PHYS 512 Problem Set 4

Teophile Lemay, 281081252

1

1.1 a)

Newton's method is a way to perform a non-linear least squares fit to some data by using the gradient of the model in parameter space to update the fit. Fitting the Lorentzian

$$d(t) = \frac{a}{1 + \frac{(t-t_0)^2}{w^2}}$$

requires the following derivatives:

$$\frac{\partial d}{\partial a} = \frac{1}{1 + \frac{(t-t_0)^2}{w^2}}$$
$$\frac{\partial d}{\partial t_0} = \frac{\frac{2a(t-t_0)}{w^2}}{\left(1 + \frac{(t-t_0)^2}{w^2}\right)^2}$$
$$\frac{\partial d}{\partial w} = \frac{\frac{2a(t-t_0)^2}{w^3}}{\left(1 + \frac{(t-t_0)^2}{w^2}\right)^2}$$

which make up the gradient of d in parameter space:

$$\boldsymbol{\nabla} d = \begin{pmatrix} \frac{\partial d}{\partial a} \\ \\ \frac{\partial d}{\partial t_0} \\ \\ \frac{\partial d}{\partial w} \end{pmatrix}$$

For any parameter guess $\mathbf{m}$ evaluating the gradient over an interval and comparing to the data, Newton's method gives a parameter update according to

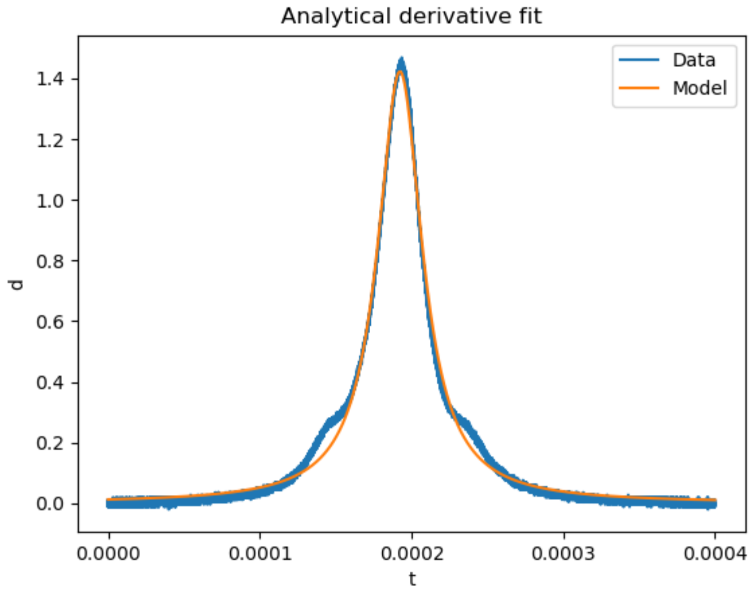
$$\boldsymbol{\nabla} d(\mathbf{m})^T N^{-1} \boldsymbol{\nabla} d(\mathbf{m}) \delta\mathbf{m} = \boldsymbol{\nabla} d(\mathbf{m})^T N^{-1}(\text{data} - d(\mathbf{m}))$$
$$\delta\mathbf{m} = (\boldsymbol{\nabla} d(\mathbf{m})^T N^{-1} \boldsymbol{\nabla} d(\mathbf{m}))^{-1} \boldsymbol{\nabla} d(\mathbf{m})^T N^{-1}(\text{data} - d(\mathbf{m}))$$
$$\delta\mathbf{m} = (\boldsymbol{\nabla} d(\mathbf{m})^T \boldsymbol{\nabla} d(\mathbf{m}))^{-1} \boldsymbol{\nabla} d(\mathbf{m})^T (\text{data} - d(\mathbf{m}))$$

Where N is a noise matrix which can be omitted by setting it to identity. Iterating Newton's method until the parameter update $\delta\mathbf{m}$ becomes smaller than some threshold gives the best fit parameters.

