

Experimentalphysik Vb (Teilchen- und Astrophysik)

Übung 06

Aufgabe 1 *Lokale Eichinvarianz der Schrödingergleichung* (7 Punkte)

Zeigen Sie für den eindimensionalen Fall, dass die mit den kovarianten Ableitungen geschriebene Schrödingergleichung lokal eichinvariant ist: Es sei eine Wellenfunktion $\Psi(t, x)$ gegeben, die die neue Schrödingergleichung

$$\frac{1}{2m}(\mathbf{i}D_x)^2\Psi = \mathbf{i}D_0\Psi$$

erfüllt. Zeigen Sie, dass dann auch die lokal gedrehte Wellenfunktion

$$\Psi' = e^{\mathbf{i}Q\alpha(t,x)}\Psi$$

die neue Schrödingergleichung erfüllt. Dabei sind

$$D_x = -\frac{\partial}{\partial x} + \mathbf{i}QA \quad \text{und} \quad D_0 = \frac{\partial}{\partial t} + \mathbf{i}QV,$$

wobei A und V zwei Felder sind, die sich gemäß

$$A' = A + \partial\alpha/\partial x \quad \text{und} \quad V' = V - \partial\alpha/\partial t$$

transformieren. Kommentieren Sie Ihre Lösung!

Sie können den Beweis per Hand führen oder das Computeralgebrasystem `sympy` benutzen. Geben Sie in diesem Fall im Moodle als Lösung zu dieser Aufgabe bitte Ihr (kommentiertes) jupyter notebook sowohl als `ipynb`-Datei als auch als `pdf`-Export ab.

Tipps: In `sympy` können Sie mit `Function` abstrakte Funktionen mehrerer Veränderlicher definieren, mit `Derivative` erzeugen Sie abstrakt die (partielle) Ableitung einer Funktion. Die Ausführung der Produktregel in den auftretenden Ausdrücken können Sie mit `doit` erzwingen. Zum Aufstellen der kovarianten Ableitungen eignen sich gewöhnliche `python`-Funktionen, die man dann später auf `sympy`-Ausdrücke anwendet.

Aufgabe 2 *Eichinvarianz und Photonmasse* (2+1=3 Punkte)

Aus den Maxwell-Gleichungen folgt in kovarianter Schreibweise die Wellengleichung für das Photonfeld A^μ zu

$$\partial_\nu \partial^\nu A^\mu - \partial^\mu \partial^\nu A_\nu = j^\mu$$

mit der Stromdichte j^μ . Dabei wird die Summenkonvention verwendet. Klassisch ergeben sich das magnetische und das elektrische Feld aus A^μ zu $\vec{B} = \vec{\nabla} \times \vec{A}$ und $\vec{E} = -\vec{\nabla} A^0 - \partial \vec{A} / \partial t$.

- a. Zeigen Sie, dass diese Wellengleichung invariant unter der folgenden lokalen Eichtransformation ist:

$$A^\mu \rightarrow A^\mu - \partial^\mu \alpha.$$

- b. Zeigen Sie, dass hingegen die Wellengleichung eines massiven Vektorfelds,

$$(\partial_\nu \partial^\nu + m^2) A^\mu - \partial^\mu \partial^\nu A_\nu = j^\mu,$$

nicht eichinvariant ist.

Für massive Vektorteilchen gibt es also keine lokale Eichinvarianz.

Aufgabe 3 *Higgs-Fabrik* (1+1+4+3=9 Punkte)

Um die Eigenschaften des Higgs-Bosons präzise zu vermessen, soll ein e^+e^- -Collider gebaut werden.

- a. Welche Vorteile hat ein e^+e^- -Collider im Vergleich zu einem $p\bar{p}$ -Collider?
- b. Der dominante Produktionsprozess des Higgs-Bosons ist die sogenannte Higgs-Strahlung, bei der ein Z -Boson ein Higgs-Boson abstrahlt. Welche Schwerpunktsenergie muss der e^+e^- -Collider mindestens haben, damit Z - und Higgs-Teilchen reell erzeugt werden können?
- c. Das Higgs-Boson zerfällt am häufigsten in die schwersten Fermionen, die reell erzeugt werden können. Zeichnen Sie den kompletten Feynmangraphen der Produktion und dieses Zerfalls. Beschreiben Sie die experimentelle Signatur in einem Teilchendetektor wie CMS.
- d. Drücken Sie den Viererimpuls p_H des Higgs-Bosons durch die Viererimpulse von e^+ , e^- und Z aus. Leiten Sie eine Gleichung ab, die die Higgs-Masse m_H als Funktion der e^+e^- -Schwerpunktsenergie \sqrt{s} , Z -Masse m_Z und Energie E_Z des Z -Bosons darstellt. Beschreiben Sie eine Methode, mit der man die Higgs-Masse messen kann, ohne die Zerfallsteilchen des Higgs-Bosons zu rekonstruieren.

