Computational Project 3

Andrew Dowling, Labani Roy

November 2023

1 Purpose

In this project, we have tried data fitting using Markov chain Montecarlo. Using metropolis algorithm, distributions of parameters of our model are shown.

2 Background

The LHCb experiment at CERN makes use of a series of different detection systems to reconstruct particles tracks created by decay products of heavy hadrons. When a particle crosses a detector an electrical signal is generated and the particles spacial position and momentum information can be extracted. From this, heavier particles such as hadrons can be reconstructed by lining up the tracks of their decay products.

Unfortunately, detectors are not perfect and there exists error in the energy and position values given by the detector. For this reason, no two particle tracks point back to the exact same location in space. Reconstructed particles are then created using two tracks that are “close enough“ to each other. This allows for false positives; two particles from seperate decays can point back to a false vertex.

A particle can be “close enough“ when it passes a series of restrictions (cuts) to ensure that candidate particles are likely to be true signal and not false reconstructions. Tighter cuts give better chances that combining particles will result in real decays, but are more susceptible to filtering out true events that have larger error.

To improve the accuracy of the detector, measuring a particles time-of-flight would give better accuracy to measuring tracks in the detector. Tracking particles in 4-D would better ensure that candidate particles came from real decays without having to sacrifice throwing out real events.

4-D tracking has long been a distant pipedream of those working at LHCb: analog signal processing could only timestamp pulses generated by particles at the ns level. No

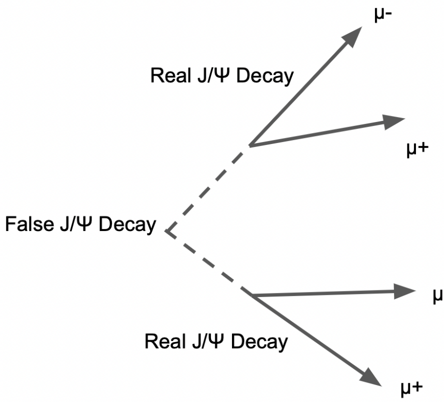

Figure 1: A false positive can occur when two unrelated tracks point back to the same location in space

meaningful information could be garnered from this with typical particle time-of-flight being 10's of ps. However, With new developments in analog to digital conversion, measuring time of arrival of particles could now be feasable. The AARDVARCv3, produced by Nalu Scientific, is analog to digital converter (ADC) capable of <10 ps precision. It can sample the waveform at 10 Gs/s, allowing for full waveform reconstruction. The digitized waveform can be fitted to a known function and a time of arrival can be found using the location of the pulse's maximum value.

The first step in determining if the AARDVARCv3 is a viable candidate for the LHCb project is by sending it a representative pulse using a waveform generator and seeing if the ADC can accurately capture the waveform. To generate the data for this report, an Tektronix AWG2041 arbitrary waveform generator generated a pulse with a amplitude of 1V and a rise time of 1ns. The Landau pdf, $L(x)$ describes the distribution of electrons produced by a charged particle passing through a silicon sensor, however, the AWG2041 does not have sufficient capabilities to interpret the function. The Landau pdf given below is one member of a family of functions with different widths and means.

$$L(x) = \frac{1}{\pi}\int_0^\infty \exp(-t\log t - xt)\sin(\pi t)dt \tag{1}$$

The shape of the pulse is instead defined by the Moyal function, $M(x)$, as it is a close approximation of the Landau pdf. Parameters μ and σ govern the peak location and the “width“ of the pulse, respectively.

$$M(x) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(\frac{x-\mu}{\sigma} + \exp(-\frac{x-\mu}{\sigma}))) \tag{2}$$

The pulse that is generated by the AWG2041 is not normalized and has some amplitude

‘A‘, as well as some constant offset voltage. The above equation has been modified to best describe the pulse's behavior as a function of time. After the modifications, the function given below is ready to be used to fit the digitized pulse so that the parameters A, μ, $sigma$, and V_0 can be extracted.

$$V(t) = A\exp(-\frac{1}{2}(\frac{t-\mu}{\sigma} + \exp(-\frac{t-\mu}{\sigma}))) + V_0 \tag{3}$$

