Approach of eigenvalues in medium

G. Dalton Bentley

http://physics.stackexchange.com/users/111801/dalton-bentley

(Dated: September 26, 2020)

Abstract

This note accompanies the `Python 3` code file Eigenvalue trajectory available at the author's GitHub repository. We briefly discuss here the relationship between solar electron production energies and electron number densities and the corresponding values of the neutrino mass eigenvalues of the effective Hamiltonian in medium. This relationship affects the possibility of a jump between mass eigenstates ν_{1m} and ν_{2m} when their energy difference is at minimum at MSW resonance.

I. INTRODUCTION

In treatments of solar neutrino physics you often encounter passing reference to "would-be level crossing" [1] or statements like

> The second possibility [nonadiabatic] is realized if in the resonance region, where the two levels [$|\nu_1^m(t)\rangle$ and $|\nu_2^m(t)\rangle$] approach each other most, the system jumps from the upper level to the lower level and after that continues to be in the state $|\nu_1^m(t)\rangle$ until the neutrino reaches the surface of the Sun [2].

The `Python` code we provide at Github graphs the conceptual trajectory of those neutrino mass state levels in solar plasma:

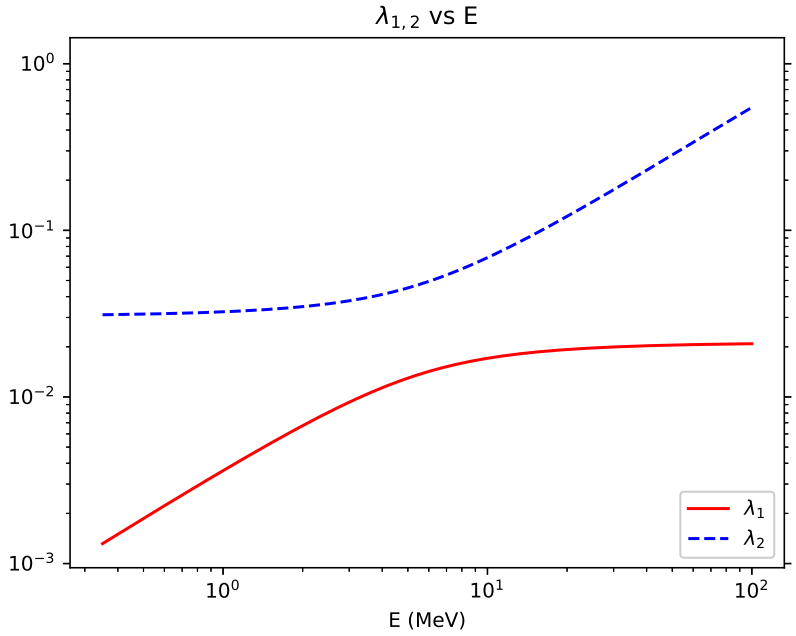

FIG. 1. The figure depicts the values of two eigenvalues, λ_1 (red line, associated with ν_1) and λ_2 (blue dashed line, associated with ν_2), of the neutrino effective Hamiltonian in the mass basis for electron number density $N_e = 5.4670 \times 10^{25}$ cm^{-3} over energy range 350 keV to 100 MeV. The y-axis is given in abstract units related to actual mass as discussed in Section II D.

We will briefly discuss here the physics relevant to the graph and computer code.

II. EFFECTIVE HAMILTONIAN AND RELATED

The effective Hamiltonian governing the evolution of neutrino flavor eigenstates in matter can be written as [3]:

$$\tilde{H}_f = \frac{1}{2E}\left[U\begin{pmatrix} m_1^2 & 0 & 0 \\ 0 & m_2^2 & 0 \\ 0 & 0 & m_3^2 \end{pmatrix}U^\dagger + \begin{pmatrix} A & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right] \tag{1}$$

U is the PMNS (Pontecorvo-Maki-Nakagawa-Sakata) matrix, which specifies the mix of flavor[1] eigenstates ν_α, $\alpha \in \{e, \mu, \tau\}$, in terms of mass eigenstates, ν_i, $i \in \{1, 2, 3\}$, as $\nu_\alpha = \sum_i U_{\alpha i}\nu_i$ where U is a 3×3 unitary matrix [3] [4]. The m_i, $i = 1, 2, 3$, denote neutrino masses (associated with the ν_i) and $A = 2\sqrt{2}G_F N_e E$ is the matter potential, i.e., the charged-current contribution to $\nu_e e^-$ forward scattering, with N_e the electron number density in medium and E the neutrino production energy.

A. U mixing matrix

Explicitly, U can be written as:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix}\begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \tag{2}$$

The elements of U can be constructed by the product of rotation matrices in neutrino flavor/mass space using Euler angles and rotation matrices as would be done for three-dimensional rotations in Euclidian space (see [5] [6]). In the following, c_{13} denotes $\cos\theta_{13}$, s_{12} is $\sin\theta_{12}$ and so on. U is then the product of three Euler rotations $U = R_{23}R_{13}R_{12}$:

$$U_{PMNS} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix}\begin{pmatrix} c_{13} & 0 & s_{13} \\ 0 & 1 & 0 \\ -s_{13} & 0 & c_{13} \end{pmatrix}\begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

[1] The flavor is the type of lepton, e, μ, τ, associated with the neutrino in a weak charged-current interaction.

