**3*g**2 + 62208
*pi*adip**2*aind*d**3*g + 2052864*pi*adip**2*aind*d**2*g**2 + 1306368*pi*adip**2*aind*d
*g**2 + 373248*pi*adip**2*aind*g**2 + 144*pi**2*adip**2*d**12 + 1080*pi**2*adip**2*d**1
1*g + 1728*pi**2*adip**2*d**11 + 2025*pi**2*adip**2*d**10*g**2 + 15552*pi**2*adip**2*d
**10*g + 8640*pi**2*adip**2*d**10 + 34020*pi**2*adip**2*d**9*g**2 + 98496*pi**2*adip**2
*d**9*g + 23040*pi**2*adip**2*d**9 + 259524*pi**2*adip**2*d**8*g**2 + 359424*pi**2*adip
**2*d**8*g + 34560*pi**2*adip**2*d**8 + 1187136*pi**2*adip**2*d**7*g**2 + 829440*pi**2*a
dip**2*d**7*g + 27648*pi**2*adip**2*d**7 + 3615840*pi**2*adip**2*d**6*g**2 + 1244160*pi
**2*adip**2*d**6*g + 9216*pi**2*adip**2*d**6 + 7682688*pi**2*adip**2*d**5*g**2 + 118886
4*pi**2*adip**2*d**5*g + 11560320*pi**2*adip**2*d**4*g**2 + 663552*pi**2*adip**2*d**4*g
+ 12192768*pi**2*adip**2*d**3*g**2 + 165888*pi**2*adip**2*d**3*g + 8646912*pi**2*adip**
2*d**2*g**2 + 3732480*pi**2*adip**2*d*g**2 + 746496*pi**2*adip**2*g**2 + 972*pi**2*adip
*aind*d**9*g**2 + 17496*pi**2*adip*aind*d**8*g**2 + 139968*pi**2*adip*aind*d**7*g**2 +
653184*pi**2*adip*aind*d**6*g**2 + 1959552*pi**2*adip*aind*d**5*g**2 + 3919104*pi**2*ad
ip*aind*d**4*g**2 + 5225472*pi**2*adip*aind*d**3*g**2 + 4478976*pi**2*adip*aind*d**2*g
**2 + 2239488*pi**2*adip*aind*d*g**2 + 497664*pi**2*adip*aind*g**2 + 486*pi**3*adip*d**1
2*g**2 + 11664*pi**3*adip*d**11*g**2 + 128304*pi**3*adip*d**10*g**2 + 855360*pi**3*adip
*d**9*g**2 + 3849120*pi**3*adip*d**8*g**2 + 12317184*pi**3*adip*d**7*g**2 + 28740096*pi
**3*adip*d**6*g**2 + 49268736*pi**3*adip*d**5*g**2 + 61585920*pi**3*adip*d**4*g**2 + 54
743040*pi**3*adip*d**3*g**2 + 32845824*pi**3*adip*d**2*g**2 + 11943936*pi**3*adip*d*g**
2 + 1990656*pi**3*adip*g**2 + 81*pi**4*d**14*g**2 + 2268*pi**4*d**13*g**2 + 29484*pi**4
*d**12*g**2 + 235872*pi**4*d**11*g**2 + 1297296*pi**4*d**10*g**2 + 5189184*pi**4*d**9*g
**2 + 15567552*pi**4*d**8*g**2 + 35582976*pi**4*d**7*g**2 + 62270208*pi**4*d**6*g**2 +
83026944*pi**4*d**5*g**2 + 83026944*pi**4*d**4*g**2 + 60383232*pi**4*d**3*g**2 + 301916
16*pi**4*d**2*g**2 + 9289728*pi**4*d*g**2 + 1327104*pi**4*g**2) + 2*(d + 3*g)**2*(729*p
i**4*d**14 + 20412*pi**4*d**13 + 265356*pi**4*d**12 + 2122848*pi**4*d**11 + 11675664*pi
**4*d**10 + 46702656*pi**4*d**9 + 140107968*pi**4*d**8 + 320246784*pi**4*d**7 + 5604318
72*pi**4*d**6 + 747242496*pi**4*d**5 + 747242496*pi**4*d**4 + 543449088*pi**4*d**3 + 27
1724544*pi**4*d**2 + 83607552*pi**4*d + 11943936*pi**4)*(16*adip**2*aind**2*d**6 + 120*
adip**2*aind**2*d**5*g + 144*adip**2*aind**2*d**4*g**2 + 288*adip**2*aind**2*d**4*g + 4
32*adip**2*aind**2*d**3*g**2 + 288*adip**2*aind**2*d**3*g + 432*adip**2*aind**2*d**2*g
**2 + 32*pi*adip**2*aind*d**9 + 240*pi*adip**2*aind*d**8*g + 192*pi*adip**2*aind*d**8 +
288*pi*adip**2*aind*d**7*g**2 + 2016*pi*adip**2*aind*d**7*g + 384*pi*adip**2*aind*d**7
+ 2592*pi*adip**2*aind*d**6*g**2 + 6912*pi*adip**2*aind*d**6*g + 256*pi*adip**2*aind*d
**6 + 9504*pi*adip**2*aind*d**5*g**2 + 12288*pi*adip**2*aind*d**5*g + 17856*pi*adip**2*a
ind*d**4*g**2 + 11520*pi*adip**2*aind*d**4*g + 17280*pi*adip**2*aind*d**3*g**2 + 4608*p
i*adip**2*aind*d**3*g + 6912*pi*adip**2*aind*d**2*g**2 + 16*pi**2*adip**2*d**12 + 120*p
i**2*adip**2*d**11*g + 192*pi**2*adip**2*d**11 + 144*pi**2*adip**2*d**10*g**2 + 1728*pi
**2*adip**2*d**10*g + 960*pi**2*adip**2*d**10 + 2160*pi**2*adip**2*d**9*g**2 + 10944*pi
**2*adip**2*d**9*g + 2560*pi**2*adip**2*d**9 + 14256*pi**2*adip**2*d**8*g**2 + 39936*pi
**2*adip**2*d**8*g + 3840*pi**2*adip**2*d**8 + 54144*pi**2*adip**2*d**7*g**2 + 92160*pi
**2*adip**2*d**7*g + 3072*pi**2*adip**2*d**7 + 129600*pi**2*adip**2*d**6*g**2 + 138240*
pi**2*adip**2*d**6*g + 1024*pi**2*adip**2*d**6 + 200448*pi**2*adip**2*d**5*g**2 + 13209
6*pi**2*adip**2*d**5*g + 195840*pi**2*adip**2*d**4*g**2 + 73728*pi**2*adip**2*d**4*g +
110592*pi**2*adip**2*d**3*g**2 + 18432*pi**2*adip**2*d**3*g + 27648*pi**2*adip**2*d**2*
g**2)
```

