Phys 512 - PS 3

Fernanda C. Rodrigues Machado

ID# 260905170

October, 2020

P1

We can expand the equation for the paraboloid and make a parameter substitution to get a function that is linear on its parameters:

Expand and change parameters

```
In[145]:= z == a ((x - x0)^2 + (y - y0)^2) + z0;
          Expand[%]
          Collect[Collect[%, -2 a], a]
          Collect[
           Expand[% /. {x0^2 + y0^2 -> (m0 - z0) / a} /. {x0 -> m1 / (-2 a)} /.
              {y0 -> m2 / (-2 a)} /. a -> m3], m3]
```

Out[146]= $z == a x^2 + a y^2 - 2 a x x_0 + a x_0^2 - 2 a y y_0 + a y_0^2 + z_0$

Out[147]= $z == a \left(x^2 + y^2 + x_0^2 + y_0^2 - 2 (x x_0 + y y_0)\right) + z_0$

Out[148]= $z == m0 + m1\, x + m2\, y + m3 \left(x^2 + y^2\right)$

The new parameters $(m0, m1, m2, m3)$ can be written in terms of the old ones (x_0, y_0, z_0, a) as

```
In[149]:= Solve[x0^2 + y0^2 == (m0 - z0) / a, m0]
          Solve[x0 == m1 / (-2 a), m1]
          Solve[y0 == m2 / (-2 a), m2]
          m3 -> a
```

Out[149]= $\{\{m0 \to a x_0^2 + a y_0^2 + z_0\}\}$

Out[150]= $\{\{m1 \to -2 a x_0\}\}$

Out[151]= $\{\{m2 \to -2 a y_0\}\}$

Out[152]= $m3 \to a$

and the old ones can be recovered by substituting:

```
In[157]:= Solve[{x0^2 + y0^2 == (m0 - z0) / a, x0 == m1 / (-2 a), y0 == m2 / (-2 a)},
  {x0, y0, z0}]
```

$$\text{Out[157]=}\ \left\{\left\{x_0 \to -\frac{m1}{2\,a},\ y_0 \to -\frac{m2}{2\,a},\ z_0 \to -\frac{-4\,a\,m0 + m1^2 + m2^2}{4\,a}\right\}\right\}$$

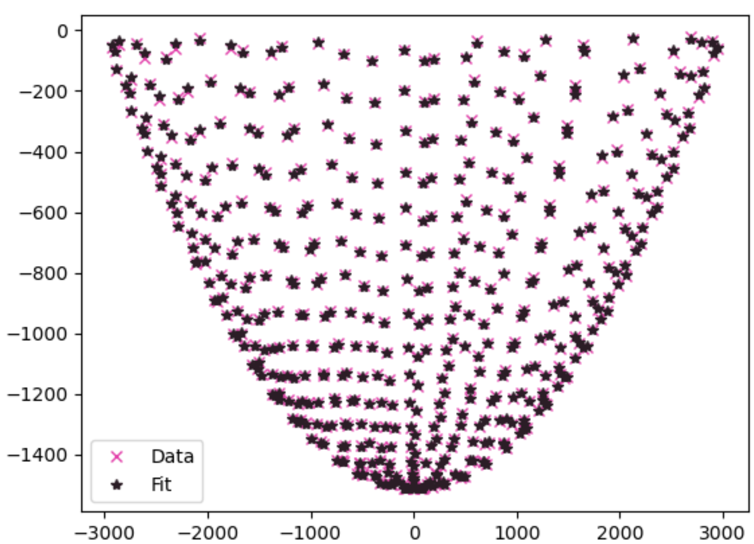
Carrying out the fit, my best-fit new parameters are:

$$\begin{aligned} m0 &= -1512.31182 \\ m1 &= 4.53599028 \times 10^{-4} \\ m2 &= -1.94115589 \times 10^{-2} \\ m3 &= 1.66704455 \times 10^{-4}\ . \end{aligned}$$

The old parameters are then given by:

$$\begin{aligned} x_0 &= -1.3604886221970875 \\ y_0 &= 58.22147608157978 \\ z_0 &= -1512.8772100367878 \\ a &= 0.00016670445477401347\ . \end{aligned}$$

Plotting the fit and the real data, I get:



with an RMS error between fit and data equal to 3.7683386487847277.

