A Log Likelihood fit for extracting the D^o meson lifetime using 1D and 2D minimisation methods

19th December 2016
Priyanka Mocherla
CID: 00929775

This project focuses on calculating the mean lifetime the D^o meson from a set of data. The lifetime is extracted by minimising the negative log likelihood of the probability density function of decay times, with respect to the free parameters: lifetime (τ) and fraction of signal (a). To solve for the lifetime, a 1D parabolic minimiser was initially implemented and the lifetime was calculated as τ = 0.405 ± 0.005 ps. After noting the presence of a background signal, the data then had to be solved for τ and a. Three 2D minimisers were used to achieve this: the Gradient, Newton and Quasi-Newton minimisers. Each minimiser returned the value of the lifetime as τ = 0.410 ± 0.005 ps, and the level of background as a = 0.016 ± 0.007 ps. The 2D minimisation produced a more accurate result, corresponding to the accepted value of τ = 0.410 ± 0.002 ps.

1 INTRODUCTION

The D^o meson ($c\bar{u}$) is a highly unstable subatomic particle. After a short time the meson will decay into less massive particles, to stabilise. This decay occurs randomly, however, by averaging the decay times of a large number of mesons, the D^o lifetime can be obtained. The D^o meson has an accepted lifetime of 0.410 ± 0.002 picoseconds. [1] This lifetime magnitude is small and therefore cannot be measured directly. The data is instead obtained by colliding particles in a particle accelerator and calculating the decay time, *t,* using relativistic dynamics.

This paper aims to extract the mean D^o meson lifetime, with its associated uncertainty from, a set of 10,000 *t* measurements. The effect of the background signal on the lifetime value will also be discussed.

2 THEORY

2.1 D^o MESON DECAY TIME DISTRIBUTION

In nature, the decay times of D^o mesons are independent but continuous and therefore follow an exponential distribution.

Experimentally however, each decay time measurement, *t*, has a measurement uncertainty, σ, which causes a 'smearing' of the distribution. This uncertainty can be represented as a Gaussian of width σ. If σ has a value close to the decay distribution width, the smearing effect becomes stronger. This means that *t* can appear negative due to the distortion, although this is not physically plausible.

The measured distribution is hence a convolution of the theoretical exponential decay function and the uncertainty function, resulting in:

$$f^m_{sig}(t) = \frac{1}{2\tau}\exp\left(\frac{\sigma^2}{2\tau^2} - \frac{t}{\tau}\right)\,\mathrm{erfc}\left(\frac{1}{\sqrt{2}}\left(\frac{\sigma}{\tau} - \frac{t}{\sigma}\right)\right) \tag{2}$$

where τ is the mean lifetime of the measured distribution, f^m_{sig}.

There is also a level of background in the dataset. This combinatorial background occurs due to random combinations of other subatomic particles in the accelerator. These have lifetimes of zero, but are smeared by σ in the same way as the D^o decays. The measured background, f^m_{bkg}, is a delta function convoluted with the uncertainty distribution:

$$f^m_{bkg}(t) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{t^2}{\sigma^2}\right) \tag{3}$$

Thus the true signal, $f(t)$, has a parameter, *a,* which needs to be calculated to measure the amount of background in the signal:

$$f(t) = af_{sig}^{m}(t) + (1-a)f_{bkg}^{m}(t) \tag{4}$$

The mean lifetime of the sample is extracted by applying the negative log likelihood (NLL) on the desired probability distribution. The NLL is given by:

$$\mathrm{NLL}(\boldsymbol{u}) = -\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \log(P(\boldsymbol{u},(t_i,\sigma_i)) \tag{5}$$

where $\boldsymbol{u}$ represents the free parameters to be calculated and $P(\boldsymbol{u},(t_i,\sigma_i))$ is the probability density function (PDF) being evaluated at each measurement. To calculate the mean lifetime, irrespective of the background, $\boldsymbol{u} = \tau$ and Equation 2 is set as the PDF. For the second part of the project, the background is taken into account and $\boldsymbol{u}$ = (τ, a) whilst the PDF becomes Equation 4.

The values of $\boldsymbol{u}$ are subsequently found by minimising the resulting NLL. These minimisations are done computationally using the algorithms outlined in Section 3.

