Quantum Computing for Nuclear Physics Final Project
– Neutrino flavor oscillations –

Nuclear Talent course
European Center for Theoretical Nuclear Physics and Related Areas, Trento, Italy
(Dated: June 16-July 4, 2025)

The aim of this project is to study a simple model for collective neutrino oscillations as a pretext to put to the test some of the things you learned during the school like the Trotter approximation and the implementation of quantum circuits. If you also happen to be interested in the physics topic on itself, you can have a look for instance at the recent review 2305.01150. If you happen to believe that two-flavor neutrino oscillations are just child's play and a mere excuse to use qubits, you might want to have a look at 2503.00607 (and of course you could do the whole project like that).

PROJECT DETAILS

As you probably all know neutrinos show flavor oscillations due to a mismatch between flavor states and energy eigenstates. If we consider the simple case where only two flavors are present, let's call them e and x, we can encode the two possible states of a neutrino into the states of a single qubit

$$|\uparrow\rangle = |0\rangle \equiv |\nu_e\rangle \qquad |\downarrow\rangle = |1\rangle \equiv |\nu_x\rangle \ . \tag{1}$$

In this flavor basis, the weak Hamiltonian in vacum can be written in terms of Pauli operators like

$$H_{vac} = \frac{\Delta m^2}{4E}\left(\sin(2\theta)X - \cos(2\theta)Z\right) \ , \tag{2}$$

where $\Delta m^2 = m_1^2 - m_2^2$ is the difference in masses squared, E is the neutrino energy, θ the mixing angle and X and Z are the respective Pauli matrices. A useful compact notation for this Hamiltonian is

$$H_{vac} = \frac{\Delta m^2}{4E}\vec{B}\cdot\vec{\sigma} \ , \tag{3}$$

where

$$\vec{B} = \left(\sin(2\theta), 0, -\cos(2\theta)\right) \ , \tag{4}$$

and $\vec{\sigma} = (X, Y, Z)$ is a vector of Pauli matrices.

In extreme astrophysical environments like core-collapse supernovae or binary neutron star mergers, the neutrino flux is so large that neutrino-neutrino scattering becomes important. A simple model (which considers only forward scattering) for this leads to the following interaction Hamiltonian acting on a pair of neutrinos (labeled here 1 and 2)

$$H_{\nu\nu} = \frac{G_F}{\sqrt{2}V}\left(1 - \cos(\theta_{12})\right)\vec{\sigma}_1\cdot\vec{\sigma}_2 \ , \tag{5}$$

where $\vec{\sigma}_k$ is the vector of Pauli matrices acting on the k-th neutrino, G_F is Fermi's constant, V is the volume of the system and $\theta_1 2$ is the relative angle between the neutrino momenta

$$\cos(\theta_{12}) = \frac{\vec{p}_1\cdot\vec{p}_2}{\|\vec{p}_1\|\|\vec{p}_2\|} \ . \tag{6}$$

If we measure energies in units of $\mu = \sqrt{2}G_F\rho$, with $\rho = N/V$ the neutrino number density, and consider a system made of N neutrinos with the same energy but different momentum direction we can write the full Hamiltonian as

$$H = \lambda\sum_{i=1}^{N}\vec{B}\cdot\vec{\sigma} + \frac{1}{2N}\sum_{i<j}\left(1 - \cos(\theta_{ij})\right)\vec{\sigma}_i\cdot\vec{\sigma}_j \ . \tag{7}$$

For this project you can take $\lambda = 1$ and angles θ_{ij} chosen as (see 2207.03189 if want to know why)

$$\theta_{ij} = \arccos(0.9)\frac{|i-j|}{N-1} \ . \tag{8}$$

You can then try to explore what happens to the neutrino flavor when you start in a given flavor state like

$$|\nu_e\rangle|\nu_x\rangle \quad \text{or} \quad |\nu_e\rangle|\nu_e\rangle|\nu_x\rangle|\nu_x\rangle \ , \tag{9}$$

as they evolve in time.

For the project, start with a single pair $N = 2$ and

1. write explicitly the Hamiltonian and find its energy spectrum (can you think of a quick way to do it analytically?).

2. study the evolution of flavor (the expectation value of Z) for all different initial neutrino flavor assignments of the pair (like $|\nu_e\rangle|\nu_x\rangle$ or $|\nu_e\rangle|\nu_e\rangle$). You can get the exact evolution numerically using

$$e^{-itH} = \sum_k |\phi_k\rangle e^{-ite_k}\langle\phi_k| \ , \tag{10}$$

where e_k and $|\phi_k\rangle$ are the eigenvalues and eigenvectors of H, respectively.

3. implement the time evolution operator using a quantum circuit and compare your results with the exact calculation done above. You could use Trotter for this (and as a simple exercise you should try it) but there is a better way that only uses 3 CNOT gates. If this seems mysterious to you, check out https://arxiv.org/abs/quant-ph/0308006

4. try to look at how entanglement evolves in time for the individual neutrinos. Remember that for a single qubit the reduced density matrix can be written explicitly as follows

$$\rho = \frac{1}{2}\left(\mathbb{1} + \vec{s}\cdot\vec{\sigma}\right) \ , \tag{11}$$

where

$$\vec{s} = \left(\langle X\rangle, \langle Y\rangle, \langle Z\rangle\right) \ , \tag{12}$$

is a 3-component vector containing the expectation values of the Pauli matrices. To quantify entanglement you can use the **purity**

$$P(\rho) = Tr\left[\rho^2\right] \ , \tag{13}$$

or the entanglement entropy

$$S(\rho) = -Tr\left[\rho\log_2(\rho)\right] \ . \tag{14}$$

Before starting the calculation, try to understand what values you expect for both of these if you calculate them with a product state like $|0\rangle|0\rangle$ or a very entangled state like a Bell pair.

Once you are done with the $N = 2$ system you should try to do $N = 4$ with the initial state $|\nu_e\rangle|\nu_e\rangle|\nu_x\rangle|\nu_x\rangle$. In this case the time evolution operator will need to be approximated using Trotter when you implement it as a quantum circuit.

- If you feel comfortable with quantum circuits you can try to devise a strategy to perform the calculation when qubits are laid out on a line like

$$(0) - (1) - (2) - (3) \tag{15}$$

so that you can do a CNOT between qubit 0 and qubit 1 but not between qubit 0 and qubit 2. The problem you'll face is similar to the one appearing when you use the Jordan-Wigner mapping for fermions (and the solution might have something to do with how you work around that problem).

  - if your solution requires to do things like using SWAP gates to move qubits close and then move them back, you might be interested in looking at 2102.12556 or 2207.03189. And yes you can do that for fermions encoded with Jordan-Wigner too! See 1711.04789

Once you performed all the simulations with $N = 2$ and $N = 4$ with the *statevector* simulator for quantum circuits (with qiskit, cirq or your own code) you should try to do the calculations a bit more realistically and perform the measurements with a finite number of shots. Make sure to include a corresponding error-bar that estimates the statistical error when you present results.

If all of this was too easy and you want a bit more of a challenge, try to perform everything with a noise simulator (like the one you can find in qiskit) and try to mitigate the resulting errors. If you go down this path you might find it useful to have a look at 2102.12556 (in particular Appendix B) or, for some more modern error mitigation strategies, to 2103.08591 and 2205.09247