

1

As a preface. All of this is wrong. I don't know what exactly went wrong, but I suspect it is somewhere in my lorentz and gradient implementation

1.1

Plotting the prediction after 5, using the initial parameters $a = 1.4$, $w = 2 \cdot 10^{-5}$, $t_0 = -2 \cdot 10^{-5}$, which are of the same order of magnitude and somewhat close to the parameters output by `scipy.optimize.curve_fit`. This is still very far from a good lorentzian:

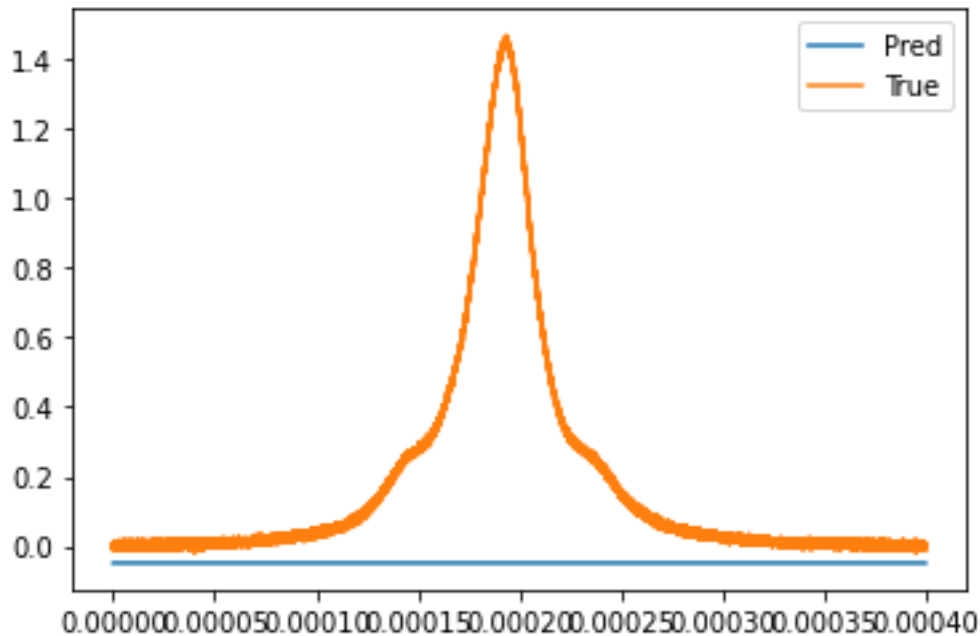


Figure 1: Expected value and signal over time

1.2

Plotting the simple residuals, we can see that it is indeed a bad fit:

