Extracting Neutrino Oscillation Parameters Using A Negative Log Likelihood Fit

Suhasan Kanagasabapathy 01338402

***Abstract*—A set of simulated T2K data was analysed to extract the parameters of muon neutrino oscillation. A negative log-likelihood fit with a Poisson probability density function was used. The parameters were the mixing angle, θ_{23}, difference between the squared masses of the two neutrinos, Δm_{23}^2 and the cross section scaling constant, γ. Various numerical methods such as parabolic minimiser and gradient descent method were used to determine the minimum of the log-likelihood function which best fit the data. The values of the parameters obtained were θ_{23} = (0.675 $\pm$ 0.012) rad and (0.896 $\pm$ 0.012) rad, Δm_{23}^2 = (2.727 $\pm$ 0.038) $\times$ 10^{-3}eV^2, γ = (1.601 $\pm$ 0.059) GeV^{-1}. This proves the existence of neutrino oscillations, hence shows the neutrinos have non-zero mass.**

I. Introduction

THE aim of the investigation is to implement a negative log-likelihood fit on the event rate of muon neutrinos observed in a simulated T2K experiment. The parameters that minimise the likelihood function will best fit the data. Various numerical methods are needed to perform the minimisation.

II. Theory

A. Neutrino Oscillation

Neutrinos are regarded as one of the fundamental particles in the Standard Model of Particle Physics. They have no electric charge and are colorless, thus only interact via the weak nuclear force. Neutrinos come in three flavours (ν_e, ν_μ and ν_τ) with each corresponding to their lepton partners; electrons, e, muons, μ and taus, τ. Initially, neutrinos were assumed to be massless in the Standard Model [1].

However, the experiments such as Super-Kamiokande and SNO measured a decrease in the expected rate of neutrino interactions from the atmosphere and the Sun. This decrease was caused by a quantum mechanical phenomenon called neutrino oscillations where neutrinos oscillate between the three flavour states as they propagate through space. The first order approximation of the 'survival probability' of a muon neutrino with an energy E (GeV) as it travels L (km) is given by Eq. 1. This refers to the probability that a muon neutrino will be detected as a muon neutrino and will not have oscillated to a tau neutrino [2].

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

where θ_{23} represents the mixing angle which determines the amplitude of the neutrino oscillation probability, and Δm_{23}^2 is the difference of the squared masses of the two neutrinos, which affects the frequency of the oscillations [2]. This oscillation is only possible when Δm_{23}^2 term is non-zero, i.e. the oscillating neutrinos must have non-zero masses. In order to determine these two parameters, long-baseline neutrino experiments measure the muon neutrino events as function of energy E at a constant distance L.

B. Negative Log-Likelihood Function

The weak interaction of the neutrinos with the detectors means the experiments are statistically limited. Therefore, the number of muon neutrino events observed are best represented by a Poisson distribution, which is the probability density function chosen $\mathcal{P}(\mathbf{u})$ where $\mathbf{u}$ is the set of unknown parameters of the function. The density function is given by

$$\mathcal{P}(k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (2)$$

where λ is the expected event rate in each bin, i of a data histogram in our experiments, and k is the observed number of events in bin i [2]. Here, the expected event rate, λ actually depends on $\mathbf{u}$, giving $\lambda(\mathbf{u})$. The likelihood $\mathcal{L}$ of n measurements, m of the events is given by

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

Our goal is to find the value of $\mathbf{u}$ which maximises the $\mathcal{L}(\mathbf{u})$. This corresponds to the best fit between the measurements given and the $\mathcal{P}$ chosen. This is analogous to taking the minimum of the *Negative Log Likelihood (NLL)* given by

$$NLL(\mathbf{u}) = -\sum_{i=1}^{n} log(\mathcal{P}(\mathbf{u}; m_i)). \quad (4)$$

Rewriting this with the function $\mathcal{P}$ given in Eq. 2 gives

$$NLL(\mathbf{u}) = \sum_{i=1}^{n}\left[\lambda_i^k - k_i + k_i log\left(\frac{k_i}{\lambda_i^k}\right)\right]. \quad (5)$$

In our analysis, numerical methods will be used to minimise the *NLL* and thus determine the minimising parameters $\mathbf{u}$.

III. Methods

A. The Data

A personalised set of simulated data obtained from T2K experiment was provided. Eq. 5 was applied to this data with k_i as measurements of the muon neutrino event rate as a function of energy and $\mathbf{u}$ = (θ_{23}, Δm_{23}^2). The data file provided consisted of two lists, each containing 200 values. One of the lists was the set of measured number of muon neutrino events k_i per bin i and the other was the simulated event rate prediction ϕ in the case where muon neutrinos do not oscillate. Each value for events corresponded to a particular energy bin with a width of 0.05 GeV. Since there were 200 such bins, the range of the energy for the events observed was 0-10 GeV. From now on, the analysis for this data was done entirely on Python. A histogram of muon neutrino events observed was plotted to visualise the data.