```
return res
```

```
def d6(d, g, adip, aind):
```

```
res = 2*(2*pi)**6*(d + 3*g)**2*(729*pi**4*d**14 + 20412*pi**4*d**13 + 265356*pi**4*
d**12 + 2122848*pi**4*d**11 + 11675664*pi**4*d**10 + 46702656*pi**4*d**9 + 140107968*pi
**4*d**8 + 320246784*pi**4*d**7 + 560431872*pi**4*d**6 + 747242496*pi**4*d**5 + 7472424
96*pi**4*d**4 + 543449088*pi**4*d**3 + 271724544*pi**4*d**2 + 83607552*pi**4*d + 119439
36*pi**4)*(9289728*d*pi**4 - 995328*adip*pi**3 + 1327104*pi**4 + 30191616*d**2*pi**4 +
60383232*d**3*pi**4 + 83026944*d**4*pi**4 + 83026944*d**5*pi**4 + 62270208*d**6*pi**4 +
35582976*d**7*pi**4 + 15567552*d**8*pi**4 + 5189184*d**9*pi**4 + 1297296*d**10*pi**4 +
235872*d**11*pi**4 + 29484*d**12*pi**4 + 2268*d**13*pi**4 + 81*d**14*pi**4 - 16422912*a
dip*d**2*pi**3 - 27371520*adip*d**3*pi**3 - 30792960*adip*d**4*pi**3 - 24634368*adip*d
```

