A log-likelihood fit for extracting neutrino oscillation parameters

Hanlin Xiao
CID: 01490409
Date: 12th Dec 2022
Imperial College London

***Abstract*—The neutrinos oscillate between flavours when they travel over long distances. This is an important property of theirs and indicates that neutrinos are not massless. In this project, a probability distribution of the neutrino oscillation, together with a set of simulated unoscillating neutrino events data set, is fitted to a simulated T2K data set by minimising the Negative Log Likelihood function of them. The extraction is done for one, two and three-parameter cases separately with different minimisation algorithms. The extraction results of the three-parameter case that give the best fit are $\theta_{23} = 0.79 \pm 0.01$, $\Delta m^2_{23} = (2.34 \pm 0.02) \times 10^{-3}$ $\mathbf{eV^2/c^2}$ and neutrino cross-section $\alpha = 1.13 \pm 0.05$ $\mathbf{GeV^{-1}}$.**

Also, the efficiencies of the minimisation algorithms are compared by their running time and visualisations of the paths. The Newton method is shown to be the most efficient algorithm for this project.

I. Introduction

The neutrino is a particle first introduced by W. Pauli[5] to ensure energy, momentum and spin conservation. The neutrino has long been considered a massless particle in Standard Model[7], while a small but finite neutrino mass could be postulated by the SeeSaw mechanism by introducing a neutral particle of a very large mass [5]. Therefore, physicists set out to search for non-zero neutrino mass. One method is by studying neutrino oscillation. Neutrinos have three flavours which are tau, muon, and electron. After travelling for a distance, the neutrinos are postulated to have a possibility to change their flavour[2]. The survival probability of a muon neutrino (the possibility that it does not change its flavour) follows the formula[7],

$$\mathcal{P}(\nu_\mu \to \nu_\mu) = 1 - \sin^2(2\theta_{23}) \sin^2\left(\frac{1.267\Delta m^2_{23} L}{E}\right) \quad (1)$$

where E is the energy of the muon, L is the distance the neutrinos travelled, θ_{23} is the mixing angle, denoting the fraction of ν_2 and ν_3 components in ν_μ and $\Delta m^2_{23} = m^2_{\nu i} - m^2_{\nu j}$ is the squared difference between the i-th and j-th mass states of definite mass[5]. The neutrinos need to have mass difference and thus neutrino oscillation. Therefore, motivated by the prediction of neutrino oscillation and the mass of the neutrino, many experiments are done. Not until 1998, at the 18th International Conference on Neutrino Physics and Astrophysics, evidence of neutrino oscillation is confirmed [3].

The aim of this project is to extract the parameters θ_{23} and Δm^2_{23} from a set of data of a simulated T2K experiment, one of the experiments that use a large detector to gather neutrino data and measure the neutrino oscillation. In order to do that, the Negative Log Likelihood between the T2K data and the product of a set of simulated unoscillating data and the oscillation probability distribution is minimised by different algorithms to obtain the parameter(s) that would best fit the two sets of data. The whole process is done for one, two, and three-parameter cases respectively. Also, the paths of minimisation are visualised to compare the efficiency of the different minimisation algorithms.

II. Methods

A. Data acquisition

There are two sets of known data, both giving the number of neutrino events vs. the energy of the neutrino. Both sets of data are given by the physics department of Imperial College as the sample data for the project. The data is different for every student. The first set of data is the simulated data of the T2K experiment, i.e. the data of oscillating neutrinos, which will be referred to as **T2K data** in this report. The second set of data is the simulated number of neutrino events assuming the neutrinos do not oscillate, which is referred to as **unoscillating data**.