3 Markov-Chain Monte Carlo and Metropolis algorithm

The main idea of Markov-chain monte carlo is that the future state of a system depends only on the current state, not the entire history of the process. Using this idea, we can sample from a probability distribution. In this project, we have used MCMC for curve fitting. Suppose, we have a model $f(\theta)$ where θ are our parameters of the model. Prior distribution is the distribution of parameters (θ) without having any prior knowledge of parameters. Likelihood is probability of our data given model and noise. So the posterior prob. is probability of our model given data points. This posterior probability dist. is our target distribution.

$$\begin{gathered} Posterior \propto likelihood \times prior \\ \Rightarrow P(f(\theta)|data) \propto P(data|f(\theta), noise) \times P(\theta) \\ \Rightarrow \log P(f(\theta)|data) \propto \log P(data|f(\theta), noise) + \log P(\theta) \end{gathered}$$

Metropolis Algorithm :

1. An initial point $\theta_{initial} = \{\theta_1, \theta_2, ...\}$ in the parameter space is chosen.
2. Calculating posterior prob. of initial set of parameters.
3. A proposal distribution $g(\theta)$ will be chosen and the next step will be chosen from this distribtuion θ_{next}.
4. Posterior probability for this point will also be calculated.
5. At this point, we need to take decision whether this point is acceptable or not.
6. acceptance prob. $= posterior_{next}/posterior_{current}$
7. Selecting a random number u from $\approx U(0, 1)$.
8. If acceptance prob. $>$ u, then this new state will the selected.

9. Then above procedure is repeated.

4 Intuition behind choosing initial values and proposal distribution

From the plot of given data, we can see that the peak is coming around $\mu = 2000$. And if we set $t = \mu$ in equation(3), then $A = 1000 \times \exp(0.5) = 1648.72$. There is little offset voltage within 10 to 50 mV. Motivating from these values, we have considered the initial set of parameters for the mcmc process to be $(V_0, A, \mu, \sigma) = (20, 1580, 1830, 72.98)$. There is noise in the datapoints. We have considered gaussian noise with mean $= 0$ and standard-deviation $= 10$.

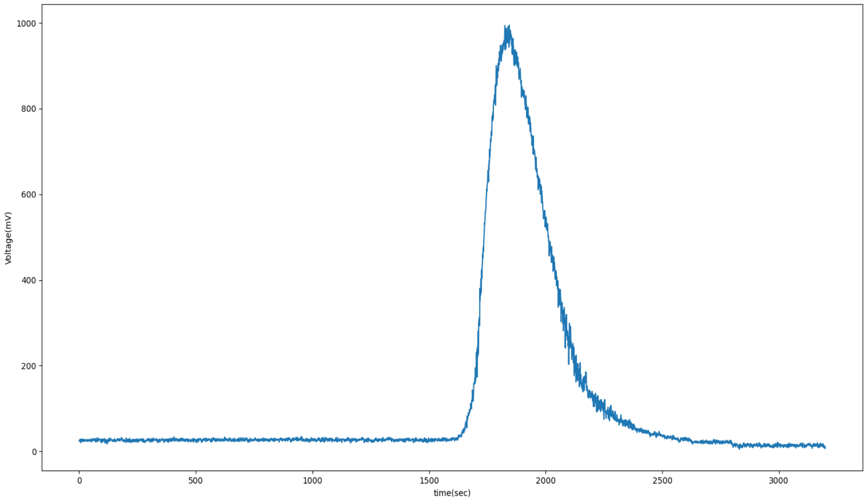

Figure 2: Plot of given data points

Proposal Distribution : At each iteration, the next step is selected from the proposal distribution. In our case, the proposal distribution for parameter V_0, A is a normal distribution (as it is a symmetric distribution, it is somewhat trivial to consider). For the other two parameters the proposal distribution is the exponential of a normal distribution. Exponential part is considered to ensure the fact that the mean and width of our model should be positive.

Prior Distribution : In case of off-shelf method, choice of prior is the gaussian distribution whose mean is around the best-fit values calculated from the optimization process and standard=deviation is 1. On the other hand, gaussian distributions of mean $= 1$ and standard deviation = 1000 are priors for handwritten mcmc. This is approximately uniform distribution.

