Likelihood-Based Determination of Neutrino Oscillation Parameters using Simulated T2K Data

Max Hart

Abstract – We begin with a brief review of the theory behind muon-tau neutrino oscillations, before setting up the extraction of the oscillation parameters as a negative-log-likelihood (NLL) minimisation problem. We then go on to discuss various direct-search and Monte-Carlo minimisation methods whose implementations are validated and compared using test functions. The most suitable among these methods are then used to minimise the trivariate NLL to obtain the mixing angle, difference in squared mass and rate of increase of cross section with energy as $\theta_{23} = 0.818 \pm 0.030$, $\Delta m_{23}^2 = (2.526 \pm 0.029) \times 10^{-3}$ eV^2 and $\alpha = 1.416 \pm 0.059$ respectively. Univariate and bivariate parameter estimation with no cross section energy dependence is also carried out beforehand. The final trivariate estimates are then used to generate new data sets on which the minimisation process is again carried out on to obtain resampled parameters. The distribution of the resampled parameters with respect to the original parameter suggests that the small data set size results in the parameter α being over-fitted in value. After introducing an additional averaging step before minimising, the mean resampled value converged to the source value as expected. By studying the mean resampled value as the number of averaging steps is varied, we predict that increasing the original experimental data set size by a factor of ~ 4 would would reduce the fitting error in α by a factor of ~ 10.

INTRODUCTION

While there are numerous reasons which suggest that the standard model incomplete, the observation of neutrino oscillations is perhaps one of the most direct [1]. The standard model predicts that all three neutrino flavours should have exactly zero mass. However, for neutrino oscillations to occur, the neutrinos must have some non-zero, finite mass. Thus, observation of this phenomenon provides a crucial first glimpse into physics beyond the standard model. Here we will be using simulated data from the T2K experiment to extract estimates for various μ-τ neutrino oscillation parameters.

THEORY

Neutrino Oscillations

When neutrinos are produced they are, by definition, in a definite flavour eigenstate. However, since neutrinos must propagate as energy eigenstates, the flavour eigenstates will be mixed upon detection. If we restrict ourselves to a simpler case of two-state mixing between the μ and τ eigenstates ν_μ and ν_τ respectively, we can express their decomposition into energy eigenstates succinctly as

$$\begin{pmatrix} \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} \cos\theta_{23} & \sin\theta_{23} \\ -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix} \begin{pmatrix} \nu_2 \\ \nu_3 \end{pmatrix} \tag{1}$$

where θ_{23} is a parameter known as the mixing angle. Suppose a state is created as ν_μ at time $t = 0$, then at a later time t the wave function $\psi(t)$ will be

$$\psi(t) = \nu_2 \cos\theta_{23} e^{i\phi_2(t)} + \nu_3 \sin\theta_{23} e^{i\phi_3(t)} \tag{2}$$

where m_i and ϕ_i are the rest masses and phases accumulated by the μ and τ states respectively. If we then decompose this back into flavour eigenstates we get

$$\begin{aligned} \psi(t) &= (\cos^2\theta_{23} e^{i\phi_2(t)} + \sin^2\theta_{23} e^{i\phi_3(t)})\nu_\mu \\ &+ \sin\theta_{23}\cos\theta_{23}(e^{i\phi_3(t)} - e^{i\phi_2(t)})\nu_\tau \end{aligned} \tag{3}$$

from which we can see the probability of observing the state ν_μ to be

$$\begin{aligned} P(t) &= |\cos^2\theta_{23} e^{i\phi_2(t)} + \sin^2\theta_{23} e^{i\phi_3(t)}|^2 \\ &= 1 - \sin^2 2\theta_{23} \sin^2((\phi_3(t) - \phi_2(t))/2) \end{aligned} \tag{4}$$

To calculate the phase difference, we start by recalling that the phase accumulated by particle i is $\phi_i(t) = E_i t - p_i x$, then the phase difference is

$$\phi_3 - \phi_2 = (E_3 - E_2)t - (p_3 - p_2)x \tag{5}$$

To express p_i in terms of E_i we can make the reasonable assumption the neutrinos are relativistic, i.e. $m_i/E_i << 1$ to give

$$p_i = \sqrt{E_i^2 - m_i^2} = E_i\sqrt{1 - m_i^2/E_i^2} \approx E_i(1 - m_i^2/E_i^2) \tag{6}$$

Assuming that $E_2 = E_3 = E$ and $t = L/c$, where L is the distance between the source and detector, we find

$$\phi_3 - \phi_2 = \frac{\Delta m_{23}^2 L}{2E} \tag{7}$$

where Δm_{23}^2 is the difference between the square of the neutrino masses. Substituting this back into (4), we finally get

$$P(E) = 1 - \sin^2(2\theta_{23})\sin^2\left(\frac{\Delta m_{23}^2 L}{4E}\right) \tag{8}$$

which can be written in a more useful form as

$$P(E) = 1 - \sin^2(2\theta_{23})\sin^2\left(\frac{1.267\Delta m_{23}^2 L \times 10^{-3}}{4E}\right) \tag{9}$$

where L is given in km, Δm_{23}^2 in eV$^2 \times 10^{-3}$ and E in GeV.