```
*5*pi**3 - 14370048*adip*d**6*pi**3 - 6158592*adip*d**7*pi**3 - 1924560*adip*d**8*pi**3  
- 427680*adip*d**9*pi**3 - 64152*adip*d**10*pi**3 - 5832*adip*d**11*pi**3 - 243*adip*d*  
*12*pi**3 - 5971968*adip*d*pi**3)  
    return res
```

Coefficients for angle functions, Eqs. (10)

In [3]:

```
pi = np.pi
def Atheta(nu,d,g,adip, aind):
    w = 2*np.pi*nu
    res = - (9*d*w**2*pi*(d + 2)**4*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g - 12
*adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*
pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi
*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*a
dip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g
+ 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*a
dip*aind*d*g))/(2*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15
120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2
- 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*a
dip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*a
dip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 +
2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g
*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip
*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*a
dip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind
*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)) - (d*(72*adip*g + 4*adip*d**3 + 2*ad
ip*d**4 - 3*pi*g*(d + 2)**5 + 108*adip*d*g + 60*adip*d**2*g + 12*adip*d**3*g)*(32*pi*ad
ip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d
**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456
*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d
**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d
**5*g**2 + 6*adip*aind*d**2*g**2)*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2
+ 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*
g + 72*adip*aind*g + 32*pi*adip*d**2 + 32*pi*adip*d**3 - 8*pi*adip*d**5 - 2*pi*adip*d**
6 + 1344*d*g*pi**2 + 8*adip*aind*d**2 + 192*pi*adip*d**2*g - 72*pi*adip*d**3*g - 66*pi*
adip*d**4*g - 12*pi*adip*d**5*g + 480*pi*adip*d*g + 48*adip*aind*d*g))/(12*pi*((32*pi*a
dip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*
d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 345
6*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d
**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*
d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*
d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 +
288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 +
4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d
**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g
+ 72*adip*aind*d*g)**2)*(d + 2)**4) - (d*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g
- 12*adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 1
8144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi
**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g +
24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*p
i*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)*
(768*g*pi**2 + 5376*d**2*g*pi**2 + 5376*d**3*g*pi**2 + 3360*d**4*g*pi**2 + 1344*d**5*g*
pi**2 + 336*d**6*g*pi**2 + 48*d**7*g*pi**2 + 3*d**8*g*pi**2 + 576*pi*adip*g + 144*adip*
aind*g + 32*pi*adip*d**3 + 32*pi*adip*d**4 - 8*pi*adip*d**6 - 2*pi*adip*d**7 + 3072*d*g
*pi**2 + 8*adip*aind*d**3 + 768*pi*adip*d**2*g + 48*pi*adip*d**3*g - 180*pi*adip*d**4*g
- 84*pi*adip*d**5*g - 12*pi*adip*d**6*g + 48*adip*aind*d**2*g + 1152*pi*adip*d*g + 144*
adip*aind*d*g))/(12*pi*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2
- 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*p
i**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4
*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36
*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi
**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*
d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*p
i*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 76
8*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adi
p*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)*(d + 2)**3)
```