5 Results/error

5.1 Distribution of parameters from the handwritten mcmc

In the plots below, distribution of parameters are shown.

- **iterations** = 10000 :

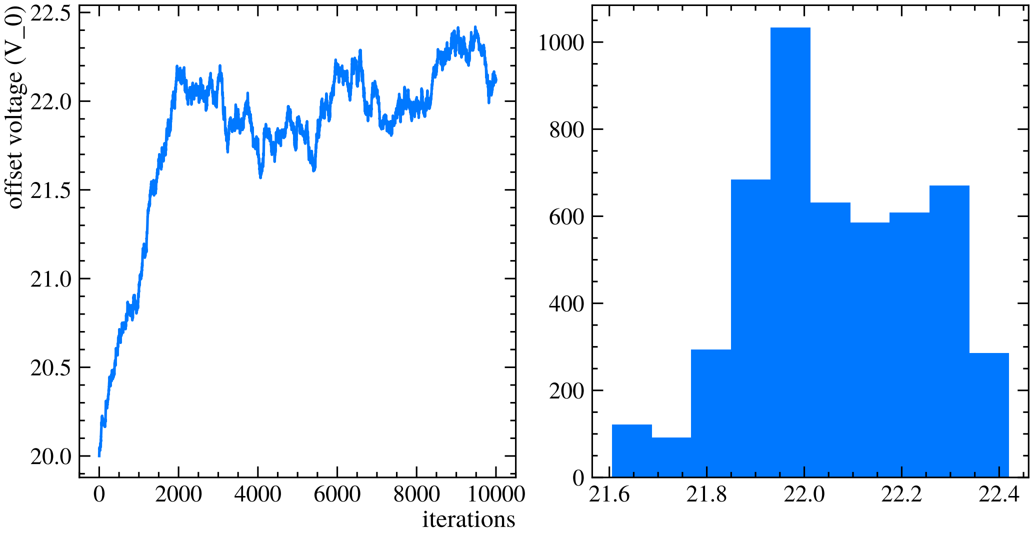

Figure 3: Trace plot and Distribution of V_0

1. For graphs in figures (3) and (4), although the chain moves to a new place in every iteration, the jump is relatively small. So it takes long time to get from one end of the distribution to another end.
2. For graph in figure (5), we can see that the chain tends to stay in one point for a longer time and sometimes it also takes more than 200 iterations to move to another place. Number of iterations in the burn-in phase is approximately 500.
3. For graph in figure (6), we can see that it explores the region with most of the density and moves to another point quickly. Number of iterations in the burn-in phase is approximately 2000.
4. For 10000 iterations the burn-in phase is not clear for offset and amplitude graphs. So, we have increased the number of iterations in the next step to see, whether it can make some effect in the graphs or not.