Detection Rates

To investigate neutrino oscillations we can measure the number of μ-neutrinos which make it to a detector after being produced at a pure μ-neutrino source some distance away. Given the random and uncorrelated nature of detections, the rate of un-oscillated neutrinos reaching the detector with an energy belonging to the i^{th} energy bin centred at E_i will follow a Poisson distribution $\mathcal{P}_i$ with a mean

$$\lambda_i = P_i \Phi_i \tag{10}$$

where Φ_i is the simulated rate of ν_μ detections in the energy bin E_i without any oscillations, assuming the cross section σ is constant with energy. If we make the more physical assumption that the cross section scales linearly with energy, (10) will instead have the form

$$\lambda_i = P_i \Phi_i \alpha E_i \tag{11}$$

where α is a dimensionless constant related to the rate of increase of σ with E.

Maximum Likelihood Estimation

To extract the neutrino oscillation parameters from our data we will proceed by using the process of maximum likelihood estimation. Let Λ_i denote the observed number of neutrinos in the energy bin E_i. The maximum likelihood principle then states that the parameters $\hat{\theta_{23}}, \hat{\Delta m^2_{23}}$ and $\hat{\alpha}$ which maximise the likelihood

$$\mathcal{L}(\vec{\Lambda}|\vec{\lambda}) = \prod_{n=i}^{N} \mathcal{P}_i(\Lambda_i|\lambda_i) \tag{12}$$

will be the best estimate for the true parameters. Monotonicity of logarithms means that this is equivalent to the often easier task of minimising the negative log-likelihood

$$\mathcal{N}(\vec{\Lambda}|\vec{\lambda}) = -\sum_{i=1}^{N} \log \mathcal{P}_i(\Lambda_i|\lambda_i) \tag{13}$$

By subbing in $\mathcal{P}_i(\Lambda_i|\lambda_i) = \lambda_i^{\Lambda_i} e^{-\lambda_i}/\Lambda_i!$ it can be shown that

$$\mathcal{N}(\vec{\Lambda}|\vec{\lambda}) = \sum_{i=1}^{N} \left[\lambda_i - \Lambda_i + \Lambda_i \log\left(\frac{\Lambda_i}{\lambda_i}\right)\right] \tag{14}$$

where it has been left implicit that the λ will depend on the fitting parameters $\{\theta_{23}, \Delta m^2_{23}, \alpha\}$ along with L and $\{\Phi_i, E_i\}$.

METHODOLOGY & IMPLEMENTATION

The data set we will be using consists of the number of observed μ-neutrinos in energy bins of width 0.05 GeV, over a energy range of 0→10 GeV, giving a total of 200 bins. To estimate the desired parameters we numerically minimise (14) using the various algorithms discussed below.

Parabolic Methods

Although we know in reality our data is parameterised by three variables when accounting the energy dependence of σ, we can begin with the constant σ case and minimise with respect to θ_{23} while holding Δm^2_{23} constant.

A simple minimisation algorithm for a 1D function $f(x)$ is the parabolic method. Here we start with three points $\{(x_0, f_0), (x_1, f_1), (x_2, f_2)\}$ to which we fit a quadratic Lagrange polynomial which will have a minimum (x_3, f_3) given by

$$x_3 = \frac{1}{2} \frac{(x_2^2 - x_1^2)f_0 + (x_0^2 - x_2^2)f_1 + (x_1^2 - x_0^2)f_2}{(x_2^2 - x_1)f_0 + (x_0 - x_2)f_1 + (x_1 - x_0)f_2} \tag{15}$$

Out of these four points we then discard the point which has the lowest corresponding f_i, and repeat the process on the remaining three points. The value of x_3 will then converge to a local minimum as desired.

This method can be extended to higher dimensional minimisation problems. To do so we minimise with respect to the first coordinate whilst holding the others constant, move to that minimum, and then minimise with respect to the second coordinate and move to its minimum, and so on. While this method is intuitive, implementing it in higher dimensions becomes increasingly complex, and it is prone to becoming less efficient as the gradient ∇f becomes misaligned with the coordinate system's axes.

Gradient Descent Methods

A more appropriate choice for simultaneous minimisation of θ_{23} and Δm^2_{23} is the gradient method. Suppose we want to minimise the n-dimensional function $f(\vec{x})$. The gradient method then works by taking small steps of size α against the gradient $\nabla f(\vec{x})$ such that it follows the path of steepest descent to the local minimum. The iterative scheme is then

$$\vec{x}_{n+1} = \vec{x}_n - \alpha \vec{\nabla} f(\vec{x}_n) \tag{16}$$

Not only does this method mitigate issues of the aforementioned higher-dimensional parabolic methods, but it also has the added benefit of being trivial to extend to the minimisation of higher-dimensional functions.