3 Numerical Methods

3.1 1D Minimiser

In general, the minimum of a function can be approximated as a parabola. While $\boldsymbol{u} = \tau$, the problem is one-dimensional, hence a 1D parabolic method was used for minimisation.

Three starting points, $\tau_{0,1,2}$ and their corresponding $\mathrm{NLL}_{0,1,2}$ values were picked close to the minimum. A 2nd order Lagrangian was fitted through these points and the minimum, τ_3, of the interpolating parabola was found using:

$$\tau_{3}=\frac{1}{2}\frac{(\tau_2^2-\tau_1^2)\mathrm{NLL}_0+(\tau_0^2-\tau_2^2)\mathrm{NLL}_1+(\tau_1^2-\tau_0^2)\mathrm{NLL}_2}{(\tau_2-\tau_1)\mathrm{NLL}_0+(\tau_0-\tau_2)\mathrm{NLL}_1+(\tau_1-\tau_0)\mathrm{NLL}_2} \tag{6}$$

The lowest three values from $\mathrm{NLL}_{0,1,2,3}$ were kept, and the algorithm repeated until the difference between the successive τ_3 values were less than a tolerance. In this investigation the tolerance was set to 10^{-5}, unless stated otherwise.

3.2 Two Dimensional Minimisers

Setting $\boldsymbol{u}$ = (τ, a) turned the NLL into a 2D function. Three 2D minimisers were used in this project. Each algorithm used was iterated until the step between successive iterations was lower than the tolerance.

3.2.1 Gradient Method

The first minimiser calculated the local gradient and stepped in its negative direction. This was implemented using:

$$\vec{x}_{n+1} = \vec{x}_n - \alpha\vec{\nabla}f_n \tag{7}$$

where $\vec{x}_n$ is the starting point (τ_n,a_n), α is a small scaling parameter and $\vec{\nabla}f_n$ is the local gradient of the NLL function. The finite difference method (FDM) formula, used to calculate the gradient, can be found in Appendix 8.1.

3.2.2 Newton Method

The Newton Method was also implemented and involved calculating the 2D Hessian:

$$\boldsymbol{H_n} = \begin{pmatrix} \frac{\partial^2 f_n}{\partial\tau^2} & \frac{\partial^2 f_n}{\partial\tau\partial a} \\ \frac{\partial^2 f_n}{\partial a\partial\tau} & \frac{\partial^2 f_n}{\partial a^2} \end{pmatrix} \tag{8}$$

and applying it to:

$$\vec{x}_{n+1} = \vec{x}_n - [\boldsymbol{H}_n]^{-1}\vec{\nabla}f_n \tag{9}$$

The method was expected to be more efficient, as it included the local curvature in calculations. The partial derivatives were calculated using an FDM, the formula for which can be found in Appendix 8.1.

3.2.3 Quasi-Newton Method

The Quasi-Newton method was similar to the Newton, however it approximated the inverse Hessian as the matrix, $\boldsymbol{G_n}$, expected to make the method less computationally intensive as no detailed matrix calculations were involved. $\boldsymbol{G_n}$ was updated using the Davidon-Fletcher-Power algorithm. Steps were iterated over:

$$\vec{x}_{n+1} = \vec{x}_n - \alpha\,\boldsymbol{G_n}\cdot\vec{\nabla}f_n \tag{10}$$

The full form of $\mathbf{G_n}$ can be found in Appendix 8.1. [2]

3.3 Uncertainty Calculations

Two methods were used to calculate the uncertainty in minimisation. The first method involved finding the absolute uncertainty by scanning across both sides of minimum until the NLL function increased by 0.5, corresponding to one standard deviation, $\tau_{\pm}$, in each direction.

The second method involved finding the curvature of the minimum and approximating it to a parabola. Then using the relation:

$$\frac{1}{{\tau_{\pm}}^2} = \frac{2[y_0(x_2 - x_1) + y_1(x_0 - x_2) + y_2(x_1 - x_0)}{(x_1 - x_0)(x_2 - x_0)(x_2 - x_1)} \tag{11}$$

the standard deviation was calculated.