B. Numerical methods

1) Negative Log Likelihood function: The **Negative Log Likelihood(NLL)** function is used to account for the likelihood of a set of data

according to a postulated model. First, consider a probability distribution $\mathcal{P}$ of n possible values. As a result, the likelihood of getting a set of data **m** of n values from this probability distribution is

$$\mathcal{L} = \prod_{i=1}^{n} \mathcal{P}(m_i), \tag{2}$$

where m_i is the i-th value in the data. Then, suppose the probability distribution is dependent on a set of parameters **u**, the likelihood equation changes to

$$\mathcal{L}(\mathbf{u}) = \prod_{i=1}^{n} \mathcal{P}(\mathbf{u}; m_i), \tag{3}$$

For the data to fit the probability distribution the most, the likelihood function needs to be maximised, or in other words, the NLL need to be minimised. By taking a logarithm of both sides of equation (3), the NLL can be obtained to be

$$\begin{aligned} \text{NLL}(\mathbf{u}) &= -\ln\left(\prod_{i=1}^{n} \mathcal{P}(\mathbf{u}; m_i)\right) \\ &= -\sum_{i=1}^{n} \ln\left(\mathcal{P}(\mathbf{u}; m_i)\right). \end{aligned} \tag{4}$$

In this project, the independent variable of the data is the energy of neutrinos. The dependent variable m_i is the number of events of the neutrino detected at the energy i. The probability of detecting m_i events at energy i can be obtained by a Possion distribution,

$$\mathcal{P}(m_i) = \frac{\lambda_i^{m_i} e^{-\lambda_i}}{m_i!}, \tag{5}$$

where λ_i is the average number of events at energy i, specifically in this case, the λ_i corresponds to the estimated number of events at energy i, which could be obtained by the product of the unoscillating data and the probability distribution of neutrino oscillation given by equation (1). Since the PDF of equation (1) depends on a number of parameters to fit, $\lambda_i = \lambda_i(\mathbf{u})$ where **u** contains the parameters to fit. Finally, by plugging equation (5) into equation (4), the NLL function is rearranged to be,

$$\begin{aligned} \text{NLL}(\mathbf{u}) &= -\ln\left(\prod_{i=1}^{n} \frac{\lambda_i^{m_i}(\mathbf{u}) e^{-\lambda_i(\mathbf{u})}}{m_i!}\right) \\ &= \sum_{i=1}^{n} [\lambda_i(\mathbf{u}) - m_i \ln \lambda_i(\mathbf{u}) + \ln m_i!\,]. \end{aligned} \tag{6}$$

To reiterate the meaning of the parameters, λ_i is the mean of the Poisson distribution of the energy i, which can be obtained by the product of $\mathcal{P}(\mu \to \mu)$ and the unoscillating data. The **u** contains the parameters to fit and m_i is the number of neutrino events at energy i in the T2k data. Also, since $\ln(m_i)!$ is simply a constant and will not have any effect on the minimisation, it will be omitted in the later calculation.

2) Parabolic Method: The parabolic method is a 1D iterative method. It is implemented by first selecting three points x_0, x_1, and x_2 as initial guesses, from the function $y = y(x)$. Then the next iterative point is found by interpolating the three points with a parabolic function and choosing the x at the minimum of the parabolic function[6],

$$x_3 = \frac{(x_2^2 - x_1^2)y_0 + (x_0^2 - x_2^2)y_1 + (x_1^2 - x_0^2)y_2}{(x_2 - x_1)y_0 + (x_0 - x_2)y_1 + (x_1 - x_0)y_2}, \tag{7}$$

where $y_i = y(x_i)$. Then the three xs that has the minimum y(x) are kept for the next iteration. Additionally, a criterion on the difference of y needs to be set for the iteration to stop.

3) Univariate Method: The univariate method is simply the multiple-dimensional version of the parabolic method. Suppose there are 2 parameters to fit. The steps of the algorithm are:

1) Select one parameter, ideally the one that will make the convergence faster(less local minimum, steeper gradient...)
2) Implement the parabolic minimisation scheme on this parameter while setting the other parameter to a fixed initially guessed value.
3) Implement the parabolic minimisation on the second parameter while fixing the first parameter.
4) Iterate the process until the criterion of difference in y is reached.