Newton & Quasi-Newton Methods

We can improve upon the efficiency of the gradient descent method by also considering the curvature of $f(\vec{x})$ in addition to its gradient. If we suppose the minimum is a displaced $\vec{\delta}$ from the current point $\vec{x}$ it can be shown that up to quadratic order

$$\mathbf{H}(\vec{x}) \cdot \vec{\delta} = -\vec{\nabla} f(\vec{x}) \tag{17}$$

where $\mathbf{H}(\vec{x})$ is the Hessian matrix with elements $H_{ij}(\vec{x}) = \partial^2 f(\vec{x})/\partial x_i x_j$. This then leads to the iterative scheme

$$\vec{x}_{n+1} = x_n - \gamma[\mathbf{H}(\vec{x}_n)]^{-1} \cdot \vec{\nabla} f(\vec{x}_n) \tag{18}$$

where we have included a 'damping factor' γ to allow us to control the step size for stability purposes. Although this method has the advantage of quadratic convergence, this comes at the expense of $\mathcal{O}(n^3)$ operations being required to invert the Hessian at each iteration.

There exists a class of schemes known as quasi-Newton methods, which avoid this costly inversion by approximating $\mathbf{H}(\vec{x}_n)^{-1} = \mathbf{H}_n^{-1} \approx \mathbf{G}_n$ to give the iteration scheme

$$\vec{x}_{n+1} = \vec{x}_n - \alpha \mathbf{G}_n \cdot \vec{\nabla} f(\vec{x}_n) \tag{19}$$

If we define

$$\vec{\delta}_n = \vec{x}_{n+1} - \vec{x}_n \tag{20}$$

and

$$\vec{\gamma}_n = \vec{\nabla} f(\vec{x}_{n+1}) - \vec{\nabla} f(\vec{x}_n) \tag{21}$$

it can be shown that to linear order,

$$\mathbf{G}_n \cdot \vec{\gamma}_n = \vec{\delta}_n \tag{22}$$

which will play the role of the update matrix. There are numerous different ways of updating $\mathbf{G}_n \rightarrow \mathbf{G}_{n+1}$, each with differing efficiencies and convergence characteristics. The most common of these are listed below.

Davidon-Fletcher-Powell (DFP) [2]

$$\mathbf{G}_{n+1} = \mathbf{G}_n + \frac{(\vec{\delta}_n \otimes \vec{\delta}_n)}{\vec{\gamma}_n \cdot \vec{\delta}_n} - \frac{\mathbf{G}_n \cdot (\vec{\delta}_n \otimes \vec{\delta}_n) \cdot \mathbf{G}_n}{\vec{\gamma}_n \cdot \mathbf{G}_n \cdot \vec{\gamma}_n} \tag{23}$$

Broyden–Fletcher–Goldfarb–Shanno (BFGS) [3]

$$\mathbf{G}_{n+1} = \left(\mathbf{I} - \frac{(\vec{\delta}_n \otimes \vec{\gamma}_n)}{\vec{\delta}_n \cdot \vec{\gamma}_n} \right) \mathbf{G}_n \left(\mathbf{I} - \frac{(\vec{\gamma}_n \otimes \vec{\delta}_n)}{\vec{\delta}_n \cdot \vec{\gamma}_n} \right) + \frac{\vec{\delta}_n \otimes \vec{\delta}_n}{\vec{\gamma}_n \cdot \vec{\delta}_n} \tag{24}$$

Symmetric-Rank 1 (SR1) [2]

$$\mathbf{G}_{n+1} = \mathbf{G}_n + \frac{(\vec{\delta}_n - \mathbf{G}_n \vec{\gamma}_n) \otimes (\vec{\delta}_n - \mathbf{G}_n \vec{\gamma}_n)}{(\vec{\delta}_n - \mathbf{G}_n \vec{\gamma}_n) \cdot \vec{\gamma}_n} \tag{25}$$

Simulated Annealing

Simulated annealing [4] is a type of Monte-Carlo method in which at each iteration a next step is proposed from a generating PDF $g(\vec{x})$ such that $\vec{x}_{n+1} = \vec{x}_n + \vec{g}(\vec{x}_n)$. We then accept this step with a probability

$$p_{accept} = \begin{cases} 1 & \Delta E \leq 0 \\ e^{-\Delta E / T} & \Delta E > 0 \end{cases} \tag{26}$$

where $\Delta E = f(\vec{x}_{n+1}) - f(\vec{x}_n)$ and T is a constant less than unity which is lowered as the path progresses. The main advantage of this algorithm over the direct search methods is that the non-zero acceptance probability for $\Delta E > 0$ means that the trajectory is able to become 'unstuck' from local minima.

