Protokoll PAP2 Versuch 213: Kreisel

Leonard Scheuer

Motivation/Versuchsziel

Das Verhalten ienes Kreisels soll untersucht werden. Insbesondere sollen dabei Effekte wie Reibungsverluste und und Dämpfung des Kreisels gemessen werden. Zur Vereinfachung wird hier hauptsächlich ein symmetischer Kreisel betrachtet werden. Desweiteren werden Trägheitsmomente und Nutationsfrequenzen verschieden präparierter Kreisel betrachtet.

Grundlagen

Kräftefreier symmetrischer Kreisel

Ein starrer um einen Festpunkt rotierenden Körper nennen wir einen **Kreisel**. Ist dieser in seinem Schwerpunkt aufgehängt, so ist er **kräftefrei**, da dann kein(gravitatives) Drehmoment entsteht. Der Drehimpuls ist hier also erhalten. Sind zwei der Hauptträgheitsmomente gleich, so nennen wir den Kreisel **symmetrisch**. Einen Kreisel charakterisieren wir anhand drei Achsen:

- **Figurenachse F**
- **Drehimpulsachse $\vec{L}$**
- **Drehachse $\vec{\omega}$**

Ändert man die an der Figurenachse angreifende Kraft, sodass der Kreiselm (kurz) nicht mehr perfekt Kräftefrei ist, so "taumelt" der Kreisel, er führt eine **Nutation**sbewegung aus. Die Figurenachse bewegt sich dabei um die raumfeste Drehimpulsachse mit $\vec{\omega_N}$ herum, besitzt aber weiterhin eine Eigenrotation $\vec{\omega}_F$. Addieren wir diese Rotationen, so ergibt sich insgesamt:

$$\vec{\omega} = \vec{\omega}_N + \vec{\omega}_F \tag{1}$$

und

$$\omega_N = \frac{L}{I_x} \tag{2}$$

wobei I_x das entsprechende Trägheitsmoment ist. Für ω_N klein, kann man dies näheren:

$$L \approx I_z\omega_F \Longrightarrow \omega_N \approx \frac{I_z}{I_x}\omega_F \tag{3}$$

Für die Winkelgeschwindigkeit Ω des Körperkegels ergibt sich:

$$\Omega = \frac{I_x - I_z}{I_x}\omega_F \tag{4}$$

Es lässt sich

$$I_x - I_z = \frac{I_z}{\omega_F/\Omega - 1} \tag{5}$$

zeigen.

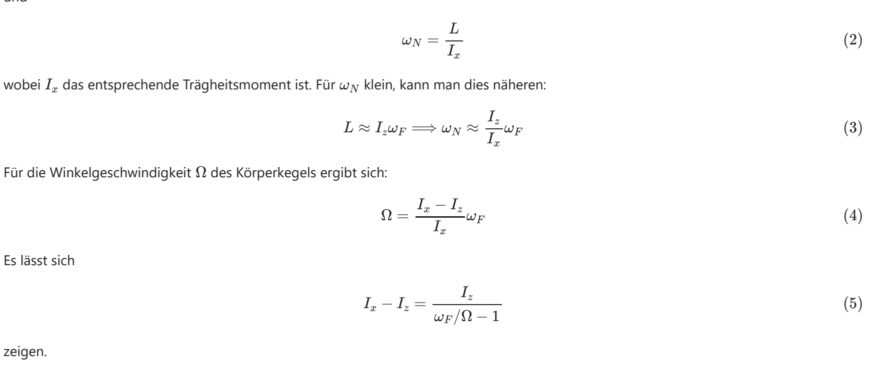


Abbildung 1: Bewegung des kräftefreien symmetischen Kreisels (Quelle: Script)