3.4 Minimiser Validation

3.4.1 1D Minimiser

The parabolic minimiser was tested using the function cosh*(x)*, where the minimum is known to occur at (0,1). Running the minimiser with a tolerance of 10^{-9} and starting points x = 0.2, 0.4 and 0.6, it returned the minimum as (0.0,1.0), as required.

3.4.2 2D Minimisers

The function $z = x^2 + y^2$ was used to test the 2D minimisers. The minimum was expected to occur at (x,y) = (0,0).

Setting the tolerance to 10^{-8}, α to 0.1 and starting at (0.4,0.7), the minimum obtained from the Gradient minimiser was (-5×10^{-6}, -5×10^{-6}). The minimiser was effective to an acceptable degree of accuracy. The convergence took 73 iterations. Using the Quasi-Newton minimiser with the same settings also returned (-5×10^{-6}, -5×10^{-6}) and convergence took 10 iterations. Hence it was predicted to be more efficient than the Gradient minimiser. Finally, the Newton method returned (5×10^{-6}, 3×10^{-6}), in 8 iterations.

4 Results and Analysis

4.1 D^0 Lifetime: without Background

Histograms were produced using data from 10,000 decay measurements. Presented in Figure 1, the decay times varied from -2 to 6 ps.

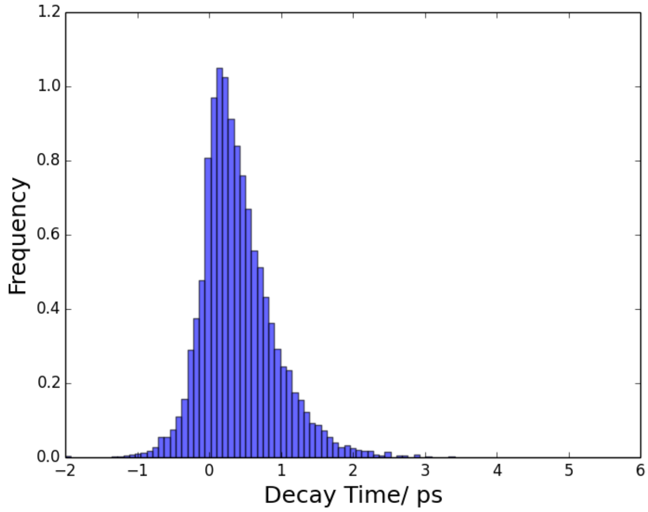

Figure 1: A histogram to show the normalised distribution of the decay times in the dataset. The distribution resembles an exponential distribution and peaks between 0 – 0.5 ps.

The measurement uncertainties are presented in Figure 2, which ranged between 0 and 0.6 ps. These uncertainties had values comparable to the decay time distribution width, accounting for the negative decay times due to smearing of results.

Using Equation 2, a function was fitted to the distribution. The behaviour of the fit function was observed by adjusting the lifetime (τ) and uncertainty (σ) parameters. Figure 3 shows the behaviour with varying τ. Increasing τ increased the peak width whilst reducing its amplitude. Smaller τ values also appeared to make the fit function appear more symmetrical and Gaussian-like. By approximately fitting the data, Figure 3 shows that the lifetime was around 0.4 ps.

The effects of varying σ were also observed and shown in Figure 4. Reducing σ reduced the extent of distribution blurring. The function also peaked more sharply around zero and decayed more rapidly.

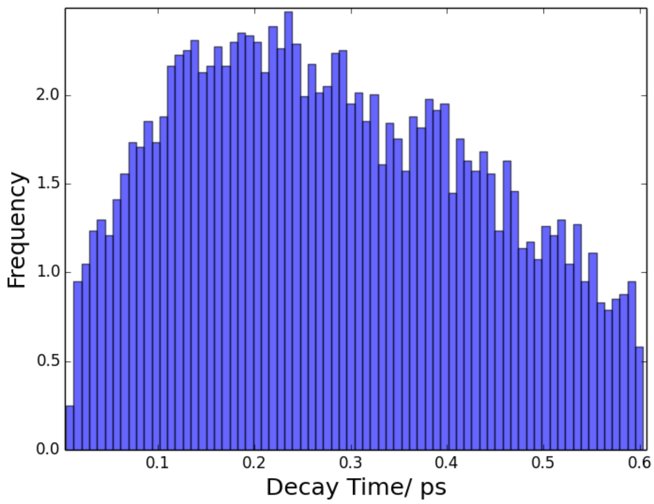

Figure 2: The normalised distribution of uncertainties of the measurements. They are between 0 - 0.6 ps however have values comparable to the decay time distribution width.

