Since the Python script for this assignment is complicated, they are split into several files. Particularly, `assignemnt5.py` contains several functions and data loading routine that are shared across other scripts, `1_lm.py` contains code for computing χ^2 for part 1) as well as LM method for part 2), `1_mcmc.py` contains MCMC code for part 3) and `1_mcmc_tauprior.py` contains MCMC code for part 4) with a constraint τ.

1) The χ^2 for the parameters in the test script as well as the one in the question are

```
Params in the test script gives chi2 15172.476842708616
Params closer to accepted value gives chi2 3196.7799089862683
Total degrees of freedom is 2501
```

The degree of freedom is 2501 which gives a upper 1σ bound of $2501 + \sqrt{2 \cdot 2501} = 2571.72$. The two values does *not* give an acceptable fit as the χ^2 value is well beyond the 1σ upper bound.

2) I implemented the LM method and performed the fit. The results are

```
Fit using LM gives:
H0 = 68.35867354940581 +- 1.1838887590635299
Omegabh2 = 0.02240840419711142 +- 0.00022959372323208598
Omegach2 = 0.11745616445690459 +- 0.002638998947398046
tau = 0.0850644158715427 +- 0.03406952006403914
As = 2.2147799020503496e-09 +- 1.4294657789898378e-10
ns = 0.9737285243931806 +- 0.006559939516802778
with chi2 2493.4424667622934
```

and are also saved in `planck_fit_params.txt`. The curvature matrix is stored in `planck_fit_cov_mat.txt`.

3) I implemented the MCMC method and performed the fit. I used the parameters given in the test script as the initial value. The curvature matrix from LM fit is used to generate the steps. The resulting MCMC chain is stored in `planck_chain.txt`. To check for convergence, we plot χ^2 and parameters as a function of steps as well as the FFT of the parameters, which are shown in Figure 1. All the parameters converges with some of them having somewhat large fluctuations. The FFT shows flattened intensity at low frequencies. The fit results are

```
Fit using MCMC gives:
H0 = 68.38896832326066 +- 1.115301882752364
Omegabh2 = 0.02241487595534897 +- 0.00023257636872724084
Omegach2 = 0.1174257010221754 +- 0.0024827352127567534
tau = 0.08825530019089566 +- 0.025911763597709317
As = 2.2330996687250968e-09 +- 1.0962086418637827e-10
ns = 0.9740647760432307 +- 0.006112070004528546
with chi2 2493.8817598711175
OmegaLambda is 0.7010069436344617 +- 0.05179142148731293
```

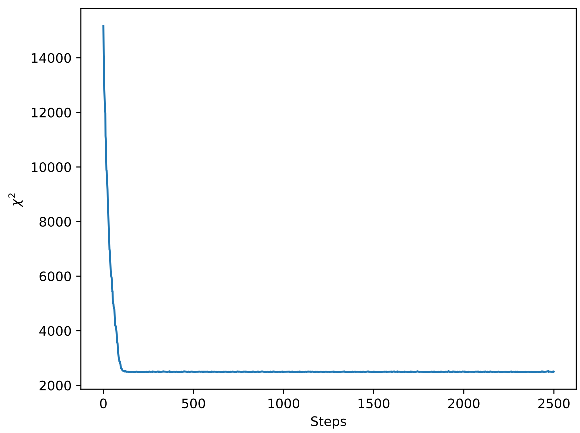
where we also computed Ω_Λ. Note

$$\Omega_\Lambda = 1 - \Omega_b - \Omega_c = 1 - \frac{\Omega_b h^2}{h^2} - \frac{\Omega_c h^2}{h^2},\ h = \frac{H_0}{100}.$$