The covariance matrix $N = (A^T N^{-1} A)^{-1}$ gives the correlated noise $\langle n_i n_j \rangle$. Since we have $a = m3$, the uncertainty in a is given by $n_3 = \sqrt{\langle n_3 n_3 \rangle}$, where $\langle n_3 n_3 \rangle$ is just the last input of the matrix N. In this way, I get the uncertainty $\delta a = 1.712133743432042 \times 10^{-8}$.

The focal length in the paraboloid is given by

$$f = \frac{1}{4a}$$

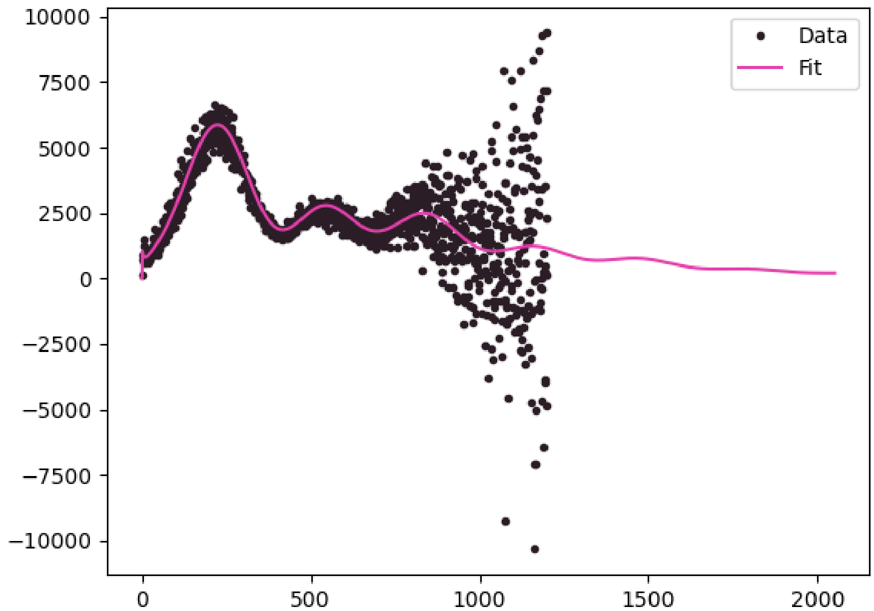
and its uncertainty (by error propagation) is

$$\delta f = \left| \frac{\delta a}{4a^2} \right| .$$

Substituting values, we get $f = 1.49966 \pm 0.00015$ m.

P2

With the given parameters, we get a $\chi^2 = 1588.2376465828454$.



P3

Using Newton's method. With optical depth fixed at $\tau = 0.05$, the best-fit values for the other parameters are:

$$\begin{aligned}
H_0 &= 69.3273685 \pm 2.40207892 \\
\omega_b h^2 &= (2.24913957 \pm 0.0539639860) \times 10^{-2} \\
\omega_c h^2 &= 0.113912236 \pm 0.00522680121 \\
A_s &= (2.04250577 \pm 0.0389837370) \times 10^{-9} \\
n_s &= 0.969769787 \pm 0.0135890077
\end{aligned}$$

and the new minimized χ^2 is equal to 1227.935636135719.

