PHYS512 - HW2

Damien Pinto (260687121)

October 2019

1:

(a) We must go to eighth order in our Chebyshev polynomial fit to obtain an error ≈We can see from the graph of the errors in Fig. 1 and Tab. 1, it seems that the Chebyshev fit offers a lower error on the extremities of the data given, and a lower maximum error, but for the trade-off of having slightly higher errors across the mid-section of the fit/data, which leads to a higher root mean squared error.

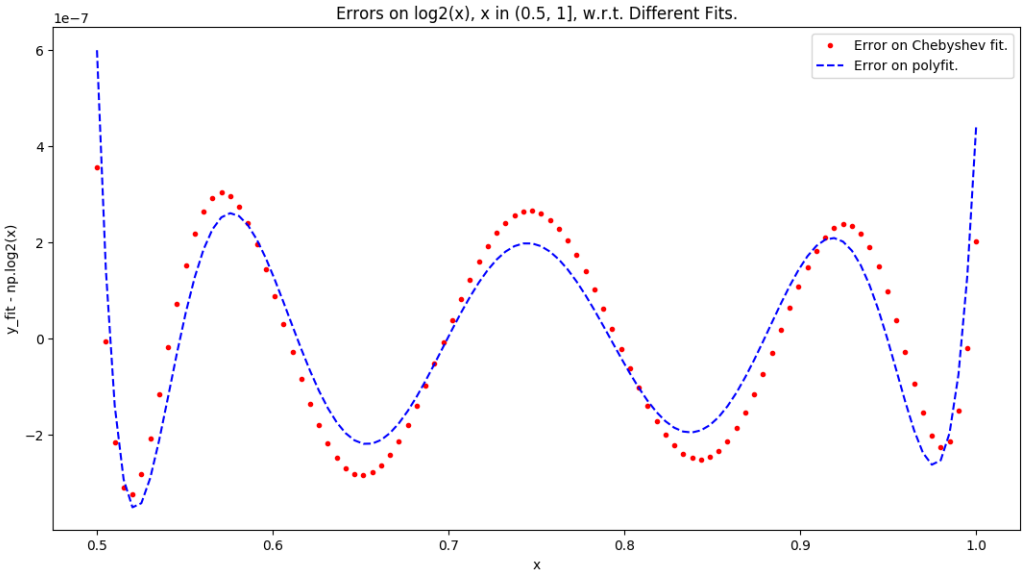

Figure 1: *Plot of erros associated with custom Chebyshev fit method (red) and np.polyfit (blue) applied to* $log_2(x)$, $x \in [0.5, 1]$ *(both of order 8)*

	Chebyshev	polyfit
Max Error	$3.58 \cdot 10^{-7}$	$6.00 \cdot 10^{-7}$
RMS Error	$1.93 \cdot 10^{-7}$	$1.80 \cdot 10^{-7}$

Table 1: *Table of maximum and roo mean squared (RMS) error for each (8th order) fit examined.*

(b) We can pretty much give our function "get_Cheby_mtx" the same range for x that it always needs, $x \in [-1, 1]$, no matter what the actual values of our independent variable are, as long as both ranges

are partionned into the same number of points. The issue comes from our data for $y = log_2(x)$ being outside the range $-1 \leq y \leq 1$, the valid range for a Chebyshev fit.

The function np.frexp expresses any number num in the form:

$$num = m \cdot 2^{exp}, \tag{1}$$

where m is the *mantissa*, bassicaly a "amplitude modulator" of the power of 2 that multiplies it. $m \in (-1, 1)$, always, because we just adjust exp such that m has the value of smallest magnitude necessary to represent the quantity we want. This means that, with the functions we are currently using as an example:

$$\begin{aligned} y &= \log_2(x) \\ &= \log_2(m \cdot 2^{exp}) \\ &= \log_2(m) + exp. \end{aligned}$$

This means, with an array of y_i data, if we can keep track of all the exp_i that go along with them, then we can just focus on fitting the $log_2(\frac{m_i+3}{4})$ (which will restrict the values of m to between 0.5 and 1) and recombining the two after the whole fitting process.