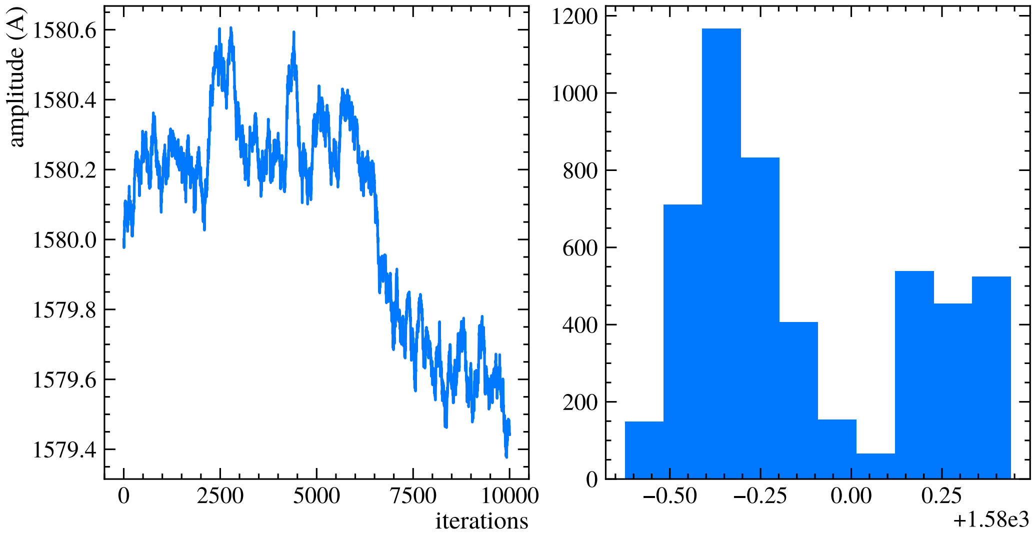

Figure 4: Trace plot and Distribution of A

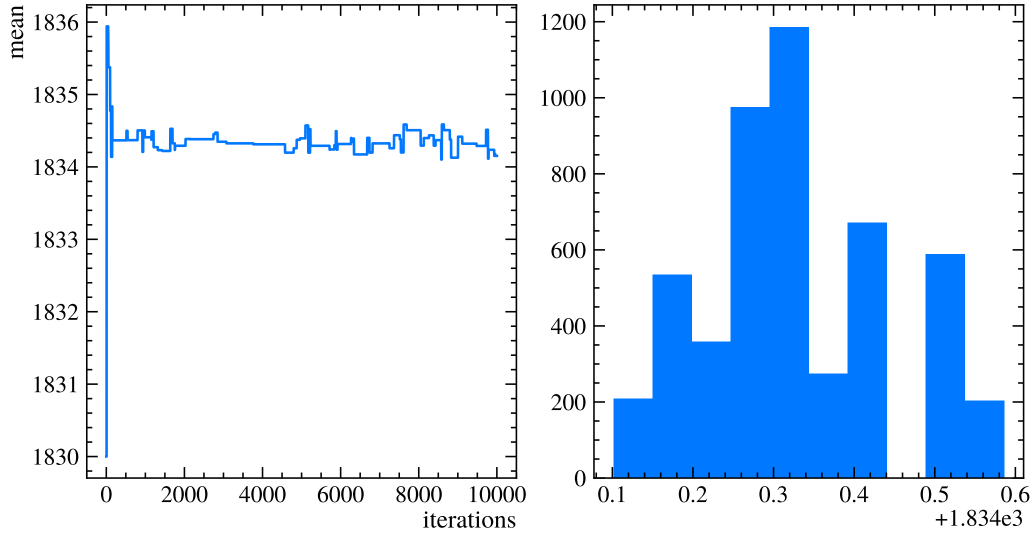

Figure 5: Trace plot and Distribution of μ

- **iterations** $= 90000$:

  1. When we are increasing the number of iterations, the trace plots for different parameters are getting clearer. From graph in figure(7), number of iterations in burn-in phase is 20000.