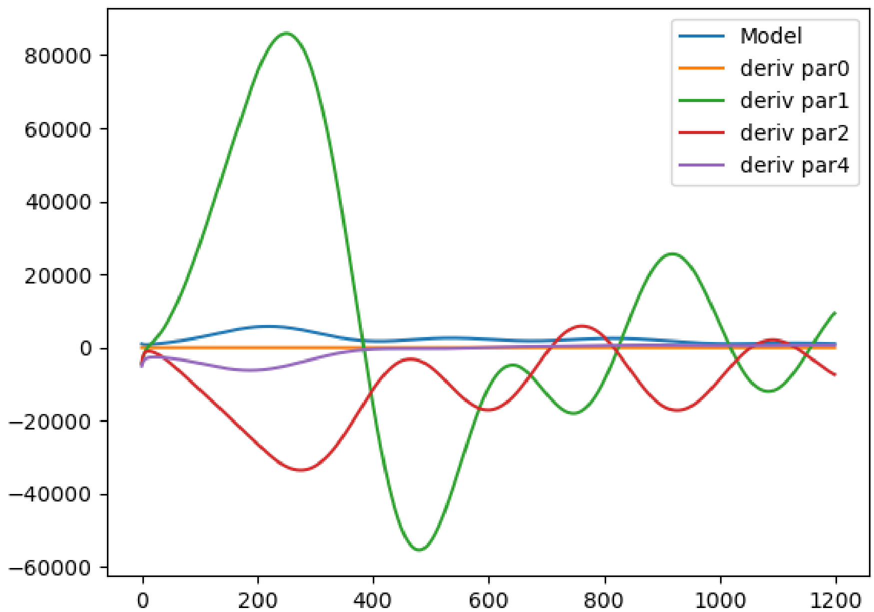


Figure 1: CMB and the derivatives for each of the parameters above, keeping tau constant. The derivative w.r.t. parameter 3 (A_s) was omitted due to its scaling.

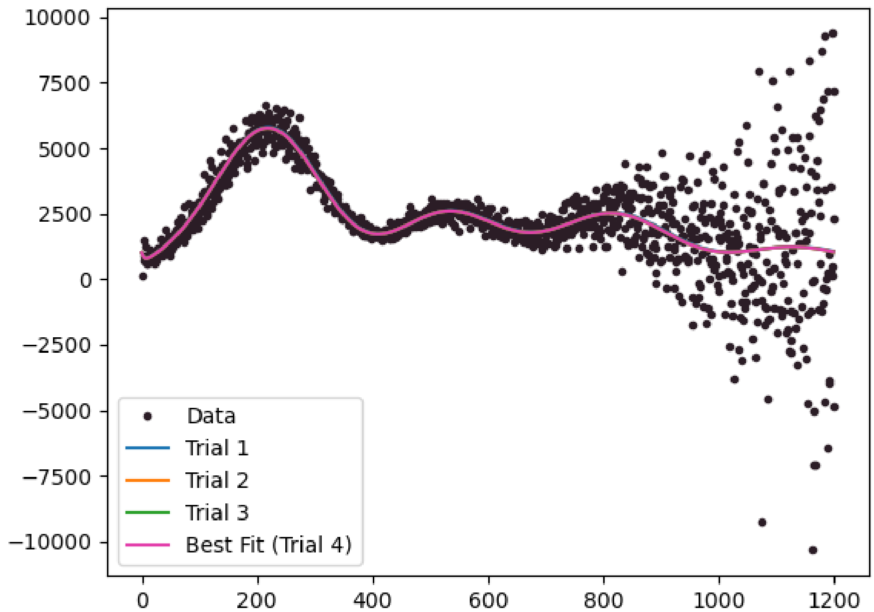


Figure 2: CMB data and 4 fits.

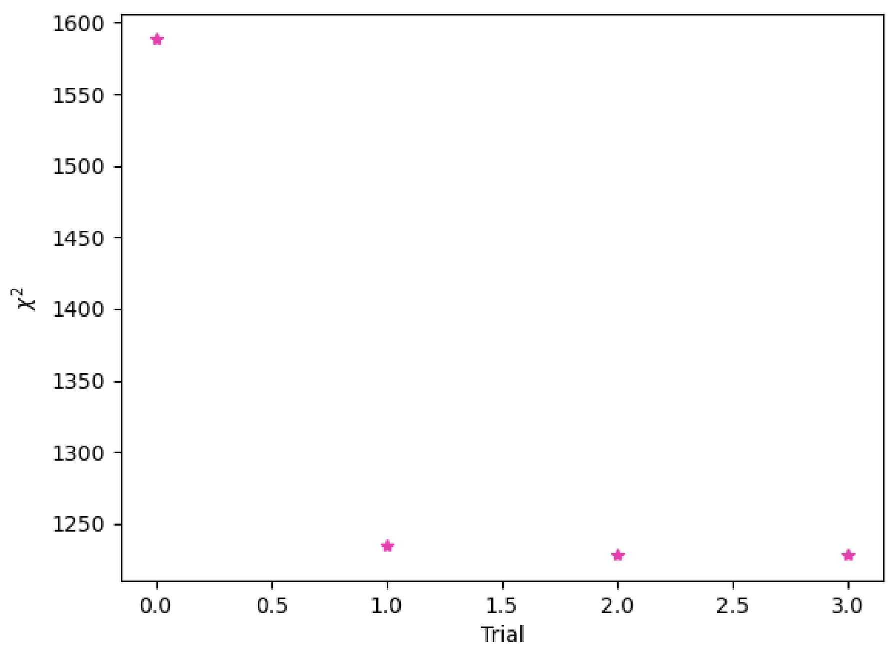


Figure 3: χ^2 for each of the 4 trials.