Der schwere, symmetrische Kreisel

Der Kreisel sei nun nicht mehr im Schwerpunkt, aber immernoch auf der Figurenachse aufgehängt. Es ergibt sich also ein Drehmoment durch die am Schwerpunkt angreifende Kraft:

$$\vec{M} = \vec{l} \times m\vec{g} \tag{6}$$

Wobei $\vec{l}$ vom Unterstützungspunkt zum Schwerpunkt reicht. Der Drehimpuls ist nun also Änderung unterworfen. Der Drehimpulsvektor bewegt sich nun in einem Kegel um die z-Achse herum, dies nennen wir **Präzession**. $\vec{\omega}_P$ sei die Präzessionsfrequenz, mit welcher $\vec{L}$ um die z-Achse rotiert. Man findet:

$$\omega_P = \frac{mgl}{I_z\omega_F} \tag{7}$$

sodass allgemein gilt:

$$\vec{M} = \overrightarrow{\omega_P} \times \vec{L} \tag{8}$$

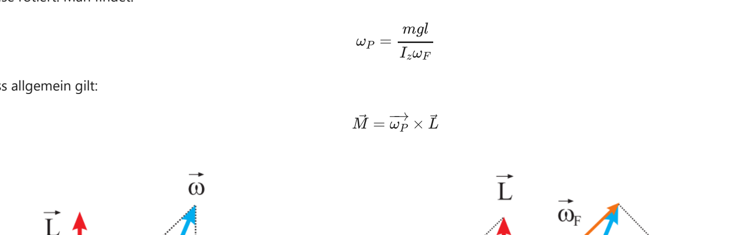


Abbildung 2: Nutationsbewegung (Quelle: Script)

Durchführung

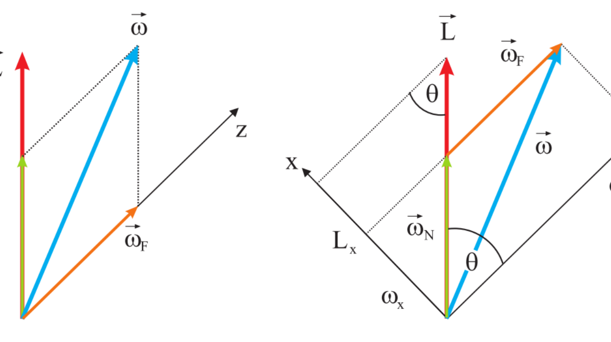


Abbildung 3: Versuchsaufbau (Quelle: Script)

Matrial

- Stahlkugel mit Aluminiumstab (Gesamtmasse: $4,164$kg, Kugelradius: $5,08$cm)
- Luftkissenpfanne
- 2 Gewichte (Masse: $9,85$g)
- Farbscheibe
- Scheibe mit konzentrischen Ringen
- Stroboskop
- Weiteres Frequenzmessgerät
- Stoppuhr
- Motor mit Netzgerät

Dämpfung des Kreisels

Am Stabende werden Gewichte angebracht, sodass der Kreisel Kräftefrei wird, also keine Präzession auftritt. Der Kreisel wird vertikal auf etwa 600 - 700min^{-1} beschleunigt. Anschließend wird für 12 Minuten alle 2 Minuten die Drehfrequenz messen.

Präzessionsmessung

Der Kreisel mit aufgesetzter Farbscheibe und einem in 20cm von der Stahlkugel angebrachten Gewicht wird vertikal auf $500\mathrm{min}^{-1}$ beschleunigt. Der Kreisel wird so nun (möglichst nutationsfrei) in drei verschiedenen Winkeln präpariert und die Periodendauer eines Umlaufs bestimmt.

Präzessionsdauer bei verschiedenen Krafteinwirkungen

Die Frequenz der Drehung um die Figurenachse und die Präzessionsdauer eines Kreisels werden bei vier Geschwindigkeiten zwischen 250 min^{-1} und 700min^{-1} gemessen:

- Ein Gewichtsstück bei 15 cm
- Ein Gewichtsstück bei 20 cm
- Zwei Gewichtsstücke bei 15 cm
- Zwei Gewichtsstücke bei 20 cm

Umlauf der momentanen Drehachse um die Figurenachse

Der Kreisel wird wieder in Nutation versetzt. Es werden mit dem Stroboskop 10 Wertepaare von ω_F und ω_N bestimmt. Das Stroboskop sendet dabei Lichtblitze einstellbarer Frequenz aus - Die kleinste eingestellte Frequenz, bei der die Farbscheibe zeitlich gleich erscheint, ist die zu messende Frequenz.