The Eq. 1 was coded into a function, P. The initial values used for the variables were: $\theta_{23} = \frac{\pi}{4}$, $\Delta m_{23}^2 = 2.4 \times 10^{-3} eV^2$, L = 295 km. A plot of P as a function of E for different values of θ_{23} was plotted to understand the effect. Similarly a plot was produced for different values of Δm_{23}^2. Next, the expected event rate, $\lambda(\mathbf{u})$ was computed using the oscillation probability as shown in Eq. 6.

$$\lambda(\mathbf{u}) = \phi \times P(\mathbf{u}) \tag{6}$$

where ϕ is the unoscillated event rate given in the data file. The initial values mentioned above was used to calculate the 'adjusted' λ. A histogram of 'adjusted' λ against E was plotted along with the observed event rate, k and the unoscillated event rate, ϕ. The difference in these plots was studied.

B. One-dimensional Minimisation

Firstly, one-dimensional minimisation was performed on *NLL* to estimate θ_{23} while Δm_{23}^2 was fixed at $2.4 \times 10^{-3} eV^2$. For this, a *NLL* function was defined to take θ_{23} as the input. Since there are bins, i with k_i being zero, the NLL was modified to to avoid $log(0)$ in the calculation of Eq. 5. A plot of *NLL* against θ_{23} was plotted to determine the approximate position of the minimum. It was found *NLL* had two minima.

For the minimisation procedure, a parabolic minimiser was coded. In this procedure, a general function y of x near its minimum was approximated to a parabola [3]. To start off, a function, named x_{min} was created with an input of y and a list which should contain three x points which are close to the minimum; x_0, x_1, x_2. The corresponding values of y were calculated; y_0, y_1, y_2. The points were then interpolated using 2nd-order Lagrangian Polynomial given by [3]

$$x_{min} = \frac{1}{2} \frac{(x_2^2 - x_1^2)y_0 + (x_0^2 - x_2^2)y_1 + (x_1^2 - x_0^2)y_2}{(x_2 - x_1)y_0 + (x_0 - x_2)y_1 + (x_1 - x_0)y_2}. \tag{7}$$

The function x_{min} returned the minimum of the parabola that contained all three points. Another function called *minimiser* was created with inputs of y and a list, similar to the situation above. The x_{min} function was used to find the minimum of the guess list. The x which gave the highest y in the list was replaced by the output of x_{min} and thus output a new list. The minimisation procedure was repeated on this new list using the 'while' loop until the difference in subsequent minimum x_{min}'s was less than 1×10^{-5}.

To test the minimisation method, it was implemented on the oscillation probability function P to find E which minimised P. A plot of P against E helped me to find an appropriate list of guesses, which was [0.6, 0.7, 0.65]. In addition to that, to ensure the method had converged to the desired minimum, a plot of P against the number of steps in the 'while' loop was plotted. The method had converged only if the P is constant after a few steps. To further verify the minimum obtained was correct, the optimizer routine from scipy was used to determine the minimum independently and this validation was done on all of minimisation result in this analysis.

The method was then performed on *NLL* to find the minimising θ_{23}. The guess lists used for first and second minima were [0.56, 0.7, 0.2] and [0.85, 0.9, 0.975] respectively. The error in the minimising value was found by considering the θ_{23} values that correspond to *NLL* changed by 0.5 compared to the value at the minimum; θ_{23}^+ in the positive direction and θ_{23}^- in the negative direction. Since the inverse of *NLL* could not be determined analytically, I made a list of θ_{23} and the corresponding *NLL* with high density so it approximates a continuous function well. Next, the *NLL* list was scanned for $(NLL_{min} + 0.5)$ within a difference of 0.01 and it returned a list of *NLL* which are close to $(NLL_{min} + 0.5)$. The minimum of the list was chosen as the *NLL* of θ_{23}^- and the maximum as the *NLL* of θ_{23}^+. The indices of the values were noted so that the θ_{23}^- and θ_{23}^+ could be obtained. The difference between these values and the true minimum, $\Delta\theta_{23}$ was the error. Instead of *NLL*, the curvature of the last parabolic estimate was used to determine θ_{23}^- and θ_{23}^+ separately. From the parabolic method, the last list of three points which best approximated the minimum was obtained. A parabolic function which could interpolate all three points was defined. The procedure was repeated with this new function and the corresponding θ_{23}^- and θ_{23}^+ were determined. The results from these two functions were compared. Finally, I chose the *NLL* function to best represent the error in this case, hence made an automatic function to calculate the standard deviation, which is the error. Since the positive and negative errors were close to each other, the average of those two was taken as the final error.