Aufgabe 4 *Neutrinooszillationen*

(4+2=6 Punkte)

Betrachten Sie wie in der Vorlesung die Eigenzustände $|\nu_\mu\rangle$ und $|\nu_\tau\rangle$ der schwachen Wechselwirkung, die mit den Masseneigenzuständen $|\nu_2\rangle$ und $|\nu_3\rangle$ über eine Mischungsmatrix verknüpft sind:

$$\begin{pmatrix} |\nu_\mu\rangle \\ |\nu_\tau\rangle \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} |\nu_2\rangle \\ |\nu_3\rangle \end{pmatrix}.$$

- a. Zeigen Sie explizit, dass die Wahrscheinlichkeit, ein bei $t = 0$ erzeugtes Myonneutrino nach der Zeit t wieder als Myonneutrino zu messen, gegeben ist durch

$$P_{\nu_\mu \rightarrow \nu_\mu}(t) = 1 - \sin^2(2\theta) \sin^2\left(\frac{E_3 - E_2}{2}t\right).$$

- b. Stellen Sie $P_{\nu_\mu \rightarrow \nu_\mu}(L)$ und $P_{\nu_\mu \rightarrow \nu_\tau}(L)$ für $\Delta m_{23}^2 = 0,0025 \text{ eV}^2$, $E = 1 \text{ GeV}$ und $\theta = 45^\circ$ im Bereich bis $L = 10\,000 \text{ km}$ in einem gemeinsamen Plot mit logarithmischer L -Achse grafisch dar.

Loesung_Eichinvarianz_Schroedinger

September 10, 2024

1 Musterlösung zur 6. Übung, Aufgabe 1

```
%config InteractiveShell.ast_node_interactivity='last_expr_or_assign'
```

```
import sympy as sp
from sympy import Function, Derivative, symbols
from sympy import exp, I
```

Wir definieren zunächst die benötigten Symbole und die Wellenfunktion, sowie die eichtransformierte der Wellenfunktion.

```
A, V, Q, t, x, m = symbols('A V Q t x m')
alpha = Function('alpha')(t, x)
Psi = Function('Psi')(t, x)
```

$\Psi(t, x)$

```
PsiPrime = exp(I*Q*alpha)*Psi
```

$\Psi(t, x)e^{iQ\alpha(t, x)}$

(1 Punkt)

Das Transformationsverhalten der beiden Felder A und V muss man wie folgt festlegen. Der Einfachheit halber beschränken wir uns auf den Fall einer Raumdimension, um das Problem gut mit `sympy` behandeln zu können.

```
Aprime = A + Derivative(alpha, x)
```

$A + \frac{\partial}{\partial x}\alpha(t, x)$

```
Vprime = V - Derivative(alpha, t)
```

$V - \frac{\partial}{\partial t}\alpha(t, x)$

(1 Punkt)

Als nächstes definieren wir die kovarianten Ableitungen und ihre Eichtransformierten wie in der Vorlesung eingeführt. Um sie wie einen Operator auf ein beliebiges Objekt anwenden zu können,

verwenden wir dazu eine gewöhnliche python-Funktion.

```
D = lambda f: -Derivative(f, x) + I*Q*A*f
```

```
<function __main__.<lambda>(f)>
```

```
Dprime = lambda f: -Derivative(f, x) + I*Q*Aprime*f
```

```
<function __main__.<lambda>(f)>
```

```
D0 = lambda f: Derivative(f, t) + I*Q*V*f
```

```
<function __main__.<lambda>(f)>
```

```
D0prime = lambda f: Derivative(f, t) + I*Q*Vprime*f
```

```
<function __main__.<lambda>(f)>
```

(1 Punkt)

Nach diesen Vorbereitungen können wir die neue Schrödingergleichung für die eichtransformierten Größen auswerten. Dazu bilden wir zunächst die linke Seite mit den gestrichenen Ableitungen und Wellenfunktion. Dabei erzwingen wir die Ausführung der Produktregel mit `doit()`.

```
NSGprime_links = sp.Rational(1,2)/m*sp.expand(I*Dprime(I*Dprime(PsiPrime))).  
    ↪doit()
```

$$\frac{A^2 Q^2 \Psi(t, x) e^{i Q \alpha(t, x)} + 2 i A Q e^{i Q \alpha(t, x)} \frac{\partial}{\partial x} \Psi(t, x) - e^{i Q \alpha(t, x)} \frac{\partial^2}{\partial x^2} \Psi(t, x)}{2m}$$