We have omitted a Dirac CP violation phase in the above as of no interest here. Carrying out the product of the three rotation matrices we obtain:

$$U = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13} \\ -s_{12}c_{23} - c_{12}s_{13}s_{23} & c_{12}c_{23} - s_{12}s_{13}s_{23} & c_{13}s_{23} \\ s_{12}s_{23} - c_{12}s_{13}c_{23} & -c_{12}s_{23} - s_{12}s_{13}c_{23} & c_{13}c_{23} \end{pmatrix} \tag{4}$$

We note in passing that a CAS (computer algebra system) like `SymPy` makes this kind of array calculation relatively painless. We generated the above (in a `Jupyter` notebook environment `SymPy` returns and displays typeset LaTex mathematics) with a few lines of `Python` (after importing `SymPy`:

```
sector_23 =  Matrix( [ [1, 0, 0], [0, c23, s23], [0, -s23, c23] ])
sector_13 =  Matrix( [ [c13, 0, s13], [0, 1, 0], [-s13, 0, c13] ])
sector_12 =  Matrix( [ [c12, s12, 0], [-s12, c12, 0], [0, 0, 1] ])
U = sector_23 * sector_13 * sector_12
```

The three angles θ_{12}, θ_{23} and θ_{13} in the rotation matrices can be defined via the elements of the neutrino mixing matrix Eq. 2 [2]:

$$c_{12}^2 \equiv \cos^2\theta_{12} \equiv \frac{|U_{e1}|^2}{1-|U_{e3}|^2}, \quad s_{12}^2 \equiv \sin^2\theta_{12} \equiv \frac{|U_{e2}|^2}{1-|U_{e3}|^2} \tag{5}$$

$$s_{13}^2 \equiv \sin^2\theta_{13} \equiv |U_{e3}|^2, \quad s_{23}^2 \equiv \sin^2\theta_{23} \equiv \frac{|U_{\mu 3}|^2}{1-|U_{e3}|^2} \tag{6}$$

$$c_{23}^2 \equiv \cos^2\theta_{23} \equiv \frac{|U_{\tau 3}|^2}{1-|U_{e3}|^2} \tag{7}$$

Reading off the relations in the top row of Eq. 2, we see the electron flavor neutrino can be written as a mixture of mass states:

$$|\nu_e\rangle = U_{e1}|\nu_1\rangle + U_{e2}|\nu_2\rangle + U_{e3}|\nu_3\rangle \tag{8}$$

With that in mind we see θ_{12} in Eq. 5 gives us the rotation of the mass basis into the flavor basis, in this case, the mix of ν_1 and ν_2 in ν_e. How are these angles θ_{12}, θ_{13} and θ_{23} obtained?

B. Origin of mixing angles

Putting it simply[2], the number of neutrinos detected is compared with the number expected (from a SSM or Standard Solar Model in the case of solar electron neutrinos, from the known output of a nuclear reactor or neutrino beam generated at a particle accelerator for terrestrial sources). The difference, if any, is compared to the number of events predicted by neutrino flavor-dependent interaction equations (most sources refer to these as oscillation equations, but to be accurate, solar neutrino detection has nothing to do with oscillation or periodic change in flavor expectation [8]).

In the case that a single mixing angle dominates the oscillation behavior in a particular experiment (for example, θ_{12} in the case of solar neutrinos), the value of the angle can be varied (in the equations) and the statistical fit of the prediction compared with observation. The equations depend also on the dominant mass squared split Δm^2_{ij} for the experiment. In the case of solar neutrinos there is additionally a dependence on the MSW effect.. The result is a range of values that can be excluded or included, both within the particular experiment and in comparison to other experiments involving the same parameters. See another of our articles, Solar Neutrino Survival Probability Scan, for discussion of expected detection rates for solar neutrinos, along with computer code to graph the calculated expectations (and cites to relevant papers in the literature).

C. Obtain eigenvalues

To obtain the eigenvalues of the effective Hamiltonian Eq. 1, one path to follow is to transform $\tilde{H}_f$ from the flavor eigenbasis to the mass eigenbasis with $\tilde{H}_\nu = U^\dagger \tilde{H}_f U$ to obtain:

$$\tilde{H}_\nu = \begin{pmatrix} A|U_{e1}|^2 + m_1^2 & AU_{e2}U_{e1}^* & AU_{e3}U_{e1}^* \\ AU_{e1}U_{e2}^* & A|U_{e2}|^2 + m_2^2 & AU_{e3}U_{e2}^* \\ AU_{e1}U_{e3}^* & AU_{e2}U_{e3}^* & A|U_{e3}|^2 + m_3^2 \end{pmatrix} \tag{9}$$

Then the resulting Hermitian matrix could be diagonalized through a unitary transformation $V^\dagger \tilde{H}_\nu V = \mathrm{Diag}\{\lambda_1, \lambda_2, \lambda_3\}$, where λ_i denote the eigenvalues of the effective Hamiltonian

[2] For an excellent discussion of how neutrino parameters are determined from experiment in the context of the NOνA research at Fermilab, see the doctoral dissertation of E. Catano-Mur [7].

and the transform matrix V is to be determined (the columns of this matrix would be the eigenvectors of $\tilde{H}_\nu$ in the vacuum mass eigenbasis [4].