C. Two-dimensional Minimisation

The oscillation probability P depends on both θ_{23} and Δm_{23}^2. Hence, a two-dimensional fit was used to extract both parameters. The NLL_{2D} function was modified to take θ_{23} and Δm_{23}^2 as inputs and a contour plot was produced to study the behaviour of the function. From the plot, it was found the modified NLL_{2D} had two global minima as well. Initially, I implemented the 'univariate' method, which is essentially the parabolic minimiser in each direction successively and iterating. Two x_{min} and two *minimiser* functions were defined, each for one direction; θ_{23} and Δm_{23}^2. These two *minimiser* functions would be used in another function called $univariate$ to perform the univariate method, starting the minimisation along the θ_{23} direction first since there are two minima in that direction. Then, the result of that minimisation would be the argument for θ_{23} in the minimisation in Δm_{23}^2 direction and so on. The iteration would continue till the difference in the θ_{23} is less than 1×10^{-5} and similarly for Δm_{23}^2, less than $1 \times 10^{-7} eV^2$. The guess list used for Δm_{23}^2 was [0.003, 0.002, 0.0025], which was obtained by looking at the plot.

Since the univariate method was inefficient, a simultaneous minimisation method was attempted to extract the parameters where where both minimisation would happen simultaneously instead of each direction successively. For this purpose, the gradient descent method was chosen where the path would follow the steepest descent towards the minimum. Firstly, the gradient at an initial point was calculated. A function called $grad$ was defined to find the local gradient of f using a finite forward-difference approximation. For example, the gradient of f at x_1 is shown by

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1 + \Delta, x_2) - f(x_1, x_2)}{\Delta} \tag{8}$$

where x_2 is the second argument of f [3]. The same was done for the gradient with respect to x_2 so the $grad$ function outputted a vector of gradient, $\boldsymbol{d}(\boldsymbol{x}_n)$. The finite difference, Δ chosen was 1×10^{-5} in x_1 and 1×10^{-7} in x_2. A function named $gradminimiser$ was defined to perform the gradient method using the iterative function

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{\alpha} \cdot \boldsymbol{d}(\boldsymbol{x}_n) \tag{9}$$

where $\boldsymbol{x}$ is the vector [x_1, x_2] and $\boldsymbol{\alpha}$ is the step size of each descent [3]. $\boldsymbol{\alpha}$ here was given as a vector to allow different values of α in each direction, since the value of x might be smaller than the α, which should be accounted for. Large α might overshoot the minimum and cause the algorithm to not converge. Instead of setting up a difference parameter, I had the iteration run for a number of steps and had this as one of the input of $gradminimiser$. This was so that I could plot the NLL_{2D} as a function of number of steps to describe if and how soon the iteration had converged to the minimum.

To validate the 2-d minimiser, I defined an analytical 2-d function z:

$$z = 2x^2 + x + 3y^2 + 3y + 4. \tag{10}$$

A contour plot was made to highlight the global minimum. The global minimum was then calculated analytically to be (-0.25, -0.5). The gradient minimiser was implemented on this validation function for 20 steps and the result was checked against the analytic result. $\boldsymbol{\alpha}$ used was [0.01, 0.01] and the list of initial guess was [-0.2, 0]. A plot of NLL_{2D} as a function of number of steps was plotted as well. Then, the gradient minimiser was implemented on NLL_{2D} to extract θ_{23} and Δm^2_{23}. The number of iterations was 20. The $\boldsymbol{\alpha}$ was [1×10^{-4}, 1×10^{-9}] and the lists of initial guesses for two minima were [0.6, 2.4×10^{-3}] and [0.9, 2.4×10^{-3}] respectively. The result was cross-checked with the result from the univariate method. The error in each values was calculated in similar way as the one-dimensional parabolic minimiser. Though, an important assumption made here was that the error in one parameter was independent of the error of the other.

D. Neutrino Interaction Cross-section

In reality, the expected rate of neutrino events $\lambda(\mathbf{u})$ is also dependent on the neutrino interaction cross section [2]. Initially, the cross section scaling factor, σ was assumed to be a dimensionless constant for all values of energies E. In fact, the cross section scaling factor would be linearly dependent on E as shown by

$$\sigma = \gamma E \tag{11}$$

where γ is the rate of increase of σ and is assumed to be positive with the dimension of GeV^{-1}. Thus, the updated expected rate of neutrino events $\lambda(\mathbf{u})$ is

$$\lambda(\mathbf{u}) = \phi \times P \times \gamma E \tag{12}$$

where $\mathbf{u}$ is [θ_{23}, Δm^2_{23}, γ]. This new $\lambda(\mathbf{u})$ along with an appropriate NLL_{3D} functions were defined. A plot of NLL_{3D} against γ was produced to obtain the approximate value of γ. The previous minimisation result for θ_{23} and Δm^2_{23} were used as initial guess. Then, the gradient method was implemented to extract $\mathbf{u}$. To do this, appropriate $grad$ and $gradminimiser$ functions were defined. They were just modified versions of previously defined ones but were made to include γ. The $\boldsymbol{\alpha}$ chosen was [1×10^{-4}, 1×10^{-9}, 1×10^{-3}], while the number of iterations was 30. The guesses used for two global minima, as previously, were [0.6, 2.4×10^{-3}, 0.6] and [0.9, 2.4×10^{-3}, 0.6] respectively. The errors in these three parameters were determined similarly to the 2-d minimisation case with the same assumption.