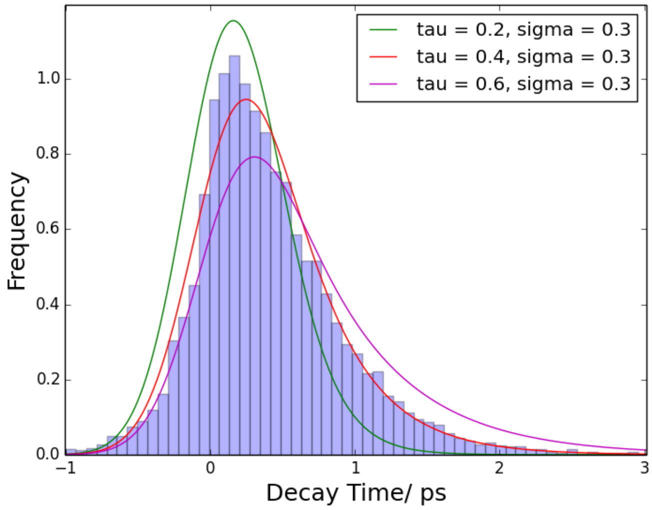

Figure 3: A plot to show the variation of the fit function with τ. As τ increases, the peak width increases and the function becomes more symmetrical.

The fit function is a PDF. If normalised, the integral over decay times should return one, and should be independent of τ and σ. Code was written to integrate the fit function for a series of τ and σ values over -50 < t < 50. Results showed that all integrals returned 1.0, as expected.

The NLL function was created using Equations 2 and 5, with $\boldsymbol{u}$ = τ. Plotting the NLL function, presented in Figure 5, showed that the minimum lay between 0.3 – 0.5 ps.

Using the method outlined in Section 3.1, a parabolic minimiser was created. By using a range of starting points near the minimum, between 0.1 and 1.0, the minimum obtained was:

τ = 0.405 ± 0.005 ps

The uncertainty was calculated using both methods explained in Section 3.3, which corresponded with each other to an accuracy of 10^{-4}.

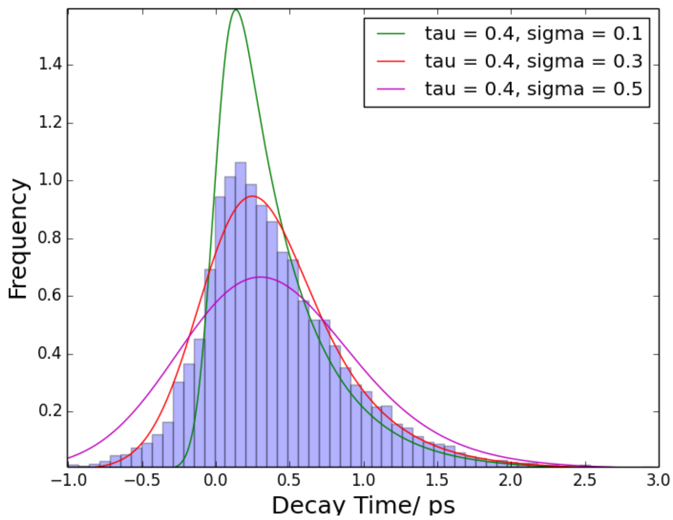

Figure 4: A plot showing the variation of the fit function with σ. As σ decreases, there is a smaller smearing effect on the results and the function tends towards an exponential distribution.