4) Newton Method: The Newton Method is a multiple-dimensional method, which uses the property of multiple-dimensional Taylor expansion. The derivation could be found in [6] or most numerical methods textbooks such as [4] and will not be described in detail here for the length of the report. The steps of the algorithm are:

1) Choose a set of initial guesses for the multidimensional parameters.
2) Update the vector of the parameters by[6]

$$\vec{x}_{n+1} = \vec{x}_n - [\mathbf{H}(\vec{x}_n)]^{-1} \cdot \vec{\nabla} f(\vec{x}_n), \tag{8}$$

where $f(\vec{x})$ is the function to minimise and $\mathbf{H}(\vec{x}_n)$ is the Hessian matrix of the function,

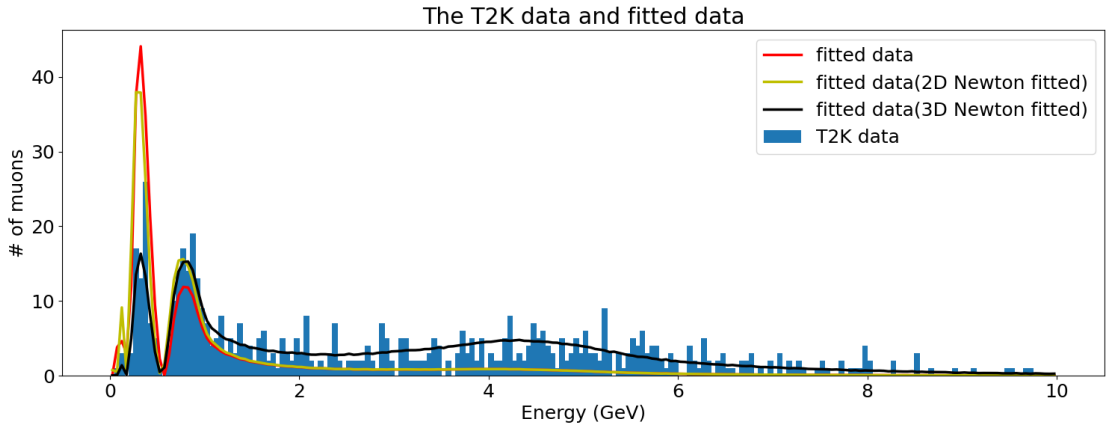

Fig. 1. This plot shows the accuracy of the fitting with different numbers of parameters. The fitting results of the Newton method are used since the numerical results obtained by different methods are similar for the same number of parameters. The number of parameters corresponds to the dimensions of the fitting. For 1D and 2D fitting, the peak at around 0.3 GeV and the trend after 2 Gev of the T2K data are not represented by the model accurately, while the 3D fitting accounted for these two parts with better precision.

$$\mathbf{H}_{ij}(\vec{x}) = \frac{\partial^2 f(\vec{x})}{\partial x_i \partial x_j} \tag{9}$$

The gradient and the Hessian matrix are calculated by forward difference scheme[6] and its extension, which also will not be described in detail here.

3) Iterate until the stopping criterion is met.

The Newton method is a higher-order gradient method, which involves the inverse of the Hessian matrix, therefore having a considerable computational cost.

5) Monte Carlo Method: The Monte Carlo method is a multidimensional minimisation scheme. It utilised the concept of thermodynamics. The function will go in random directions and even the paths that may increase the value of the objective function might be accepted for a certain probability. This trait allows the function to escape local minima. The steps of the algorithm are:

1) Select a set of initial guesses $\vec{x}$ as the starting point of the function, a step size h for each iteration and a T_{max} corresponds to the maximum temperature of a thermal dynamic system. In the algorithm, the T_{max} also denotes the number of the iterations
2) Start from temperature $T = T_{max}$ and move $\vec{x}_i$ to a random direction for the small step size h, i.e.

$$\vec{x}_{i+1} = \vec{x}_i + h\vec{\delta}, \tag{10}$$

where the i-th element in $\vec{\delta}_0$ is a random number between 0 and 1 times x_i (to scale the size of the step to be consistent with the relative scale of different x components).