IV. Results

A. The Data

Figure 1 shows the histogram of muon neutrino events observed in the bins of energies E. It shows the maximum value for events is around 40, indicating the experiment was statistically limited indeed. Hence it supports the decision of choosing Poisson distribution for the likelihood fit. Next, the effect of changing θ_{23} and Δm^2_{23} on the oscillation probability P can be seen on figure 2 and figure 3. The plots show Δm^2_{23} affects the frequency of oscillation whereas θ_{23} affects the amplitude of the oscillation. The amplitude was the maximum when $\theta_{23} = \pi/4$. Both plots also show that the oscillation dominates in the lower energy region. Following the definition of $\lambda(\mathbf{u})$, the plot of the 'adjusted' event rate as a function of E was produced as shown in figure 4. The significant deviation of the unoscillated flux from the observation shows a strong proof of the neutrino oscillation. Even with a guess for the parameters, the 'adjusted' curve follows the observation reasonably well, indicating our oscillation probability function P was sensible. Besides, it suggests the initial guesses were not so far away from the true values. However, the observed events at higher energies were not quite captured by P and it will be tackled later on.

B. One-dimensional Minimisation

The plot of *NLL* as a function of θ_{23} alone is shown by figure 5. It clearly shows the function has two minima in the chosen range, at around 0.7 and 0.9 radians and they are symmetric on the line $\theta_{23} = \pi/4$. The number of minima indicates the contribution of $sin^2(2x)$ term since it is symmetric on $x = \pi/4$ too. Once the minimiser had been set up, it was tested on the oscillation probability P. Figure 6 clearly shows the value of P is constant after the second step, indicating the algorithm has converged. The minimising E for P was 0.5782 GeV which corresponds to the minimum in figure 3 at θ_{23}= 0.785 rad. This result was cross-checked with

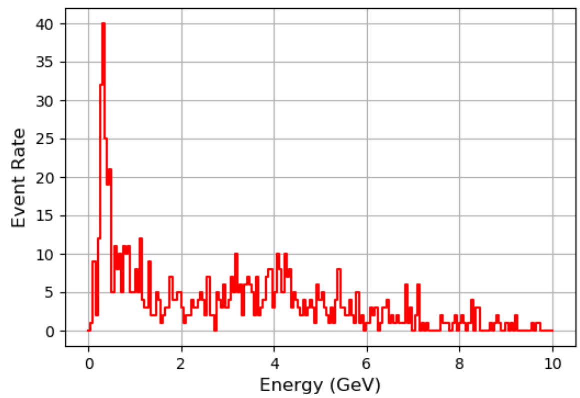

Fig. 1: The muon neutrino events per energy bin with a width of 0.05 GeV. The rate in this analysis means 'per bin'.

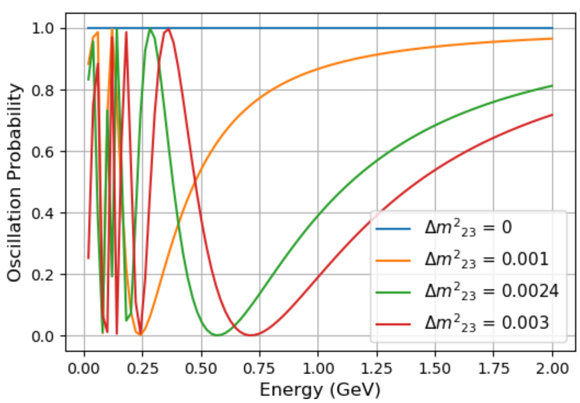

Fig. 2: Varying Δm_{23}^2 with a unit of eV^2. Only the low energy region is focused here since the plots approach $P = 1$ at higher energies.

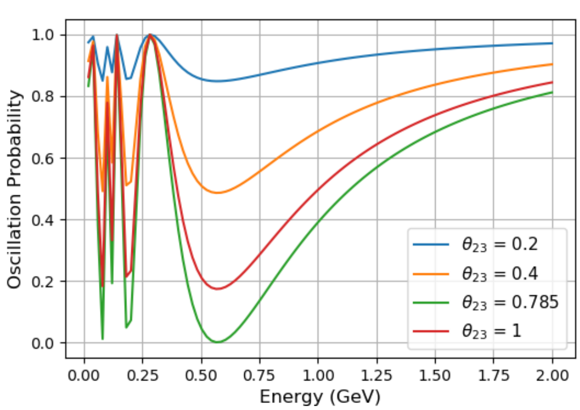

Fig. 3: Varying θ_{23} in radians.