If I let all parameters float, I get as best-fit values:

$$
\begin{aligned}
H_0 &= 68.8015331 \pm 3.25778331 \\
\omega_b h^2 &= (2.23770286 \pm 0.0765761984) \times 10^{-2} \\
\omega_c h^2 &= 0.114822064 \pm 0.00658297176 \\
\tau &= 0.0194766363 \pm 0.150860338 \\
A_s &= (1.92249969 \pm 0.598622072) \times 10^{-9} \\
n_s &= 0.965829151 \pm 0.0228317247
\end{aligned}
$$

and χ^2 equal to 1235.5552207546873. These errors and χ^2 are worse than the case above, where τ was constant.

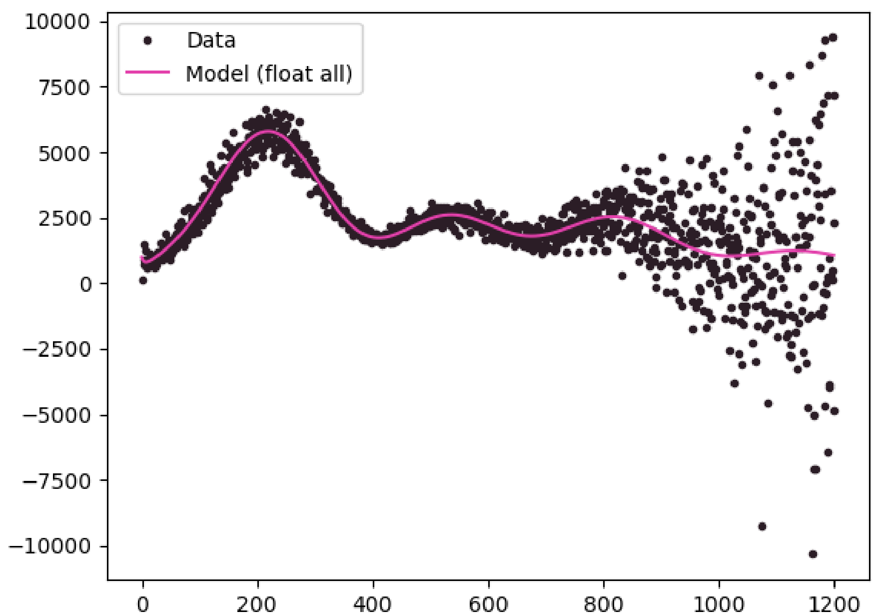


Figure 4: CMB data and fit with all parameters floating.

P4

Using a Markov-chain Monte Carlo with 5000 steps[1], I get the new parameters:

$$
\begin{aligned}
H_0 &= 70.3112543 \pm 2.62403787 \\
\omega_b h^2 &= (2.26686607 \pm 0.0588006489) \times 10^{-2} \\
\omega_c h^2 &= 0.112163971 \pm 0.00519791688 \\
\tau &= 0.0810869807 \pm 0.0473308810 \\
A_s &= (2.17328497 \pm 0.196466768) \times 10^{-9} \\
n_s &= 0.977929447 \pm 0.0154416237
\end{aligned}
$$

and χ^2 equal to 1233.099189381654. Since we could only accept steps that would give positive τ, the actual number of accepted steps was equal to 1784, giving an acceptance rate of 0.3568. When comparing to the parameters given above for Newton's method floating τ, there is a clear improvement in parameters errors.