Verifying a Minimum Point

Several of the methods we have discussed simply search for points where the gradient is zero and do not discriminate between maxima, minima and saddle points. To distinguish between these we can note that for $\vec{x}$ to be a minimum then the Hessian matrix $\mathbf{H}(\vec{x})$ must be strictly positive definite. This is equivalent to the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ of $\mathbf{H}(\vec{x})$ all being strictly positive. To find the the eigenvalues of $\mathbf{H}(\vec{x})$ we employ a an explicit algorithm described in [5].

Finite Difference Methods

Almost all of the methods we have mentioned involve the calculation of derivatives. By using central difference schemes we can approximate first and section partial derivatives as

$$\frac{\partial f(\vec{x})}{\partial x^i} = \frac{f_{i+1} - f_{i-1}}{2h} + \mathcal{O}(h^2) \tag{27}$$

and

$$\frac{\partial^2 f(\vec{x})}{\partial x^i x^j} = \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4hk} + \mathcal{O}(h^2) \tag{28}$$

respectively [6]. Here $f_{i\pm1,j\pm1}$ denotes the value of f evaluated at steppings of $\pm h$ and $\pm k$ away from the central point $\vec{x}$ in the $\vec{x}^i$ and $\vec{x}^j$ directions respectively. Since the calculated minimum will be insensitive to differing initial starting point, we need not consider the effect of error accumulation when choosing an optimum value of h.

Validation

To verify that we have correctly implemented these minimisation algorithms into code, we can apply them to functions whose minima are known analytically and compare the answers. A popular choice is the Rosenbrock function [7] defined as

$$f(\vec{x}) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \tag{29}$$

where the x_i are components of the N-vector $\vec{x}$. If we apply the previously discussed minimisation methods to this function for the 2D case we obtain the paths shown in figure 1.

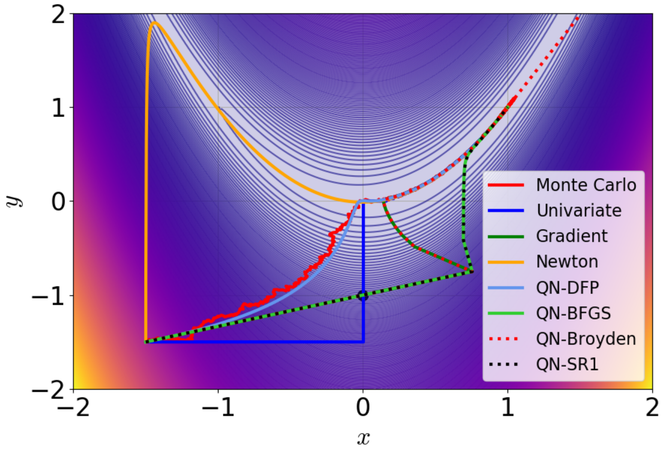

Fig. 1: A contour plot showing the paths taken by various minimisation algorithms from the starting point $(-1.5, -1.5)$ to the global minimum $(1, 1)$. The gradient and Broyden methods share similar paths, as do the BFGS and SR1 routines. This function is demonstrative of how the Newton and univariate methods can become rather inefficient under unfavourable conditions. Here we have deliberately removed the end sections of the DFP and Broyden paths as their divergent behaviour causes clutter otherwise.

All of the methods found the minimum successfully, bar the DFP and Broyden quasi-Newton methods which had a tendency to become unstable. Although the latter two were successful for other functions, for the Rosenbrock function

Method	Minimum	Iterations	Time	Loop Time
Univariate	Yes	4650	321	0.069
Gradient	Yes	25480	403	0.016
Newton	Yes	270	17	0.063
DFP	No	/	/	/
BFGS	Yes	16419	900	0.055
Broyden	No	/	/	/
SR1	Yes	16482	639	0.039
Monte Carlo	Yes	10,000	193	0.019

Table I: A table summarising the number of iterations and time taken for each algorithm when minimising the 2D Goldstein-Price function as illustrated in figure 1. The starting point for all of the algorithms was (-1.5, -1.5). All algorithms which converged were set to terminate after reaching an accuracy of $\mathcal{O}(10^{-8})$.

they required the step size to be exceedingly small to ensure stability, which prevented them from reaching the minimum in any reasonable number of iterations. The paths taken by the methods tested are shown in figure 1, and a summary of their performance is given in table I. When carrying out a similar test for the 3D Rosenbrock function we see similar results, except the Broyden method was now stable and the DFP/SR1 methods unstable. When applied to the NLL function the DFP method was stable. Since all methods have been seen to be consistent for at least one function we can be confident they have been implemented correctly. The fact that not all of them are universally stable highlights how we muct dynamically choose the correct method for the function in question. A detailed plot and table for the 3D testing has been omitted for brevity.