3) Accept the $\vec{x}_{i+1}$ according to the probability,

$$\mathcal{P}_{acc} = \begin{cases} 1 & \text{if } \Delta E < 0) \\ e^{\frac{-\Delta E}{k_b T}} & \text{if } \Delta E > 0) \end{cases} \tag{11}$$

4) If the $\vec{x}_{i+1}$ is not accepted, repeat step 2 on $\vec{x}_i$
5) No matter the $\vec{x}_{i+1}$ is accepted or not, each time the $\vec{x}_{i+1}$ is generated, update $T = T-1$. Therefore, at lower T, the possibility of accepting a larger $\vec{x}_{i+1}$ is lower.
6) Stop the function until T = 0.

6) Error analysis: There are two ways of calculating the error of the i-th minimised parameters. The first method is by Taylor expand the NLL function to the second order to approximate it by a Gaussian function, and the standard deviation(error) can be estimated as,

$$\sigma = \frac{1}{\sqrt{\frac{\partial^2}{\partial^2 x_i}\text{NLL}(x_i)}} \tag{12}$$

The other method is implemented by finding θ_{23}^+ and θ_{23}^-, the positions of the i-th component of $\vec{x}$ at which the NLL function increases by 0.5 from the minimum. The error is approximated by the difference $\sigma = \theta_{23}^+ - \theta_{23}^-$. The results in this project will use the first methods for estimations of the errors.

III. Results

The project aims to extract parameters from the simulated T2K data by minimising the NLL function. The process is done for one, two

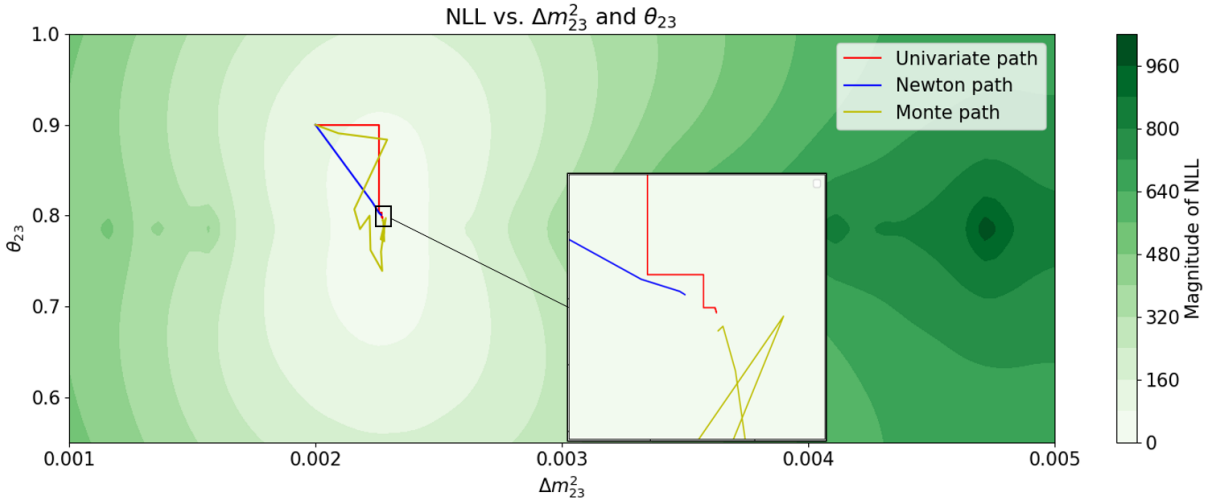

Fig. 2. The contour plot shows the convergent paths of the different minimisation algorithms. The background shows the magnitude of the NLL function, which is represented by the colour. The Univariate method(red) makes right-angle turns because it is minimised on one parameter at a time. The Newton method(blue) goes from the initial guess position to the minimum in a straightforward way. The Monte Carlo method(yellow) is zigzag but gradually leads to the minimum position finally, as shown in the zoomed-in subplot. Also, it could be spotted that there is no clear local minimum that may trap the convergent path, so it is not necessary to use the Monte Carlo method in this case.

and three-parameter cases respectively. For each case, different sets of parameters are defined, different minimisation methods are used and the resultant minimised values of the NLL function are compared.