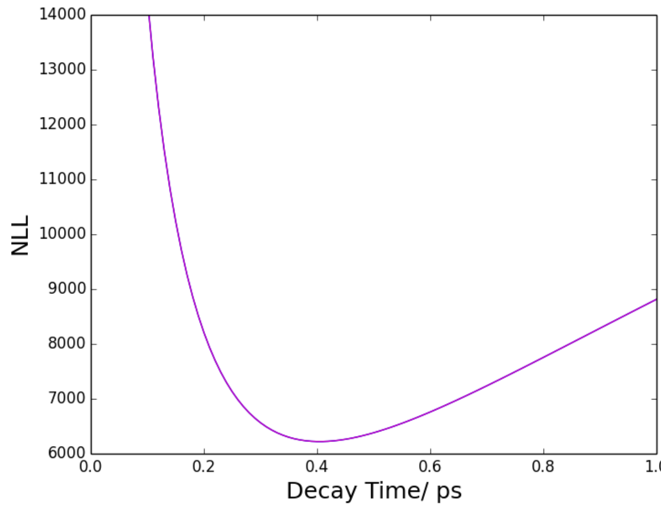

Figure 5: A plot of the NLL against decay times. The plot shows that the minimum lies between 0.3 - 0.5. The parabolic minimiser found the minimum to be 0.405 ps.

4.2 D^0 Lifetime: With Background

To include the effect of background signal in calculations, the PDF in Equation 4 was implemented. Plotting the NLL as a function of τ and fraction of signal (a) allowed an approximation of the new minimum, presented in Figure 6.

To find the position of the new minimum the 2D minimisers detailed in Section 3.2 were applied and the absolute uncertainty was calculated at the minimum. The final values for the parameters were:

τ = 0.410 ± 0.005 ps
a = 0.984 ± 0.007

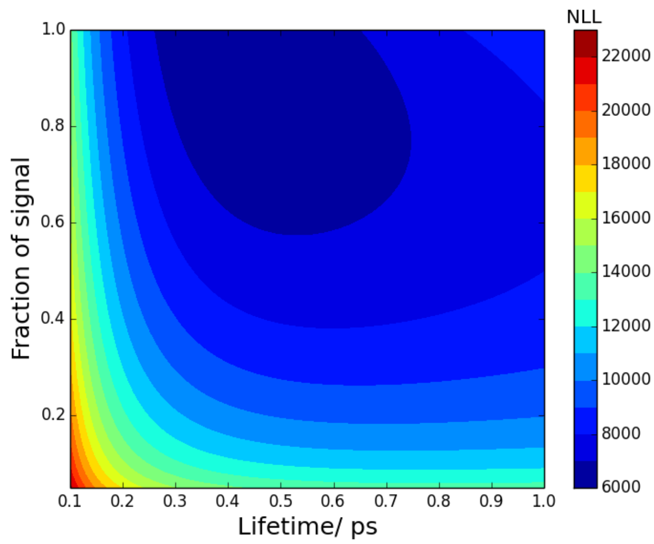

Figure 6: A graph of the NLL, as a function of τ and a. The graph shows that the new minimum lies between 0.4 < τ < 0.6 and 0.6 < a < 1.0.

5 Discussion

5.1 D^0 Lifetime

Ignoring the background signal in calculations gave τ = 0.405 ± 0.005 ps. Compared to the literature value of 4.010 ps, the calculated result was within one $\tau_{\pm}$. Including the background produced τ = 0.410 ± 0.005 ps, more accurate than the first calculation. This shows that the background caused an underestimation of τ, likely due to the fact that the lifetime of the background sources were zero, which shifted the distribution towards zero.

The fraction of background in the signal was 0.016 ± 0.007, which caused a 1.2% discrepancy in the final results. Disregarding the background gave a lifetime with the accepted value within its uncertainty, however it was a significant systematic error. By taking it into account the accuracy of τ increased as it corroborated well with the accepted value.

5.2 Uncertainty Analysis

For the 1D minimisation, both of the uncertainty methods returned a value of ± 0.005 ps.

The absolute uncertainty method took into account the fact that the NLL was not symmetrical about the minimum. In this case, it was found that τ_+ and τ_- were 0.0047 ps and 0.0048 ps respectively. Nevertheless, this was not significant in the final statement of results. However the scanning involved in calculating the uncertainty was time intensive.