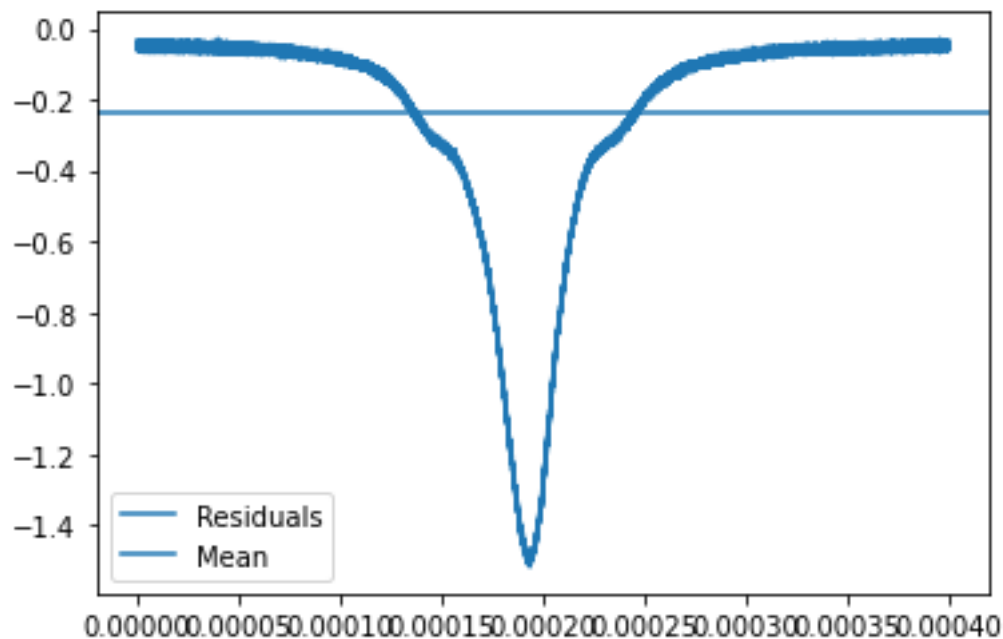


Figure 2: Residuals and mean of residuals

Event though the mean is not terrible, there is obviously a pattern to our residuals, since the fit is not good

1.3

Changing the function, we see that our function is quite similar to the first one, very far from the true values, and its residuals have the same shape as the first ones:

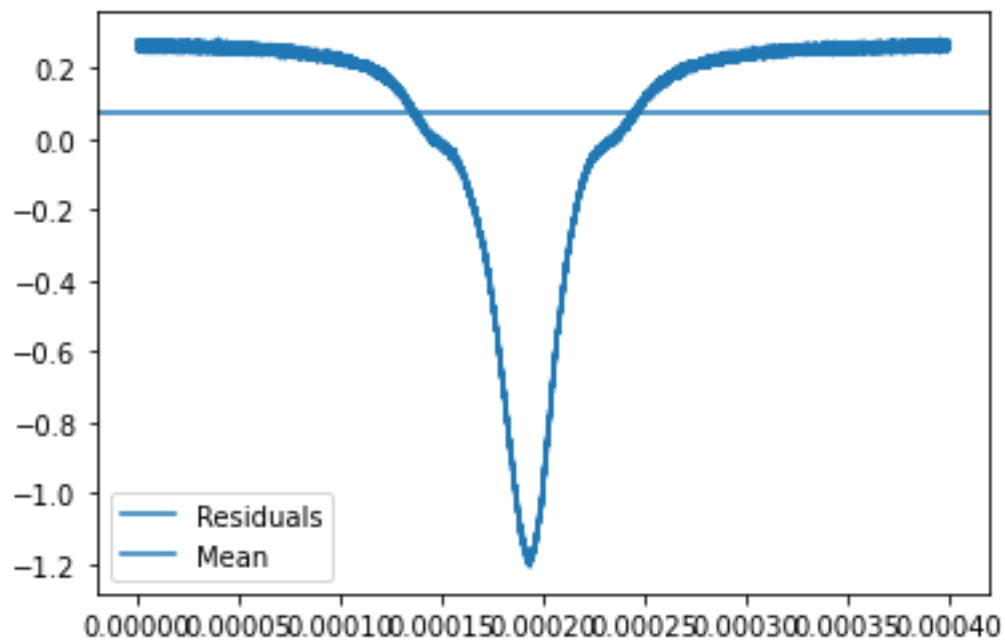


Figure 3: Residuals of numerical derivatives

1.4

Modelling our three-peak lorentzian, we see that there is very little change in our signal or residuals:

