# Blatt01_A2

April 29, 2024

## 1 Aufgabe 2:

sorry für Denglisch

[19]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[20]:
```
E = 50 * 10 **9
m = 511 * 10**3
gamma = E/m
beta = np.sqrt(1-gamma**(-2))
```

**(a)** Berechnung von $\beta^2$

[21]:
```
beta_square = beta**2
print(f"beta square = {beta_square}")
```

```
beta square = 0.9999999998955516
```

mit $\beta^2 = 0.9999999998955516$ folgt

$$\frac{2+\sin^2(\theta)}{1-\beta^2\cos^2(\theta)} \approx \frac{2+\sin^2(\theta)}{1-\cos^2(\theta)}$$

$$\Rightarrow \text{Unstable für: } \theta \to 0 \quad \text{bzw.} \quad \theta \in (1\cdot 10^{-9}, 1\cdot 10^{-7}) \quad \text{(s. Plot)}$$

**(b)** unter Verwendung der Hinweise lautet die umgeschriebene Funktion

$$f = \frac{2\gamma^2+\gamma^2\sin^2\theta}{1+\gamma^2\beta^2\sin^2\theta}$$

**(c)**

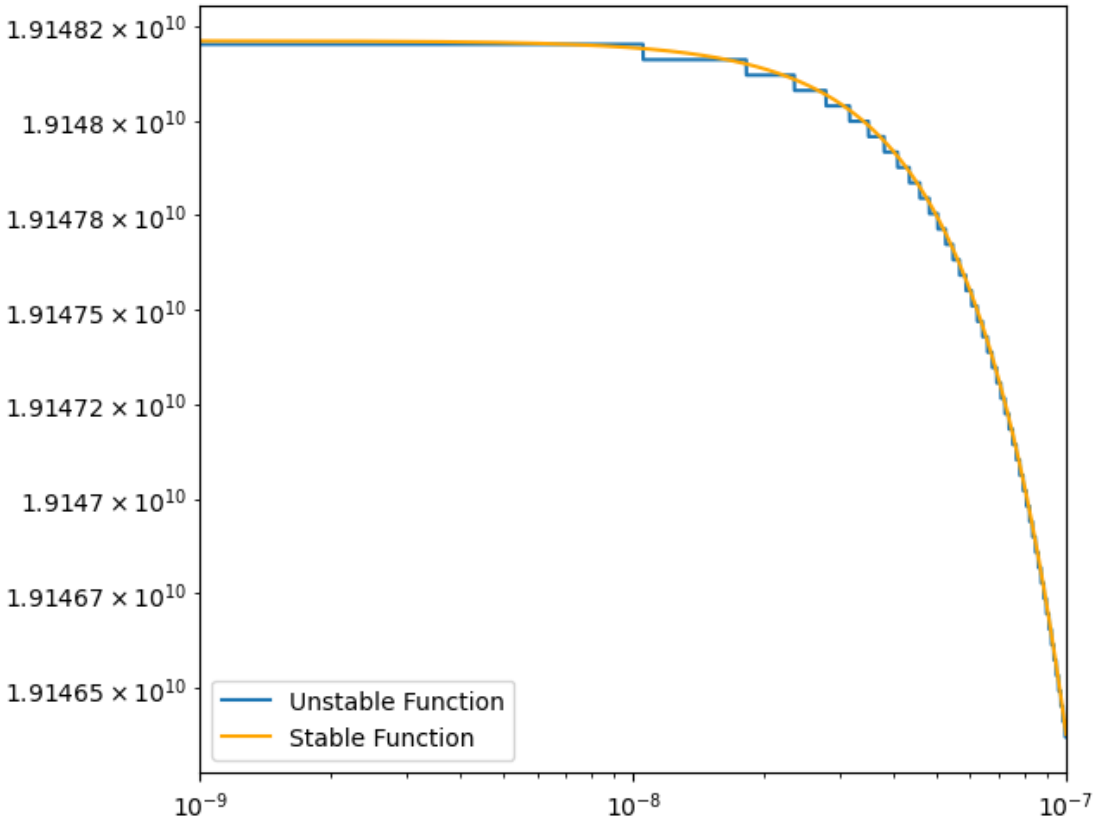
[22]:
```
def function_unstable(theta):
    return (2 + np.sin(theta)**2) / (1 - beta**2 * np.cos(theta)**2)

def function_stable(theta):
    return (2 * gamma**2 + gamma**2 * np.sin(theta)**2 ) / (1 + gamma**2 *␣
 ↪beta**2 * np.sin(theta)**2)
```

[23]:
```
x = np.logspace(-9,-7, 10000)

fig1, (ax1) = plt.subplots(1, 1, layout="constrained")
ax1.plot(x, function_unstable(x), label="Unstable Function")
ax1.plot(x, function_stable(x), color= "orange", label="Stable Function")
ax1.set_xscale("log")
ax1.set_yscale("log")
plt.xlim(1e-9,1e-7)
ax1.legend(loc="best")
```

[23]:
```
<matplotlib.legend.Legend at 0x1140f5f10>
```




**(d)** In General: Condition Nummer $K = \left| x \frac{f'(x)}{f(x)} \right|$ Here:

$$\frac{\mathrm{d}}{\mathrm{d}\theta} f(E,\theta) = -\frac{2\cos\theta\sin\theta(\beta^2\sin^2\theta + \beta^2\cos^2\theta + 2\beta^2 - 1)}{\left(\beta^2\cos^2(\theta) - 1\right)^2}$$

$$= -\frac{2\cos\theta\sin\theta(3\beta^2 - 1)}{\left(\beta^2\cos^2(\theta) - 1\right)^2}$$

$$\Rightarrow K = \left| -\theta \cdot \frac{2\cos\theta\sin\theta(3\beta^2-1)}{\left(\beta^2\cos^2(\theta)-1\right)^2} \cdot \frac{1-\beta^2\cos^2\theta}{2+\sin^2\theta} \right|$$

