Prepared for submission to JHEP

Neutrino Oscillation with Python

Ka Young Ban

Yonsei University,
South Korea, Seoul

E-mail: ban94gy@naver.com

Abstract: Neutrino oscillation is a quantum mechanical phenomenon whereby a neutrino created with a specific lepton flavor (electron, muon, or tau) can later be measured to have a different flavor. The probability of measuring a particular flavor for a neutrino varies periodically as it propagates through space. We will confirm how three neutrino probabilities, $\nu_\alpha \rightarrow \nu_\beta$ such that α *and* β is $e,$ μ *or* τ, were drawn and compare with Wikipedia reference. To draw graphs, we will use Python code, but there are not special code technique. And introduce some of the history, the advantages and disadvantages of Python.

Contents

1 Introduce of Python

1.1 History of Python

The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC (programming language) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). Python was named for the BBC TV show Monty Python's Flying Circus.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible Python 2.6 and 2.7.

1.2 Advantages of Python

Python is easy to integrate with C, C ++ Fortran code. This means it is easy to load and use written code without having to create a new one.

Pierre Carbonell, a Python developer and blogger who runs the PyPL language index, said "The main characteristic of Python programs is easy to read," That means Python's stability helps open source development.

1.3 Disadvantages of Python

Python is an interpreter language that is slower than compiled languages like JAVA and C ++. Therefore, if you need a very short response time, like a real-time trading system, it is better to develop in a compiled language such as C ++.

Python is not a good language for processing with simultaneous multithreading or CPU-intensive threads. This is due to the Global Interpreter Lock (GIL), that is the interpreter excutes only one byte code at a time.

2 Neutrino Oscillation

Neutrino oscillation arises from a mixture between the flavor and mass eigenstates of neutrinos. That is, the three neutrino states that interact with the charged leptons in weak interactions are each a different superposition of the three neutrino states of definite mass. Neutrinos are created in weak processes in their flavor eigenstates. As a neutrino propagates through space, the quantum mechanical phases of the three mass states advance at slightly different rates due to the slight differences in the neutrino masses. This results in a changing mixture of mass states as the neutrino travels, but a different mixture of mass states corresponds to a different mixture of flavor states. So a neutrino born as, say, an electron neutrino will be some mixture of electron, mu, and tau neutrino after traveling some distance.

Since the quantum mechanical phase advances in a periodic fashion, after some distance the state will nearly return to the original mixture, and the neutrino will be again mostly electron neutrino. The electron flavor content of the neutrino will then continue to oscillate as long as the quantum mechanical state maintains coherence. Since mass differences between neutrino flavors are small in comparison with long coherence length for neutrino oscillations this microscopic quantum effect becomes observable over macroscopic distances.

Neutrino oscillation is a function of the ratio L/E, where L is the distance traveled and E is the neutrino's energy. Neutrino sources and detectors are far too large to move, but all available sources produce a range of energies, and oscillation can be measured with a fixed distance and neutrinos of varying energy. The preferred distance depends on the most common energy, but the exact distance is not critical as long as it is known. The limiting factor in measurements is the accuracy with which the energy of each observed neutrino can be measured. Because current detectors have energy uncertainties of a few percent, it is satisfactory to know the distance to within 1%.

2.1 Propagation and Interference

In its simplest form it is expressed as a unitary transformation relating the flavor and mass eigenbasis and can be written as

$$|\nu_\alpha\rangle = \sum_i U^*_{\alpha i} |\nu_i\rangle$$
$$|\nu_i\rangle = \sum_\alpha U_{\alpha i} |\nu_\alpha\rangle$$

where

$* |\nu_\alpha\rangle$ is a neutrino with definite flavor $\alpha = e(electron),\ \mu(muon)\ or\ \tau(tauon)$

$* |\nu_i\rangle$ is a neutrino with definite mass $m_i,\ i = 1, 2, 3$

$* U_{\alpha i}$ represents the 'Pontecorvo–Maki–Nakagawa–Sakata matrix' (also called the 'PMNS matrix', 'lepton mixing matrix', or sometimes simply the 'MNS matrix').