Figure 1 shows the result of a Newton's method fit of the data in "sidebands.npz" to the Lorentzian using the analytical gradient derived above. The iterations were stopped once the magnitude of the parameter update δm reached the threshold of 10^{-10} which took 13 steps. The initial values were chosen by inspection of the data to be $(a = 10, t_0 = 0.0002, w = 0.0001)$. The best fit parameters for the Lorentzian are

$$a = 1.423$$
$$t_0 = 1.924 \cdot 10^{-4}$$
$$w = 1.792 \cdot 10^{-5}\ .$$

Figure 1: Analytical Newton's method fit



1.2 b)

For non-linear least squares fitting, we know

$$\nabla A^T N^{-1} \nabla A \delta m = \nabla A^T N^{-1}(d - A(m))$$

where A is the model, N is a diagonal matrix such that $N_{i,i} = \sigma_i^2$ (assuming random Gaussian noise in data), d is the measurement data, and m are model parameters. Setting $m \to m_t$ the "true" parameters such that $A(m_t) = d_t$ where d_t is the noiseless data, the equation above can be rearranged to

$$\delta m = (\nabla A^T N^{-1} \nabla A)^{-1} \nabla A^T N^{-1}(d - d_t) \ .$$

Obviously, $d - d_t = n$ is the noise in the data so

$$\delta m = (\nabla A^T N^{-1} \nabla A)^{-1} \nabla A^T N^{-1} n$$

Following the same steps as for linear least squares fitting, the covariance of the parameters $\left\langle (\delta m)^2 \right\rangle$ is

$$\left\langle (\delta m)^2 \right\rangle = (\nabla A^T N^{-1} \nabla A)^{-1}$$

Figure 2 shows a histogram of the difference between the data and the model. Assuming the best fit parameters are close to m_t this also describes the noise in the data. Since the noise is not centered at 0, I chose to estimate the noise at each data point as the root mean square of the difference between the model and the data. Thus, the resulting N matrix is a diagonal matrix with all entries equal to $rms(\text{model} - \text{data})$. Evaluating the error in each parameter as the square roots of the diagonal elements of the covariance matrix gives error

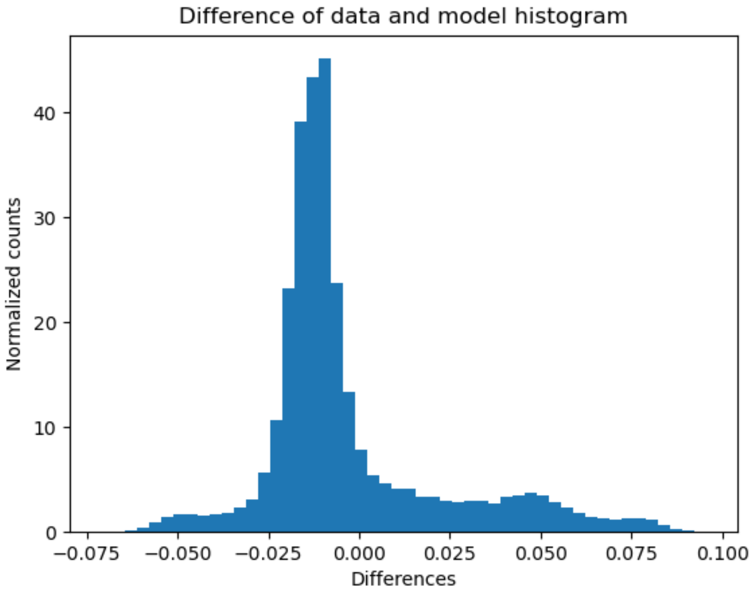
$$\sigma a = 2.678 \cdot 10^{-3}$$

$$\sigma t_0 = 3.373 \cdot 10^{-8}$$

$$\sigma w = 4.777 \cdot 10^{-8} \ .$$

All code for parts a) and b) can be found in the file "Q1_analytical_lorentzian.py".