Obtaining the analytic eigenvalues of the effective Hamiltonian is not a trivial exercise, but they have been worked out [4] (that source cites the original work in this area and the subsequent modifications). Though we generally abhor introducing new symbols into neutrino physics (without breakthrough updates in the underlying physics), we will follow [3] introducing a generic definition of neutrino mass-squared difference:

$$\Delta_* \equiv \eta\Delta_{31} + (1-\eta)\Delta_{32} \tag{10}$$

where Δ_{31} and Δ_{32} are the familiar squared mass splittings Δm^2_{13} and Δm^2_{23} (we will deal with only the normal ordering, so may write the indices in different order depending on the reference document, but we intend always $m_1 < m_2 < m_3$). η is a gauge parameter introduced to allow optimization for a particular context. We will use the so-called "normal" scheme $\eta = 1$ rather than $\eta = \cos^2\theta_{12}$ or variations $0 \leq \eta \leq 1$. With $\eta = 1$, Δ_* is simply Δ_{31}, referring to Eq. 10. Before we implement that substitution (and a few more), let us decompose the effective Hamiltonian Eq. 1 as follows:

$$\tilde{H}_f = \frac{1}{2E}\begin{pmatrix} m_1^2 & 0 & 0 \\ 0 & m_1^2 & 0 \\ 0 & 0 & m_1^2 \end{pmatrix} + \frac{\Delta m^2_{31}}{2E}\left[U \begin{pmatrix} 0 & 0 & 0 \\ 0 & \Delta m^2_{21}/\Delta m^2_{31} & 0 \\ 0 & 0 & 1 \end{pmatrix} U^\dagger + \begin{pmatrix} A/\Delta m^2_{31} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \end{pmatrix}\right] \tag{11}$$

We define additional new symbols $\hat{A} = A/\Delta_*$ (recall $A = 2\sqrt{2}G_F N_e E$ is the matter potential) and $\alpha = \Delta_{21}/\Delta_*$. We chose our scheme as $\eta = 1$, so, as we noted above, $\Delta_* \equiv \eta\Delta_{31} + (1-\eta)\Delta_{32}$ from Eq. 10 will simplify to $\Delta_* = \Delta_{31}$ (or Δm^2_{13}). The symbol Δ_{12} is the usual squared mass splitting Δm^2_{12}. With our choice of η, $\alpha = \Delta_{21}/\Delta_*$ is really $\Delta m^2_{12}/\Delta m^2_{13}$ and $\hat{A} = A/\Delta_*$ represents $A/\Delta m^2_{13}$. We belabor the meaning of these symbols in an effort to make clear the multiple layers of indirection. We now claim that Eq. 11 (which originated from Eq. 1)), the effective Hamiltonian, can be rewritten as:

$$\tilde{H}_f = \frac{m_2^2 - \eta\Delta_{21}}{2E}\mathbb{I} + \frac{\Delta_*}{2E}M_f \ , \tag{12}$$

where $\mathbb{I}$ is the rank three identity matrix Diag$(1,1,1)$ and M_f is given by:

$$M_f = U \begin{pmatrix} (\eta-1)\,\alpha & 0 & 0 \\ 0 & \eta\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} U^\dagger + \begin{pmatrix} \hat{A} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{13}$$

The first term on the rhs of Eq. 12 , being a diagonal matrix with identical terms, can be put aside here as irrelevant to oscillatory flavor change. Continuing with [3], they find it more convenient to work with the mass basis, $M_\nu = U^\dagger M_f U$, where the first term on the rhs of Eq. 13 becomes diagonal:

$$M_\nu = \begin{pmatrix} (\eta-1)\,\alpha & 0 & 0 \\ 0 & \eta\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} + \hat{A} \begin{pmatrix} |U_{e1}|^2 & U_{e1}^* U_{e2} & U_{e1}^* U_{e3} \\ U_{e2}^* U_{e1} & |U_{e2}|^2 & U_{e2}^* U_{e3} \\ U_{e3}^* U_{e1} & U_{e3}^* U_{e2} & |U_{e3}|^2 \end{pmatrix} \tag{14}$$