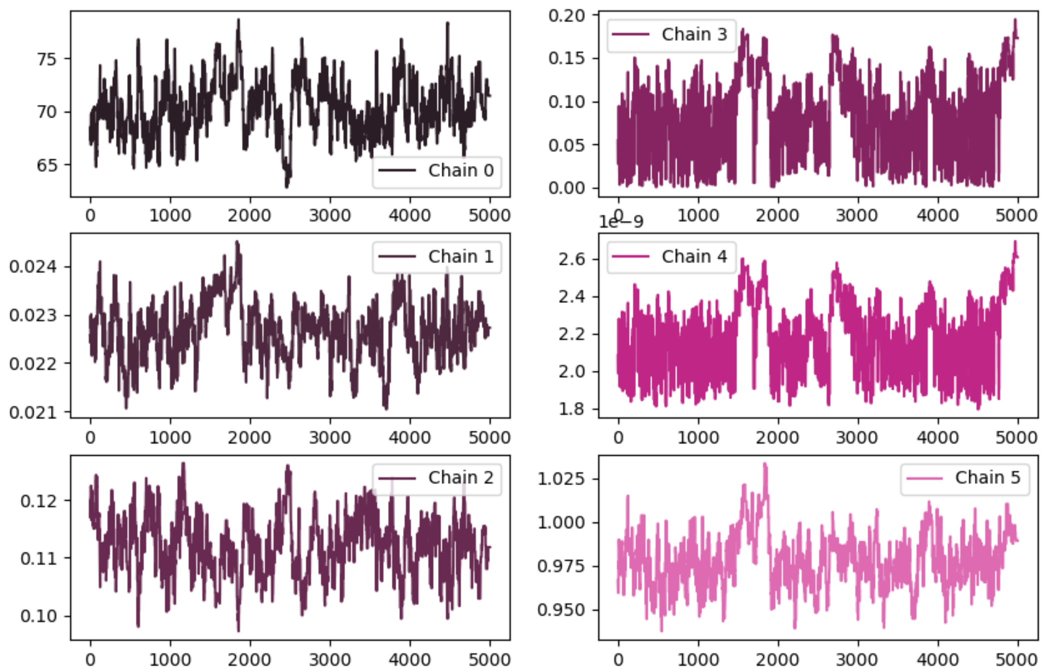


Figure 5: Chains for each parameter.

[1] I am doing only 5000 steps and only one chain because I am using a > 10 years-old MacBook, and it took around 18h to run this.

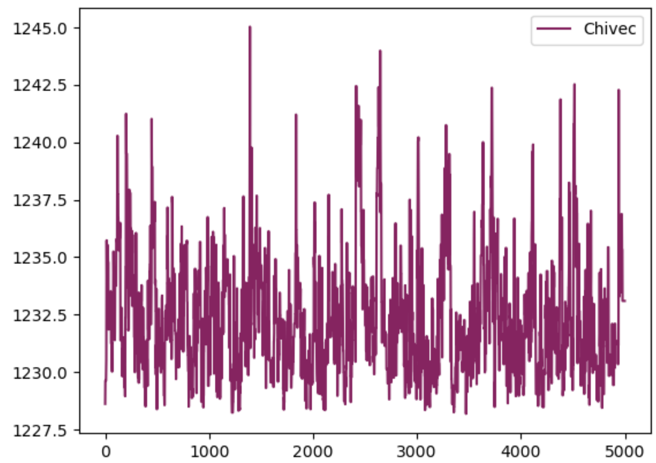


Figure 6: χ^2 evolution during chain.

From the FFTs of the chains (Fig. 7), it seems that the convergence was not very good for this number of steps. Unfortunately I don't have access to a faster computer at this moment.