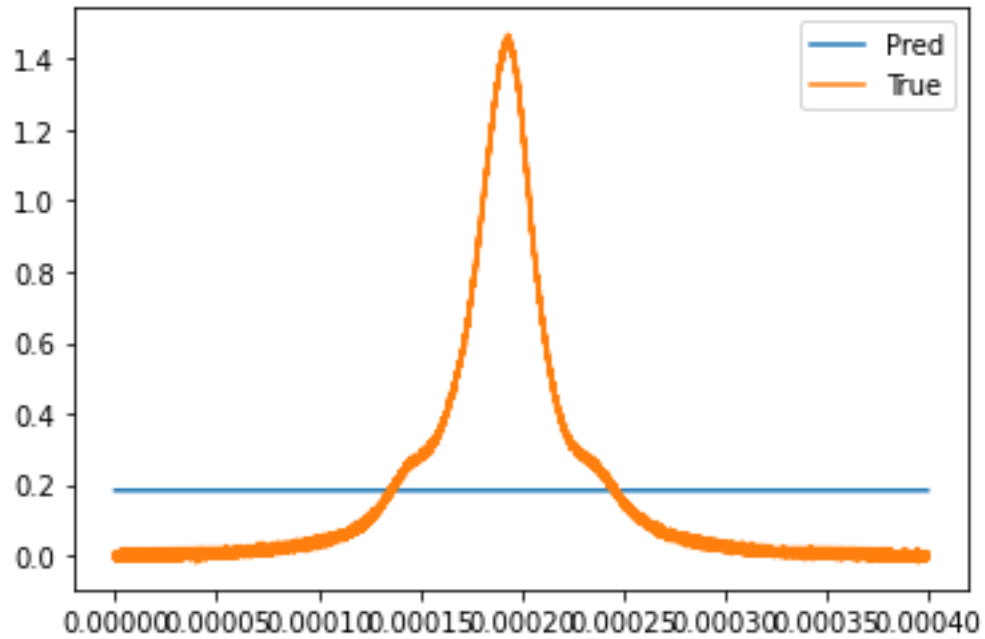


Figure 4: Expected and true values for our three-peak lorentzian

With similar residuals to the rest

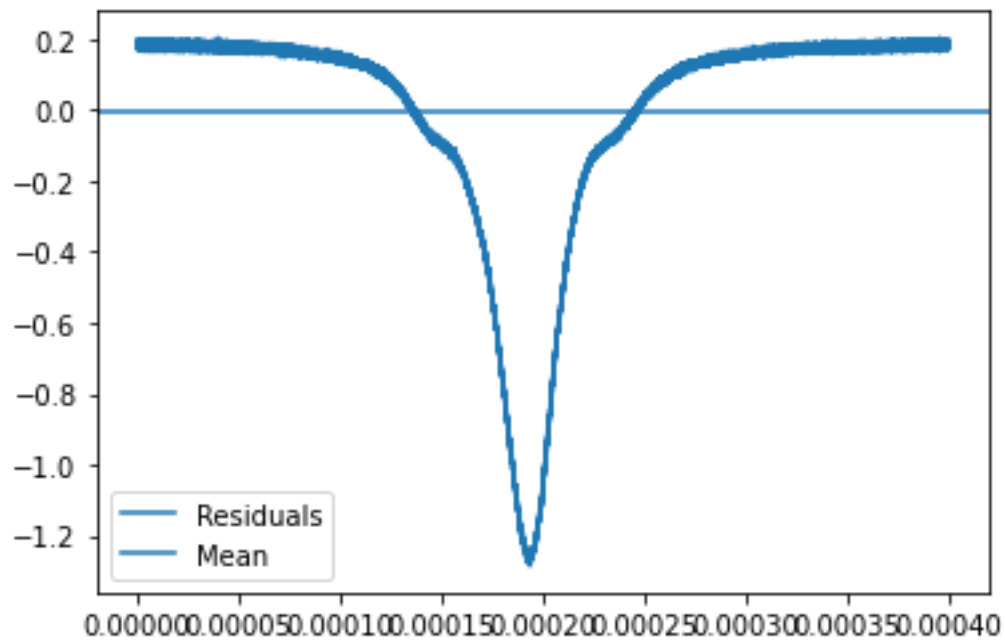


Figure 5: Residuals of three-peak lorentzian

All in all bad fits for some reason that escapes my grasp. I did not think it was wise to add a lot of other features to this code given how badly it is behaving on some very standard inputs like a test lorentzian.