What did all this buy us? They are able to present the eigenvalue problem as a problem of solving for the roots of the cubic equation:

$$\lambda^3 + b\lambda^2 + c\lambda + d = 0\,, \tag{15}$$

where the coefficients of λ are:

$$\begin{aligned} b &= -1 - \alpha\,(2\eta - 1) - \hat{A} \\ c &= \left(1 - |U_{e3}|^2\right)\hat{A} - \alpha\left\{1 + \hat{A}\,|U_{e2}|^2 + \hat{A}\,|U_{e3}|^2 - \eta\left[2 + \alpha\,(\eta-1) + \hat{A} + \hat{A}\,|U_{e3}|^2\right]\right\} \\ d &= -\alpha\left[\hat{A}\eta\,|U_{e1}|^2 + \hat{A}\,(\eta-1)\,|U_{e2}|^2 + \alpha\eta\,(\eta-1)\left(1 + \hat{A}\,|U_{e3}|^2\right)\right] \end{aligned} \tag{16}$$

They arrive at the following solution for the eigenvalues of M_f :

$$\begin{aligned} \lambda_1 &= -\frac{b}{3} - \frac{1}{3\Delta_*}\sqrt{x^2 - 3y}\left[z + \sqrt{3\,(1-z^2)}\right] \\ \lambda_2 &= -\frac{b}{3} - \frac{1}{3\Delta_*}\sqrt{x^2 - 3y}\left[z - \sqrt{3\,(1-z^2)}\right] \\ \lambda_3 &= -\frac{b}{3} - \frac{2}{3\Delta_*} z\sqrt{x^2 - 3y}\,, \end{aligned} \tag{17}$$

where

$$x = \Delta_* \left[1 + (2-\eta)\,\alpha + \hat{A}\right]$$

$$y = \Delta_*^2 \left\{ \hat{A}\left(1 - |U_{e3}|^2\right) + \alpha \left[1 + \hat{A} - \hat{A}\left(1 - |U_{e3}|^2\right)\left(\eta - \frac{|U_{e1}|^2}{1-|U_{e3}|^2}\right) + \alpha^2\,(1-\eta)\right]\right\}$$

$$z = \cos\left\{\frac{1}{3}\arccos\frac{\Delta_*\left[2x^3 - 9xy + 27\Delta_*^3\alpha\hat{A}\,(1+\alpha-\eta\alpha)\,|U_{e1}|^2\right]}{2|\Delta_*|\sqrt{(x^2-3y)^3}}\right\}\,, \tag{18}$$

The $\{\lambda_1, \lambda_2, \lambda_3\}$ eigenvalues of M_f *correspond to* $\{\tilde{m}_1^2, \tilde{m}_2^2, \tilde{m}_3^2\}$.

D. Relation of eigenvalues to mass

The relationship between these eigenvalues and the actual squared mass in matter $\tilde{m}_i^2$ of the associated neutrino mass eigenstate ν_i^m is [3]

$$\lambda_i = \left[\tilde{m}_i^2 - \left(m_2^2 - \Delta_{21}\right)\right]/\Delta_{31} \tag{19}$$

This expression basically restores from Eq. 12 the Δ_* factor on M_f and the numerator of the diagonal term (as an additive term on M_f). In other words, the λ_i of M_f are dimensionless, and only become directly comparable to a neutrino squared mass in medium through the relation derived from Eq. 19:

$$\tilde{m}_i^2 = \lambda_i\Delta_{31} + \left(m_2^2 - \Delta_{21}\right) = \lambda_i\Delta m_{13}^2 + m_1^2 \tag{20}$$

Notice that the λ_i in Eq. 20 in effect specifies the fraction of Δm_{13}^2 (the largest vacuum split in normal ordering) to be added to the vacuum m_1^2 mass to generate an individual mass in matter $\tilde{m}_i^2$. We would therefore require an estimate of the actual vacuum squared mass m_2^2 or m_1^2 to see a projection of the eigenvalue of the Hamiltonian in matter as a single squared mass value.

Oscillation experiments, being interference experiments, only reveal ranges of possible squared mass splits, Δm_{ij}^2, the difference between two squared neutrino masses m_i^2 and m_j^2. Kinematic tests, on the other hand, e.g., nuclear β-decay, provide potentially a look at the actual neutrino mass involved, i.e., the endpoint energy of the electron is reduced by the

mass of the daughter electron anti-neutrino[3] .

The eigenvalue equations we obtained from [3] are intended to support the need for increased accuracy and transparency of the analytic oscillation equations[4] used in, e.g., long baseline accelerator neutrino oscillation experiments with MSW effects due to transit through Earth matter, rather than to identify actual individual squared mass values. However, we find the corresponding mass value of interest, so address that incidentally in checking the methods used in the eigenvalue trajectory code below.

III. EIGENVALUE GRAPH CODE

A. Description of graph produced

To produce Fig. 1, we held the production N_e electron number density constant at $N_e = 5.4670 \times 10^{25}$ cm^{-3} and scanned the production energy E (modify these parameters as desired). The graph portrays the difference in energy level of the two neutrino mass eigenstates depending on the energy and electron number density at production (λ_3 is also calculated and available). For example, on the graph it appears that the eigenvalues have a relatively smaller energy gap separating them over the region $\approx 3 - 7\,\text{MeV}$[5] .