Messdaten: siehe Anhang

Auswertung

Vorversuch

1. Das Kreiselende bewegt sich orthogonal zur Kraftausübung. Der Kreisel verbleibt nach "Verschieben" in der Position, da durch Anwenden einer Kraft zur Drehachse hin kein Drehmoment entsteht, der Kreisel also weiter wie zuvor als kräftefrei betrachtet werden kann.
2. Hier lässt sich nun an einem Punkt auf der Farbscheibe die Farbe klar erkennen. Dieser Dreht sich nur um sich selbst. Aufgrund der Nutaion kann man jedoch ein Farbwechsel in dem Punkt beobachten.
3. Hier lassen sich kozentrische Kreise erkennen, dessen Mittelpunkt der Schnittpunkt der Ebene mit dem Drehimpulsvektor darstellt, wie erwartet.
4. Es lässt sich hier beobachten, dass die Präzessionsrichtung abhängig ist von der Wirkrichtung des Drehmomentes, mit Zusatzgewichten liegt der Schwerpunkt oberhalb der Unterstützung, das wirkende Drehmoment erfährt also einen Vorzeichenwechsel.

Dämpfung des Kreisels

Für die Dämpfungskonstante δ gilt:

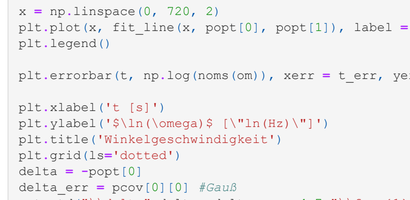
$$\omega(t) = \omega_0 e^{-\delta t} \tag{9}$$

Wir tragen ω daher logarithmisch auf um δ zu erhalten:

In [1]:
```python
#Funktionen zur späteren Verwendung.
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy as np
from IPython.display import Markdown, display, Latex
import pandas as pd
from scipy.optimize import curve_fit
from uncertainties import ufloat
from uncertainties.umath import *
from uncertainties import unumpy as unp
from uncertainties.unumpy import (nominal_values as noms,
                                  std_devs as stds)

table_counter=1

def comma_to_float(valstr):
  return float(valstr.decode("utf-8").replace(',','.'))

def round_up(n, decimals = 0):
    multiplier = 10 ** decimals
    return np.ceil(n * multiplier) / multiplier

def outwstd(name,c,Dc,zp=0,decimals=0,unit=""):
    if zp==0:
        display(Markdown("$"+name+" = (" +"{:.{}f}".format(round(c*10**(-zp),decimals),decimals)+" \pm "+"{:.{}
    else:
        display(Markdown( "$ "+name+" = ( " +"{:.{}f}".format(round(c*10**(-zp),decimals),decimals)+" \pm "+"{:

def out_sys_std(name,c,zp=0,decimals=0,unit=""):
    if zp==0:
        pass
        display(Markdown( "$ "+name+" = ( " +"{:.{}f}".format(round(c[0]*10**(-zp),decimals), decimals)+" \pm '
    else:
        display(Markdown( "$ "+name+" = ( " + "{:.{}f}".format(round(c[0]*10**(-zp),decimals),decimals)+" \pm '

def out_no_error(name,c,zp=0,decimals=0,unit=""):
    if zp==0:
        display(Markdown("$"+name+" = " +"{:.{}f}".format(round(c*10**(-zp),decimals),decimals)+unit+"$"))
    else:
        display(Markdown( "$ "+name+" = " +"{:.{}f}".format(round(c*10**(-zp),decimals),decimals)+" \cdot 10^{

def df_out(df,title):
    global table_counter
    display(Markdown("**Tabelle "+str(table_counter)+": " +title+ "**"))
    display(Markdown(df.to_markdown()))
    table_counter+=1

def sigma(g1,Dg1,g2,Dg2):
    return abs((g1-g2)/np.sqrt(Dg1**2+Dg2**2))

def quad_add(arr):
    quad_sum=0
    for i in arr:
        quad_sum+=i**2
    return (quad_sum**0.5)/len(arr)

def fit_line(x,a,b):
    return a*x+b

def fit_propline(x,a):
    return a*x
```