A. 1D minimisation

Recall that, by equation (6), the NLL($\mathbf{u}$) depend on $\lambda_i(\mathbf{u})$, which is obtained by

$$\boldsymbol{\lambda}(\mathbf{u}) = \mathcal{P}(\mathbf{u}; \nu_\mu \to \nu_\mu) \cdot G(E), \qquad (13)$$

where $G(E)$ is the distribution of unocsillating data. For $\mathcal{P}(\mathbf{u}; \nu_\mu \to \nu_\mu)$ with known L and variable E, the potential parameters are θ_{23} and Δm^2_{23}. In the 1D case, the θ_{23} is minimised by the parabolic minimisation scheme and with a fixed value of Δm^2_{23} given in the project instruction[7].

By setting the value of $\Delta m^2_{23} = 2.4 \times 10^{-3}$ and performing the minimisation on the NLL function with initial guess $\theta_{23} = \pi/4$, the minimisation result is shown in Table 1. Also, the resultant fitting is shown in Fig. 1 as well. However, it could be spotted in the plot that the amplitude of data is not accounted for precisely after 2 GeV.

B. 2D minimisation

The 2D minimisation used θ_{23} and Δm^2_{23} as the parameters in the Univariate, Newton and Monte Carlo methods. The numerical results are listed in Table 1, and the minimisation paths are shown in Fig. 2.

For the Univariate method, the Δm^2_{23} is minimised first because the gradient is steeper along that direction, which can be observed from the density of the contour lines in Fig. 2. For the Monte Carlo method, the T_{max} is taken to be 1000 to ensure the minimum is reached accurately. The step sizes for gradient calculations and convergent criterion are both set to be $1e^{-4}$. The k_b in the Monte Carlo method is set to be $1e^{-5}$. And the initial guesses are all set to be $[\Delta m^2_{23}, \theta_{23}] = [0.002, 0.9]$ to show a clear convergent path and to allow the algorithm to converge.

As shown in Fig. 1, the peak at the front and the bump after 2 GeV are still not accounted for accurately. The improvement compared to 1D case is limited, since the default Δm^2_{23} in 1D minimisation is already close to the minimum. Regarding the efficiency of the algorithms, judging from the path visualisation and running time, $t_{Uni} > t_{Monte} > t_{Newton}$, it can be seen that the Newton method is the most efficient one, the Monte Carlo method is the second, and the Univariate method is the least efficient one. The advantage of the Monte Carlo method is that it could help escape local minima, but it can be seen from the contour plot that there is not any local minimum. So the Newton method would be the most efficient in this case.

C. 3D Minimisation

The 3D minimisation took takes into account another parameter that governs the expected rate