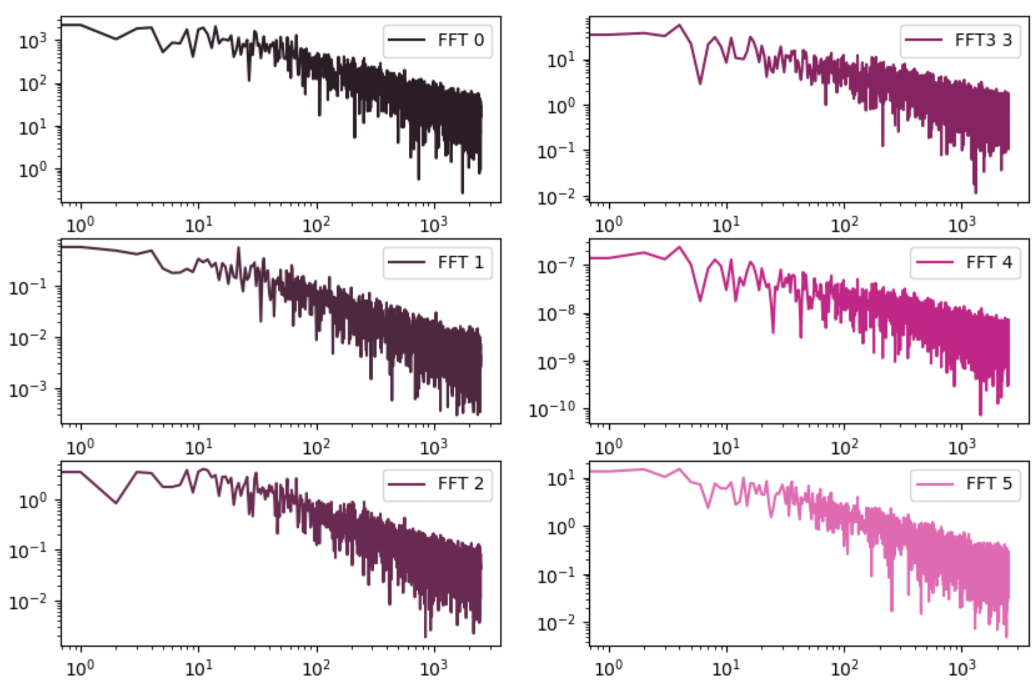


Figure 7: FFTs of chains.

P5

Now I am running the chain again with 5000 steps, but starting with τ equal to 0.0544, and accepting only values of τ that fall within 3 standard deviations of the given value (i.e., accepting only steps that get $\tau = 0.0544 \pm (0.0073 \times 3)$). Since now the allowed steps are much more restricted due to the condition in τ, the number of accepted steps decreased to 870, which gives an acceptance rate of 0.174. The new parameters given by the chain are:

$$
\begin{aligned}
H_0 &= 68.8141901 \pm 2.02671287 \\
\omega_b h^2 &= (2.23783957 \pm 0.0472483326) \times 10^{-2} \\
\omega_c h^2 &= 0.114735698 \pm 0.00438943358 \\
\tau &= 0.0543745742 \pm 0.0125939975 \\
A_s &= (2.06303935 \pm 0.0590706493) \times 10^{-9} \\
n_s &= 0.967221120 \pm 0.0119549595
\end{aligned}
$$

and χ^2 of 1233.4932280787098. Even with the smaller acceptance rate, we can see that all the errors decreased in comparison with the previous chain (with all six parameters floating). Therefore the importance of having a prior.