Die rechte Seite der neuen Schrödingergleichung mit den eichtransformierten Größen lautet entsprechend:

```
NSGprime_rechts = sp.expand(I*D0prime(PsiPrime).doit())
```

$$-QV\Psi(t, x)e^{iQ\alpha(t, x)} + ie^{iQ\alpha(t, x)}\frac{\partial}{\partial t}\Psi(t, x)$$

(1 Punkt)

Wenn die neue Schrödingergleichung erfüllt ist, muss die Differenz der beiden Seiten verschwinden:

```
NSGprime=NSGprime_links - NSGprime_rechts
```

$$QV\Psi(t, x)e^{iQ\alpha(t, x)} - ie^{iQ\alpha(t, x)}\frac{\partial}{\partial t}\Psi(t, x) + \frac{A^2 Q^2 \Psi(t, x) e^{i Q \alpha(t, x)} + 2 i A Q e^{i Q \alpha(t, x)} \frac{\partial}{\partial x} \Psi(t, x) - e^{i Q \alpha(t, x)} \frac{\partial^2}{\partial x^2} \Psi(t, x)}{2m}$$

Wir müssen zunächst ausklammern:

```
NSGprime=sp.factor(NSGprime)
```

$$\frac{i \left(A^2 Q^2 \Psi(t, x) - 2 A Q \frac{\partial}{\partial x} \Psi(t, x) + 2 i Q V m \Psi(t, x) + 2 m \frac{\partial}{\partial t} \Psi(t, x) - i \frac{\partial^2}{\partial x^2} \Psi(t, x) \right) e^{i Q \alpha(t, x)}}{2 m}$$

Um jetzt die Gültigkeit der neuen Schrödingergleichung für die nicht-transformierten Größen auszunutzen, müssen wir die beiden Seiten zunächst analog ausdrücken und dann eine geeignete Umformung vornehmen.

Linke Seite:

```
NSG_l = sp.expand(sp.Rational(1,2)/m*I*D(I*D(Psi)).doit())
```

$$\frac{A^2 Q^2 \Psi(t, x)}{2 m} + \frac{i A Q \frac{\partial}{\partial x} \Psi(t, x)}{m} - \frac{\frac{\partial^2}{\partial x^2} \Psi(t, x)}{2 m}$$

Rechte Seite:

```
NSG_r = sp.expand(I*D0(Psi).doit())
```

$$-Q V \Psi(t, x) + i \frac{\partial}{\partial t} \Psi(t, x)$$

Wir können jetzt die beiden Seiten gleichsetzen und dann z.B. nach $V Q \Psi$ umformen ...

```
VQPsi=sp.expand(sp.solve(sp.Eq(NSG_l, NSG_r), V*Q*Psi)[0])
```

$$-\frac{A^2 Q^2 \Psi(t, x)}{2 m} - \frac{i A Q \frac{\partial}{\partial x} \Psi(t, x)}{m} + i \frac{\partial}{\partial t} \Psi(t, x) + \frac{\frac{\partial^2}{\partial x^2} \Psi(t, x)}{2 m}$$

... und den so erhalten Ausdruck in unsere Differenz oben einsetzen:

```
sp.expand(NSGprime.subs(V*Q*Psi, VQPsi))
```

0

(3 Punkte)

Alternativ können wir für die eichtransformierten und die nicht-transformierten Größen jeweils die linken und rechten Seiten der neuen Schrödingergleichung durcheinander dividieren.

Für die linken Seiten:

```
sp.factor(NSGprime_links)
```

$$\frac{\left(A^2 Q^2 \Psi(t, x) + 2 i A Q \frac{\partial}{\partial x} \Psi(t, x) - \frac{\partial^2}{\partial x^2} \Psi(t, x) \right) e^{i Q \alpha(t, x)}}{2 m}$$

```
R_l = sp.simplify(sp.factor(NSGprime_links) / NSG_l)
```

$$e^{i Q \alpha(t, x)}$$

Für die rechten Seiten:

```
sp.factor(NSGprime_rechts)
```

$$i \left(i Q V \Psi(t, x) + \frac{\partial}{\partial t} \Psi(t, x) \right) e^{i Q \alpha(t, x)}$$

```
R_r = sp.simplify(sp.factor(NSGprime_rechts / NSG_r))
```