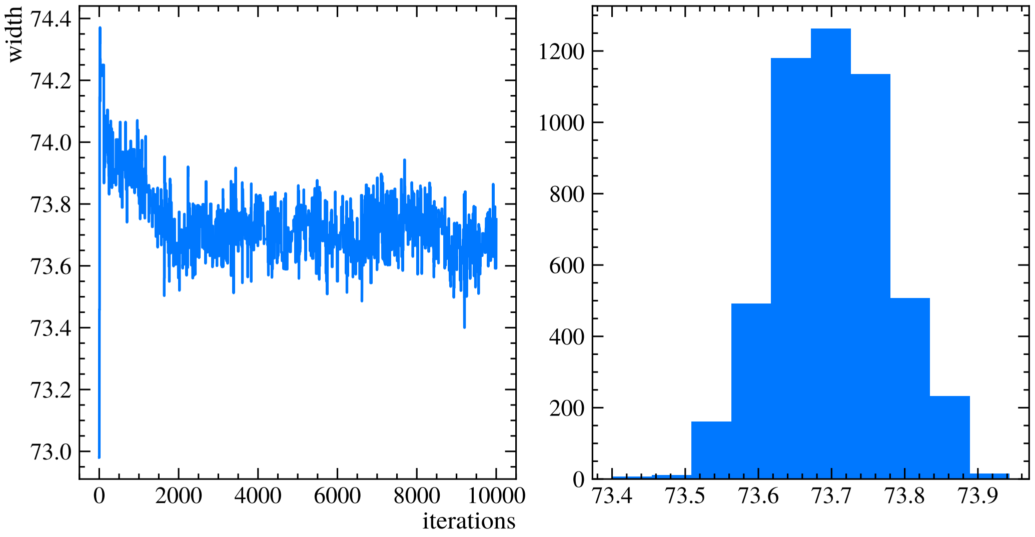

Figure 6: Trace plot and Distribution of σ

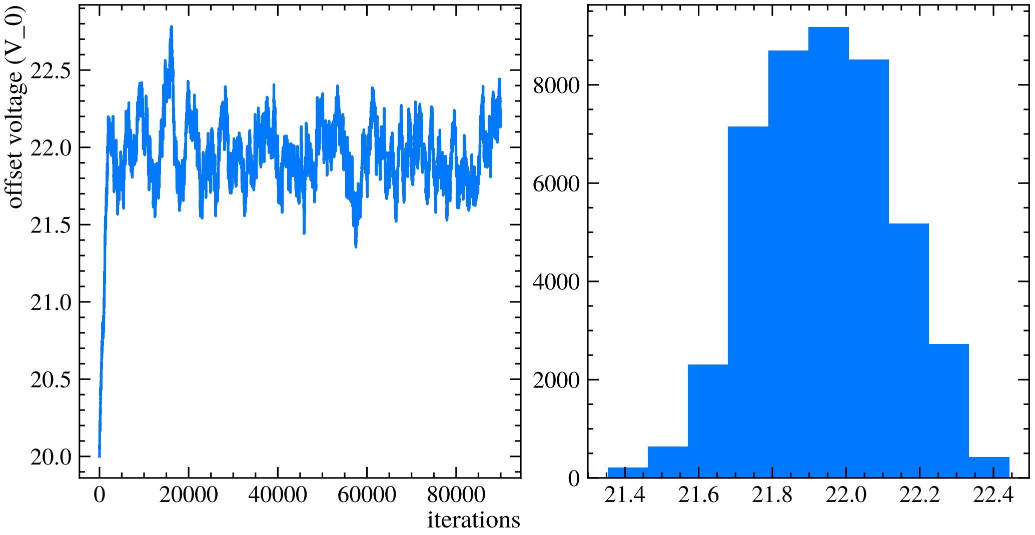

Figure 7: Trace plot and Distribution of V_0

6 Validation

6.1 Convergence checking

For the convergence of different walkers in the 4-dimensional parameter-space, **off-shelf method** is used. Each of them are heading toward the true distribution in time. In the plots below, convergence of different walkers are shown.