```

return res
def Btheta(nu,d,g,adip, aind):
    w = 2*np.pi*nu
    res = (d*w*(72*adip*g + 4*adip*d**3 + 2*adip*d**4 - 3*pi*g*(d + 2)**5 + 108*adip*d*
g + 60*adip*d**2*g + 12*adip*d**3*g)*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi
**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*ad
ip*g + 72*adip*aind*g + 32*pi*adip*d**2 + 32*pi*adip*d**3 - 8*pi*adip*d**5 - 2*pi*adip*
d**6 + 1344*d*g*pi**2 + 8*adip*aind*d**2 + 192*pi*adip*d**2*g - 72*pi*adip*d**3*g - 66*
pi*adip*d**4*g - 12*pi*adip*d**5*g + 480*pi*adip*d*g + 48*adip*aind*d*g)*(384*g*pi**2 +
2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g
*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip
*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*a
dip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind
*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g))/(4*pi*((32*pi*adip*d**3*g - 12096*d*w**2
*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d
**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip
*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g
**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d
**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g
*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*
aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*
g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4
*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)*
(d + 2)**4) - (3*d*w*pi*(d + 2)**4*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g - 12*
adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*
d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 -
378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*
adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip
*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2))/(2*((
32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2
- 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**
2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip
*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*p
i*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2
+ 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*
pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip
*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*p
i*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*a
dip*d*g + 72*adip*aind*d*g)**2)) + (d*w*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g
- 12*adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**
3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 28
8*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*
pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**
3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g +
72*adip*aind*d*g)*(768*g*pi**2 + 5376*d**2*g*pi**2 + 5376*d**3*g*pi**2 + 3360*d**4*g*pi
**2 + 1344*d**5*g*pi**2 + 336*d**6*g*pi**2 + 48*d**7*g*pi**2 + 3*d**8*g*pi**2 + 576*pi*
adip*g + 144*adip*aind*g + 32*pi*adip*d**3 + 32*pi*adip*d**4 - 8*pi*adip*d**6 - 2*pi*ad
ip*d**7 + 3072*d*g*pi**2 + 8*adip*aind*d**3 + 768*pi*adip*d**2*g + 48*pi*adip*d**3*g -
180*pi*adip*d**4*g - 84*pi*adip*d**5*g - 12*pi*adip*d**6*g + 48*adip*aind*d**2*g + 1152
*pi*adip*d*g + 144*adip*aind*d*g))/(4*pi*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 181
44*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**
2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24
*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*
adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2
+ 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*
d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*p
i*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*a
dip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*ad
ip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)*(d + 2)**3)
return res
def Aphi(nu,d,g,adip, aind):

```

```

w = 2*np.pi*nu
res = (d*(72*adip*g + 4*adip*d**3 + 2*adip*d**4 - 3*pi*g*(d + 2)**5 + 108*adip*d*g
+ 60*adip*d**2*g + 12*adip*d**3*g)*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g - 12*
adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*
d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 -
378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*
adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip
*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2))/(12*
((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**
2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi
**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*ad
ip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6
*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**
2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*
g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*ad
ip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432
*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi
*adip*d*g + 72*adip*aind*d*g)**2)) + (d*(d + 2)*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36
*pi*d*g - 12*adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(48*pi*g - 36*adip*g - 2*adip*d**3
+ 96*pi*d*g - 36*adip*d*g + 72*pi*d**2*g + 24*pi*d**3*g + 3*pi*d**4*g - 12*adip*d**2*g)
*(32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**
2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi
**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*ad
ip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6
*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2))/(12*((32*pi*adip*d**3*g - 12096*d*w**2*pi*
**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w
**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**
4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2
+ 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2
*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi
**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind
*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi
**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g +
15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)) - (9
*w**2*pi*(d + 2)**3*(72*adip*g + 4*adip*d**3 + 2*adip*d**4 - 3*pi*g*(d + 2)**5 + 108*ad
ip*d*g + 60*adip*d**2*g + 12*adip*d**3*g)*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*
g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*
pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi
*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*
g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72
*adip*aind*d*g))/(2*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 -
15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi
**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*p
i*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*p
i*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**
2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d
**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi
adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*
pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*
aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2))
return res
def Bphi(nu,d,g,adip, aind):
w = 2*np.pi*nu
res = - (3*w*pi*(d + 2)**3*(72*adip*g + 4*adip*d**3 + 2*adip*d**4 - 3*pi*g*(d + 2)*
*5 + 108*adip*d*g + 60*adip*d**2*g + 12*adip*d**3*g)*(32*pi*adip*d**3*g - 12096*d*w**2*
pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**
5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*
d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g*
*2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d
**2*g**2))/(2*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d
**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27

```