$$e^{iQ\alpha(t,x)}$$

Wieder bilden wir die (jetzt triviale) Differenz:

```
R_l - R_r
```

0

Physik Vb (Teilchen- und Astrophysik) Musterlösung zu Übung 6, Aufgabe 2

Wir betrachten zunächst die Wellengleichung für das Photonfeld A^μ ,

$$\partial_\nu \partial^\nu A^\mu - \partial^\mu \partial^\nu A_\nu = j^\mu.$$

Nach Einsetzen der Eichtransformation

$$A^\mu \rightarrow A^\mu - \partial^\mu \alpha$$

auf der linken Seite erhält man

$$\begin{aligned} & \partial_\nu \partial^\nu (A^\mu - \partial^\mu \alpha) - \partial^\mu \partial^\nu (A_\nu - \partial_\nu \alpha) \\ &= \partial_\nu \partial^\nu A^\mu - \partial_\nu \partial^\nu \partial^\mu \alpha - \partial^\mu \partial^\nu A_\nu + \partial^\mu \partial^\nu \partial_\nu \alpha \\ &= j^\mu + \partial^\mu \partial^\nu \partial_\nu \alpha - \partial_\nu \partial^\nu \partial^\mu \alpha \\ &= j^\mu + \partial^\mu \partial^\nu \partial_\nu \alpha - \partial^\mu \partial^\nu \partial_\nu \alpha \\ &= j^\mu, \end{aligned}$$

d.h. auch das transformierte Photonfeld erfüllt die Wellengleichung.
(2 Punkte)

Betrachten wir hingegen die Wellengleichung eines massiven Vektorfelds,

$$(\partial_\nu \partial^\nu + m^2) A^\mu - \partial^\mu \partial^\nu A_\nu = j^\mu,$$

so enthält diese zusätzlich den Massenterm $m^2 A^\mu$. Unter der Eichtransformation führt dieser zum zusätzlichen Auftreten eines Terms

$$-m^2 \partial^\mu \alpha,$$

der durch keine anderen Terme kompensiert wird. Diese Wellengleichung ist also nicht eichinvariant.

(1 Punkt)

Physik Vb (Teilchen- und Astrophysik) Musterlösung zu Übung 6, Aufgabe 3

- a.
- An einem e^+e^- -Collider kann die Schwerpunktsenergie genau eingestellt werden. Bei einem $p\bar{p}$ -Collider wechselwirken hingegen die Partonen (Quarks und Gluonen), deren Impulsanteile von Kollision zu Kollision variieren.
 - Dadurch dass die Elektronen und Positronen nicht aus Partonen bestehen und zudem keine Farbladung im Anfangszustand existiert, ist die Ereignissignatur im Detektor deutlich klarer. Es existieren fast ausschließlich die Signaturen des Endzustands des zentralen Prozesses.

(1 Punkt für mindestens einen richtigen Aspekt)

- b. Die minimale Schwerpunktsenergie entspricht der Summe der Massen von Z -Boson- und Higgs-Boson, sodass beide Bosonen “on-shell” produziert werden. Die minimale Schwerpunktsenergie beträgt somit

$$m_Z + m_H = 91 \text{ GeV} + 125 \text{ GeV} = 216 \text{ GeV}.$$

(1 Punkt)

- c. Das Higgs-Boson zerfällt vorrangig in zwei bottom-Quarks. Im Detektor misst man somit zum einen den Zerfall des Z -Bosons, welches in zwei geladene Leptonen, zwei Neutrinos oder zwei Quarks zerfällt:
- Bei Elektronen misst man eine Spur im Tracker, sowie eine Energiedeposition im elektromagnetischen Kalorimeter.
 - Myonen hinterlassen ebenfalls eine Spur im Tracker, sowie eine Spur im Myondetektor.
 - Tau-Leptonen zerfallen frühzeitig in Elektronen, Myonen, Neutrinos und Pionen welche dann eine entsprechende Signatur im Detektor hinterlassen.
 - Neutrinos können nicht gemessen werden, wodurch ein signifikanter Anteil an fehlender transversaler Energie beobachtbar ist.
 - Quarks und Pionen erzeugen Jets. Geladene Teilchen hinterlassen eine Spur im Tracker. Das elektromagnetische sowie das hadronische Kalorimeter zeigen signifikante Energieeinträge. Die jeweiligen Zerfallsprodukte bilden eine invariante Masse, die der Masse des Z -Bosons entspricht.

(2 Punkte, je nach Vollständigkeit der Diskussion)

Zum anderen misst man den Zerfall des Higgs-Bosons, in diesem Falle in zwei bottom-Quarks. Bottom-Quarks hinterlassen messbar versetzte Vertices („Sekundärvertices“), sodass man die Spuren der Endzustandsteilchen

(Jets) zu einem vom Primärvertex verschiedenen Punkt rekonstruieren kann. Die beiden bottom-Quarks bilden eine invariante Masse, die der Masse des Higgs-Bosons entspricht.

(1 Punkt)

Das entsprechende Feynmandiagramm ist in Abbildung 1 zu sehen. Dabei ist $q \in \{u, d, s, c, b\}$.