B. Verify accuracy

The minimum in the difference between λ_2 and λ_1 in the graph data occurs at neutrino energy 2.1638 MeV, step 182 of 10000 increments over the energy range 350 keV to 100 MeV.

To verify the accuracy of the eigenvalue graph data minimum we will use a three-flavor MSW resonance energy equation derived from equations (95) and (96) of the 2014 final Borexino experiment Phase-I results [10]:

$$E_{RES} = \frac{\cos 2\theta_{12}\Delta m^2}{2\sqrt{2}G_F N_e \cos^2\theta_{13}} \quad \text{3-flavor, solar MSW resonance} \tag{21}$$

[3] This is currently supposed to be an average over all the mass states comprising the electron neutrino [9]. The suggestion has been made though that, very near the endpoint of the decay, conservation of energy might require solely the emission of the lightest neutrino mass state, ν_1 [6].

[4] Although existing equations for oscillation probabilities in Earth matter, for example, can be solved with numerical methods, it is difficult to see the underlying physics and thereby fully undertand the results.[3].

[5] The minimum difference $\Delta\lambda_{12} = \lambda_2 - \lambda_1$ occurs at 2.1638 MeV on the x-axis, but cannot be seen accurately on the graph.

With units and conversion factors explicit:

$$E_{res}\,[\mathrm{GeV}] = \frac{\Delta m^2_{12}\left[\mathrm{GeV}^2\right]\cos 2\theta_{12}}{2\sqrt{2}G_F\left[\mathrm{GeV}^{-2}\right](\hbar c)^3\,[\mathrm{GeV}^3\cdot\mathrm{cm}^3]\,N_e\,[\mathrm{cm}^-3]\cos^2\theta_{13}} \tag{22}$$

The ν_1-ν_2 vacuum mass split is $\Delta m^2_{21} = 7.5\ \times 10^{-5}\ \mathrm{eV}^2$, but we want to work in GeV^2 here, so divide by 10^{18} to obtain $\Delta m^2_{21} = 7.5\ \times 10^{-23}\ \mathrm{GeV}^2$. The Fermi coupling constant is $G_F = 1.1664\ \times 10^{-5}\,\mathrm{GeV}^{-2}$. We have production electron number density[6] $N_e = 5.4670\ \times 10^{25}\ \mathrm{cm}^{-3}$. $\cos 2\theta_{12} = 0.3913$. $2\sqrt{2} = 2.828427$. $\hbar c = 1.9732\ \times 10^{-14}\,\mathrm{GeV}\cdot\mathrm{cm}$ (cube this). $\cos^2\theta_{13} = 0.9781$. Plug in the numbers:

$$E_{res}\,[\mathrm{GeV}] = \frac{(7.5\ \times 10^{-23})(0.3913)}{(2.828427)(1.1664\ \times 10^{-5})(7.68350\ \times 10^{-42})(5.4670\ \times 10^{25})0.9781} \tag{23}$$

We obtain $E_{RES} = 2.1655\,\mathrm{MeV}$ with this equation, validating the eigenvalue minimum (at $2.1638\,\mathrm{MeV}$) within the $\sim 10\,\mathrm{keV}$ resolution of the graph.

C. Use non-linear mass solution

Recall our discussion of the relation of the λ_i eigenvalues to the actual squared mass in matter in Section II D, Eq. 20:

$$\tilde{m}^2_i = \lambda_i\Delta_{31} + \left(m^2_2 - \Delta_{21}\right) = \lambda_i\Delta m^2_{13} + m^2_1$$

For our purpose here we can use software minimization techniques with bounds and constraints to obtain possible solutions for the absolute vacuum masses of neutrino mass states, i.e., solutions for the individual m_i that yield the required known vacuum squared mass splittings for $m^2_{12} = 7.5\times 10^{-5}\,\mathrm{eV}^2$ and $m^2_{31} = 2.457\times 10^{-3}\,\mathrm{eV}^2$ and do not exceed the upper bound on the sum of the mass states 0.23 eV from the Planck cosmological data [11]. Using an implementation of a sequential least squares programming function (SLSQP) within the `scipy.optimize.minimize` function provided by the open-source scientific applications software `SciPy` we obtained a set of such possible mass values (this set was solved for a

[6] In another of our articles, available at Bounds on Neutrino Packet Lengths, we incidentally demonstrate how to read in and work with Standard Solar Model data, e.g., electron number density .

maximum rather than minimum meeting the constraints):

$$
\begin{aligned}
m_1 &= 0.07132812 \text{ eV} \\
m_2 &= 0.07185194 \text{ eV} \\
m_3 &= 0.08681994 \text{ eV} \\
\Delta m^2_{21} &= 7.5 \times 10^{-5} \text{ eV}^2 \\
\Delta m^2_{31} &= 2.45 \times 10^{-3} \text{ eV}^2
\end{aligned}
$$

$$
\sum_i m_i = m_3 + m_2 + m_1 = 0.2300000 \leq 0.23 \text{ eV} \qquad \text{(Planck 2013 CMB upper limit)}
$$