By approximating the minimum as a parabola and

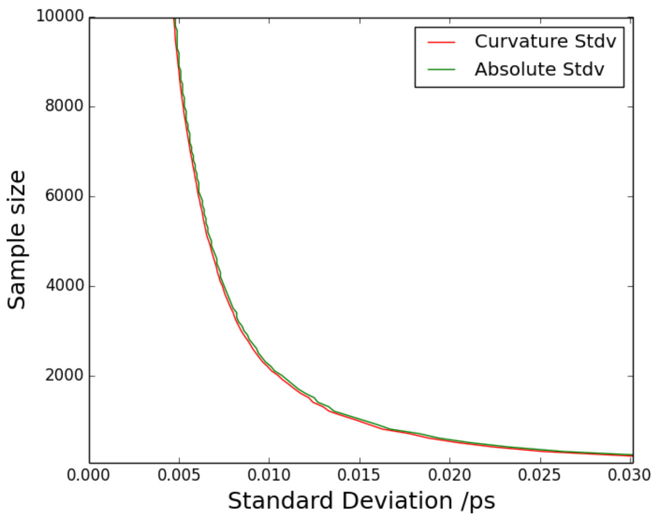

Figure 7: The standard deviation calculated using the curvature method and the absolute method. They largely correspond, although the absolute method is slightly greater than the curvature method at any given point

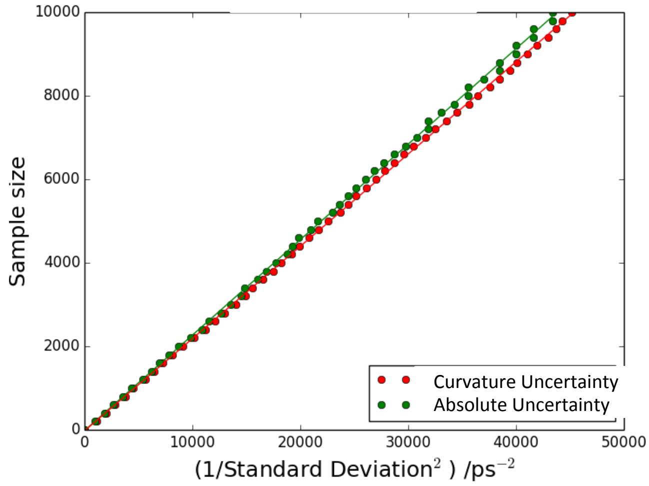

Figure 8: A graph showing the linear relationship between the Sample size and 1/variance of the sample. The linearity shows that the central limit theorem is a good approximation for the dataset.

using the curvature uncertainty, the result was obtained much faster. As mentioned, for this dataset, the standard deviations of both methods corresponded with each other to three decimal places as the minimum was largely symmetrical.

The behaviour of the standard deviation was observed, as a function of the number of measurements included in the NLL fit. Figure 7 shows that the uncertainty followed an inverse relationship. For large (>30 measurements) sample sizes of independent random variables, the central limit theorem can be applied, regardless of the initial distribution. Therefore, the variance of each sample should vary as 1/N where N is the number of measurements in the sample. Plotting N as a function of 1/variance, should therefore result in a straight line. [3] This is exhibited in Figure 8.

Fitting a line through the data points returned the following results

$$\textbf{Absolute Uncertainty: N} = \frac{0.23}{\tau_{\pm}^2} - 13 \qquad (12)$$

$$\textbf{Curvature Uncertainty: N} = \frac{0.22}{\tau_{\pm}^2} - 24 \qquad (13)$$

For an accuracy of $\tau_{\pm} = 10^{-3}$ ps, both methods required over 200,000 measurements to be taken, 20 times more than the data available for this investigation.

5.3 2D Minimiser Analysis

To find the position of the minimum in Figure 6, the minimisers were run for 50 trials, and the results analysed.

The Gradient and Quasi-Newton methods followed the same minimisation paths, as shown in Figure 9. From the 50 trials plotted, 11 Quasi-Newton trials