(1 Punkt)

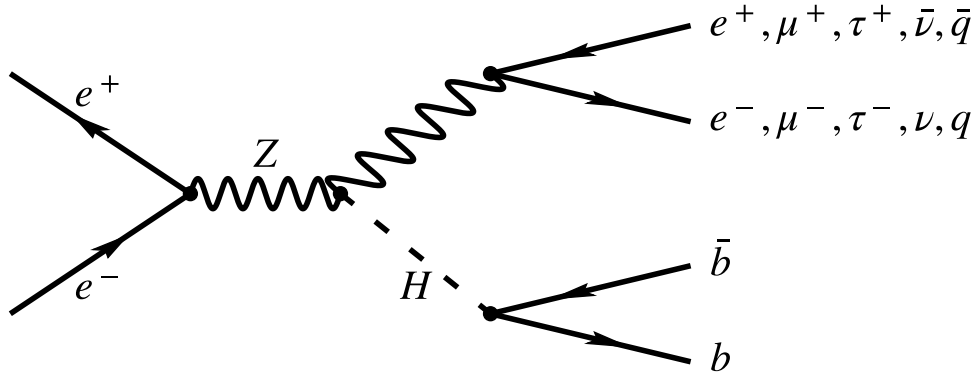


Abbildung 1: Feynmandiagramm zur Higgsstrahlung im e^+e^- -Collider. Dabei ist $q \in \{u, d, s, c, b\}$.

d. Aus der Viererimpulsbilanz, $p_{e^+} + p_{e^-} = p_Z + p_H$, folgt:

$$p_H = p_{e^+} + p_{e^-} - p_Z.$$

Im Schwerpunktsystem, das zugleich das Laborsystem ist, gilt $E_{e^+} = E_{e^-} \equiv E$ und damit liefert das Einsetzen der Viererimpulse:

$$p_H = \begin{pmatrix} E + E - E_Z \\ \vec{p}_{e^+} + \vec{p}_{e^-} - \vec{p}_Z \end{pmatrix}.$$

Außerdem gilt $\vec{p}_{e^+} = -\vec{p}_{e^-}$.

(1 Punkt)

Damit liefert Quadrieren:

$$m_H^2 = (2E - E_Z)^2 - \vec{p}_Z^2 = (2E)^2 - 4EE_Z + E_Z^2 - \vec{p}_Z^2 = s - 2\sqrt{s}E_Z + m_Z^2,$$

wobei wir noch $\sqrt{s} = 2E$ verwendet haben.

(1 Punkt)

Da die Schwerpunktsenergie und m_Z bekannt sind, lässt sich somit mit Hilfe der Verteilung der Energie des rekonstruierten Z -Bosons die Masse des Higgs-Bosons bestimmen.

(1 Punkt)

Loesung_Neutrinooszillationen

September 10, 2024

1 Musterlösung zur 6. Übung, Aufgabe 4

```
%config InteractiveShell.ast_node_interactivity='last_expr_or_assign'
```

```
import sympy as sp
from sympy import I
from sympy.matrices import Matrix
from sympy import sin, cos, exp
```

Hier definieren wir alle im folgenden benötigten Symbole:

```
theta, v0_mu, v0_tau = sp.symbols('theta nu_mu__0 nu_tau__0', real=True)
x, y, z, t, E2, E3, E, deltam2, L = sp.symbols('x y z t E_2 E_3 E Delta_m__2_
↳L', real=True)
```

1.1 a)

Wir definieren zunächst die Transformationsmatrix für die beiden Neutrinosorten, die die Masseneigenzustände in die Flavoreigenzustände überführt:

```
R = Matrix(2,2,[cos(theta), sin(theta), -sin(theta), cos(theta)])
```

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Danach geben wir die Zeitentwicklungsmatrix für die Masseneigenzustände an:

```
evo = sp.diag(exp(-I*E2*t), exp(-I*E3*t))
```

$$\begin{bmatrix} e^{-iE_2 t} & 0 \\ 0 & e^{-iE_3 t} \end{bmatrix}$$

Jetzt definieren wir uns als Startvektor die beiden Flavoreigenzustände zum Zeitpunkt $t = 0$. (Notation: $\nu_i^0 \equiv |\nu_i(0)\rangle$)

```
v0 = Matrix(2, 1, [v0_mu, v0_tau])
```

$$\begin{bmatrix} \nu_\mu^0 \\ \nu_\tau^0 \end{bmatrix}$$

Der Vektor $\begin{pmatrix} |\nu_\mu(t)\rangle \\ |\nu_\tau(t)\rangle \end{pmatrix}$ ergibt sich dann wie in der Vorlesung gezeigt zu

```
amp = R * evo * R**(-1) * v0
```