In [2]:
```python
t, omp, omp_err = np.loadtxt('daempfung.txt', unpack = True, skiprows = 1)

om=unp.uarray(omp,omp_err)

om *= 2 * np.pi / 60
t_err = np.zeros(len(t)) + 1

popt, pcov = curve_fit(fit_line, t, unp.log(noms(om)), sigma = stds(om)/noms(om)) #Gauß, t_err neglegible

x = np.linspace(0, 720, 2)
plt.plot(x, fit_line(x, popt[0], popt[1]), label = 'Ausgleichsgerade')
plt.legend()

plt.errorbar(t, np.log(noms(om)), xerr = t_err, yerr = stds(om)/noms(om), fmt = 'bx', capsize = 3)

plt.xlabel('t [s]')
plt.ylabel('$\ln(\omega)$ [\"ln(Hz)\"]')
plt.title('Winkelgeschwindigkeit')
plt.grid(ls='dotted')
delta = -popt[0]
delta_err = pcov[0][0] #Gauß
outwstd("\\delta",delta, delta_err,-4,7,"\\frac{1}{s}")
```

$\delta = (6.2074088 \pm 0.0000010)10^{-4}\frac{1}{s}$



Wir bestimmen nun die Halbwertszeit mit:

$$t_{1/2} = \frac{\ln(2)}{\delta} \tag{10}$$

$$\Delta t_{1/2} = \ln(2)\frac{\Delta\delta}{\delta^2} \tag{11}$$

In [3]:
```python
t_hw = np.log(2)/delta
t_hw_err = np.log(2)*delta_err/(delta**2)
outwstd("t_{1/2}",t_hw, t_hw_err,0,5,"s")
```

$t_{1/2} = (1116.64498 \pm 0.00018)s$

Interpretation der Präzessionsdauern bei verscheidenen Startwinkeln

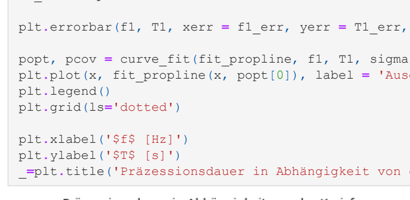
Wir wollen nun die gemessenen Präzessionsdauern diskutieren. Theoretisch erwarten wir dafür:

$$\omega_P = \frac{I_F\omega_F}{mgl} \tag{12}$$

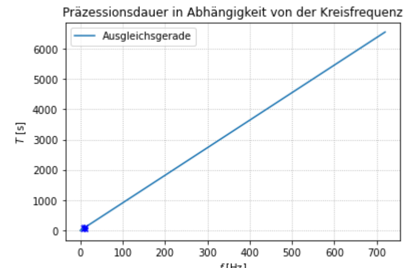
$$\therefore T = \frac{2\pi I_F}{mgl}\omega_F = \frac{4\pi^2 I_F}{mgl}f \tag{13}$$

Insbesondere erwarten wir also $f \propto T$. Betrachten wir diesen Zusammenhang, so ist dies schwer erkennbar:

In [4]:
```python
f1, winkel, T1 = np.loadtxt('praezession_a.txt', unpack = True, skiprows = 1)
winkel_err=np.ones(len(winkel))*5
f1 = f1/60 #SI units
f1_err = (np.zeros(len(f1))+2)/60
T1_err = np.zeros(len(T1)) + 5

plt.errorbar(f1, T1, xerr = f1_err, yerr = T1_err, fmt = 'bx', capsize = 3)

popt, pcov = curve_fit(fit_propline, f1, T1, sigma = T1_err)
plt.plot(x, fit_propline(x, popt[0]), label = 'Ausgleichsgerade')
plt.legend()
plt.grid(ls='dotted')

plt.xlabel('$f$ [Hz]')
plt.ylabel('$T$ [s]')
_=plt.title('Präzessionsdauer in Abhängigkeit von der Kreisfrequenz')
```



In [5]:
```python
Tf=unp.uarray(T1,T1_err)/unp.uarray(f1,f1_err)

plt.errorbar(winkel, T1/f1, xerr = winkel_err, yerr =stds(Tf), fmt = 'bx', capsize = 3)
plt.grid(ls='dotted')

plt.xlabel('Winkel [$°$]')
plt.ylabel('$T/f[s^2]$')
_=plt.title('Präzessionsdauer/Frequenz in Abhängigkeit vom Anfangswinkel')
```



Die in letzterem Diagramm dargestellten Werte sollten alle konstant sein. Auch dies ist schwer zu erkennen. Jedoch sind auch die Messfehler groß.