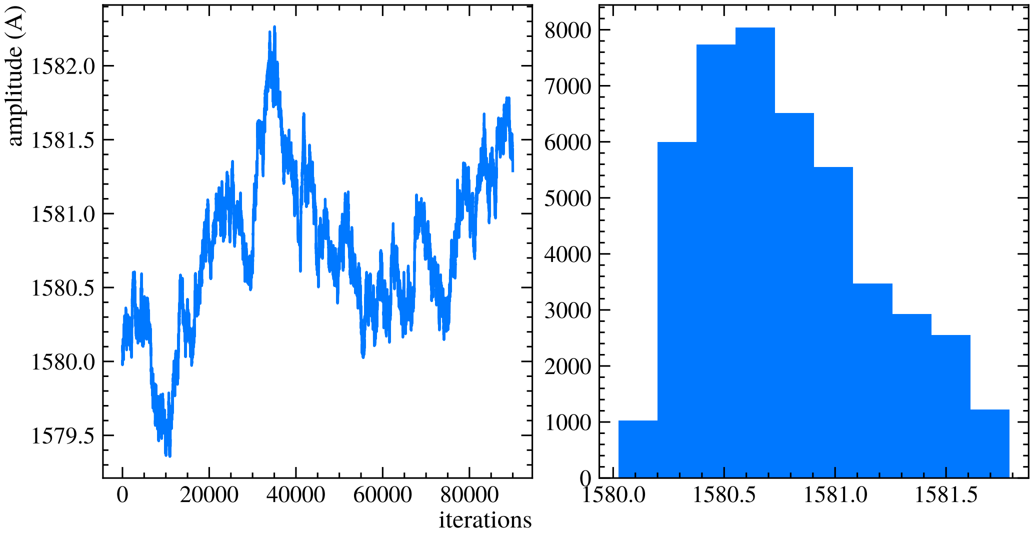

Figure 8: Trace plot and Distribution of A

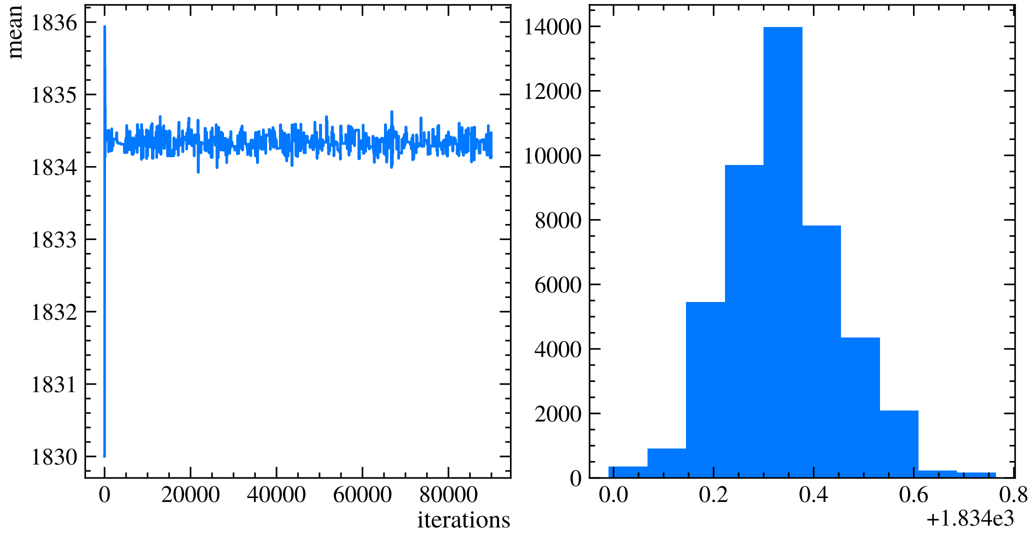

Figure 9: Trace plot and Distribution of μ

6.2 Auto-correlation time

Auto-correlation time is the number of steps required for the chain to produce independent samples. For any kind of sampler we always need to have smaller auto-correlation length. One reason maybe to remove "burn-in" period. We always have different starting points and they will drift differently. Once, we are near to the mode (or posterior prob. distribution), the point will be bouncing around that mode. If the points are far away, in that case there will be strong pull back towards the mode. Points close to the mode will have smaller pull and hence smaller auto-correlation length.

Auto-correlation time in our case are $(A, \mu, V_0, \sigma) \equiv (53.87, 39.72, 54.26, 47.46)$ when number