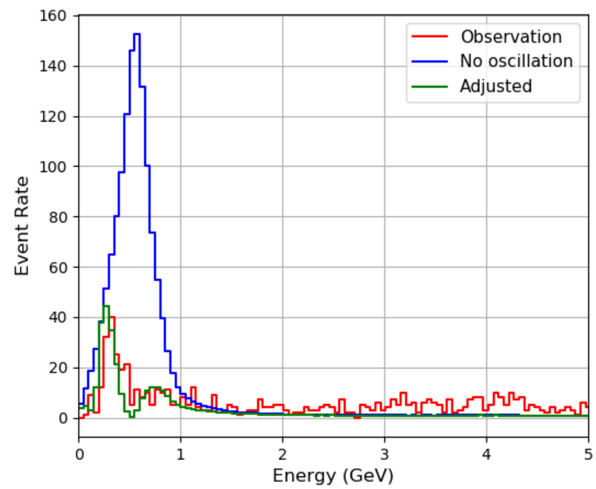

Fig. 4: The neutrino event distribution comparing the observation with the unoscillated flux and the 'adjusted' λ. The 'adjusted' fit is plotted as a histogram to highlight the discrete nature of the data.

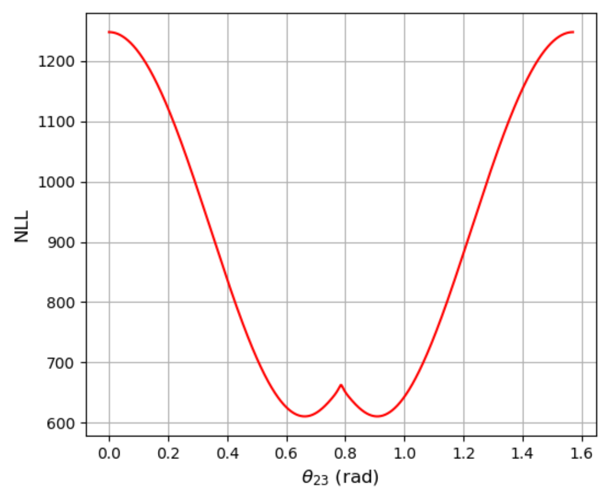

Fig. 5: The negative log-likelihood *NLL* as a function of θ_{23}.

scipy optimization routine and further validated the answer, hence, the algorithm.

The minimisation was then implemented on *NLL* and the result for both minima were 0.662635 and 0.90816 (rounded up to 6 decimal places). The results agree with the positions of the minima in figure 5. The plot in figure 7 shows the algorithm has converged to both the minima. Furthermore, the scipy routine returned the same result up to 4 decimal places. The accuracy was calculated as mentioned previously, and the θ_{23}^+ and θ_{23}^- were determined using *NLL* and the parabolic function. Both functions returned identical answers since the parabolic function fits *NLL* near the minimum to a very high degree. The *NLL* was chosen to represent the final error, $\Delta\theta_{23}$ in the two directions as shown on table I. The final result on the minimisation of *NLL* with respect to θ_{23} is shown on table II. The error on the measurement of θ_{23} should increase as the

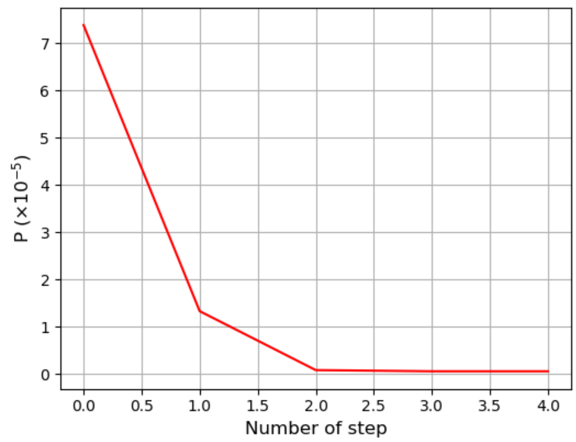

Fig. 6: The oscillation probability P as a function of the number of steps in parabolic minimisation method.

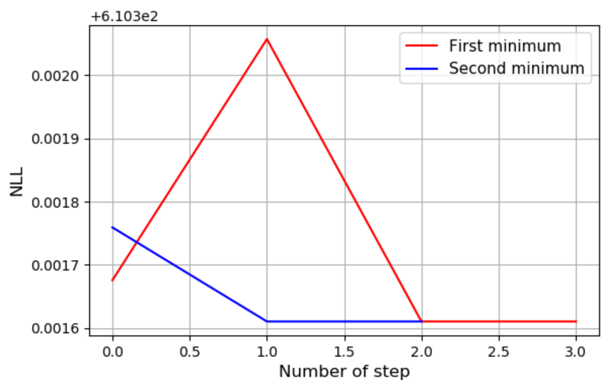

Fig. 7: The negative log-likelihood *NLL* as a function of the number of steps in parabolic minimisation method. The '+6.103e2' on the left indicates the values in the vertical axis should be added to 610.3. This occurred since the differences in the values were too small.