Bestimmung des Trägheitsmoments

Wir wollen nun mit (13) I_F bestimmen. Ist k die Steigung in dem nachfolgenden Diagramm, so ergibt sich:

$$k = \frac{2\pi I_F}{gml} \tag{14}$$

$$\Leftrightarrow I_F = \frac{k \cdot gml}{2\pi} \tag{15}$$

$$\overset{\text{Gauß}}{\Rightarrow} \Delta I_F = I_F\sqrt{\left(\frac{\Delta k}{k}\right)^2 + \left(\frac{\Delta l}{l}\right)^2} \tag{16}$$

wobei wir ω_F durch die Anfangsfrequenz und den Abfall durch die Dämpfung beschreiben:

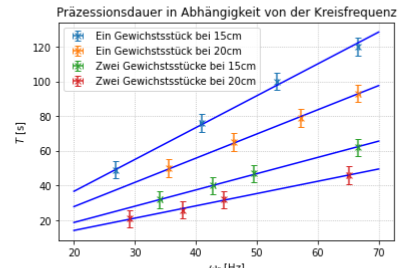
$$\bar{\omega}_F = \frac{1 + e^{-\delta T_p}}{2}\omega_{F,0} = (1 + e^{-\delta T_p})\pi \cdot f_0 \tag{17}$$

$$\Delta\bar{\omega}_F = \bar{\omega}_F\sqrt{\left(\frac{\Delta f_0}{f_0}\right)^2 + \left(\frac{e^{-\delta T_P} \cdot T_P}{1 + e^{-\delta T_P}}\Delta\delta\right)^2} \tag{18}$$

Wir mitteln schließlich noch über alle vier Messreihen und erhalten:

In [6]:
```python
f, T = np.loadtxt('praezession_b.txt', unpack = True, skiprows = 1, comments = '>')
f = f/60
f_err = (np.zeros(len(f))+2)/60
om_bar = (1+np.exp(- delta * T)) * np.pi * f
om_bar_err = om_bar * np.sqrt((f_err/f)**2 + (np.exp(- delta * T) * T * delta_err / (1+np.exp(- delta * T)))**2

T_err = np.zeros(len(T)) + 5

k = np.zeros(4)
k_err = np.zeros(4)
x = np.linspace(20, 70, 2)

Messreihe=["Ein Gewichtsstück bei 15cm","Ein Gewichtsstück bei 20cm","Zwei Gewichtsstücke bei 15cm","Zwei Ge

for i in range(0,4):
    plt.errorbar(om_bar[4*i:4*(i+1)], T[4*i:4*(i+1)], xerr = om_bar_err[4*i:4*(i+1)], yerr = T_err[4*i:4*(i+1)]
    popt, pcov = curve_fit(fit_propline, om_bar[4*i:4*(i+1)], T[4*i:4*(i+1)], sigma = T_err[4*i:4*(i+1)])
    k[i] = popt[0]
    k_err[i] = pcov[0][0]**0.5
    plt.plot(x, fit_propline(x, k[i]), 'b')

plt.xlabel('$\omega_F$ [Hz]')
plt.ylabel('$T$ [s]')
plt.title('Präzessionsdauer in Abhängigkeit von der Kreisfrequenz')
plt.grid(ls='dotted')
_=plt.legend()

l = np.array([0.15, 0.2, 0.15, 0.2])
l_err = np.zeros(4)+0.2*10**(-3)
m = np.array([1,1,2,2]) * 9.85 * 10**(-3)
I_F = k * 9.80984 * m * l / (2 * np.pi)
I_F_err = I_F * np.sqrt((l_err/l)**2 + (k_err/k)**2)
I_F_final = np.mean(I_F)
I_F_final_err = np.sqrt(np.sum(I_F_err**2))/len(I_F_err)
outwstd("I_F",I_F_final, I_F_final_err,-3,3,"kgm^2")
```