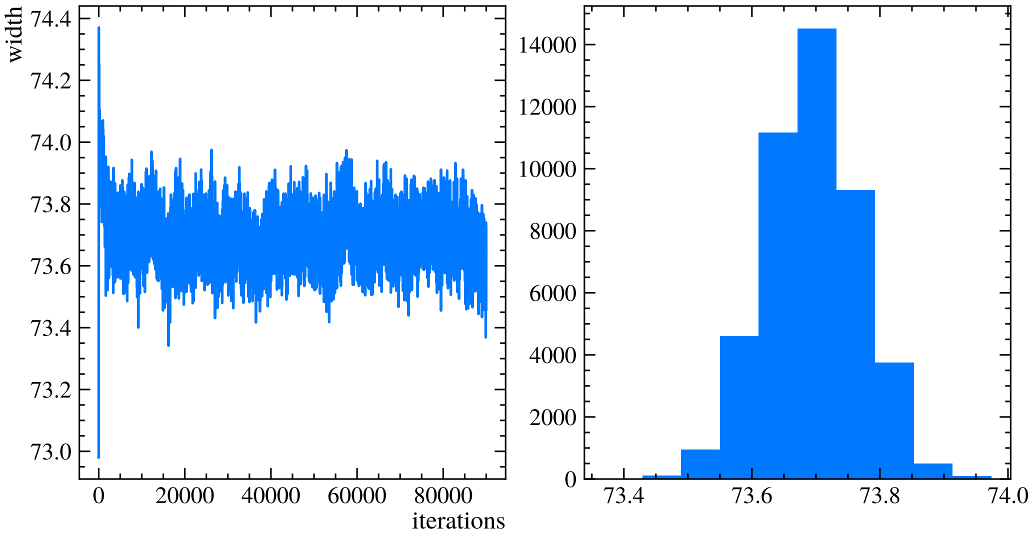

Figure 10: Trace plot and Distribution of σ

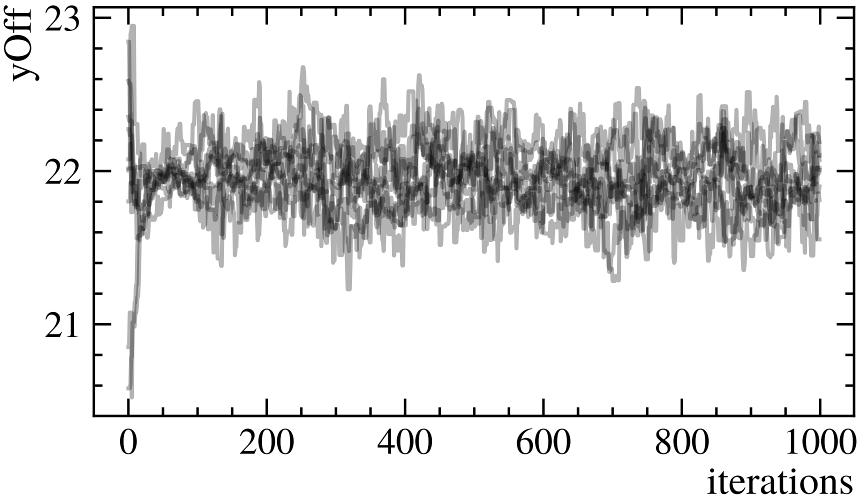

Figure 11: Convergence of different walkers for the parameter V_0

of steps are = 10000 and number of walkers = 50.
MCMC error will be lower for weakly correlated samples.

6.3 Comparison with the optimization process

Using optimization process, the best fitted parameters of our model (3) are $(V_0, A, \mu, \sigma) \equiv (21.94, 1581.99, 1834.32, 73.65)$. If we look at means of each parameter distributions, they are coming close to the values, we got from the optimization process.