Let us demonstrate use of the relation in Eq. 20) with these possible vacuum mass solutions. What is the mass split in matter, $\tilde{m}^2_{12} = \tilde{m}^2_2 - \tilde{m}^2_1$, at the beginning of the scan in the graph in Fig. 1, i.e., index $[0] = 350\,\text{keV}$ (the computer code provides us incidentally with arrays populated with the λ_i values)? We would think that it should be close to the vacuum split $m^2_{12} = 7.5 \times 10^{-5}\,\text{eV}^2$ since the neutrino production energy is relatively low, but there will be some MSW modification because the electron density is high at $N_e = 5.4670 \times 10^{25}$ cm^{-3}. First obtain $\tilde{m}^2_2$:

$$
\begin{aligned}
\tilde{m}^2_2 &= \lambda_2 \Delta_{31} + \left(m^2_2 - \Delta_{21}\right) \\
&= \lambda_2[0](2.457 \times 10^{-3}\,\text{eV}^2) + \left[(0.07185194 \text{ eV})^2 - (7.5 \times 10^{-5}\,\text{eV}^2)\right] \\
&= (0.031139)(2.457 \times 10^{-3}) + \left[(5.1627 \times 10^{-3}) - (7.5 \times 10^{-5})\right] \\
&= (7.65087 \times 10^{-5}) + (5.0877 \times 10^{-3}) \\
\tilde{m}^2_2 &= 5.164210 \times 10^{-3}\,\text{eV}^2
\end{aligned}
$$

The λ_i are dimensionless so you may work in eV^2 or GeV^2 (or other) above, but be consistent in the selection within the calculation. Repeat the above calculation, substituting λ_1 to obtain $\tilde{m}^2_1$:

$$
\begin{aligned}
\tilde{m}^2_1 &= \lambda_1 \Delta_{31} + \left(m^2_2 - \Delta_{21}\right) \\
&= \lambda_1[0](2.457 \times 10^{-3}\,\text{eV}^2) + \left[(0.07185194 \text{ eV})^2 - (7.5 \times 10^{-5}\,\text{eV}^2)\right] \\
&= (0.001317)(2.457 \times 10^{-3}) + \left[(5.1627 \times 10^{-3}) - (7.5 \times 10^{-5})\right] \\
&= (3.235286 \times 10^{-6}) + (5.0877 \times 10^{-3}) \\
\tilde{m}^2_1 &= 5.090937 \times 10^{-3}\,\text{eV}^2
\end{aligned}
$$

Take the difference of the results,

$$\Delta\tilde{m}^2_{12} = \tilde{m}^2_2 - \tilde{m}^2_1 = 5.164210 \times 10^{-3}\,\mathrm{eV}^2 - 5.090937 \times 10^{-3}\,\mathrm{eV}^2 = 7.32729 \times 10^{-5}\,\mathrm{eV}^2$$

That appears to be a reasonable result, decreasing slightly from the vacuum splitting of $\Delta m^2_{21} = 7.5 \times 10^{-5}$ eV^2, as we expected from Fig. 1 and the physics of the matter potential, i.e., ν_1 wants to increase until it crosses ν_2, implying the difference between the two decreases with energy or electron density, but with a non-zero vacuum mix angle, $\theta_{12} = 33.48°$, the two mass eigenstates instead repel at the point of the would-be level crossing, unless the neutrino passes through an electron density discontinuity sufficient to violate the adiabaticity requirement[7].

D. Code listing

The following `Python` computer code produces the graph in Fig. 1 using the [3] (pp 7-8) analytical solution to the cubic eigenequation, $\lambda^3 + b\lambda^2 + c\lambda + d = 0$, describing the eigenvalues for the M_f subcomponent of the effective three-flavor Hamiltonian in medium, i.e., with matter potential as discussed above in Section II. Use assumes you have `SciPy`, `NumPy`, `Matplotlib`, and `Python 3` and know how to make code adjustments if you have different versions:

```
# coding: utf-8

from scipy import constants
# we want to be forced to prepend math. to all Python math functions
import math
import cmath

import numpy as np

import matplotlib.pyplot as plt
```