$I_F = (4.299 \pm 0.017)10^{-3}kgm^2$



Wir vergleichen dies mit dem theoretischen Trägheitsmoment der Kugel:

$$I = \frac{2}{5}mr^2 \tag{19}$$

In [7]:
```python
I_theo=(2/5)*4.164*0.0508**2
out_no_error("I_{theo}",I_theo,zp=-3,decimals=2,unit="kgm^2")
out_no_error("\sigma",sigma(I_theo,0,I_F_final,I_F_final_err),zp=0,decimals=2,unit="")
```

$I_{theo} = 4.30 \cdot 10^{-3}kgm^2$

$\sigma = 0.04$

Man findet hier also eine gute Übereinstimmung.

Bestimmung von $I_\perp$ über Umlauffrequenz Ω

Wir wollen nun das Trägheitsmoment orthogonal zur Figuenachse ermitten. Dafür verwenden wir (4) und erhalten bei Auftragung von Ω gegen ω_F die Steigung

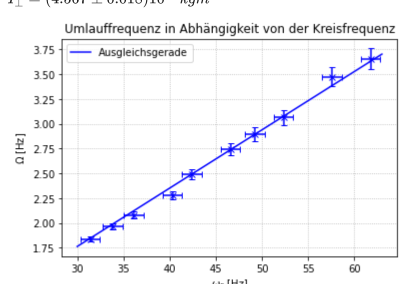
$$m = \frac{I_\perp - I_F}{I_\perp} \tag{20}$$

$$\therefore I_\perp = \frac{I_F}{1 - m} \tag{21}$$

$$\therefore \Delta I_\perp = I_\perp\sqrt{\left(\frac{\Delta I_F}{I_F}\right)^2 + \left(\frac{\Delta m}{1 - m}\right)^2} \tag{22}$$

In [8]:
```python
om2, om_T = np.loadtxt('momDrehachse.txt', unpack = True, skiprows = 1)

om2 *= 2 * np.pi / 60 #winkelgeschwindigkeit
om2_err = np.zeros(len(om2)) + 10 * 2 * np.pi / 60
om_T_err = 20 * np.pi / om_T**2 * 0.5
om_T = 20 * np.pi / om_T #winkelgeschwindigkeit

plt.errorbar(om2, om_T, xerr = om2_err, yerr = om_T_err, fmt = 'bx', capsize = 3)

popt, pcov = curve_fit(fit_propline, om_T, om2, sigma = om2_err) #om_T_err neglegible

x = np.linspace(30, 63, 2)

plt.plot(x, fit_propline(x, 1/popt[0]), 'b', label = 'Ausgleichsgerade')

plt.xlabel('$\omega_F$ [Hz]')
plt.ylabel('$\Omega$ [Hz]')
plt.title('Umlauffrequenz in Abhängigkeit von der Kreisfrequenz')
plt.grid(ls='dotted')
plt.legend()
I_perp = I_F_final/(1-1/popt[0])
I_perp_err = I_perp * np.sqrt((I_F_final_err/I_F_final)**2 + (pcov[0][0]/(popt[0]**2*(1-1/popt[0])))**2)
outwstd("I_\\perp",I_perp, I_perp_err,-3,3,"kgm^2")
```

$I_\perp = (4.567 \pm 0.018)10^{-3}kgm^2$



Bestimmung von $I_\perp$ über Nutation

Jetzt nutzen wir (3) um $I_\perp$ zu bestimmen:

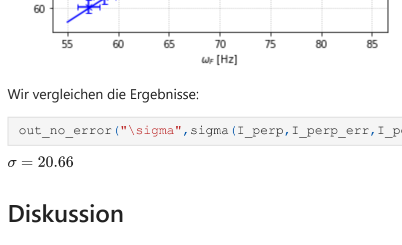
$$m\frac{\omega_N}{\omega_F} = \frac{I_F}{I_\perp} \tag{23}$$

$$\therefore I_\perp = \frac{I_F}{m} \tag{24}$$

$$\therefore \Delta I_\perp = I_\perp\sqrt{\left(\frac{\Delta I_F}{I_F}\right)^2 + \left(\frac{\Delta m}{m}\right)^2} \tag{25}$$