Results, Validation & Discussion

Univariate Minimisation

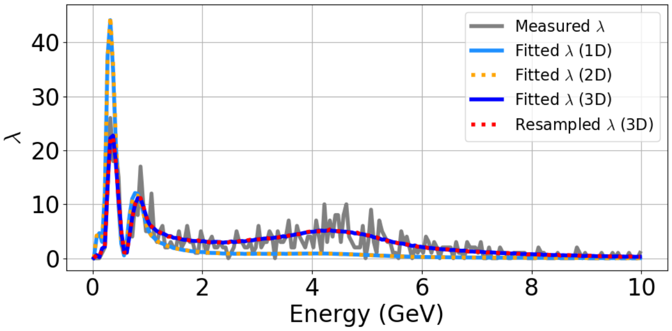

Fig. 2: A plot showing the number of un-oscillated neutrinos λ detected in each energy bin over an energy range of 0→200 GeV. The experimental profile is shown along with the predicted profiles obtained by subbing the values from table II into (10) and (11). The predicted profile obtained from using the mean resampled values with 100 averaging steps is also shown, and can be seen to be in good agreement with the blue profile from which the resampled data was generated.

We begin by minimising with respect to θ_{23} only, leaving $\Delta m^2_{23} = 2.4$ eV$^2 \times 10^{-3}$ and ignoring cross section energy dependence. If we then apply the parabolic method we find the value which minimises (14) to be $\hat{\theta_{23}} = 0.815 \pm 0.014$, as shown in figure 3. When subbing these values into (10) we can see the produced λ_i are in rough agreement with what we would expect.

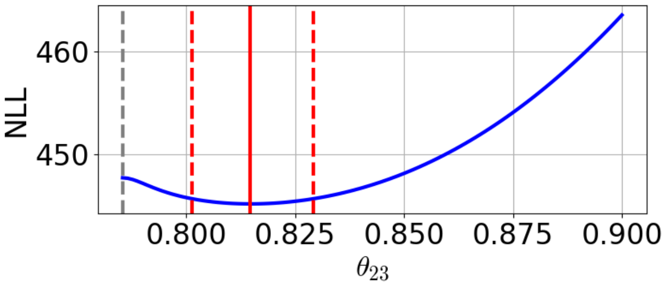

Fig. 3: A plot showing how the 1D NLL varies with θ_{23} when we set $\Delta m^2_{23} = 2.4\text{eV}^2 \times 10^{-3}$. The minimum of the NLL is shown by the solid red line while the uncertainty bounds are shown either side by the dashed red line. The NLL is symmetric about $\theta_{23} = \pi/4$ which is shown by the grey dashed line.

Bivariate Minimisation

Ignoring the cross-section energy dependence as before, we now minimise (14) with respect to both θ_{23} and Δm^2_{23} simultaneously. The 2D univariate, gradient descent, Newton, simmulated annealing and DFP methods all gave the same result of $\hat{\theta_{23}} = 0.815 \pm 0.014$ and $\Delta \hat{m}^2_{23} = 2.416 \pm 0.030$ eV$^2 \times 10^{-3}$, with their paths shown in figure 4. The choice of $\Delta \hat{m^2_{23}}$ for the 1D minimisation means that we see little improvement over the 1D case.

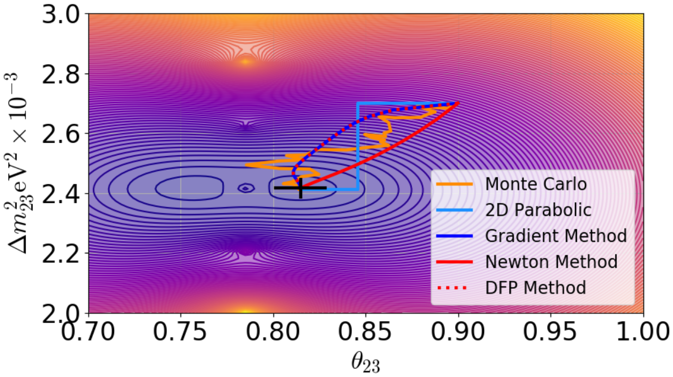

Fig. 4: A contour plot of the 2D NLL along with the paths taken by the various minimisers when starting from the point $(\theta_{23}, \Delta m^2_{23}) = (0.9, 2.5)$. Higher colour temperature indicates a larger NLL. We can again observe the symmetry of the NLL about the line $\theta_{23} = \pi/4$, along which we can also see repeated singularities as Δm^2_{23} is varied. The simulated annealing path is labelled 'Monte Carlo'.

Dimensions	$\hat{\theta_{23}}$	$\hat{m^2_{23}}$ eV$^2 \times 10^{-3}$	$\hat{\alpha}$
1D	0.815 ± 0.014	(2.4)	/
2D	0.815 ± 0.014	2.416 ± 0.030	/
3D	0.818 ± 0.030	2.526 ± 0.029	1.416 ± 0.059

Table II: A summary of how the parameter estimates change as we increase the number of parameters we allow to vary. Note that the 1D and 2D cases do not have entries for $\hat{\alpha}$ as these cases assume that the cross section remains constant with energy. The bracketed value was a fixed input.