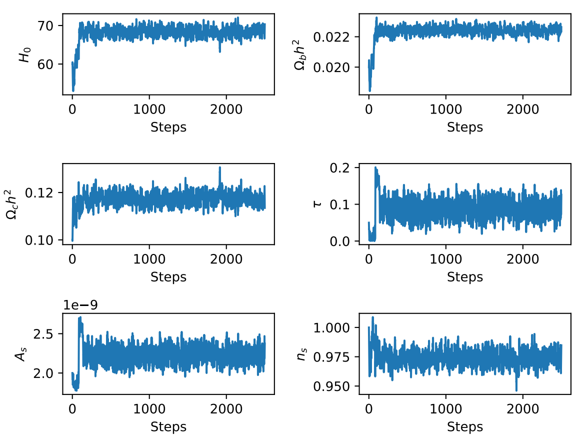
thus the uncertainty is

$$\sigma_{\Omega_\Lambda} = \sqrt{\left(\frac{\sigma_{\Omega_b h^2}}{\Omega_b h^2}\right)^2 + \left(2\frac{\sigma_h}{h}\right)^2 + \left(\frac{\sigma_{\Omega_c h^2}}{\Omega_c h^2}\right)^2 + \left(2\frac{\sigma_h}{h}\right)^2},\ \sigma_h = \frac{\sigma_{H_0}}{100}.$$

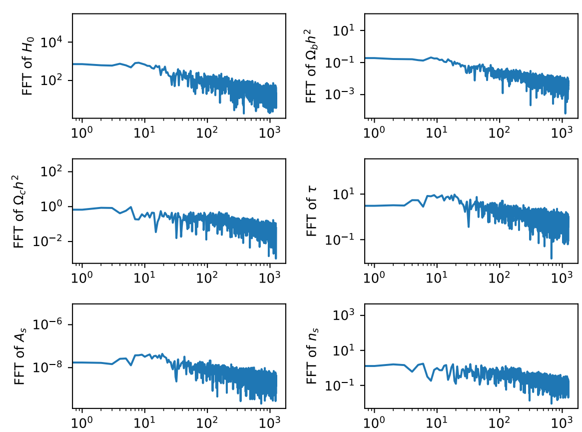
4) I implemented the MCMC method and performed the fit. I used the parameters given in the test script as the initial value. I performed importance sampling on the MCMC chain from part c) to generate



(a) Plot of χ^2 as a function of steps.



(b) Plot of parameters as a function of steps.



(c) Plot of the FFT of the parameters.

Figure 1: Figures for check MCMC convergence.

the new covariance matrix which is then used to generate the steps. The constraint is implemented by setting the accept probability to be

$$\exp\left\{-\frac{1}{2}\left[\chi^2_{\rm new} - \chi^2_{\rm old} + \left(\frac{\tau - \tau_{\rm prior}}{\sigma_{\tau_{\rm prior}}}\right)^2\right]\right\}$$

where $\tau_{\rm prior}, \sigma_{\tau_{\rm prior}}$ are the prior τ and its uncertainty. The resulting MCMC chain is stored in `planck_chain_tauprior.txt`. To check for convergence, we plot χ^2 and parameters as a function of steps as well as the FFT of the parameters, which are shown in Figure 2. All the parameters converges with some of them having somewhat large fluctuations, but still smaller than the one in part 3). The FFT shows flattened intensity at low frequencies. The fit results are

```
Constraint tau fit using MCMC gives:
H0 = 67.81382240103268 +- 0.9810211833386103
Omegabh2 = 0.022331576029242208 +- 0.0002021734589382418
Omegach2 = 0.11868518686231985 +- 0.002210110331606522
tau = 0.056167065810322214 +- 0.007357215349912556
As = 2.09874448112524e-09 +- 3.168027373384711e-11
ns = 0.9710334268892657 +- 0.005528632019569583
with chi2 2494.2712596105166
```