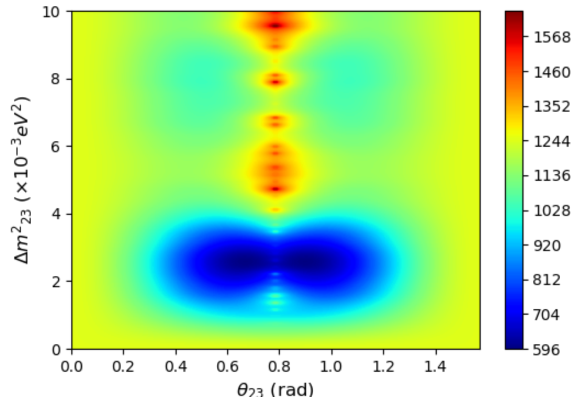

Fig. 8: Contour plot of two dimensional negative log-likelihood NLL_{2D} against θ_{23} and Δm^2_{23}. The colourbar on the right corresponds to the value of NLL_{2D}.

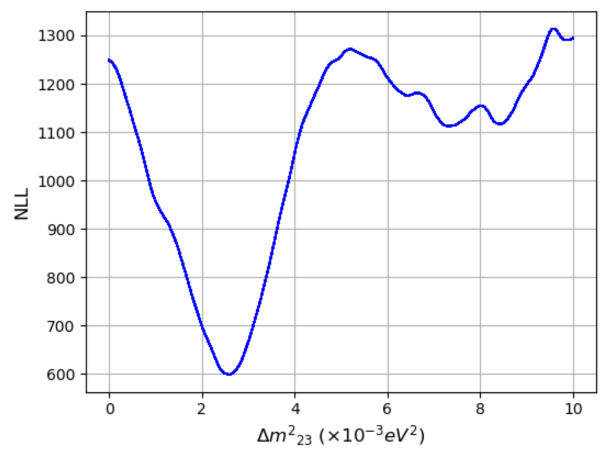

Fig. 9: NLL_{2D} as a function of Δm^2_{23} at a fixed θ_{23} = 0.67 rad.

true minimum approaches $\pi/4$ since the $(NLL + 0.5)$ value in one of the directions (positive or negative) could overshoot the value at $\pi/4$ as seen in figure 5. However, in our case, the true minima are reasonably further away from $\pi/4$.

Parameter	First Min	Second Min
$\Delta\theta^+_{23}$	0.011268	0.011309
$\Delta\theta^-_{23}$	0.011364	0.011314

TABLE I: The $\Delta\theta_{23}$ values in the positive and negative directions for both minima, rounded up to 6 decimal places.

Minimum	θ_{23} (rad)
First	0.663 ± 0.011
Second	0.908 ± 0.011

TABLE II: The θ_{23} values with their uncertainties in both minima.

C. Two-dimensional Minimisation

The contour plot of NLL_{2D} against θ_{23} and Δm^2_{23} in figure 8 shows there are two global minima which corresponds to the minima of both parameter simultaneously. Figure 9 shows the position of minima with respect to Δm^2_{23} more clearly. The results from the univariate method is shown in table III. The minimising θ_{23} is indeed slightly different from the result from one-dimensional minimisation in table II. The univariate method's result agrees with scipy's routine up to 4 decimal places.

Parameter	First Minimum	Second Minimum
θ_{23} (rad)	0.670	0.901
Δm^2_{23} ($10^{-3} eV^2$)	2.591	2.591

TABLE III: The result from the univariate method rounded up to 3 decimal places for both minima.

The gradient descent method was then validated by implementing it on the function in Eq. 10. Figure 10 shows the contour plot of the function, z. In figure 11 after 5 steps or so, the algorithm has converged to the minimum point of [-0.25000317, -0.5] which is close to the analytical result up to 5 decimal places. This proves the gradient descent routine works as expected. The minimisation result of NLL_{2D} is shown on table IV. The uncertainty of θ_{23} is the same as the one

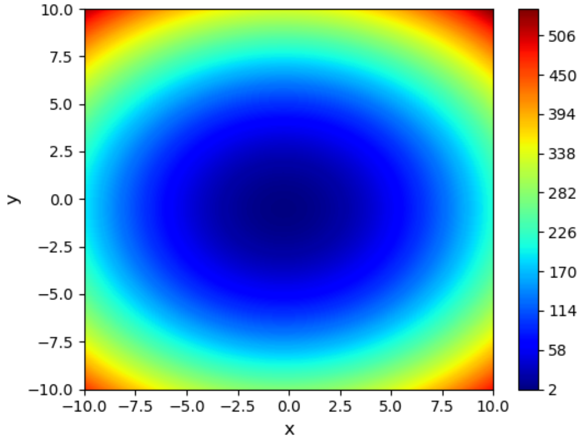

Fig. 10: Contour plot of validation function z in Eq. 10 with respect to x and y.