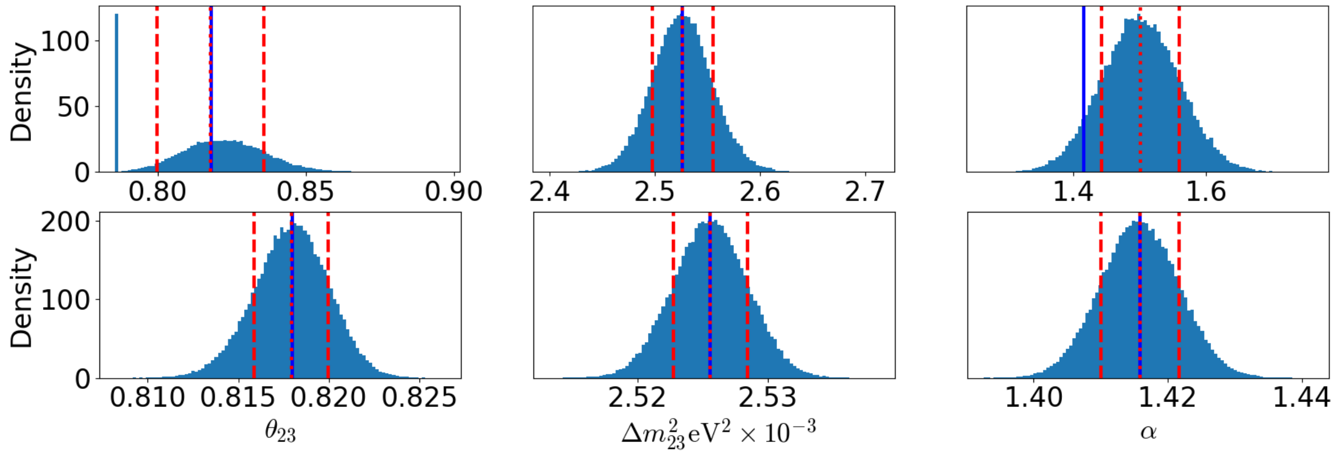

Fig. 5: Histograms showing the distribution of the resampled parameters. The top row shows the outcome when no averaging is applied for 5500 data sets, while the bottom row shows the outcome when we average over 100 data sets before minimising 12500 times. In all cases the position of the source parameter is shown by the solid blue line while the mean of the resampled parameters and the standard deviations either side of it are shown by dashed red lines.