Since $|\nu_i\rangle$ are mass eigenstates, their propagation can be described by plane wave solutions of the form

$$|\nu_i(t)\rangle = e^{-i(E_i t - \vec{p_i}\cdot\vec{x})}|\nu_i(0)\rangle$$

where

* quantities are expressed in natural units (c = 1, $\hbar = 1$)

* E_i is the energy of the mass-eigenstate i

* t is the time from the start of the propagation

* $\vec{p_i}$ is the three-dimensional momentum

* $\vec{x}$ is the current position of the particle relative to its starting position

In the ultrarelativistic limit, $|\vec{p_i}| = p_i \gg m_i$, we can approximate the energy as

$$E_i = \sqrt{p_i^2 + m_i^2}\ \simeq\ p_i + \frac{m_i^2}{2p_i}\ \approx\ E + \frac{m_i^2}{2E}$$

where E is the total energy of the particle.

This limit applies to all practical (currently observed) neutrinos, since their masses are less than 1 eV and their energies are at least 1 MeV, so the Lorentz factor γ is greater than 10^6 in all cases. Using also t, where L is the distance traveled and also dropping the phase factors, the wavefunction becomes:

$$|\nu_i(L)\rangle = e^{-im_i^2 L/2E}|\nu_i(0)\rangle.$$

Eigenstates with different masses propagate with different frequencies. The heavier ones oscillate faster compared to the lighter ones. Since the mass eigenstates are combinations of flavor eigenstates, this difference in frequencies causes interference between the corresponding flavor components of each mass eigenstate. Constructive interference causes it to be possible to observe a neutrino created with a given flavor to change its flavor during its propagation. The probability that a neutrino originally of flavor α will later be observed as having flavor β is

$$P_{\alpha\to\beta} = |\langle\nu_\beta(t)|\nu_\alpha\rangle|^2 = \left|\sum_i U_{\alpha i}^* U_{\beta i} e^{-im_i^2 L/2E}\right|^2$$

2.2 Two Neutrino Case

Writing it explicitly in terms of mixing angles is extremely cumbersome if there are more than two neutrinos that participate in mixing. Fortunately, there are several cases in which only two neutrinos participate significantly. In this case, it is sufficient to consider the mixing matrix

$$U = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

Then the probability of a neutrino changing its flavor is,

$$P_{\alpha\to\beta,\alpha\neq\beta} = \sin^2(2\theta)\sin^2\left(\frac{\Delta m^2 L}{4E}\right) \text{ (natural units).}$$

Or, using SI units and the convention introduced above,

$$P_{\alpha\to\beta,\alpha\neq\beta} = \sin^2(2\theta)\sin^2\left(1.27\frac{\Delta m^2 L}{E}\frac{[\mathrm{eV}^2]\,[\mathrm{km}]}{[\mathrm{GeV}]}\right).$$

This formula is often appropriate for discussing the transition $\nu_\mu \leftrightarrow \nu_\tau$ in atmospheric mixing, since the electron neutrino plays almost no role in this case. It is also appropriate for the solar case of $\nu_e \leftrightarrow \nu_x$, where ν_x is a superposition of ν_μ and ν_τ. These approximations are possible because the mixing angle θ_{13} is very small and because two of the mass states are very close in mass compared to the third.

In the approximation where only two neutrinos participate in the oscillation, the probability of oscillation follows a simple pattern, The blue curve shows the probability of the