Figure 2: Noise distribution for analytical Lorentzian fit



1.3 c)

To fit the single Lorentzian model to the data using numerical derivatives, I use a central derivative to calculate partial derivatives with respect to each parameter which make up the gradient. The step size is not optimized for each numerical derivative. Generally, the ideal step size for a numerical central difference derivative of $f(x)$ is given by

$$h \approx \left(\frac{\epsilon f(x)}{f'''(x)}\right)^{1/3} x$$

where ϵ is the machine precision (Numerical Recipes, third edition). My implementation assumes that $f/f''' \approx 1$ which simplifies the step size choice to $h = (\epsilon)^{1/3}x$. As shown in figure 3, the numerical derivative fit of the model produces a very similar fit to the fit made using analytical derivatives. This suggests that the assumption made for step size choices did not introduce significant errors. The best fit parameters from the numerical Newton's method fit are

$$a = 1.423$$
$$t_0 = 1.924 \cdot 10^{-4}$$
$$w = 1.972 \cdot 10^{-5}\ .$$

The best fit parameters from the analytical and numerical Newton's method fits are nearly identical. Taking the difference of the un-rounded values gives

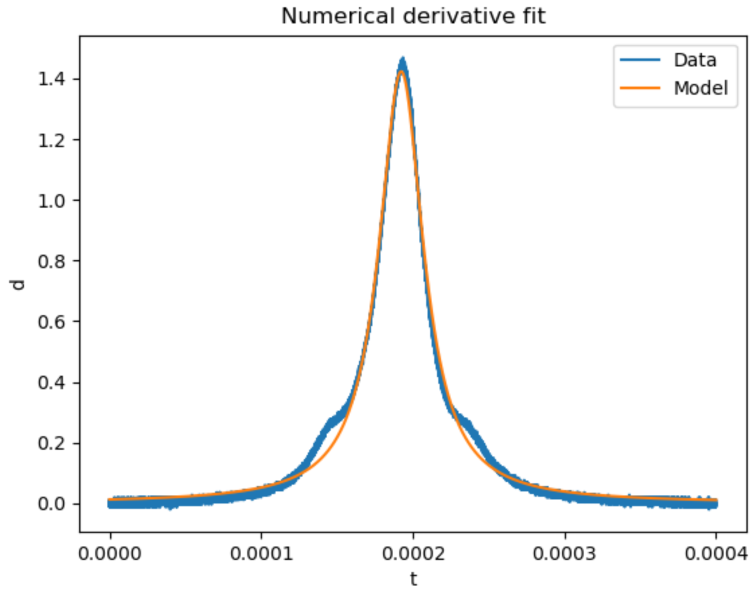
$$\delta a = 2.3 \cdot 10^{-12}, \quad \delta t_0 = 9.1 \cdot 10^{-16}, \quad \delta w = 5.8 \cdot 10^{-17}.$$

Since the numerical fit parameters fall well within the ranges of the uncertainties for the analytical fit parameters, the answers are not statistically different from each other. Code for part c) can be found in "Q1_numerical_lorentzian.py".

1.4 d)

The numerical fit for the sum of three Lorentzians model was performed in the same way as for the single lorentzian model (figure 4) and error was estimated using the same procedure as used in part b). The best fit parameters for