In [9]:
```python
om_F, om_N = np.loadtxt('nutation.txt', unpack = True, skiprows = 1)
om_F *= 2 * np.pi / 60
om_N *= 2 * np.pi / 60 #winkelgeschwindigkeit
om_F_err = np.zeros(len(om2)) + 10 * 2 * np.pi / 60
om_N_err = np.zeros(len(om2)) + 10 * 2 * np.pi / 60

plt.errorbar(om_F, om_N, xerr = om_F_err, yerr = om_N_err, fmt = 'bx', capsize = 3)

popt, pcov = curve_fit(fit_propline, om_F, om_N, sigma = np.sqrt(om_N_err**2+(I_F_final/I_perp*om_F_err)**2))

x = np.linspace(55, 85, 2)

plt.plot(x, fit_propline(x, popt[0]), 'b', label = 'Ausgleichsgerade')

plt.xlabel('$\omega_F$ [Hz]')
plt.ylabel('$\omega_N$ [Hz]')
plt.title('Nutationsfrequenz in Abhängigkeit von der Kreisfrequenz')
plt.grid(ls='dotted')
plt.legend()

I_perp2 = I_F_final/popt[0]
I_perp2_err = I_perp2 * np.sqrt((I_F_final_err/I_F_final)**2 + (pcov[0][0]/popt[0])**2)
outwstd("I_\\perp",I_perp2, I_perp2_err,-3,3,"kgm^2")
```

$I_\perp = (4.094 \pm 0.016)10^{-3}kgm^2$



Wir vergleichen die Ergebnisse:

In [10]:
```python
out_no_error("\sigma",sigma(I_perp,I_perp_err,I_perp2,I_perp2_err),zp=0,decimals=2,unit="")
```

$\sigma = 20.66$

Diskussion

In diesem Versuch haben wir uns zunächst im Vorversuch ein Gefühl für das Verhalten eines Kreisels verschafft. Anschließend haben wir duch Messung über 12 Minuten die Dämpfung des Kreisels bestimmt. Der Fehler der Dämpfungskonstanten ist sehr klein (6 Größenordnungen kleiner als die Größe selbst). Danach wollten wir die Präzessionseigenschaften des Kreisels untersuchen. Wir hätten theoretisch erwartet, dass die Präzessionsdauern unabhängig vom Winkel sind, dies konnte jedoch nicht verifiziert werden. Die Fehler sind jedoch so groß, dass es auch nicht als widerlegt gesehen kann. Vermutlich liegen hier bei zumindest einem der Messwerte ein Messfehler vorliegt. Wir haben schließlich das Trägheitsmoment um die Figurenachse bestimmt, der ermittelte Wert stimmt äußerst gut mit dem theoretischen Wert überein ($< 0.05\sigma$). Schließlich haben wir das Trägheitsmoment orthogonal zur Figurenachse auf zwei verschiedene Arten bestimmt. Zuerst haben wir dafür die Umlauffrequenz Ω genutzt, dann die Nutation. Beide Werte unterscheiden sich Signifikant ($> 20\sigma$). (Mindestens) eine der Messungen scheint einen systematischen Fehler aufzuweisen. Errechnet man das theorische Trägheitsmoment, so erhält man einen nahezu identischen Wert zur ersten Messung ($4,56 \cdot 10^{-3}kgm^2$, Abweichung $< 0.5\sigma$). Das deutet darauf hin, dass der Fehler bei der Messung der Nutationsfrequenzen auftrat. Die Aufnahme der Messwerte mit dem Stoboskop könnte einen systematischen Fehler aufweisen. Auch wurde eine kleinwinkelnäherung in der zugrundeliegenden Theorie verwendet, eventuell war der Nutationskegel während der Messung zu weit geöffnet, damit diese ihre vollständige Gültigkeit behält.

Bemerke: Die große Scheibe ist zu schwer, der Schwerpunkt ist auch bei kleinstem Abstand zur Kugel nicht unterstützt. Daher wird nur die kleine verwendet.