[7] In another of our articles we cited earlier, we briefly calculate the probability of an adiabaticity violation in the solar plasma. See Solar Neutrino Survival Probability Scan

```
import matplotlib as mpl

# Fermi coupling constant in GeV^-2
G_f = constants.value('Fermi coupling constant')
# speed of light in m/s
c = constants.value('speed of light in vacuum')
hbar = constants.value('Planck constant over 2 pi in eV s')
hbar_GeVs = hbar / 1e9
c_cm_s = c * 1e2
hbarcGeVcm = hbar_GeVs * c_cm_s
root2 = math.sqrt(2)
# Avogadro's number 6.022140857e+23
N_0 = constants.value('Avogadro constant')

# values from 2016 global fit
# eV^2 converted to GeV^2 since energy in GeV below
delta_m12_sqr = 7.5e-5 * 1e-18
delta_m31_sqr = 2.457e-3 * 1e-18
delta_m32_sqr = delta_m31_sqr - delta_m12_sqr

# want to use radian version of theta12 PMNS mix angle
theta_12 = math.radians(33.48)
theta_23 = math.radians(42.3)
theta_13 = math.radians(8.5)

# use Li, Zhang, Zhou nomenclature for variable dense equations
# c12, for example, is cos theta_12, etc.

c12 = math.cos(theta_12)
c12_squared = c12**2
c13 = math.cos(theta_13)
s13 = math.sin(theta_13)
```

```
s12 = math.sin(theta_12)
s12_squared = s12**2
s13_squared = s13**2
c13_squared = c13**2

s23_squared = math.sin(theta_23)**2 # not used, but wanted to see it

Delta_32 = delta_m32_sqr
Delta_31 = delta_m31_sqr
Delta_21 = delta_m12_sqr

# first row of PMNS matrix is used, squared
Ue1_squared = c12_squared*c13_squared
Ue2_squared = s12_squared*c13_squared
Ue3_squared = s13_squared

n_e = 5.4670e25 #8B max emit point density cm^-3

# energy scan range in GeV; number of steps
E_min = 0.350e-3 # 350 keV in units GeV
E_max = 100e-3  # 100 MeV in units Gev

n_E_steps = 10000

# x_ax contains the energy scan steps, GeV
x_ax = np.linspace(E_min,E_max, n_E_steps, endpoint=True)

# 2*sqrt2 * G_f * n_e, will be multiplied by E on each step below
# returns GeV
# n_e in cm^-3 , hbarc in GeV cm
Matter_potential = 2.0 * root2 * G_f * n_e * hbarcGeVcm**3
```

```
# scan neutrino energies and calculate matter potential A for each
# recall that  A = 2*sqrt2 * G_f * E * n_e
# returns GeV^2, expecting GeV in for potential and energy steps
A = np.multiply(x_ax, Matter_potential)

# select your scheme here by assigning eta the desired scheme parameter

eta = 1.0 # normal scheme
eta2 = 2*eta

# Delta_ast is a constant, a scalar; it varies depending on scheme eta
Delta_ast = eta * Delta_31 + (1-eta) * Delta_32

# alpha_c is a constant, a scalar
alpha_ast = Delta_21 / Delta_ast

# *** A_hat_ast is a VECTOR (array) consisting of 2EV(steps) / Delta_ast ***
A_hat_ast = np.divide(A,Delta_ast)

# b is a VECTOR (array) since A_hat_ast is
# get the constant part done prior to the array operation
b_constant_part = -1 - alpha_ast*(eta2 - 1)
# subtract array A_hat_ast from constant part
b = b_constant_part - A_hat_ast

# x is a VECTOR (array) since its input A_hat_ast is of that ilk
# get some of the constant parts done prior to array operation
x_constant_part = Delta_ast * (  1 + ( 2- eta) * alpha_ast)
# so we have Delta_ast*A_hat_ast[], then add to the constant part
x =  np.add(np.multiply(A_hat_ast,Delta_ast),x_constant_part)

# y will be a VECTOR (array) since it consumes A_hat_ast
```

```
# some constant parts
Delta_ast_squared = Delta_ast**2
one_minus_Ue3_sqrd = 1 - math.fabs(Ue3_squared)
eta_min_Ue3_fract = eta - math.fabs(Ue1_squared)/one_minus_Ue3_sqrd
alpha_sqr_one_min_eta = alpha_ast**2 * (1 - eta)

y = Delta_ast_squared * ( np.multiply(A_hat_ast,one_minus_Ue3_sqrd) + \
        alpha_ast * ( np.add(A_hat_ast,1) - \
        np.multiply(A_hat_ast,one_minus_Ue3_sqrd) * eta_min_Ue3_fract )\
                        + alpha_sqr_one_min_eta )



# z will be VECTOR (array) since contains matrix A_hat_ast
# work out from the A_hat_ast array part
z_Ahat_factor = 27 * Delta_ast**3 * alpha_ast
z_Ahat_component_outer = np.multiply(A_hat_ast, z_Ahat_factor)
z_Ahat_first_expression = (1 + alpha_ast - eta*alpha_ast)*\
                        math.fabs(Ue1_squared)
z_Ahat_consolidate_1 =  \
    np.multiply(z_Ahat_component_outer,z_Ahat_first_expression)
# prepare a couple of bracket terms to add to the consolidated A_hat_ast
nine_xy = np.multiply(np.multiply(x,y),9)
two_x_cubed = np.multiply(np.power(x,3),2)
# sum the numerator stuff so far, subtracting last to avoid negatives
z_numerator_part_1 = two_x_cubed + z_Ahat_consolidate_1 - nine_xy
# multiply by the Delta_ast factor
z_numerator = np.multiply(z_numerator_part_1, Delta_ast)
# prepare denominator, evil square root first
z_denom_radicand = np.power( np.square(x) - np.multiply(y,3), 3 )
z_denom_sqrrt = np.sqrt( z_denom_radicand )
z_denominator = np.multiply(z_denom_sqrrt, (2*np.absolute(Delta_ast)) )
z_fract_arg_of_accos = np.divide( z_numerator, z_denominator )
```

```
# z_fract_arg_of_accos should be clean domain [-1.0,+1.0] now
z_arccos_part = np.arccos(z_fract_arg_of_accos)
# apply the 1/3 factor to that
z_braced_part = np.multiply(z_arccos_part, (1.0/3.0))
# take the cosine of that result and ship it out
z = np.cos(z_braced_part)

# do the part common to all three eigenvalues
L_common_b_by_3 = np.divide(b,3) # a matrix
L_common_one_by_3Delta = -1.0 / (3.0*Delta_ast) # a scalar NEGATIVE

L_common_xy_sqrt = np.sqrt( np.square(x) - np.multiply(y,3) )

# and a part common to first two eigenvalues
one_minus_zsqr_part = np.subtract(1.0, np.square(z))
L12_common_root = np.sqrt( np.multiply(3.0, one_minus_zsqr_part) )

# assemble eigenvalue 1, the last shall be first
lam_1 = np.add(z,L12_common_root) * L_common_xy_sqrt *\
        L_common_one_by_3Delta - L_common_b_by_3


# assemble eigenvalue 2, the rightmost first, polarity here only difference
lam_2 = np.subtract(z,L12_common_root) *\
        L_common_xy_sqrt * L_common_one_by_3Delta - L_common_b_by_3


# assemble eigenvalue 3, somewhat different than first two
two_by_3delta = 2.0 / (3*Delta_ast)
lam_3 = np.multiply(z,L_common_xy_sqrt) * two_by_3delta - L_common_b_by_3
```

```
# ditch previous runs or end up with multiple plots or pieces of plots
plt.clf()

fg1, axes = plt.subplots(1, 1) # create a figure

# scale x up to MeV from GeV
x_ax_MeV = np.multiply(x_ax,1e3)

# plot eigenvalue 1 first
axes.plot(x_ax_MeV, lam_1, color='r')

# eigenvalue 2
axes.plot(x_ax_MeV, lam_2, color='b', linestyle='dashed')

# eigenvalue 3 (plotting in invisible white to tweak plot of other 2)
axes.plot(x_ax_MeV, lam_3, color='w', linestyle='dotted')

# want to see the tick markers (not labels) on left and right
axes.yaxis.set_ticks_position('both')

axes.legend([r'$\lambda_1$', r'$\lambda_2$'])

axes.set_yscale('log')
axes.set_xscale('log')

axes.set_title(r'$\lambda_{1,2}$ vs E')
axes.set_xlabel('E (MeV)')

plt.show()
plt.close()
```