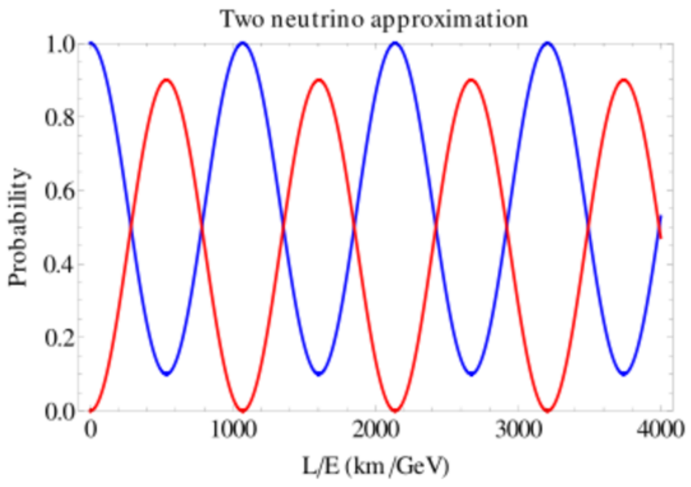


Figure 1: Two neutrino probabilities in vacuum

original neutrino retaining its identity. The red curve shows the probability of conversion to the other neutrino. The maximum probability of conversion is equal to $sin2\theta$. The frequency of the oscillation is controlled by $\triangle m^2$

2.3 Three Neutrino Case

Three neutrino case is the main content for python code. When the standard three-neutrino theory is considered, the matrix is 3×3. If one or more sterile neutrinos are added (see later), it is 4×4 or larger. In the 3×3 form, it is given by

$$\begin{aligned}
U &= \begin{bmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{bmatrix} \begin{bmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta} & 0 & c_{13} \end{bmatrix} \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e^{i\alpha_1/2} & 0 & 0 \\ 0 & e^{i\alpha_2/2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{bmatrix} \begin{bmatrix} e^{i\alpha_1/2} & 0 & 0 \\ 0 & e^{i\alpha_2/2} & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

where $c_{ij} = cos\theta_{ij}$, and $s_{ij} = sin\theta_{ij}$.
Above equation in '2.1 Propagation and interference',

$$P_{\alpha\to\beta} = |\langle\nu_\beta(t)|\nu_\alpha\rangle|^2 = \left|\sum_i U^*_{\alpha i}U_{\beta i}e^{-im_i^2 L/2E}\right|^2$$

we use these two equations to plot three neutrino oscillation probability graph.

3 Python Code

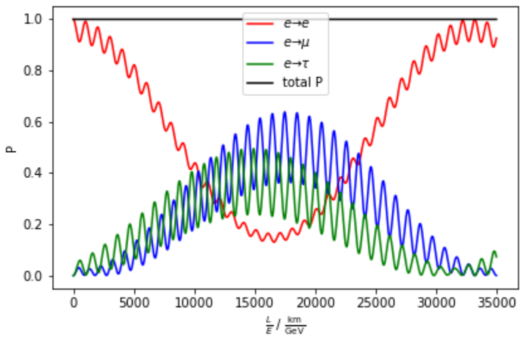
This is python code for calculating and plotting probabilities

```python
import numpy as np
import matplotlib.pyplot as plt

# JHEP 11 (2014) 052 [arXiv:1409.5439]
theta_12 = np.deg2rad(33.48)
theta_23 = np.deg2rad(42.3)
theta_13 = np.deg2rad(8.50)

d = np.deg2rad(306)

d_m_12 = np.sqrt(7.50e-5) # eV^2
d_m_13 = np.sqrt(2.457e-3) # eV^2

masses = [0, d_m_12**2, d_m_13**2]

s12, c12 = np.sin(theta_12), np.cos(theta_12)
s23, c23 = np.sin(theta_23), np.cos(theta_23)
s13, c13 = np.sin(theta_13), np.cos(theta_13)

U = np.matrix([[c12*c13, s12*c13, s13*np.exp(-1j*d)],
[-s12*c23 - c12*s23*s13*np.exp(1j*d), c12*c23 - s12*s23*s13*np.exp(1j*d), s23
    *c13],
 [s12*s23 - c12*c23*s13*np.exp(1j*d), -c12*s23 - s12*c23*s13*np.exp(1j*d), c23
    *c13]])