Figure 3: Numerical derivative fit



the triple Lorentzian model are

$$a = 1.443$$

$$t_0 = 1.926 \cdot 10^{-4}$$

$$w = 1.607 \cdot 10^{-5}$$

$$b = 0.1039$$

$$c = 0.06473$$

$$dt = 4.457 \cdot 10^{-5}.$$

The parameter errors for this fit are

$$\sigma a = 0.00221$$

$$\sigma t0 = 2.61 \cdot 10^{-8}$$

$$\sigma w = 4.68 \cdot 10^{-8}$$

$$\sigma b = 0.00210$$

$$\sigma c = 0.00206$$

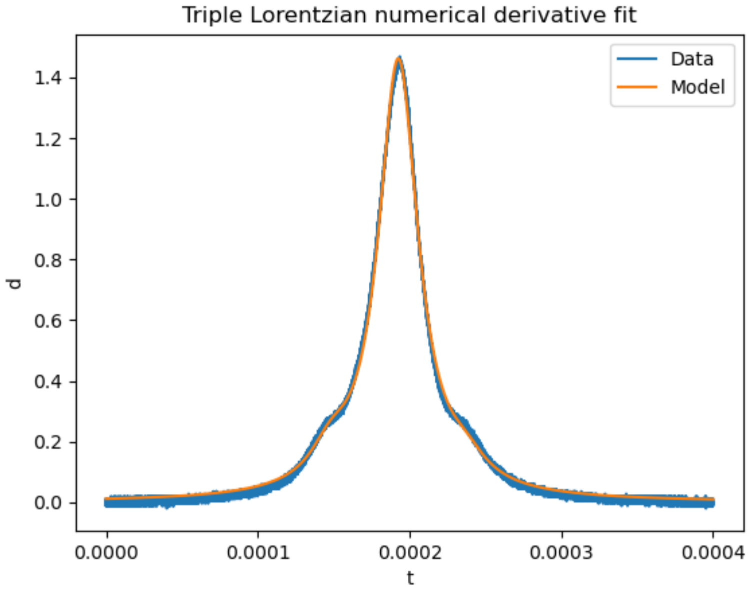
$$\sigma dt = 3.15 \cdot 10^{-7}.$$

Code for this part can be found in ”Q1_triple_lorentzian.py”.

1.5 e)

Figure 5 shows the residuals for the triple Lorentzian model. Based on the figure, the noise is clearly correlated which breaks the assumptions used for the least-squares fit. Therefore, I do not believe that the model does not perfectly describe the data. In addition, the error estimation for the best fit parameters assumes uncorrelated Gaussian noise wit uniform variance. Figure 5 shows that this is clearly not the case, so the error bars for the parameters are likely erroneous.

Figure 4: Triple Lorentzian numerical fit



1.6 f)

I don't really have time to do this one unfortunately. I would love to understand it better though!

1.7 g)

I performed an MCMC fit of the model over 100000 steps. Figure 6 shows the evolutions of all 6 parameters for the triple Lorentzian model, and all parameters converge around their final value, albeit with non-uniformly random fluctuations at convergence. Figure 7 shows the result of the MCMC fit which has parameters

$$a = 1.460$$

$$t_0 = 1.926 \cdot 10^{-4}$$

$$w = 1.597 \cdot 10^{-5}$$

$$b = 0.1143$$

$$c = 0.07192$$

$$dt = 4.449 \cdot 10^{-5}$$

with error

$$\sigma a = 0.171$$

$$\sigma t_0 = 7.03 \cdot 10^{-7}$$

$$\sigma w = 1.06 \cdot 10^{-6}$$

$$\sigma b = 0.0475$$

$$\sigma c = 0.0424$$

$$\sigma dt = 6.48 \cdot 10^{-6}.$$

Code for this question can be found in "Q1_mcmc.py"

1.8 e)