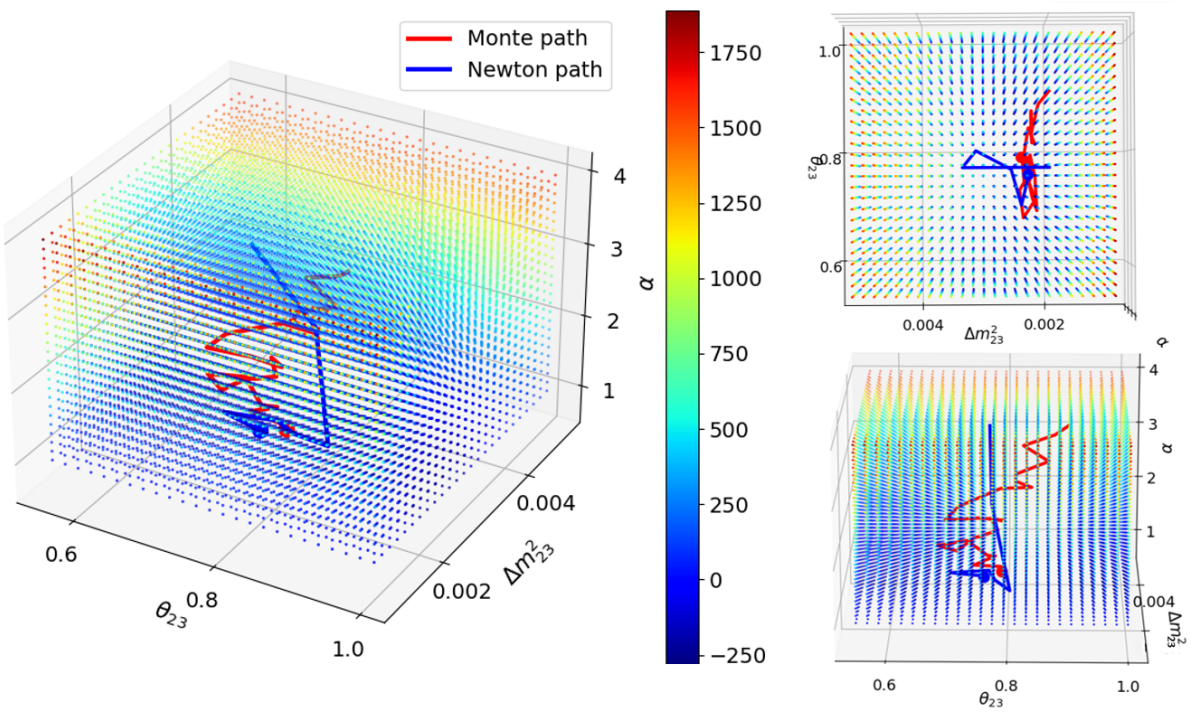

Fig. 3. This is a 4D visualisation of the minimisation path of the Newton method and the Monte Carlo methods, with two subplots from different angles. The magnitude of the NLL function is represented by the colour of the scattered points as shown in the colour bar. In this case, the Newton method does not go straight to the position of the minimum. This is due to the more complicated magnitude distribution of the NLL function in the three-parameter case. Similar to the two-parameter case, the Monte Carlo path is zigzag but still reaches the similar position of minimum as the Newton method.

of neutrino events, which is the neutrino interaction cross-section α. α is assumed to be linear with the neutrino energy and a new λ (expected number of neutrino events) can be defined by,

$$\lambda_i^{new}(\mathbf{u}) = \lambda_i^{old}(\mathbf{u}) \cdot \alpha \cdot E_\nu \qquad (14)$$

where α is the neutrino cross-section and E_ν is the energy of the neutrino. And the corresponding NLL_{new} by plugging equation (14) into equation (6) is the new function to be minimised with three parameters.

For the efficiency of convergence, the Newton method and the Monte Carlo method are used to perform the minimisation. The default constants for fitting are the same as in the 2D case.

The numerical results are listed in Table I and the minimisation paths are shown in Fig. 3. Similar to the 2D case, the Newton method reaches the minimum in a more efficient manner, which is also shown by the running time recorded in Table I. Note that the running time of the Monte Carlo method is directly related to the manually chosen T_{max}. However, according to my testing, roughly for $T_{max} > 800$ the algorithm could stably reach a minimum of uncertainty 279 ± 2, so the $T_{max} = 1000$ could represent the efficiency of the Monte Carlo method fairly.

The NLL_{min} in 3D case drastically decreased to around -280. This could also be indicated by the closeness of black fit line to the T2K data in Fig. 1. The peak in the front and the bump at the tail are both better accounted for in this case.

Additionally, it should be noted that the value of the Negative Log Likelihood function becomes negative in the 3D case because the last term in equation (6) is omitted.

IV. Conclusion

The model of the simulated T2K data is obtained by the product of the unoscillating simulated data and the probability distribution of neutrino oscillation eq. (1). By minimising the Negative Log Likelihood function of the T2K data compared to the model, the parameters that will fit the model best to the T2K data are obtained. The process is done for one, two and three-parameter cases corresponding to the involvement of the mixing angle θ_{23}, the difference of the square of mass eigenstates Δm^2_{23}, and the neutrino interaction cross-section α. Different minimisation algorithms are used and their efficiencies are investigated.