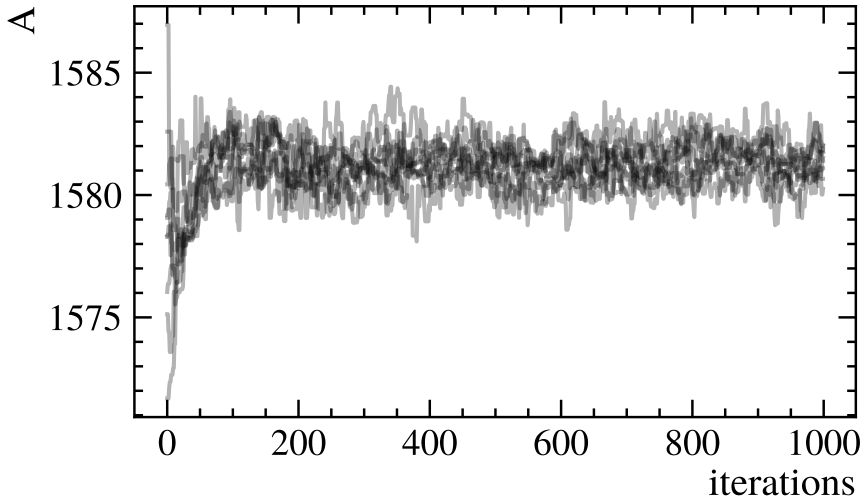

Figure 12: Convergence of different walkers for the parameter A

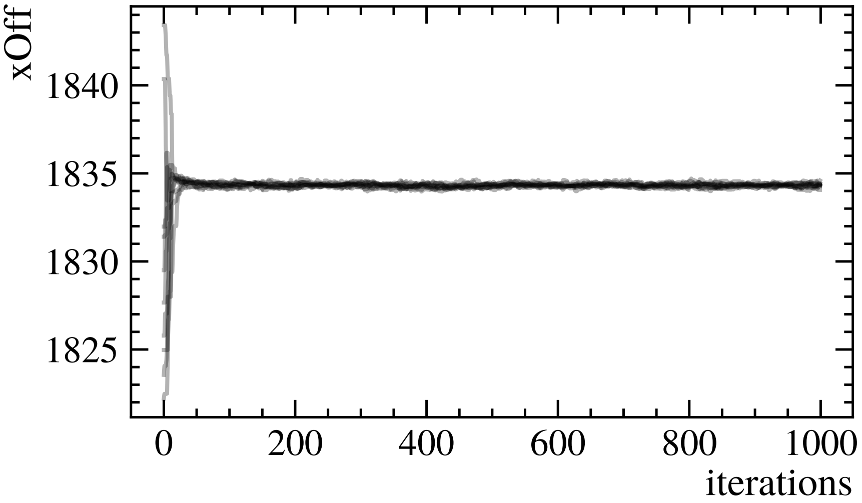

Figure 13: Convergence of different walkers for the parameter μ

7 Comparison between handwritten mcmc and off-shelf method

In case of off-shelf method, number of iterations in burn-in phase is around 50 - 100. But for the handwritten case, number of iterations in burn in is around 2000. One possible reasons maybe the choice of prior. In case of off-shelf method, choice of prior is the gaussian distribution whose mean is around the best-fit values calculated from the optimization process. So the prob. of the points near to the mean will be higher and the points will then try converge to the posterior distribution quickly. On the other hand, gaussian distributions of

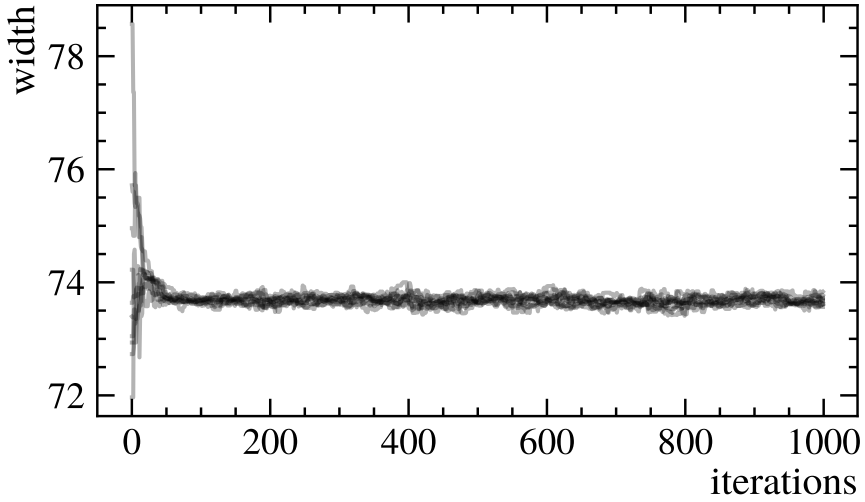

Figure 14: Convergence of different walkers for the parameter σ

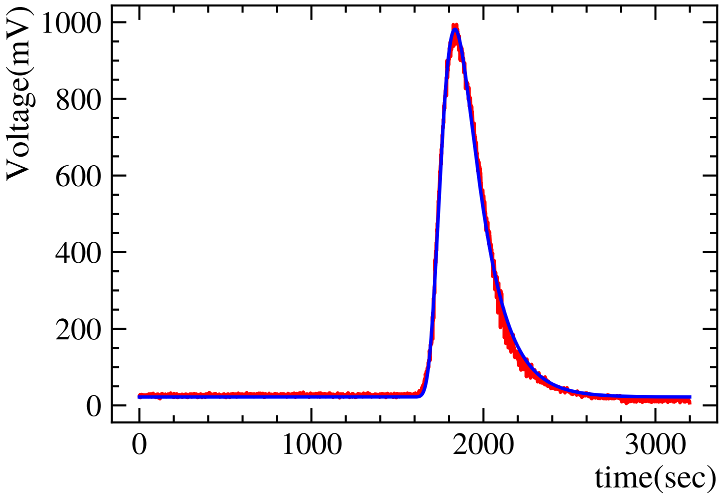

Figure 15: Datafitting using optimization process

mean = 1 and standard deviation = 1000 are priors for handwritten mcmc. This is approximately uniform distribution.