PROBLEM 4

To get a new parameter covariance matrix, I changed the $m0$ τ parameter to 0.054.
From the LM code, this yielded the following parameters:

H_0 6.77098995×10^{1}
$\Omega_b h^2$ $2.23411419 \times 10^{-2}$
Ω_{ch}^2 $1.18952531 \times 10^{-1}$
τ $5.49112640 \times 10^{-2}$
A_s $2.09480912 \times 10^{-9}$
n_s $9.68401263 \times 10^{-4}$

Notice how τ is within the 0.054 ± 0.0074 range.

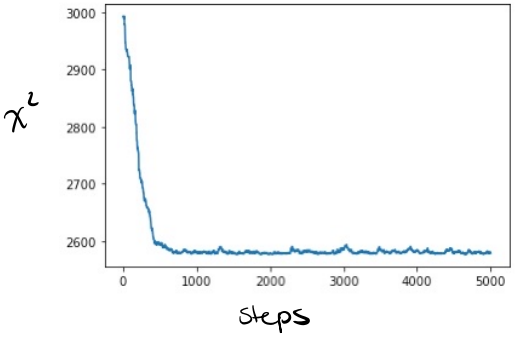
The covariance matrix was obtained exactly the same way as described in problem 3. The matrix:

```
[[ 2.31753290e-01  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 3.44363406e-05  2.89126879e-05  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [-4.88287526e-04  7.71873039e-05  1.01712147e-04  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 3.94158216e-03  3.52999159e-04 -4.37477223e-04  6.34924189e-03
   0.00000000e+00  0.00000000e+00]
 [ 1.40061374e-11  1.85292032e-12 -1.06743733e-12  2.65169267e-11
   6.01806204e-13  0.00000000e+00]
 [ 1.29731203e-03  7.47930367e-05 -7.43564800e-05  1.24723159e-04
  -2.36380911e-05  4.17792489e-04]]
```

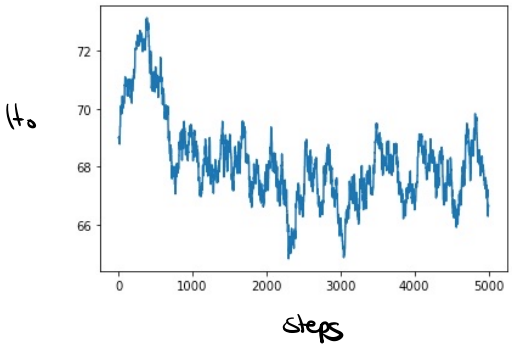
In order to redo the chain in q3 with new constraints on τ, I made sure to only accept steps when $0.0466 < \tau < 0.0614$.
With a total of 5000 steps, the percentage of accepted steps was 46.96%, which is still quite high, but at least not the 70% from the prior question.

Some plots of this method:

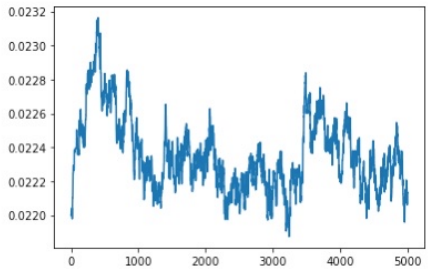
Graph 1: χ^2



Graph 2: H_0



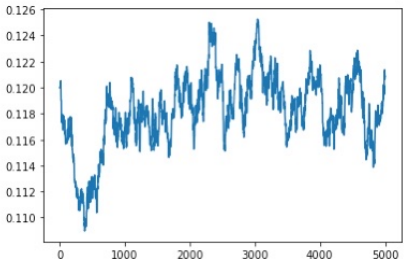
Graph 3: $\Omega_b h^2$



$\Omega_b h^2$

steps

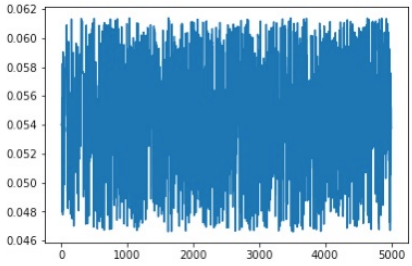
Graph 4: $\Omega_c h^2$



$\Omega_c h^2$

steps

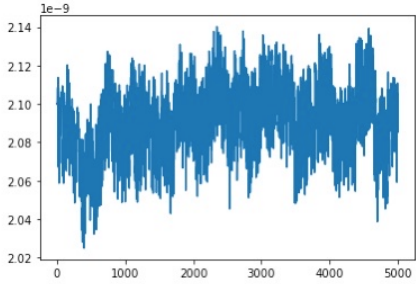
Graph 5: τ



τ

steps

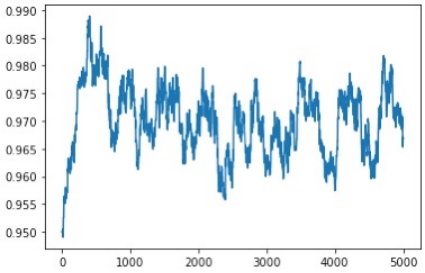
Graph 6: A_s



A_s

steps

Graph 7: n_s



n_s

steps

Notice how the τ graph is very noisy – it's fluctuating mostly between the constraints, meaning that the τ parameter will be much better.

This chain yielded on average the following parameters:

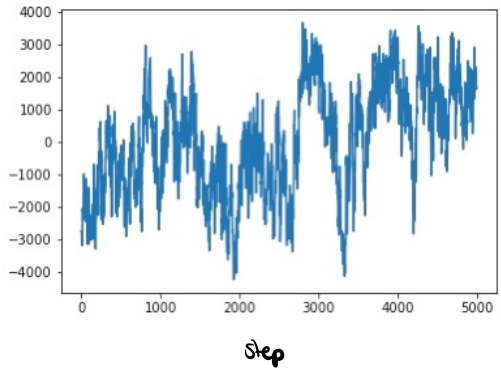
H_0 6.81648942×10^{1}
$\Omega_b h^2$ $2.23552732 \times 10^{-2}$
$\Omega_c h^2$ $1.18191149 \times 10^{-1}$
τ $5.46509243 \times 10^{-2}$
A_s $2.09148003 \times 10^{-9}$
n_s $9.76022251 \times 10^{-1}$

with avg χ^2 of 2597 (it took slightly more time to converge).

For importance sampling: See code.

→ so this is chivec − chivec.mean()

Graph 1: χ^2 for parameters



χ^2

Seems to fluctuate around 0. (but then more positive)

The weight does not seem to be working. I decided to do:

weight = np.exp((new_shift / temp_std)^2)

But wasn't working.

I am supposed to be getting parameters through important sampling if we adjust a constraint.

Considering running my mcmc code takes approx 4.5 hours, it would be best to use importance sampling, assuming it works.