$$\begin{bmatrix} \nu_\mu^0 \left(\left(-\frac{\sin^2(\theta)}{\cos(\theta)} + \frac{1}{\cos(\theta)} \right) e^{-iE_2 t} \cos(\theta) + e^{-iE_3 t} \sin^2(\theta) \right) + \nu_\tau^0 \left(e^{-iE_3 t} \sin(\theta) \cos(\theta) - e^{-iE_2 t} \sin(\theta) \cos(\theta) \right) \\ \nu_\mu^0 \left(-\left(-\frac{\sin^2(\theta)}{\cos(\theta)} + \frac{1}{\cos(\theta)} \right) e^{-iE_2 t} \sin(\theta) + e^{-iE_3 t} \sin(\theta) \cos(\theta) \right) + \nu_\tau^0 \left(e^{-iE_3 t} \cos^2(\theta) + e^{-iE_2 t} \sin^2(\theta) \right) \end{bmatrix}$$

(1 Punkt)

Die Wahrscheinlichkeit dafür, ein bei $t = 0$ erzeugtes Myonneutrino nach der Zeit t wieder als Myonneutrino zu messen, ist gegeben durch

$$P_{\nu_\mu \rightarrow \nu_\mu}(t) = |\langle \nu_\mu(t) | \nu_\mu(0) \rangle|^2.$$

$\nu_\mu(t)$ ist der obere Eintrag des gerade berechneten Vektors. Um das Skalarprodukt mit dem Anfangszustand ν_μ^0 zu bilden, wollen wir manuell die Orthogonalität der Flavoreigenzustände ausnutzen. Dafür zerlegen wir den oberen Eintrag in seine Bestandteile, um daran abzulesen, wie wir auf den gewünschten Ausdruck, d.h. die Klammer hinter dem ν_μ^0 in der oberen Zeile zugreifen können.

```
amp[0].args
```

```
(nu_mu_0*((-sin(theta)**2/cos(theta) + 1/cos(theta))*exp(-I*E_2*t)*cos(theta) +
exp(-I*E_3*t)*sin(theta)**2),
nu_tau_0*(exp(-I*E_3*t)*sin(theta)*cos(theta) -
exp(-I*E_2*t)*sin(theta)*cos(theta)))
```

Damit erhalten wir $\langle \nu_\mu(t) | \nu_\mu(0) \rangle$ zu

```
a = amp[0].args[0].args[1].simplify()
```

$$e^{-iE_3 t} \sin^2(\theta) + e^{-iE_2 t} \cos^2(\theta)$$

Und damit $P_{\nu_\mu \rightarrow \nu_\mu}(t) = |\langle \nu_\mu(t) | \nu_\mu(0) \rangle|^2$:

```
p_mu = a*sp.conjugate(a)
```

$$\left(e^{iE_2 t} \cos^2(\theta) + e^{iE_3 t} \sin^2(\theta) \right) \left(e^{-iE_3 t} \sin^2(\theta) + e^{-iE_2 t} \cos^2(\theta) \right)$$

(1 Punkt)

Wir fassen zunächst Terme zusammen.

```
p_mu = p_mu.expand().powsimp()
```

$$e^{-iE_2 t + iE_3 t} \sin^2(\theta) \cos^2(\theta) + e^{iE_2 t - iE_3 t} \sin^2(\theta) \cos^2(\theta) + \sin^4(\theta) + \cos^4(\theta)$$

Jetzt beginnt die etwas mühsame Arbeit, um den Ausdruck in die in der Vorlesung gezeigte und in der Aufgabe geforderte Form zu bringen. Dazu erinnern wir uns zunächst an die relevanten trigonometrischen Identitäten.

$$e^{ix} + e^{-ix} = 2 \cos x$$

```
sp.simplify(exp(I*x)+exp(-I*x))
```

$$2 \cos(x)$$

$$\cos 2x = 1 - 2 \sin^2 x$$

```
sp.expand_trig(cos(2*x)).subs(cos(x)**2, 1-sin(x)**2)
```

$$1 - 2 \sin^2(x)$$

$$\sin 2x = 2 \sin x \cos x$$

```
sp.expand_trig(sin(2*x))
```

$$2 \sin(x) \cos(x)$$

Damit setzen wir ein.

```
p_mu = p_mu.subs(sin(theta)*cos(theta), sin(2*theta)/2)
```

$$\frac{e^{-iE_2t+iE_3t} \sin^2(2\theta)}{4} + \frac{e^{iE_2t-iE_3t} \sin^2(2\theta)}{4} + \sin^4(\theta) + \cos^4(\theta)$$