```

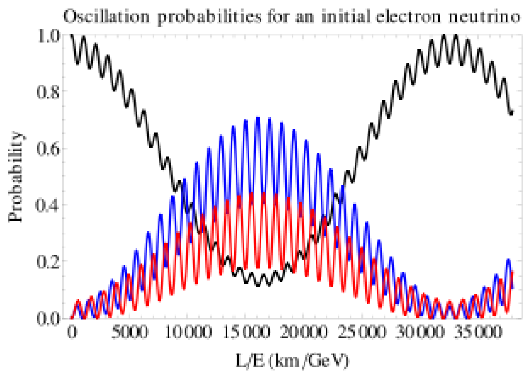
```python
def P(a, b, L, E):
  result = 0
  for i in range(3):
  # 1.27 accounts for factors of c and h_bar
  result = result + U[a-1, i].conjugate() * U[b-1, i] * np.exp(-1j * 1.27 * masses[i] * 2 * L / E)
return abs(result)**2

plt.figure(1)
x = np.linspace(0, 35000, 2000)
E = 1
plt.plot(x, P(1, 1, x, E), label='$e \\to e$', color='red')
plt.plot(x, P(1, 2, x, E), label='$e \\to\\mu$', color='blue')
plt.plot(x, P(1, 3, x, E), label='$e \\to\\tau$', color='green')
plt.plot(x, P(1, 1, x, E) + P(1, 2, x, E) + P(1, 3, x, E), label='total P', color='black')
plt.ylabel('P')
plt.xlabel('$\\frac{L}{E}\ /\ \\frac{\\mathrm{km}}{\\mathrm{GeV}}$')
plt.legend(loc='upper center')
plt.tight_layout()
plt.savefig('e.png')

plt.figure(2)
x = np.linspace(0, 35000, 2000)
E = 1
plt.plot(x, P(2, 1, x, E), label='$\\mu\\to e$', color='red')
plt.plot(x, P(2, 2, x, E), label='$\\mu\\to\\mu$', color='blue')
plt.plot(x, P(2, 3, x, E), label='$\\mu\\to\\tau$', color='green')
plt.plot(x, P(2, 1, x, E) + P(2, 2, x, E) + P(2, 3, x, E), label='total P', color='black')
plt.ylabel('$P_{\\mu\\to\\ell}$')
plt.xlabel('$\\frac{L}{E}\ /\ \\frac{\\mathrm{km}}{\\mathrm{GeV}}$')
plt.legend(loc='upper center')
plt.tight_layout()
plt.savefig('mu.png')

plt.figure(3)
x = np.linspace(0, 35000, 2000)
E = 1
plt.plot(x, P(3, 1, x, E), label='$\\tau \\to e$', color='red')
plt.plot(x, P(3, 2, x, E), label='$\\tau \\to\\mu$', color='blue')
plt.plot(x, P(3, 3, x, E), label='$\\tau \\to\\tau$', color='green')
plt.plot(x, P(3, 1, x, E) + P(3, 2, x, E) + P(3, 3, x, E), label='total P', color='black')
plt.ylabel('$P_{\\tau\\to\\ell}$')
plt.xlabel('$\\frac{L}{E}\ /\ \\frac{\\mathrm{km}}{\\mathrm{GeV}}$')
plt.legend(loc='upper center')
plt.tight_layout()
plt.savefig('tau.png')
```