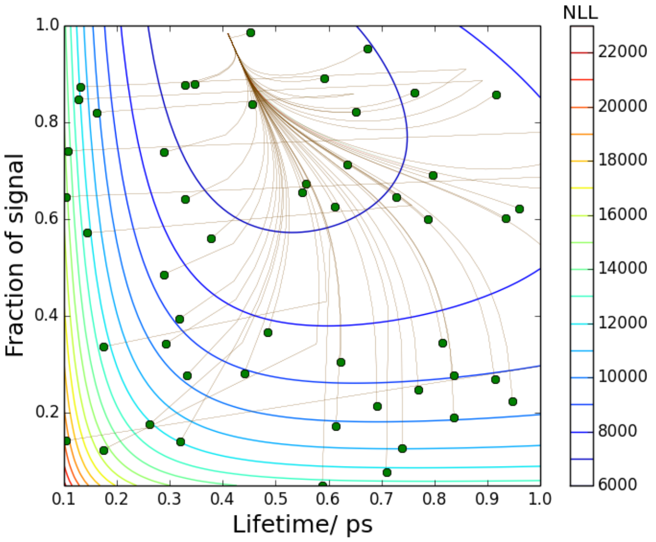

Figure 9: The minimisation paths of the Quasi-Newton (red) and the Gradient (green) methods. Both methods shared the same minimisation path, so cannot be distinguished clearly from each other.

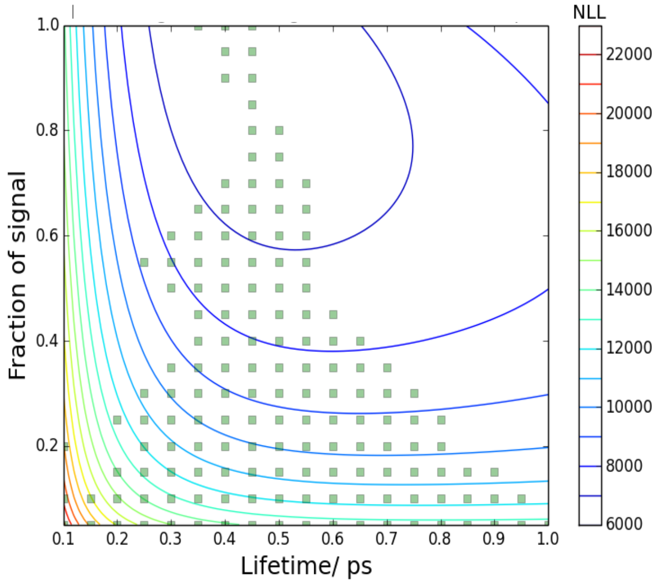

Figure 11: The approximate allowed region of starting points, given by the location of the green squares, to ensure the convergence of the Newton minimisation method.

converged faster than the Gradient trials, as opposed to equally, with a maximum of 2 fewer iterations. However the Quasi-Newton method was more time intensive, which did not compensate for the fewer iterations. In general, the number of iterations for these two methods ranged from 50-100.

The Newton method was much more efficient and the convergence paths for 10 sample points are presented in Figure 10. The paths for this sample took an average of eight iterations to converge. However a fundamental problem with the method was that it did not work for all defined starting points. This could be due to the fact that the Hessian at these points was not positive definite, a necessary property for convergence to occur. Figure 11 displays the allowed region for convergence, found by limiting the point of convergence to be within the bounds of the problem.

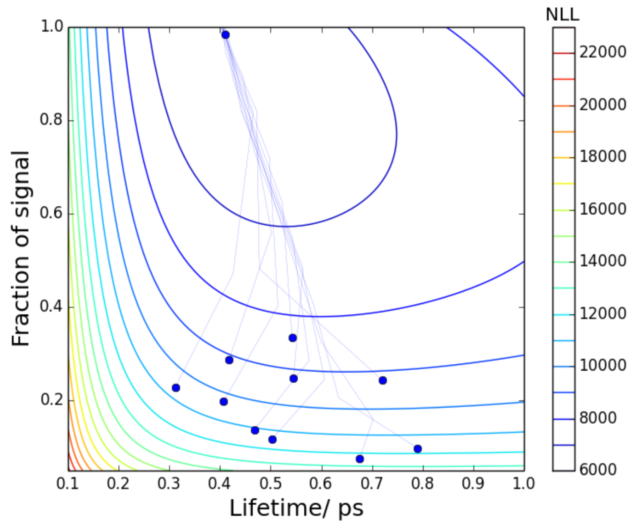

Figure 10: The minimisation paths of the Newton method for starting points in the accepted region.

Therefore the boundary conditions for convergence were $0 < a < 1$ and $\tau > 0$. After applying these conditions, the results converged to $\tau = 0.410$ ps and $a = 0.984$, corresponding to the results of the other minimisers.