Noch eine Identität: $\sin^4 x + \cos^4 x = 1 - 2 \sin^2 x \cos^2 x$.

```
test = sin(x)**4+cos(x)**4 - (1-2*sin(x)**2*cos(x)**2)
```

$$\sin^4(x) + 2 \sin^2(x) \cos^2(x) + \cos^4(x) - 1$$

```
sp.simplify(test)
```

$$0$$

Auch das setzen wir ein:

```
p_mu = p_mu.subs(sin(theta)**4+cos(theta)**4, (1-2*sin(theta)**2*cos(theta)**2))
```

$$\frac{e^{-iE_2t+iE_3t} \sin^2(2\theta)}{4} + \frac{e^{iE_2t-iE_3t} \sin^2(2\theta)}{4} - 2 \sin^2(\theta) \cos^2(\theta) + 1$$

Umschreiben als Funktion von 2θ :

```
p_mu = p_mu.subs(sin(theta)*cos(theta), sin(2*theta)/2)
```

$$\frac{e^{-iE_2t+iE_3t} \sin^2(2\theta)}{4} + \frac{e^{iE_2t-iE_3t} \sin^2(2\theta)}{4} - \frac{\sin^2(2\theta)}{2} + 1$$

Am schwierigsten ist es, `sympy` dazu zu bringen, die komplexen Exponentialfunktionen durch einen Kosinus zu ersetzen. Dazu nehmen wir eine Substitution $z = E_3t - E_2t$ vor und müssen das über Umwege tun.

```
p_mu = p_mu.subs(E2*t, x).subs(E3*t, y).subs(I*x-I*y, I*z)
```

$$\frac{e^{iz} \sin^2(2\theta)}{4} - \frac{\sin^2(2\theta)}{2} + 1 + \frac{e^{-iz} \sin^2(2\theta)}{4}$$

Leider steht `sympy` immer noch auf dem Schlauch. Wir dividieren den Ausdruck kurz durch $\sin^2 2\theta$, rufen dann `simplify` auf und multiplizieren wieder mit $\sin^2 2\theta$.

```
p_mu = sp.simplify((p_mu/sin(2*theta)**2).simplify()*sin(2*theta)**2)
```

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

Jetzt hat es geklappt.

```
p_mu = p_mu.subs(cos(z), 1-2*sin(z/2)**2)
```

$$-\sin^2(2\theta) \sin^2\left(\frac{z}{2}\right) + 1$$

Nach dieser letzten Umformung substituieren wir zurück.

```
p_mu = p_mu.subs(z, (E2-E3)*t)
```

$$-\sin^2(2\theta) \sin^2\left(\frac{t(E_2 - E_3)}{2}\right) + 1$$

Das ist der gesuchte Ausdruck.

(2 Punkte)

Aus der totalen Wahrscheinlichkeit erhalten wir sofort $P_{\nu_\mu \rightarrow \nu_\tau}(t)$:

```
p_tau = 1-p_mu
```

$$\sin^2(2\theta) \sin^2\left(\frac{t(E_2 - E_3)}{2}\right)$$

Jetzt drücken wir noch wie in der Vorlesung t durch die Flugstrecke L und die Energien durch die Differenz der Massenquadrate aus. Wir rechnen hier zunächst noch ganz einfach in natürlichen Einheiten.

```
p_mu_f = p_mu.subs(E2-E3, deltam2/2/E).subs(t, L)
```

$$-\sin^2(2\theta) \sin^2\left(\frac{\Delta_m^2 L}{4E}\right) + 1$$

```
p_tau_f = p_tau.subs(E2-E3, deltam2/2/E).subs(t, L)
```

$$\sin^2(2\theta) \sin^2\left(\frac{\Delta_m^2 L}{4E}\right)$$

1.2 b)

Um die Wahrscheinlichkeiten für die gegebenen Werte als Funktion von L plotten zu können, wandeln wir zunächst unsere `sympy`-Ausdrücke in gewöhnliche `python`-Funktionen um:

```
p_mu_func = sp.lambdify((deltam2, E, theta, L), p_mu_f)
```

```
<function _lambdifygenerated(Delta_m__2, E, theta, L)>
```

```
p_tau_func = sp.lambdify((deltam2, E, theta, L), p_tau_f)
```

```
<function _lambdifygenerated(Delta_m__2, E, theta, L)>
```

Wir importieren noch benötigte Pakete und legen Parameter für gut lesbare Achsenbeschriftungen fest.

```
import numpy as np
import matplotlib.pyplot as plt
import astropy.constants as const
import astropy.units as u
```

```
plt.rcParams['font.size'] = 24.0
plt.rcParams['font.family'] = 'sans-serif'
plt.rcParams['font.sans-serif'] = 'Arial'
plt.rcParams['font.weight'] = 'bold'
plt.rcParams['axes.labelsize'] = 'medium'
plt.rcParams['axes.labelweight'] = 'bold'
plt.rcParams['axes.linewidth'] = 1.2
plt.rcParams['lines.linewidth'] = 2.0
```