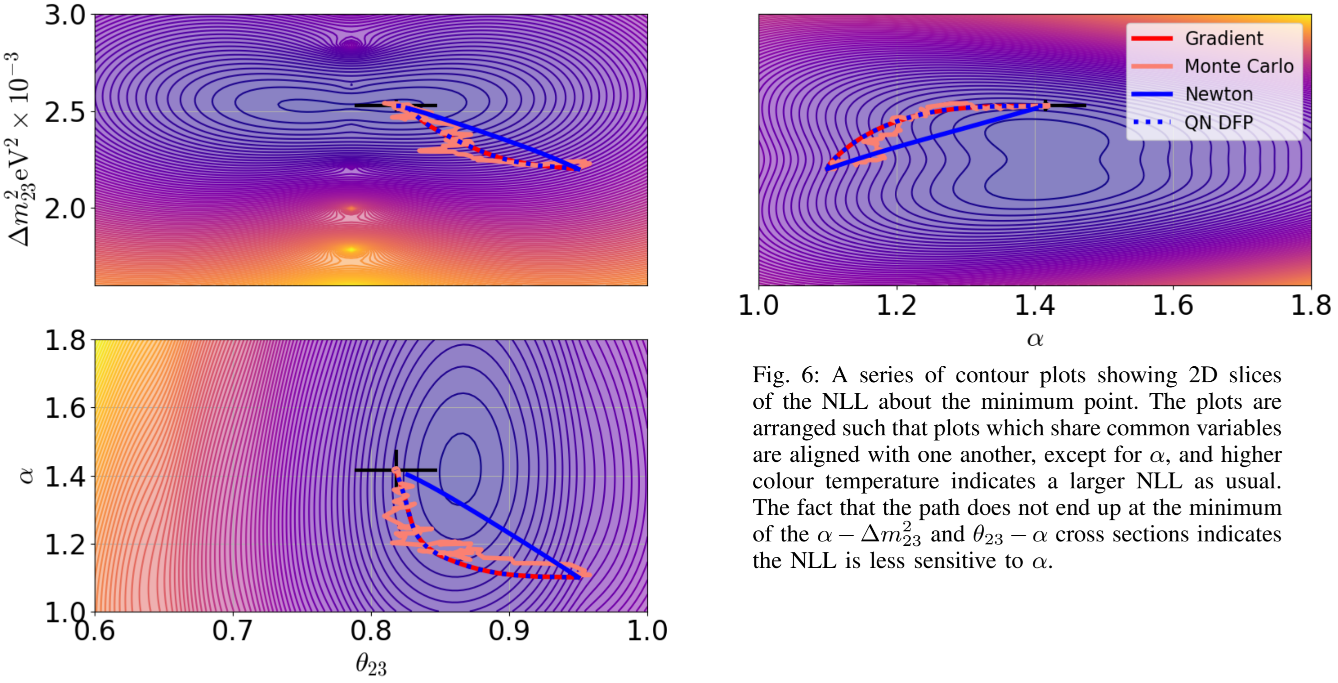

Fig. 6: A series of contour plots showing 2D slices of the NLL about the minimum point. The plots are arranged such that plots which share common variables are aligned with one another, except for α, and higher colour temperature indicates a larger NLL as usual. The fact that the path does not end up at the minimum of the $\alpha - \Delta m_{23}^2$ and $\theta_{23} - \alpha$ cross sections indicates the NLL is less sensitive to α.

Trivariate Minimisation

We now move on the most physically relevant case, in which we include the linear scaling of the cross section with energy, as given in (11). Although this entails minimising a three-dimensional NLL function, the methods employed in the 2D case (bar the univariate method) are easily extended to higher dimensions.

The gradient descent, Newton, simmulated annealing, and DFP methods all gave the same optimal values of $\hat{\theta}_{23} = 0.818 \pm 0.030$, $\Delta \hat{m}_{23}^2 = 2.526 \pm 0.029$ eV^2 $\times 10^{-3}$ and $\hat{\alpha} = 1.416 \pm 0.059$, with their paths taken shown in figure 6. The Hessian evaluated at this point was found to be strictly positive definite, thus verifying it is a minimum point to within some truncation error. To verify this was a global minimum, a scan was done over a the 3D region bounded by $\theta_{23} = 0.7 \rightarrow 1.0$, $\Delta m_{23}^2 = 2.0 \rightarrow 3.0$ and $\alpha = 1.0 \rightarrow 2.0$ at regular intervals. As expected this was the global minimum, although a considerable number of trajectories became stuck in the saddle point at $\theta_{23} = \pi/4$ When using these values to generate λ_i using (11) we can see the result is in better agreement with the data compared to the previous cases.

Error Calculation

In all cases we calculate the error of the $x_i{}^{\mathrm{th}}$ component of a given minimum point by taking the average distance over which the NLL changes by 0.5 in either direction along constant x_i. We do not need to consider error from the minimiser itself; as unless otherwise stated, all direct search methods were set to terminate only after reaching an accuracy of $\mathcal{O}(10^{-8})$.

When using the parabolic minimiser we can also estimate the error by using the curvature of the last parabolic estimate. For the 1D result we find that this method gives the error in $\hat{\theta_{23}}$ as ~ 0.073. This method is less accurate, and inextensible to other methods, and so we will not discuss it further.

Verification

Although we have been careful to test our minimisation procedures, we can make further attempts to validate the process via resampling. To do this we use our obtained parameter estimates to calculate λ_i for each energy bin using (11). We then generate samples from Poisson distributions with these means to in effect generate a new data set which has been seeded using our parameter estimates.

If we then carry our parameter estimation procedure on this new data set we should expect the parameters to be close to the 'source' parameters from which the data was generated. If we repeat this generating-fitting procedure many times we should find, by virtue of the central limit theorem, for the new parameters to be normally distributed about the source parameters. The whole resampling process has been summarised in figure 7. When this is done for our final trivariate parameter estimates our resampled parameters have distributions shown in the upper row of figure 5.

While the result for $\Delta \hat{m_{23}^2}$ is as we would expect, $\hat{\theta_{23}}$ shows an unexpected spike at $\hat{\theta_{23}} = \pi/4$ and $\hat{\alpha}$ is shifted away from the source value of $\hat{\alpha}$.

The former of these issues can be traced to the fact that there is a saddle point at $\hat{\theta_{23}} = \pi/4$, as evident from figure 6. Since the location of the minimum for the generated data will vary, we can sometimes end up becoming 'stuck' on the saddle point with no hope of finding the global minimum.

The mis-match between the source value and resampled mean for $\hat{\alpha}$ is a symptom of the nature of the data itself. The fact we are working with low-sample Poisson distributed values means that the NLL is far more sensitive to variations in θ_{23} and Δm_{23}^2 than variations in α. This is corroborated by figure 6 in which we can visually see how the overall minimum point does not lie within the minimum region for two of the three plots.

We can mitigate the effects of low statistics by adding in an averaging step in which we take the mean of many generated data sets to produce an averaged data set which will be more sensitive to the values of the source parameters. We can see from figure 5 that when this step is implemented the source values and resampled values are in good agreement.

Since the averaging will move the minimum of the generated data much closer to the starting value in the minimisation (which is always taken as the source parameters), it has the added benefit of mitigating the case where trajectories become stuck on the saddle point. Note that we should only be concerned with the mean values in figure 5 as the spread can be made arbitrarily small by increasing the number of averages taken.