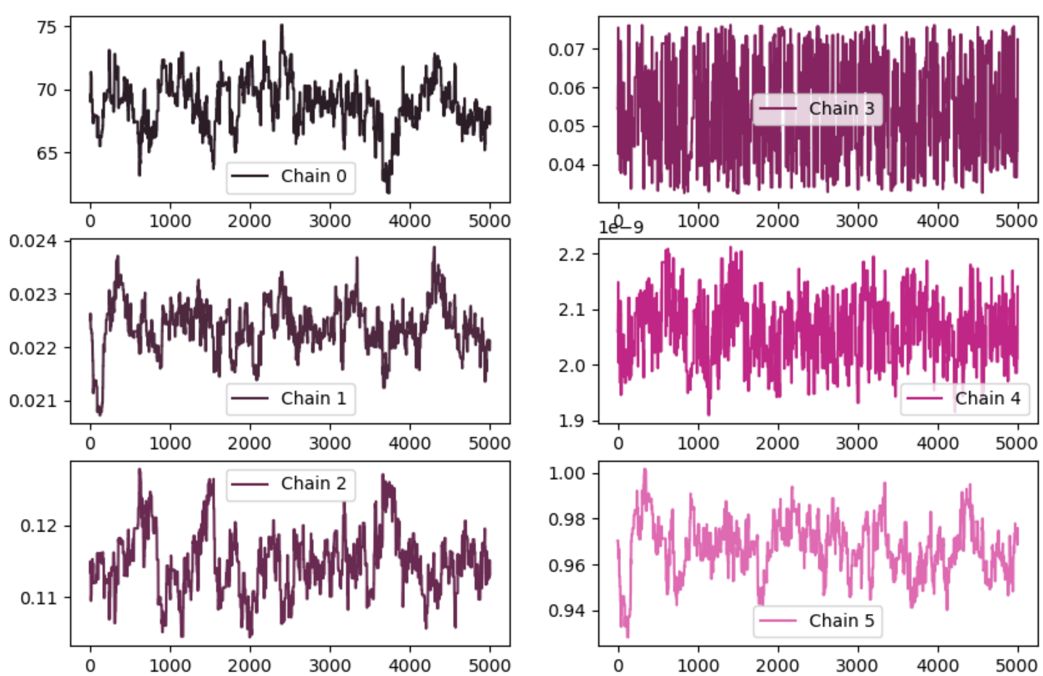


Figure 8: Chains for each parameter with prior for τ.

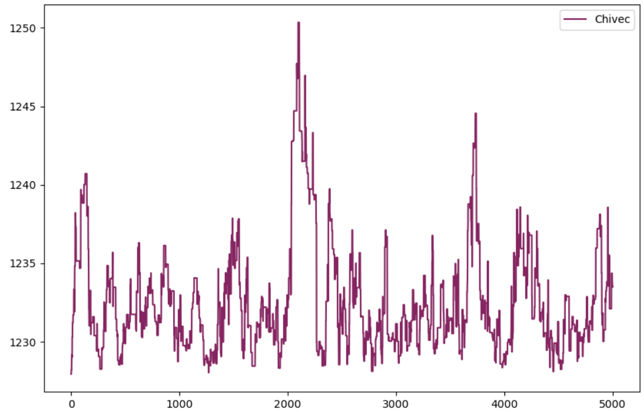


Figure 9: χ^2 evolution during chain with prior for τ.

The FFTs of the chains (Fig. 10) show less noise when compared to the previous (Fig. 7), but still not a good convergence. For that, we'd need to run more chains with more steps.

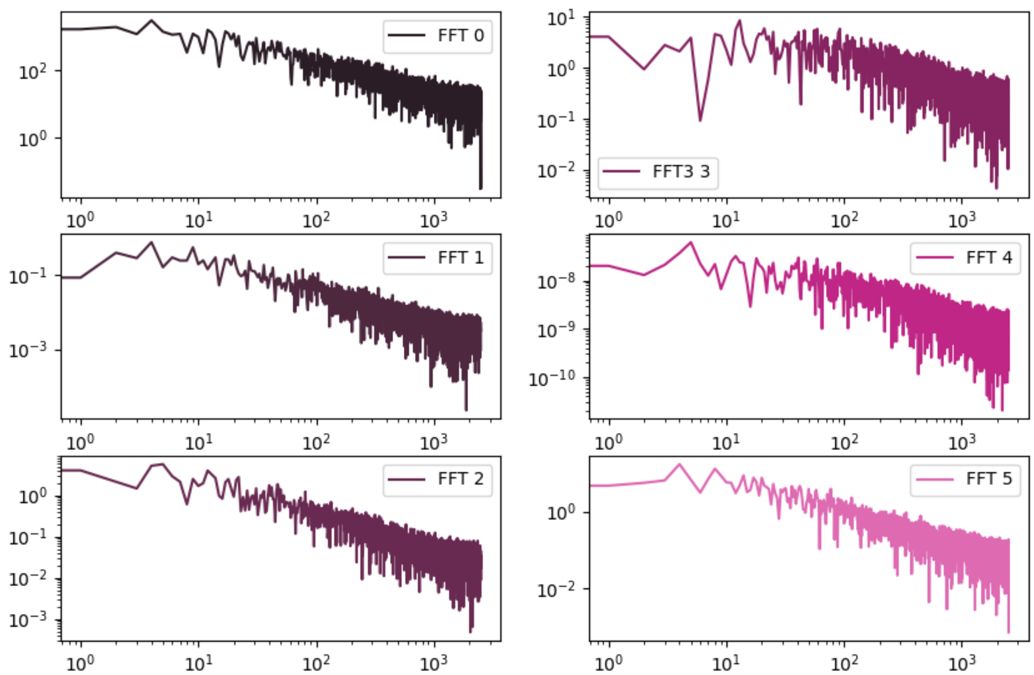


Figure 10: FFTs of chains with prior for τ.