Die im Argument der Winkelfunktionen auftretenden Ausdrücke der Form $\Delta_m^2 L/E$ haben Einheiten von eV m. Wir müssen also im Argument durch $\hbar c = 1$ dividieren.

```
(const.hbar*const.c).to('eV m')
```

1.9732698×10^{-7} eV m

Definition der vorgegebenen Werte:

```
values = {'Delta_m__2': 0.0025*u.eV**2, 'E': 1.0*u.GeV, 'theta': 45.*u.deg}
```

```
{'Delta_m__2': <Quantity 0.0025 eV2>,
'E': <Quantity 1. GeV>,
'theta': <Quantity 45. deg>}
```

Wir können testweise einen Wert für die Oszillationswahrscheinlichkeit ausrechnen. Die einzige nach Einsetzen der Konstanten verbleibende freie Variable ist L . Wir dividieren im Argument durch $\hbar c$ und fügen manuell noch explizit die Einheit rad hinzu, weil `astropy` sich sonst weigert, die Winkelfunktion auszurechnen.

```
p_tau_func(**values, L=500.*u.km/const.hbar/const.c*u.rad)
```

0.99983438

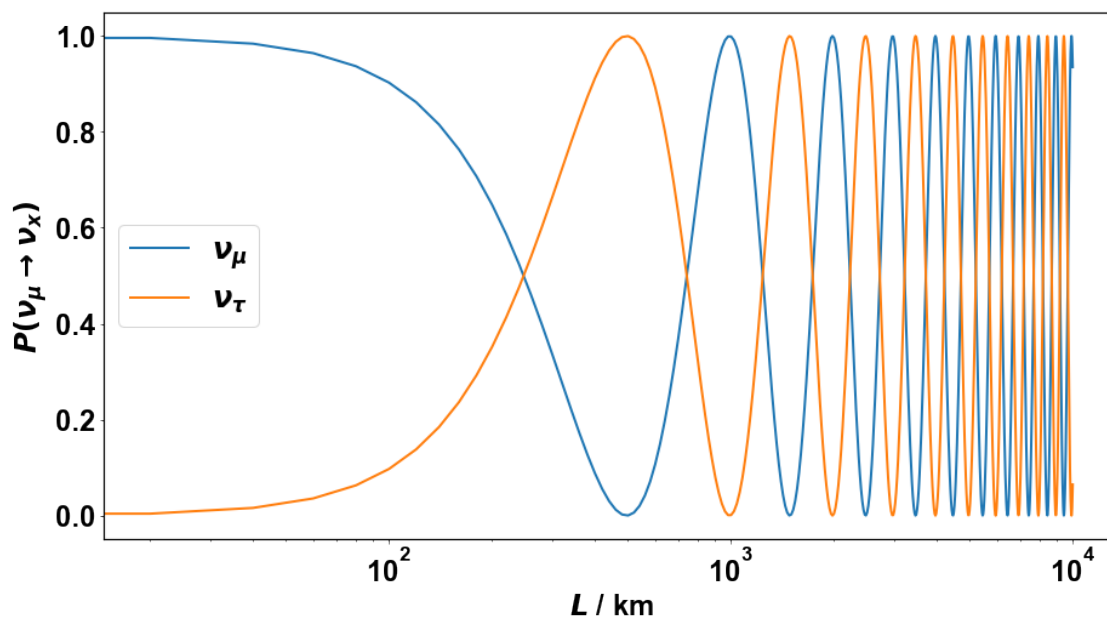
Offenbar haben wir einen Wert nahe der halben Oszillationslänge gewählt:

```
Losz = (4.*np.pi*values['E']/values['Delta_m__2']*(const.hbar*const.c)).to('km')
```

991.87359 km

Jetzt können wir die beiden Oszillationswahrscheinlichkeiten als Funktion von L plotten.

```
plt.figure(figsize=(15,8))
Lx = np.linspace(0., 10000., 500)
plt.plot(Lx, p_mu_func(**values, L=Lx*u.km/const.hbar/const.c*u.rad),  
         ↪label=r'$\nu_\mu$')
plt.plot(Lx, p_tau_func(**values, L=Lx*u.km/const.hbar/const.c*u.rad),  
         ↪label=r'$\nu_\tau$')
plt.xlabel(r'$L$ / km')
plt.ylabel(r'$P(\nu_\mu \rightarrow \nu_x)$')
plt.legend()
plt.xscale('log')
```



(1 Punkt für im Prinzip richtigen Plot)

(1 Punkt für logarithmische L -Achse, sinnvolle Achsenbeschriftungen und Legende)