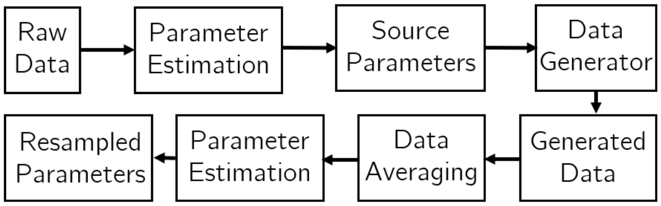

Fig. 7: A flowchart showing the general flow of data during the resampling process. Note that the data averaging step is omitted when we wish to generate data sets of the same size as the raw data set, as is the case in the top row of figure 5.

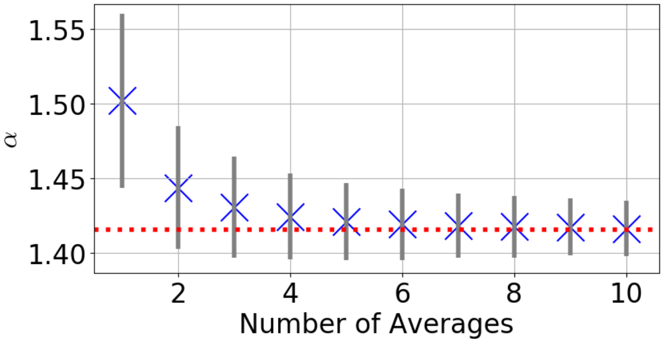

Fig. 8: A plot showing how the mean resampled value of α converges to the source value shown by the dotted red line as we increase the number of averaging steps. The standard deviations of the resampled values are shown by the grey error bars.

Improving the Sensitivity to α

From our resampling process we have learned that our value of $\hat{\alpha}$ is over-fitted when the data set size is small. This raises the important question of how much more experimental data we require for sensitivity to increase such that $\hat{\alpha}$ is well fitted. To gain an estimate for this we can see how the mean resampled α approaches the source value as we increase the number of data sets we average over. Such a process is illustrated in figure 8, where we can see that after ~ 4 or more averaging steps the mean resampled value of $\hat{\alpha}$ starts to appreciably converge (within 1%) to the source value.

Zero-Energy Cross Section

Throughout the 3D fitting process we assumed the cross section to be directly proportional to energy as in (11), and thus we have in effect assumed rest neutrinos have zero cross section. To test whether this assumption has any effect on our result we can add in an offset term β such that

$$\lambda_i = P_i \Phi_i(\alpha E_i + \beta) \tag{30}$$

Upon carrying out a 4D minimisation using the gradient method we find that $\beta \sim 0.01$ and the value of the NLL is changed by less than 0.1%. We can thus safely neglect this complication in favour of our results for the 3D case.

CONCLUSION

We found best estimates of the mixing angle, mass-squared difference and rate of cross section increase with energy to be $\hat{\theta_{23}} = 0.818 \pm 0.030$, $\hat{\Delta m^2_{23}} = 2.526 \pm 0.029 \mathrm{eV}^2 \times 10^{-3}$ and $\hat{\alpha} = 1.416 \pm 0.059$ respectively. After working with multiple minimiser methods it is clear that the gradient descent method is the most suitable minimiser owing to its robustness, and our lack of need for high efficiency given the relatively small data set size.

From the resampling process with no data set averaging we can see that while θ_{23} and Δm^2_{23} are fitted well, the relatively high variance of small-mean Poisson statistics results in α being consistently over-fitted. When instead fitting the average of multiple generated data sets to reduce the relative variance, the mean resampled α converged to the source value as expected. It was seen that the resampled value was within $\sim 1\%$ of the source value when the number of averaging steps was ~ 4 or greater. This suggests that if the original data set size is increased to the order of 10^3 samples, the rate of cross section increase with energy may be extracted with greatly improved fidelity ($\lesssim 1\%$). This may also open up the possibility of offset mentioned in (30) becoming resolvable.

REFERENCES

[1] V. Barger, D. Marfatia, and K. Whisnant, *The Physics of Neutrinos*. Princeton University Press, 2012.

[2] W. C. Davidon, “Variable metric method for minimization,” *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 1–17, 1991.

[3] F. E. Curtis and X. Que, “A quasi-newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees,” *Mathematical Programming Computation*, vol. 7, pp. 399–428, Dec 2015.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 671–680, May 1983.

[5] O. K. Smith, “Eigenvalues of a symmetric 3 &times; 3 matrix,” *Commun. ACM*, vol. 4, pp. 168–, Apr. 1961.

[6] C. H. Richardson, *An introduction to the calculus of finite differences*. New York: Van Nostrand, 1954. 142 p.

[7] H. H. Rosenbrock, “An Automatic Method for Finding the Greatest or Least Value of a Function,” *The Computer Journal*, vol. 3, pp. 175–184, 01 1960.

Word Count: ~ 2400