For this investigation, all methods were able to converge to the same minimum values for a and τ, however the Gradient method proved to be the most versatile method to implement, even if not the most efficient.

6 Conclusion

To summarise, the mean lifetime of the D^o meson was calculated from a set of 10,000 experimental measurements of the decay times. The distribution of the sample was expected be exponential, smeared by a Gaussian form of uncertainty.

The lifetime was initially calculated without taking the background signal into account. This resulted in a 1D minimisation of the NLL of the PDF. The mean lifetime was calculated to be 0.405 ± 0.005 ps, which had the accepted value of 0.410 ps within its uncertainty. The uncertainty in measurement was calculated using two methods: one based absolute uncertainty of the NLL at the minimum and the other made by approximating the minimum as a parabola and measuring the curvature. These both produced corresponding values.

Taking the background signal into account, a new lifetime was calculated by minimising the 2D NLL. The result obtained was 0.410 ± 0.005 ps with a background of 0.016 ± 0.007. The background therefore was a significant systematic error, reducing the accuracy of the lifetime and underestimating the value of the lifetime when compared to the accepted value.

For the 2D minimisation, the Gradient method was found to be the most useful minimiser, iterating faster than the Quasi-Newton minimiser. Whilst the Newton method was more efficient than both other methods, as it did not work for all starting points, the Gradient minimiser became the best method to implement for this dataset.

7 References

[1]Patrignani et al., [Online] *Particle Data Group,* Charmed Mesons, 2016, [Accessed on: 18th December 2016] Available at: <http://pdg.lbl.gov/2016/tables/rpp2016-tab-mesons-charm.pdf>

[3] Uchida Y., November 2016, Chapter 6: Minimisation and Maximisation of Functions, Computational Physics, Imperial College London

[3]W. LaMorte, *The Role of Probability*, [Online] Central Limit Theorem, Last updated: 24th July 2016 [Accessed on: 18th December 2016] Available at: <http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Probability/BS704_Probability12.html>

2488 words

8 Appendix

8.1 Gradient Calculations

8.1.1 1st Order Derivatives

$\vec{\nabla} f_n$ can also be written as:

$$\vec{\nabla} f_n = \begin{pmatrix} \frac{\partial f_n}{\partial \tau} \\ \frac{\partial f_n}{\partial a} \end{pmatrix} \tag{14}$$

for each dimension. The partial derivatives were calculated using the following finite difference method:

$$\frac{\partial f_n}{\partial \tau} \approx \frac{f(\tau_n + h, a_n) - f(\tau_n, a_n)}{h} \tag{15}$$

and

$$\frac{\partial f_n}{\partial a} \approx \frac{f(\tau_n, a_n + h) - f(\tau_n, a_n)}{h} \tag{16}$$

where h is a small valued step-size, set to 0.00001 in this paper.

8.1.2 2nd Order Derivatives

To calculate the second order derivatives used in the 2D Hessian, the following FDM was used:

$$\frac{\partial^2 f_n}{\partial x_i \partial x_j} = \frac{f(\vec{x}_n + h_i + h_j) - f(\vec{x}_n + h_i) - f(\vec{x}_n + h_j) + f(\vec{x}_n)}{h_i h_j} \tag{17}$$

where (I,j) = (τ, τ), (τ,a), (a, τ) and (a,a) respectively.

8.1.3 Davidon-Fletcher-Power algorithm

To update the $\mathbf{G_n}$ matrix, the following algorithm was used:

$$\boldsymbol{G}_{n+1} = \boldsymbol{G}_n + \frac{(\vec{\partial}_n \otimes \vec{\partial}_n)}{\vec{\partial}_n \cdot \vec{\gamma}_n} - \frac{\boldsymbol{G}_n \cdot (\vec{\partial}_n \otimes \vec{\partial}_n) \cdot \boldsymbol{G}_n}{\vec{\gamma}_n \cdot \boldsymbol{G}_n \cdot \vec{\gamma}_n} \tag{18}$$

where:

$$\vec{\partial}_n = \vec{x}_{n+1} - \vec{x}_n \tag{19}$$

and

$$\vec{\gamma}_n = \vec{\nabla} f_{n+1} - \vec{\nabla} f_n \tag{20}$$