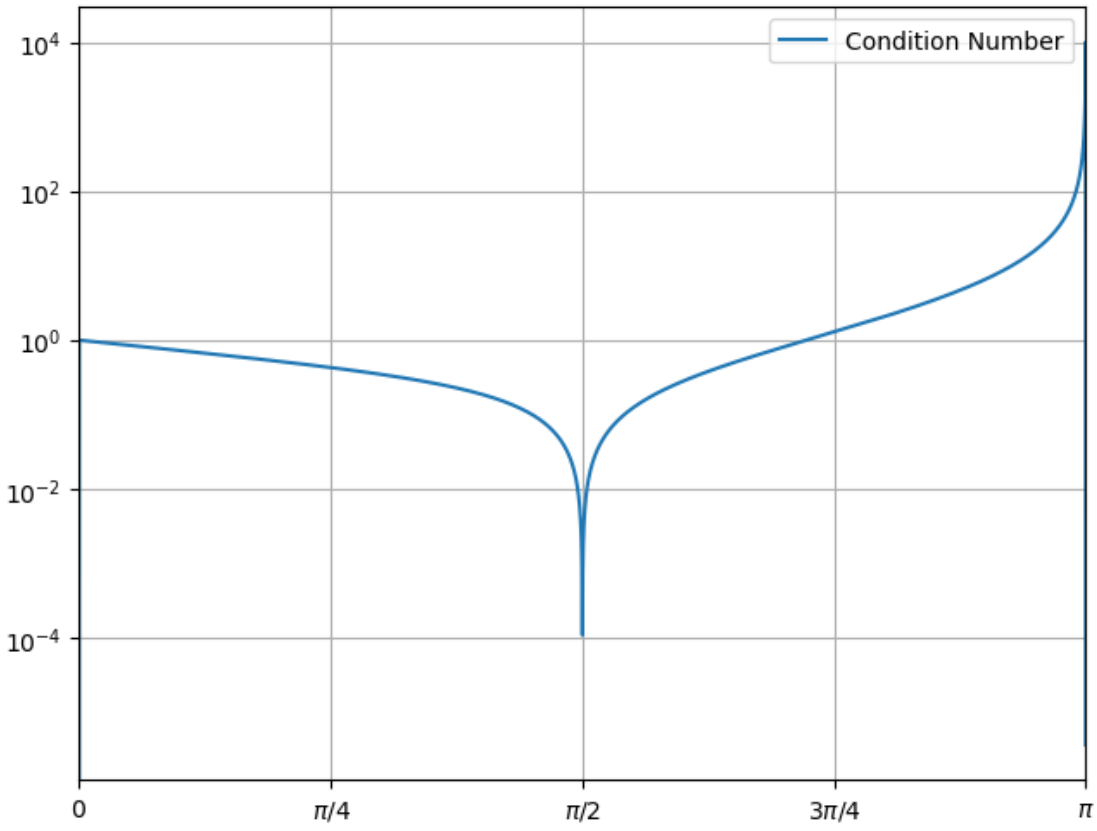
keine lineare und keine exponentielle $\theta$-Abhängigkeit

**(e)**

[24]:
```
def condition_number(x):
    return np.abs(-x * (2 * np.cos(x) * np.sin(x)*(3 * beta**2 -1)* (1 -␣
 ↪beta**2 * np.cos(x)**2))/((beta**2 * np.cos(x)**2 - 1)**2 *(2 + np.
 ↪sin(x))**2))

x2 = np.linspace(0, np.pi, 10000)
# x2 = np.linspace(-np.pi, 2*np.pi, 10000)
labels = [r"$0$", r"$\pi/4$", r"$\pi/2$", r"$3\pi/4$", r"$\pi$"]
fig3, ax3 = plt.subplots(1, 1, layout="constrained")
ax3.plot(x2, condition_number(x2), label="Condition Number")
ax3.set_xticks(np.arange(0, np.pi+0.01, np.pi/4))
ax3.set_xticklabels(labels)
ax3.set_yscale("log")
ax3.grid()
plt.xlim(0 ,np.pi)
ax3.legend(loc="best")
```

[24]:
```
<matplotlib.legend.Legend at 0x112e645f0>
```




$\Rightarrow$ ill-conditioned for $K >> 1 \Rightarrow \theta \in [\frac{7}{8}\pi, \pi]$

**(f)** Stability bezieht sich auf den Algorithmus und Condition bezieht sich auf das (mathematische) Problem. Unter Stabilität versteht man den Einfluss von Rundungsfehlern auf eine störanfällige maschinelle Rechenmethode. "Instabilities" lassen sich umgehen, indem den Algorithmus umgeschrieben wird, wie in Aufgabenteil 2b. Z.B. kann es zu Rundungsfehlern kommen, weil man sich einer Polstelle nähert, wie in 2a-c. "Condition" beschreibt den Einfluss eines anfänglichen Fehlers auf eine präzisere maschinelle Rechenmethode. Die Condition lässt sich allerdings nicht durch Umschreiben des Algorithmus lösen. Z.B. ist ein Problem "well-conditioned", wenn bei einer kleinen Veränderung der Übergabe, sich das Ergbenis nur minimal ändert und "ill-conditioned", wenn bei einer kleinen Veränderung der Übergabe, sich das Ergebnis startk verändert.