Figure 5: Triple Lorentzian residuals

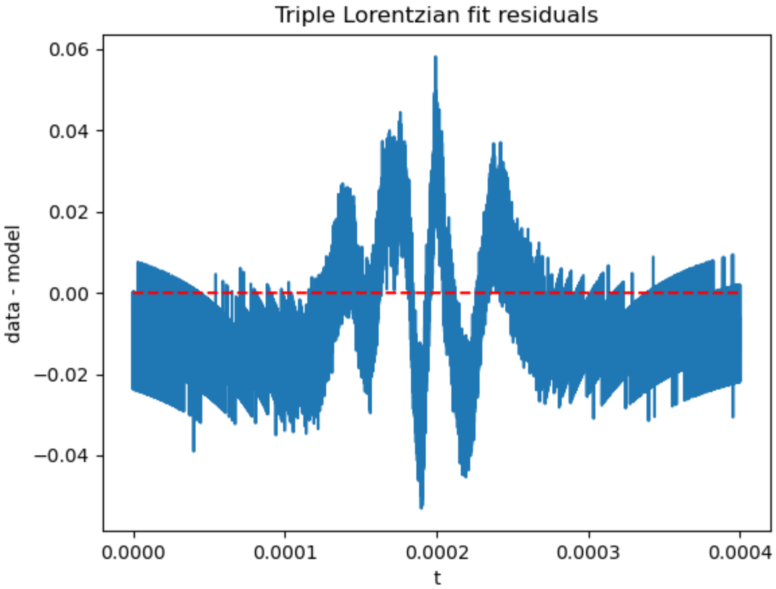


Figure 6: MCMC parameter evolution

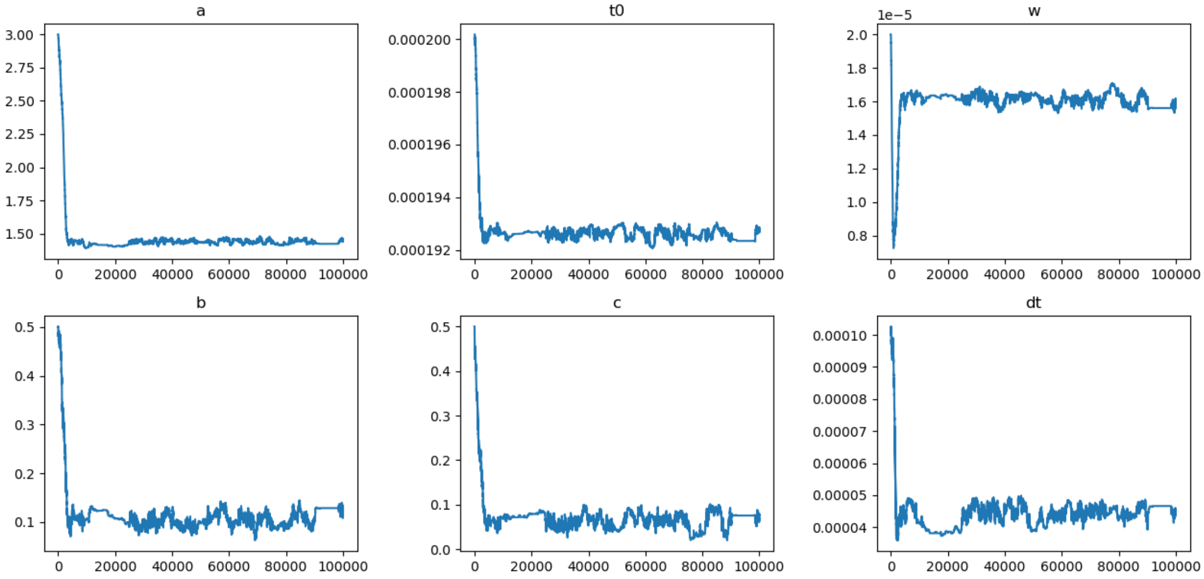
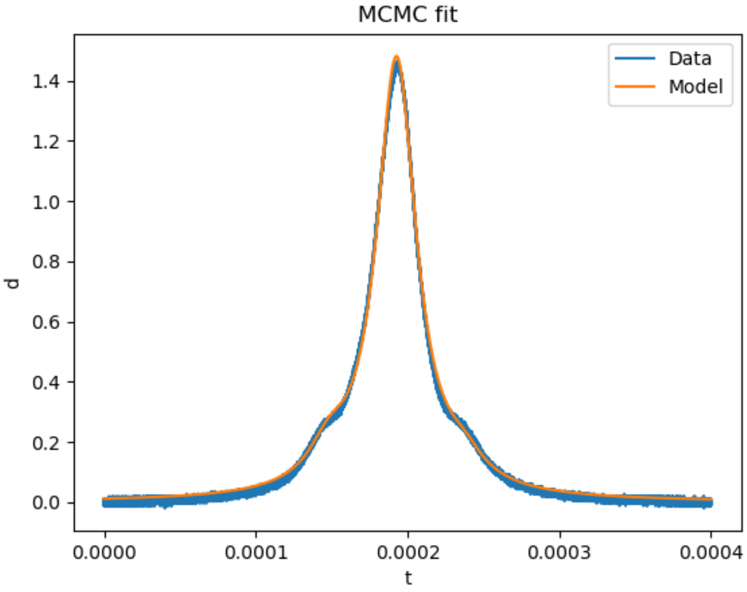


Figure 7: Triple Lorentzian MCMC fit


MCMC fit
Data
Model
d
t