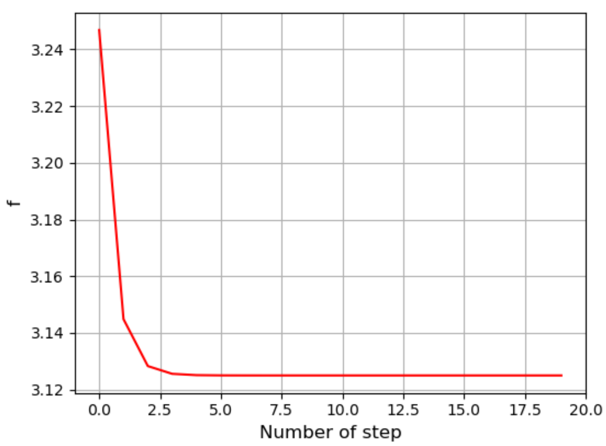

Fig. 11: z as a function of the number of steps in gradient descent method.

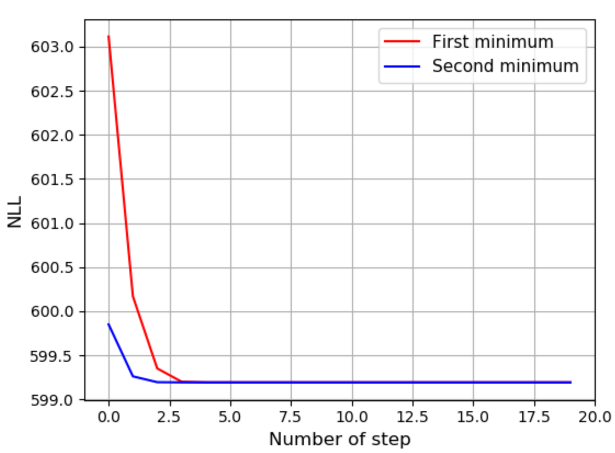

Fig. 12: NLL_{2D} as a function of the number of steps in gradient descent method. The convergence occurs after around 3rd step.

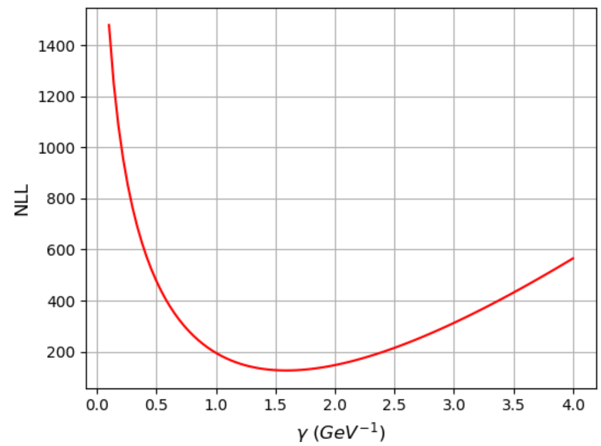

Fig. 13: The three-dimensional negative log likelihood NLL_{3D} as a function of γ at fixed values of θ_{23} = 0.67 rad and Δm^2_{23} = 0.00259 eV^2.

from one-dimensional minimisation. The values Δm^2_{23} in both minima are identical since there is only one global minima of NLL_{2D} with respect to Δm^2_{23} alone. The plot in figure 12 further validates its convergence. The result is also identical to the one from univariate method, since both should converge to the correct minima. However, the gradient descent method is significantly faster thus more efficient than the univariate method.

Parameter	First Minimum	Second Minimum
θ_{23} (rad)	0.670 $\pm$ 0.011	0.901 $\pm$ 0.011
Δm^2_{23} ($10^{-3}eV^2$)	2.591 $\pm$ 0.035	2.591 $\pm$ 0.035

TABLE IV: The result from the gradient descent method.