Listing 1: Python Code

4 Result

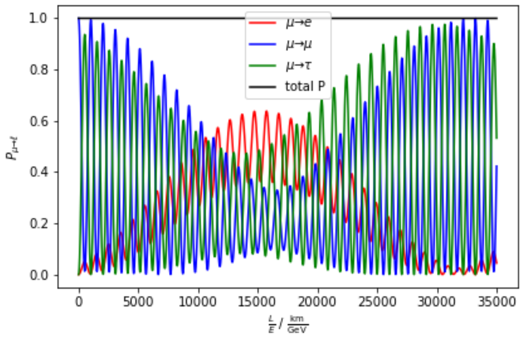


(a) Python plotting

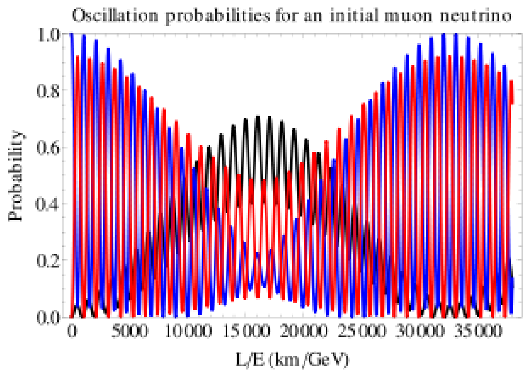


(b) In Wikipedia

Figure 2: Electron neutrino oscillations

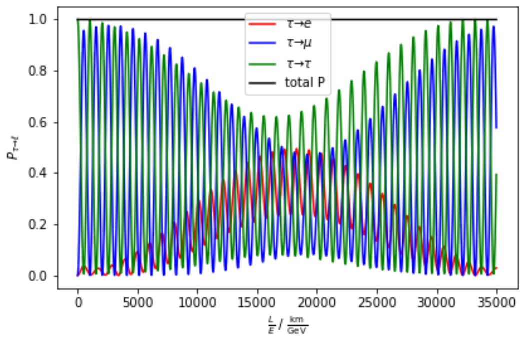


(a) Python plotting

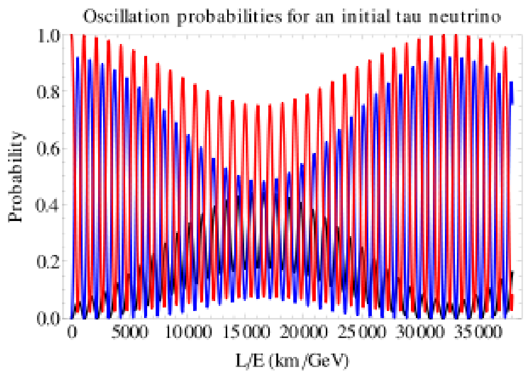


(b) In Wikipedia

Figure 3: Muon neutrino oscillations



(a) Python plotting



(b) In Wikipedia

Figure 4: Tau neutrino oscillations

Left side, Python plotting, black means total probability(= 1), red means electron neutrino, blue means muon neutrino and green means tau neutrino. And right side, In Wikipedia, black means electron neutrino, blue means muon neutrino and red means tau neutrino.

In Wiki, these values were used,
* $\sin^2 2\theta_{12} = 0.861$
* $\sin^2 2\theta_{23} = 0.97$
* $\sin^2 2\theta_{13} = 0.10$ (Controls the size of the small wiggles.)
* $\delta = 0$
* $\triangle m_{12}^2 = 7.59 \cdot 10^{-5} eV^2$
* $\triangle m_{32}^2 \approx \triangle m_{32}^2 = 2.32 \cdot 10^{-3} eV^2$
* Normal mass hierarchy.

But I use the data in 'JHEP 11 (2014) 052 [arXiv:1409.5439], page.8'
* $\theta_{12} = np.deg2rad(33.48) = 33.48 * \pi/180 \approx 0.584$
* $\theta_{23} = np.deg2rad(42.3) = 42.3 * \pi/180 \approx 0.738$
* $\theta_{13} = np.deg2rad(8.50) = 8.50 * \pi/180 \approx 0.148$
* $\delta = np.deg2rad(306) = 306 * \pi/180 \approx 5.341$
* $\triangle m_{12} = \sqrt{7.50e - 5} eV^2$
* $\triangle m_{13} = \sqrt{2.457e - 3} eV^2$

(np.deg2rad(x) = x * pi / 180 in python)

Because I used a little different data with wikipedia, the wiki and python graphs are different.

References

[1] *https://en.wikipedia.org/wiki/Neutrino_oscillation*

[2] *https://en.wikipedia.org/wiki/History_of_Python*

[3] M. C. Gonzalez-Garcia and Michele Maltoni and Thomas Schwetzd, *Updated fit to three neutrino mixing: status of leptonic CP violation* (2014), [arXiv: 1409.5439v2]

[4] *https://ko.sharelatex.com/learn/Code_listing*