2:

(a) If we're using an exponential decay model, then the model isn't linear given that exponential functions don't follow the requisite rules for linearity (eg. $\exp ax + bx \neq \exp ax + \exp bx$).
Looking at the data, we know that, if we use a decaying exponential, we'll need an t-offset (t_0) to recenter our function at the peak since it is fairly advanced in the data. We can also see that we need a constant y-offset (y_0) to compensate for the visible background noise. Then we have the general parameters for an exponential decay: an amplitude A and rate of decay in time (B). So,

$$y(t) = A \exp \frac{-(t - t_0)}{B} + y_0 \tag{2}$$

I'm only going to look at the data starting at the summit of the spike as I think we can safely assume that that is where the physics/flux from a stellar flare modeled by an exponential decay that we're detecting is actually operating (and not at the beginning of the data where we're getting noise fluctuations).
We can see that the background noise sort of oscillates around ≈ 1.0, so we can set y_0 as that. For A, I just too the maximum value of of our data minus y_0. For x_0 I used the x-value at the same index as the maximum y-value. I thought of using a power to apply to the exponential (i.e. $(t-t_0) \to (t-t_0)^p$), but the data seems to indicate just $p = 1$. Finally, for the decay rate B, I looked at the maximum value of our data (≈ 1.25), where that happened ($t \approx 1706.52$), and where the exponential component decreased its contribution to half ($y \approx 1.125, x \approx 1706.54$). This means that:

$$\begin{aligned} \frac{1}{2} &= \exp \frac{-0.02}{B} \\ \log(\frac{1}{2}) &= \frac{-0.02}{B} \\ B &= \frac{-0.02}{\log(\frac{1}{2})} \\ &\approx 0.03. \end{aligned}$$

Using this, we get a guess that looks like Fig. 2.