[1] P. Hernández , "Neutrino physics," [hep-ph] (2017), Comprehensive coverage of neutrino physics, collected lectures by author, e.g., from CERN CLASHEP2015 , 1708.01046v1.

[2] C. Patrignani *et al*, "Review of particle physics," Chin. Phys. C **40**, 100001 (2016), chapter 14, Neutrino mass, mixing and oscillations and other chapters for constants, methods, etc.

[3] Yu-Feng Li *et al*, "Looking into analytical approximations for three-flavor neutrino oscillation probabilities in matter," [hep-ph] (2016), 1610.04133v2.

[4] Zhi-zhong Xing, "New formulation of matter effects on neutrino mixing and cp violation," (2000), hep-ph/0002246v3.

[5] George B. Arfken and Hans J. Weber, *Mathematical Methods for Physicists, sixth edition* (Elsevier Academic Press, 2005) pp. 940–941.

[6] G. Fantini, A. Gallo Rosso, F. Vissani, V. Zema, "THE FORMALISM OF NEUTRINO OSCILLATIONS: AN INTRODUCTION," [hep-ph] (2018), 1802.05781v1.

[7] Erika Catano-Mur, *Constraints on neutrino oscillation parameters with the NOvA experiment*, Ph.D. thesis, Iowa State University, Ames, Iowa (2018).

[8] A. Yu. Smirnov, "Solar neutrinos: Oscillations or No-oscillations?" [hep-ph] (2017), p. 9, 1609.02386v2.

[9] G. Drexlin *et al*, "Current direct neutrino mass experiments," [physics.ins-det] (2013), 1307.0101v1.

[10] G. Bellini *et al* [Borexino Collaboration], "Final results of borexino phase-I on low-energy solar-neutrino spectroscopy," [hep-ph] (2014), pp. 2, 5, 49-50, 1308.0443v2.

[11] P.A. Ade *et al* [Planck Collaboration], "Planck 2013 results. xvi. cosmological parameters," (2014).