It is shown that the two-parameter fits lead to limited improvement of the one-parameter

TABLE I
NUMERICAL RESULTS OF THE DIFFERENT FITTING SCHEMES

	NLL_{min}	θ_{23}	Δm^2_{23} (eV^2/c^4)	α	time(ms)
1D(Parabolic)	63.76	0.79 ± 0.03	2.4×10^{-3}(fixed)	—	3
2D(Univariate)	51.79	0.80 ± 0.03	$(2.27 \pm 0.02) \times 10^{-3}$	—	90.1
2D(Newton)	51.82	0.80 ± 0.03	$(2.27 \pm 0.02) \times 10^{-3}$	—	4
2D(Monte)	51.79	0.78 ± 0.05	$(2.28 \pm 0.02) \times 10^{-3}$	—	14
3D(Newton)	-279.29	0.76 ± 0.03	$(2.29 \pm 0.03) \times 10^{-3}$	1.25 ± 0.05	15
3D(Monte)	-281.15	0.79 ± 0.01	$(2.34 \pm 0.02) \times 10^{-3}$	1.13 ± 0.05	47

fit because the default value of Δm^2_{23} is already close to the minimum. However, the three-parameter fit largely improved the accuracy of the fit, which can be observed directly from Fig. 1. The numerical results of the fitting are shown in Table.I

Regarding the efficiency of the algorithms, it has been shown that the Newton method is the most efficient, the Monte Carlo method is the second, and the Univariate method is the least efficient for a two-parameters fitting in this data set. Also, the Newton method is more efficient than the Monte Carlo method in the 3-parameter fitting. These are demonstrated both by the running time in Table. I and the path visualisation in Fig. 2 and Fig. 3. The 3D Monte Carlo method gives the minimum NLL function value, corresponding to the best fit. Also, the value of $\sin^2(\theta_{23}) = 0.50 \pm 0.01$. Therefore, comparing it to the literature values[1] where $\Delta m^2_{23} = (2.51 \pm 0.10) \times 10^{-3}\ eV^2/c^4$ and $\sin^2(\theta_{23}) = 0.511 \pm 0.055$, the values are relatively similar, for Δm^2_{23} and $\sin^2(\theta_{23})$ both within the range of 2σ of the literature measurement. The sources of the error could come from the approximation of the neutrino cross-section to be linear with neutrino energy, and the fact that the data is simulated so there could be some statistical inaccuracy.

In summary, the project extracted key parameters in neutrino oscillation and supported that neutrinos are not massless. Also, the efficiency of the Newton method is shown to be higher than the Monte Carlo method and the Univariate method in this data set and model.

REFERENCES

[1] K Abe, J Adam, H Aihara, T Akiri, C Andreopoulos, S Aoki, A Ariga, T Ariga, S Assylbekov, D Autiero, et al. Precise measurement of the neutrino mixing parameter θ 23 from muon neutrino disappearance in an off-axis beam. *Physical review letters*, 112(18):181801, 2014.

[2] Samoil M Bilenky and B Pontecorvo. Lepton mixing and neutrino oscillations. *Physics Reports*, 41(4):225–261, 1978.

[3] Yoshiyuki Fukuda, T Hayakawa, E Ichihara, K Inoue, K Ishihara, Hirokazu Ishino, Y Itow, T Kajita, J Kameda, S Kasuga, et al. Evidence for oscillation of atmospheric neutrinos. *Physical review letters*, 81(8):1562, 1998.

[4] Joe D Hoffman and Steven Frankel. *Numerical methods for engineers and scientists*. CRC press, 2018.

[5] Takaaki Kajita. Atmospheric neutrinos and discovery of neutrino oscillations. *Proceedings of the Japan Academy, Series B*, 86(4):303–321, 2010.

[6] Mark Scott and Paul Dauncey. Lecture notes of computational physics. 2022.

[7] Mark Scott and Paul Dauncey. Project1: A log-likelihood fit for extracting neutrino oscillation parameters. 2022.