1 Vorversuch

1.1 Beobachtung Vorversuch a)

Nach Justierung kommt es bei einem zur Seite Drücken des Kugellagers mit einem Fingers zu einer Bewegung der Figurenachse am Finger entlang, egal in welche Richtung der Druck ausgeübt wird, nach Wegziehen des Fingers bleibt die Drehachse parallel zur Figurenachse in dieser Position. Werden zwei Finger verwendet, so wird die Figuren- und Drehachse in die Position verlagert, in die sie geschoben wird. Es kommt zu keiner Präzession und Nutation.

1.2 Beobachtung Vorversuch b)

Die drehende Farbscheibe sieht bis auf in einem Punkt verwaschen aus: Dieser liegt in dem Punkt, um den sich der Kreisel dreht. Die Farbe ist nicht konstant, sondern ändert sich in konstanten Zeitabständen.

Mit der Ringscheibe ist es möglich, einen Punkt ohne Farbwechsel zu erzeugen.

1.3 Beobachtung Vorversuch c)

Während bei vertikaler Position der Kreisel jeder Punkt der Scheibe verwaschen aussieht, ist wie zuvor bei der Seite mit den zentrierten konzentrischen Kreisen bei einer Nutationsbewegung ein nicht verwaschener Punkt zu erkennen.

Versetzt man den Kreisel in zusätzliche Präzession, so wandert der farbfeste Punkt auf einer "girlandenförmigen" Bahn.

In den Fällen sind jeweils verwaschene Kreise zu sehen, wobei im ersten Fall dieser zentriert ist, in den anderen nicht.

1.4 Beobachtung Vorversuch d)

Wird der Kreisel ohne Farbscheibe im Uhrzeigersinn angedreht und die Aluminiumstange in die Horizontale versetzt, so präzessiert der Kreisel gegen den Uhrzeigersinn ohne und mit dem Uhrzeigersinn mit zusätzlichen Gewichten.

Andrehen des Kreisels in die andere Richtung lässt das Gegenteil geschehen.

2 Dämpfung des Kreisels

Es kam zu Fehlern beim Ablesen der Frequenz durch unterschiedliche Winkel!

Fehler Frequenz: 2 Umdrehung/min

Fehler verstrichene Zeit: 1s

Verstrichene Zeit [s]	Frequenz [Umdrehung/min]
0	683

Verstrichene Zeit [s]	Frequenz [Umdrehung/min]
122	623
240	585
360	543
484	502
601	468
720	435

3 Präzession

a) Fehler der Frequenz: 2 Umdrehung/min

Fehler Winkel: 5°

Fehler Präzessionsdauer: 5s

Frequenz [Umdrehung/min]	Winkel	Präzessionsdauer [s]
533	90°	77
535	45°	88
560	23°	82

b) Als Winkel 90° gewählt

Fehler der Frequenz: 2 Umdrehung/min

Fehler Präzessionsdauer: 5s

Fehler des Abstands: 0,2cm

Frequenz [Umdrehung/min]	Präzessionsdauer [s]
1 Gewicht bei 15cm	
660	120
525	100
260	49
400	76
1 Gewicht bei 20cm	
655	93
560	79
450	65
345	50
2 Gewichte bei 15cm	
648	62
480	47
414	40
329	32
2 Gewichte bei 20cm	
630	46
364	26

Frequenz [Umdrehung/min]	Präzessionsdauer [s]
430	32
280	21

4 Umlauf der momentanen Drehachse um die Figurenachse

Umlaufrichtung im Uhrzeigersinn

Fehler Frequenz: 10 Umdrehung/min

Fehler Umlaufdauer: 0.5s

Frequenz [Umdrehung/min]	10 Umlaufdauern [s]
590	17.2
550	18.1
500	20.5
470	21.7
445	22.9
405	25.2
385	27.6
345	30.2
323	32.0
300	34.2

5 Nutation

Fehler beide jeweils 10 Umdrehung/min

Frequenz Figurenachse (Stab) [Umdrehung/min]	Frequenz Scheibe [Umdrehung/min]
790	845
755	790
725	765
685	715
665	685
645	670
630	660
605	638
560	590
545	575