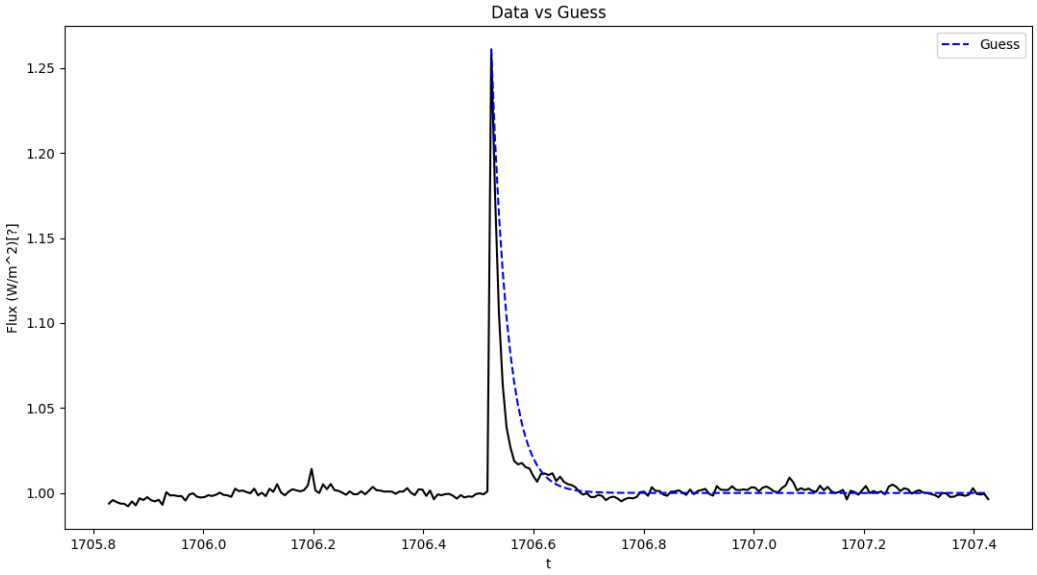

Figure 2: *First guess for exponential decay that models M-dwarf flare data.*

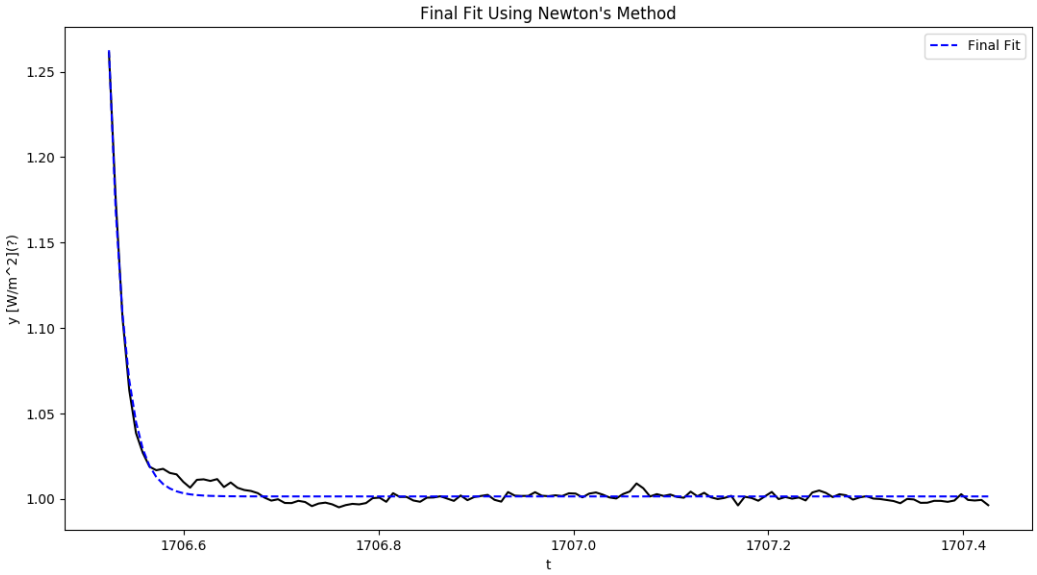

Figure 3: *Plot of best fit obtained with Newton's method.*

Table 2: Table of Fit Parameters

Best Fit Parameters	
A	1.263 ± 0.003
B	0.0155 ± 0.0003
y_0	1.0001 ± 0.0003
χ^2	0.0016
Condition Number*	146.5
Max Error	0.01
RMS Error	0.004

* The ratio of the largest and smallest eigenvalues of the matrix of derivatives of our model w.r.t. fit variables.

(b) Using this as an initial guess for program that uses Newton's method of gradient descent, we get the results seen in Fig. **??** and Tab. **??**.

For what it's worth, visually the fit looks reasonable, and the fit parameters numerically seem to resemble what we would expect them to. One worrisome value is how small the χ^2 value is. Perhaps I have over-determined the problem, although it's hard to see how given that all the fit parameters accomplish different mathematical roles in shaping the model function. The condition number doesn't seem astronomically large and so that bodes well. The Max and RMS errors seem reasonable also given the scale of the original data. If we consider that the flare-up is roughly "perfect" in it's form, and that most noise is provided by the noise inherent in the data-taking process, then we can look at the scale of the noise before the event to get an idea of what the uncertainties are. The standard deviation of said data is ≈ 0.003, and the errors on our fit parameters are of similar orders of magnitude. This makes some sense as this noise would probably be one of the main sources of uncertainty in the magnitude of the flare's peak (A), and is the main reason for the parameter y_0. The first of these two has an error of similar magnitude, and the second an order less (which I think makes sense as all the point of the data after $t \approx 1706.65$ serve to determine its value). The effect of the general noise on B is a little too tricky for me to be able to propose a relation between the two. This standard deviation of the previous was used to calculate the errors on the fit parameters actually, because, as we saw in class:

$$\begin{aligned}(m - m_t)(m - m_t)^T &= (A^T N^{-1} A)^{-1} \\ &= \frac{\partial \chi^2}{\partial m},\end{aligned}$$

which forms a matrix of the correlations of the various fit parameters. Each value on the diagonal of this matrix can be taken as an estimation of the variance on the value of one of the fit parameters. This is how the errors on each parameter were found using the data before the flare (and assuming that is representative of the general noise in the system).