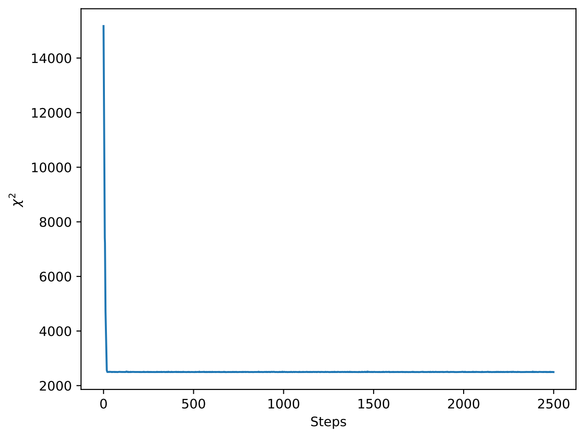
Comparing with the fit results using importance sampling from the MCMC chain in 3)

```
Constraint tau fit using importance sampling gives:
H0 = 68.20399219860941 +- 0.9153292710577352
Omegabh2 = 0.022343737976411406 +- 0.00017111459675008507
Omegach2 = 0.11764179747777596 +- 0.002049784235910284
tau = 0.05960258986628063 +- 0.01234171911370243
As = 2.1079127744393275e-09 +- 5.418394367153923e-11
ns = 0.9735251874326112 +- 0.004545197130848247
with chi2 2494.310856716881
```

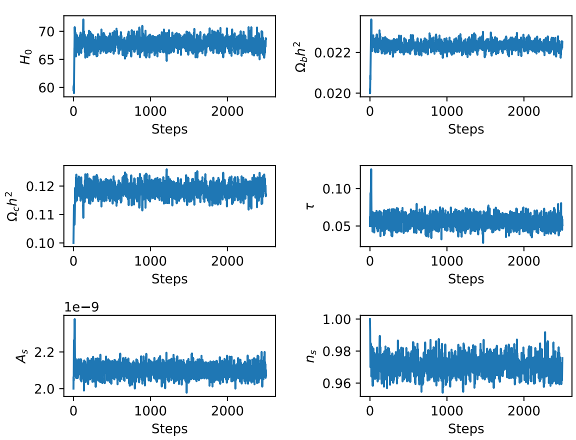
we can see all of them agree within 1σ.

BONUS 2 Since the MCMC steps are Gaussian distributed, we can safely assume a Gaussian distributed steps around the MCMC parameters after the chain is converged (we take the last 2000 steps out of the total 2500 steps). This means the 5σ error bar is just $\pm 5\sigma$ range (as long as the lower bound is positive, which is true for our fit results). Using the MCMC chain with constraint τ, we find

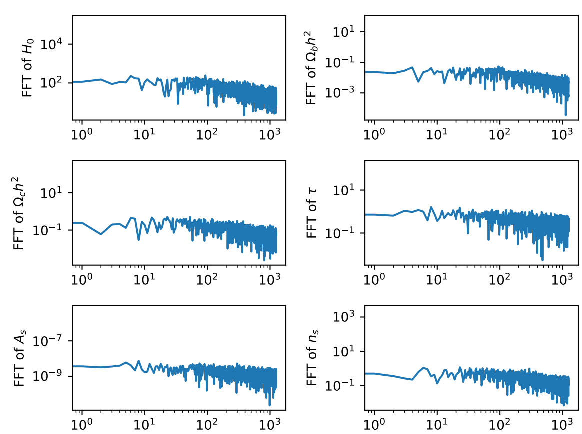
```
Constraint tau fit using MCMC have 5sigma error bars:
H0: [62.90871648433963, 72.71892831772573]
Omegabh2: [0.021320708734551, 0.023342443323933416]
Omegach2: [0.10763463520428725, 0.12973573852035247]
tau: [0.019380989060759433, 0.09295314255988499]
As: [1.9403431124560046e-09, 2.2571458497944755e-09]
ns: [0.9433902667914178, 0.9986765869871137]
```



(a) Plot of χ^2 as a function of steps.



(b) Plot of parameters as a function of steps.



(c) Plot of the FFT of the parameters.

Figure 2: Figures for check MCMC convergence with a constraint τ.