D. Neutrino Interaction Cross-section

Figure 13 shows the NLL_{3D} against γ with a fixed θ_{23} and $\Delta {m^2}_{23}$. The approximate position of the minimum is on about γ = 1.5 GeV^{-1}. More importantly, the likelihood value at the minimum is less than the value found using the parameters in the two-dimensional minimisation from figure 8. This indicates the observed events are better fitted under the consideration of the cross section. The figure 14 indicates 3-d minimisation converges slower than 2-d one. Table V displays the results from the gradient descent method on NLL_{3D} along with their uncertainties. The values agree up to 4 decimal places with the results from scipy's routine. The uncertainties in these values are very slightly larger than the two dimensions case above. The resulting parameters, $\mathbf{u}$ were used to plot $\lambda(\mathbf{u})$ along with the previous results as shown in figure 15. It can clearly be seen the cross section scaling accounts for the deviation at around E = 4 GeV very well, unlike the previous two minimisations. There are fluctuations from the observed data around the best fit due to the nature of the data being statistically small. The gradient descent method for this case takes considerably longer time to run as well, due to the increased number of dimensions.

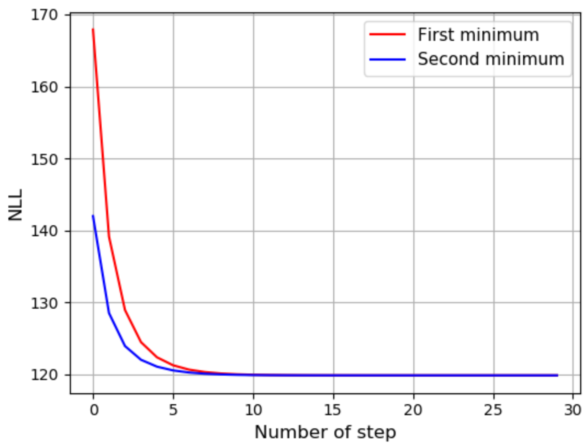

Fig. 14: NLL_{3D} as a function of the number of steps in gradient descent method. The convergence occurs after around 7th step.

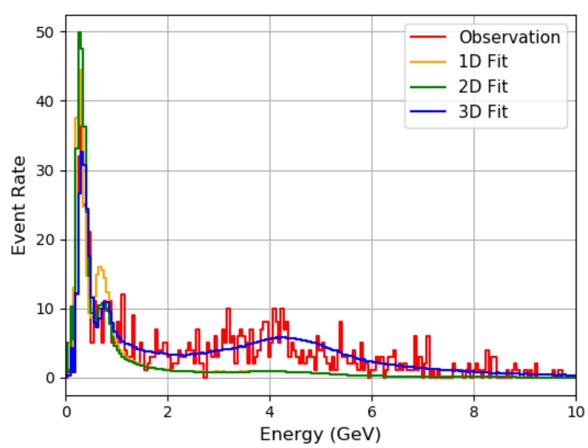

Fig. 15: The neutrino event distribution. 1D fit is the case where only θ_{23} was allowed to vary, while Δm^2_{23} was fixed. 2D fit is the case where both θ_{23} and Δm^2_{23} varied simultaneously, whereas 3D fit corresponds to the case where γ was considered as well.

Parameter	First Minimum	Second Minimum
θ_{23} (rad)	0.675 $\pm$ 0.012	0.896 $\pm$ 0.012
Δm^2_{23} (10^{-3}eV^2)	2.727 $\pm$ 0.038	2.727 $\pm$ 0.038
γ (GeV^{-1})	1.601 $\pm$ 0.059	1.601 $\pm$ 0.059

TABLE V: The result from the gradient descent method on NLL_{3D}.

V. Conclusion

A set of simulated T2K data was analysed to extract the parameters of muon neutrino oscillation. A negative log-likelihood fit with a Poisson probability density function was used. The parameters were the 'mixing' angle, θ_{23}, difference between the squared masses of the two neutrinos, Δm^2_{23} and the cross section scaling constant, γ. Various numerical methods such as parabolic minimiser and gradient descent method were used to determine the minimum of the log-likelihood function which best fit the data. The values of the parameters were θ_{23} = (0.675 $\pm$ 0.012) rad and (0.896 $\pm$ 0.012) rad, Δm^2_{23} = (2.727 $\pm$ 0.038) 10^{-3}eV^2, γ = (1.601 $\pm$ 0.059) GeV^{-1}. Overall, this result proves the neutrinos do oscillate and they have non-zero masses.

The algorithms implemented were relatively fast since appropriate choices were made for the initial guesses and the values for the parameters of the minimising routine, such as the step size in gradient descent method. To further improve the efficiency, more sophisticated minimisers such as Newton's method can be implemented, however this does not necessarily return a more accurate answer.

References

[1] Barger, V., Marfatia, Danny, and Whisnant, Kerry Lewis. *The Physics of Neutrinos*. Princeton: Princeton UP, 2012. Print.

[2] Yoshi Uchida and Mark Scott. *Project 1: A log-likelihood fit for extracting neutrino oscillation parameters*. Third Year Computational Physics, Imperial College London, 2019.

[3] Yoshi Uchida and Mark Scott. *Chapter 7 Minimisation and Maximisation of Functions Lecture Notes*. Third Year Computational Physics, Imperial College London, 2019.