```

*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*
d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*
d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 201
6*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi
**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d*
**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip
*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d*
**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2)) - (d*w*(72*adip*g + 4*adip*d**3 + 2*adi
p*d**4 - 3*pi*g*(d + 2)**5 + 108*adip*d*g + 60*adip*d**2*g + 12*adip*d**3*g)*(24*pi*g -
18*adip*g - 2*adip*d**2 + 36*pi*d*g - 12*adip*d*g + 18*pi*d**2*g + 3*pi*d**3*g)*(384*g*
pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 4
2*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48
*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 +
768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*a
dip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)))/(4*((32*pi*adip*d**3*g - 12096*d
*w**2*pi**2 - 18144*d**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 22
68*d**5*w**2*pi**2 - 378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi
*adip*d**4*g + 24*pi*adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d
**2*g**2 + 72*pi*adip*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*a
ind*d**2*g**2)**2 + 9*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d
**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*
adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 13
44*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip
*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)
**2)) - (d*w*(d + 2)*(24*pi*g - 18*adip*g - 2*adip*d**2 + 36*pi*d*g - 12*adip*d*g + 18*
pi*d**2*g + 3*pi*d**3*g)*(48*pi*g - 36*adip*g - 2*adip*d**3 + 96*pi*d*g - 36*adip*d*g +
72*pi*d**2*g + 24*pi*d**3*g + 3*pi*d**4*g - 12*adip*d**2*g)*(384*g*pi**2 + 2016*d**2*g*
pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**5*g*pi**2 + 42*d**6*g*pi**2 + 3*d
**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*adip*d**3 + 48*pi*adip*d**4 + 24*p
i*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip*aind*d**3 + 768*pi*adip*d**2*g +
432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*d**5*g + 24*adip*aind*d**2*g + 720
*pi*adip*d*g + 72*adip*aind*d*g)))/(4*((32*pi*adip*d**3*g - 12096*d*w**2*pi**2 - 18144*d
**2*w**2*pi**2 - 15120*d**3*w**2*pi**2 - 7560*d**4*w**2*pi**2 - 2268*d**5*w**2*pi**2 -
378*d**6*w**2*pi**2 - 27*d**7*w**2*pi**2 - 3456*w**2*pi**2 + 48*pi*adip*d**4*g + 24*pi*
adip*d**5*g + 4*pi*adip*d**6*g + 4*adip*aind*d**3*g + 48*pi*adip*d**2*g**2 + 72*pi*adip
*d**3*g**2 + 36*pi*adip*d**4*g**2 + 6*pi*adip*d**5*g**2 + 6*adip*aind*d**2*g**2)**2 + 9
*w**2*(384*g*pi**2 + 2016*d**2*g*pi**2 + 1680*d**3*g*pi**2 + 840*d**4*g*pi**2 + 252*d**
5*g*pi**2 + 42*d**6*g*pi**2 + 3*d**7*g*pi**2 + 288*pi*adip*g + 72*adip*aind*g + 32*pi*a
dip*d**3 + 48*pi*adip*d**4 + 24*pi*adip*d**5 + 4*pi*adip*d**6 + 1344*d*g*pi**2 + 4*adip
*aind*d**3 + 768*pi*adip*d**2*g + 432*pi*adip*d**3*g + 126*pi*adip*d**4*g + 15*pi*adip*
d**5*g + 24*adip*aind*d**2*g + 720*pi*adip*d*g + 72*adip*aind*d*g)**2))
return res

def theta(nu,d,g,adip, aind, t):
    res = Atheta(nu,d,g,adip, aind)*np.sin(2*np.pi*nu*t) + Btheta(nu,d,g,adip, aind)*np
.cos(2*np.pi*nu*t)
    return res

def phi(nu,d,g,adip, aind, t):
    res = Aphi(nu,d,g,adip, aind)*np.sin(2*np.pi*nu*t) + Bphi(nu,d,g,adip, aind)*np.cos
(2*np.pi*nu*t)
    return res

```

Closed trajectories in theta-phi space, Fig. (4)

In [4]:

