jupyter

December 31, 2024

[35]:
```python
import numpy as np
from sympy import *
init_printing(use_latex="mathjax")
```

[10]:
```python
from sympy import *
theta,E2,E3,t,hbar = symbols("theta E2 E3 t hbar", real=True)

# Nr.4
# (a)
psi = Matrix([1,0])
R = Matrix([[cos(theta), sin(theta)],[-sin(theta), cos(theta)]])
Rinv = R.adjoint()
E = Matrix([[E2,0],[0,E3]])

A = psi.T * R * exp(-I*E*t/hbar) * Rinv * psi
P = conjugate(A) * A
simplify(P[0])
```

[10]:
$$\left(e^{\frac{iE_2t}{\hbar}}\cos^2\left(\theta\right)+e^{\frac{iE_3t}{\hbar}}\sin^2\left(\theta\right)\right)\left(e^{-\frac{iE_3t}{\hbar}}\sin^2\left(\theta\right)+e^{-\frac{iE_2t}{\hbar}}\cos^2\left(\theta\right)\right)$$

[73]:
```python
# (b)
import scipy.constants as c
import matplotlib.pyplot as plt
plt.rcParams.update({"xtick.top": True , "ytick.right": True,
                     "xtick.minor.visible": True, "ytick.minor.visible": True,
                     "xtick.direction": "in" , "ytick.direction": "in",
                     "axes.labelsize": "large", "text.usetex": True, "font.
 ↪size": 13
                     })

theta = np.pi/4
Delta_m_sq = 0.0025 / c.c**2 * c.e # J
Delta_E = c.c * Delta_m_sq
E = 1e9 # eV
L = np.logspace(0,4,1000) # km
t = L / c.c

P_mu = 1 - np.sin(2*theta)**2 * np.sin(Delta_E / 2 * t / c.hbar)**2
```

```
P_tau = 1-P_mu

plt.plot(L, P_mu, label="$P_{\\nu_\\mu \\to \\nu_\\mu}(L)$")
plt.plot(L, P_tau, label="$P_{\\nu_\\mu \\to \\nu_\\tau}(L)$")
plt.xscale("log")
plt.xlim(min(L), max(L))
plt.xlabel("$L$ in km")
plt.ylabel("$P$",rotation=0)
plt.legend()
plt.title("Neutrino oscillations")
plt.savefig("neutrino_oscillations.pdf")
```