(c) Looking at the whole data set makes me confident that the precision of the y_0 variable is merited as it seems to reliably oscillate around 1.0. Given the number of points that are only affected by this noise it makes sense that it is possible to determine the mean of this noise to a high degree of precision. The precision in A could be put in doubt as stellar the event evidently causes a quite sudden spike in the data. What maximum value the system/instrument being used detects could very much be affected by the sampling rate, and we only have one point to determine it, so I'm not sure the high degree of precision here is warranted physically, although it might be numerically. I'm not entirely sure what to say about B, other than there definitely seems to be a "wiggle" at the base of the spike in our data (who's presence can be seen slightly in the residuals in Fig. 4), meaning there is a noteworthy chance that there is a certain oscillating relationship in our data that our simplistic decaying exponential

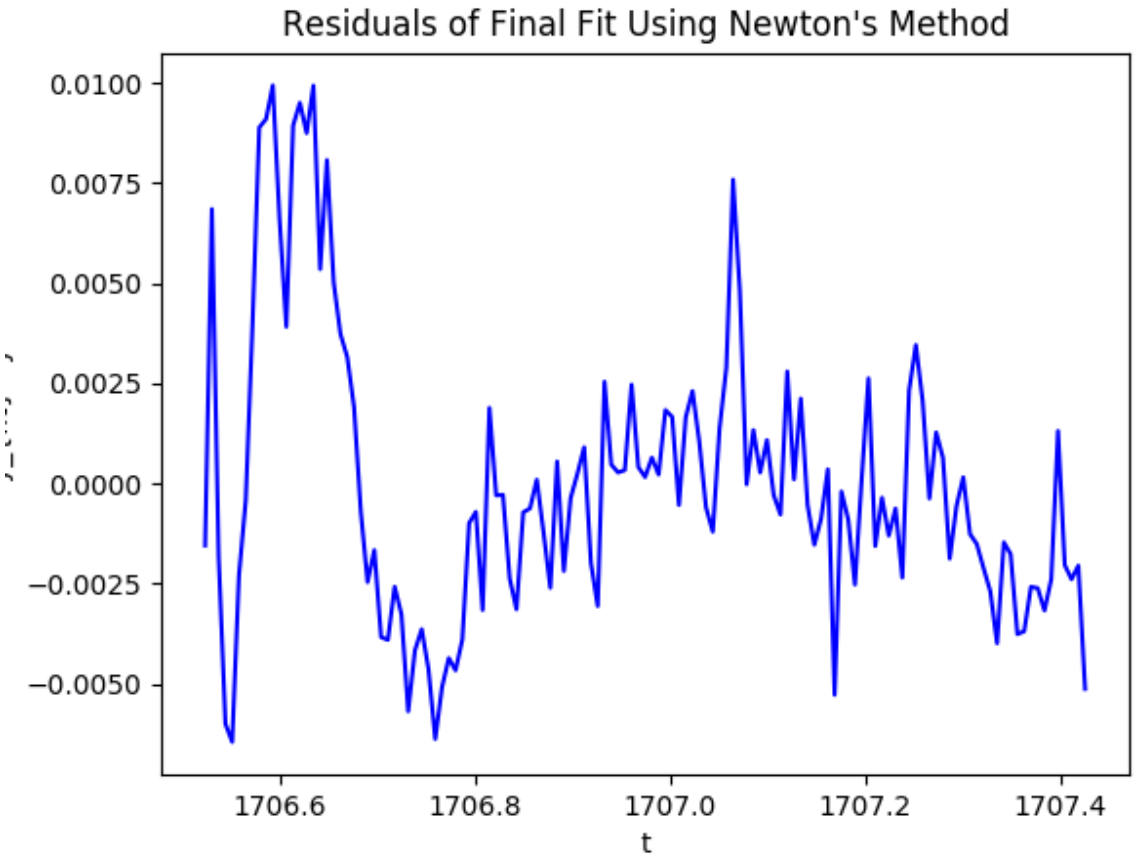

Figure 4: *Residuals between the best fit and the true values in our data.*

model doesn't capture, in which case our fitting algorithm did the best it could. In that case, the degree of precision we obtained might not be totally warranted.