```
# PARAMETERS
valchi = 1.125*10**(-12);
valh = 0.2e-12;
valmu0 = 4*np.pi*10**(-7);
valB0 = 0.00215;
valg = np.log(100.);
valr = 1.5e-6;
vald = 2.;
valmu = 0.001;
valln = valr;
valadip = valmu0*valchi/valr**3
valaind = valmu0*valh/(valr**3*valB0)
vall = vald*valr
valtm=np.pi*valmu*vall**3/9./valB0/valh
```

In [5]:

```
listnu = valtm*np.array([1, 10, 20, 30])

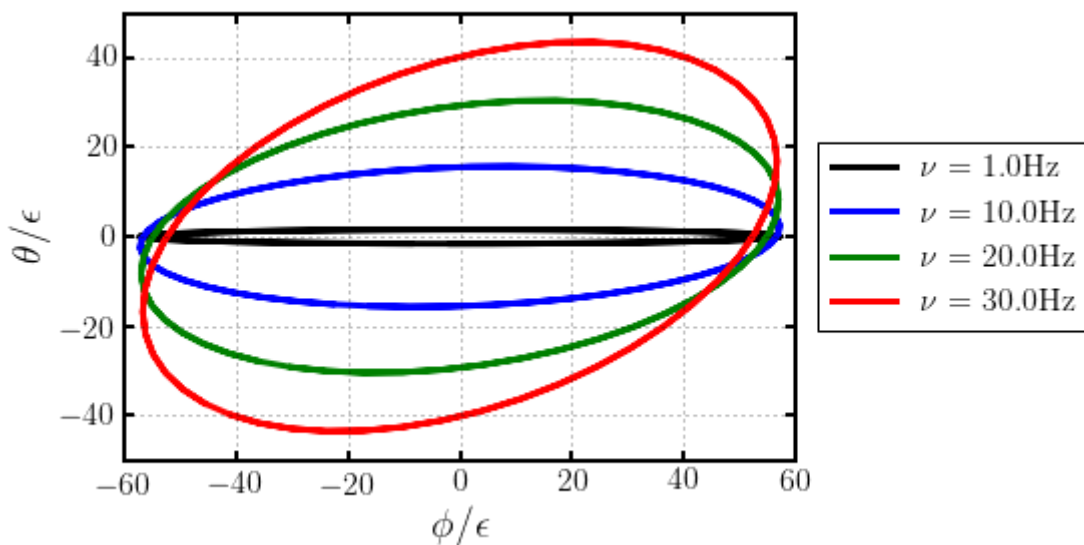
listst = ['-','--','-.', ':']

fig = plt.figure()
ax = plt.subplot(111)

for n in range(len(listnu)):
    nu = listnu[n]
    listt = 2*(1/nu)*np.arange(0,1,0.01)
    trajtheta = theta(nu, vald, valg, valadip, valaind, listt)*180./np.pi
    trajphi = -phi(nu, vald, valg, valadip, valaind, listt)*180./np.pi

    ax.plot(trajtheta, trajphi, marker='',lw = 3, ls = '--',color=clrs[n], label=r'$\nu$
= '+str(nu/(valtm))+ 'Hz')
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1.01, 0.5))
ax.grid()
ax.set_ylabel(r'$\theta/\epsilon$')
ax.set_xlabel(r'$\phi/\epsilon$')
#ax.set_legend(loc=2)
plt.savefig('Analytic_trajectory.jpg', format='jpg', dpi=1000, bbox_inches='tight', pad
_inches=0.1)

plt.show()
```



In []: