PHYS 512 Assignment 4

Michelle Lam, SN: 261005326

October 19, 2021

Problem 1

Refer to code **ps4_Problem1.py.**

Using the parameters from the given test script, the χ^2 gives 15268 fro 2501 degrees of freedom, whereas the new parameters yield a lower $\chi^2 = 3272.2$.

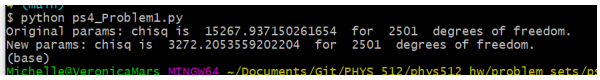

Figure 1: Output of **ps4_Problem1.py.**

Problem 2

Refer to code: **ps4_Problem2.py.**

Using Levenberg-Marquardt method to get best fit parameters and 2-sided numerical derivative function, I used 50 iterations to find the fit parameters, error in parameters, and inverse curvature matrix. The starting $\chi^2 = 3271.7$ and after 50 iteration our $\chi^2 = 2585.6$

The fit parameters and error were outputted in planck_fit_params.txt The curvature matrix were outputted in param_covar.txt

- for the numerical derivative, the parameter step size was taken to be $d(param) = param/100$
- we used the parameters from problem 1 as the initial parameters here
- parameter error: $dm = (A'^T N^{-1} A')^{-1} A'^T N^{-1} r$, where r is the residuals, and A' is the output from our numerical derivative of the parameters

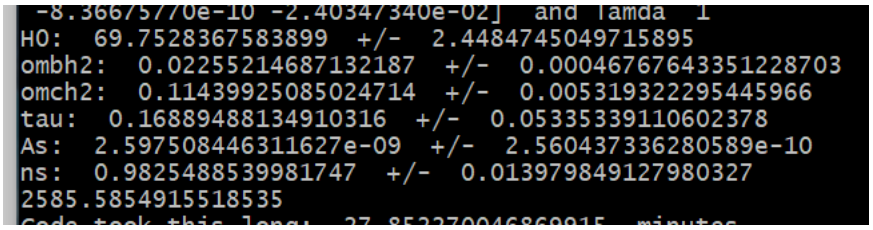

Figure 2: Output of **ps4_Problem2.py.**

Problem 3

Refer to code **ps4_Problem3.py.** To plot, run **ps4_plot_q3.py** The chain is saved as chain_chi_10000steps.py

To find the step size, we took the inverse curvature outputted from problem 2, i.e. our parameter covariance matrix. Using np.linalg.cholesky which gives us L, lower triangular matrix, we got our new step size $d\tilde{p}$

$$d\tilde{p} = Lr \tag{1}$$

where r is a vector of random values (normally distributed)

We keep track of where our parameters and chisq walk around, which we output in a file called chain_chi_10000steps.txt

The dark energy (Σ_Γ) was calculated from our parameters using these equations:

$$\Omega_B + \Omega_C + \Omega_\Lambda = 1 \tag{2}$$

$$h = \frac{H_0}{100} \tag{3}$$

$$\Omega_\Lambda = 1 - \frac{(ombh2 + ombh2)}{h^2} \tag{4}$$

And the uncertainty was calculated using the the exponential, addition, and exponential rules for propagating uncertainties. **Dark energy: 0.69 ± 0.04**, this agrees with the estimated amount of dark energy in the universe ≈ 68 percent (according to google). Although from the outputted graphs **it doesn't seem like all the parameters have fully converged, because the low frequency or starting part of the fft graphs aren't flat. Therefore I think only some of the parameters have converged and needed more steps for the other parameters to converge.**

Here is the output for Problem 3:

```
Michelle@VeronicaMars MINGW64 ~/Documents/Git/PHYS_512/phys512_hw/problem
$ C:/Users/Michelle/Anaconda3/python.exe c:/Users/Michelle/Documents/Git/
4/ps4_plot_q3.py
chisq:  2579.215081820794
dark energy is: 0.6920593232684369 +/- 0.035579000923967444
7
H0 :  67.27351698530974 +/- 0.646722302283479
ombh2 :  0.022245199352345146 +/- 0.00018400196460095406
omch2 :  0.11987347731345858 +/- 0.0014984010028071787
tau :  0.04750486018975785 +/- 0.013775107242524256
As :  2.0695957606225408e-09 +/- 5.593550716094159e-11
ns :  0.9678522192065832 +/- 0.004306778800059544
(base)
Michelle@VeronicaMars MINGW64 ~/Documents/Git/PHYS_512/phys512_hw/problem
```

Figure 3: Parameter and chisq values at end of chain for problem 3

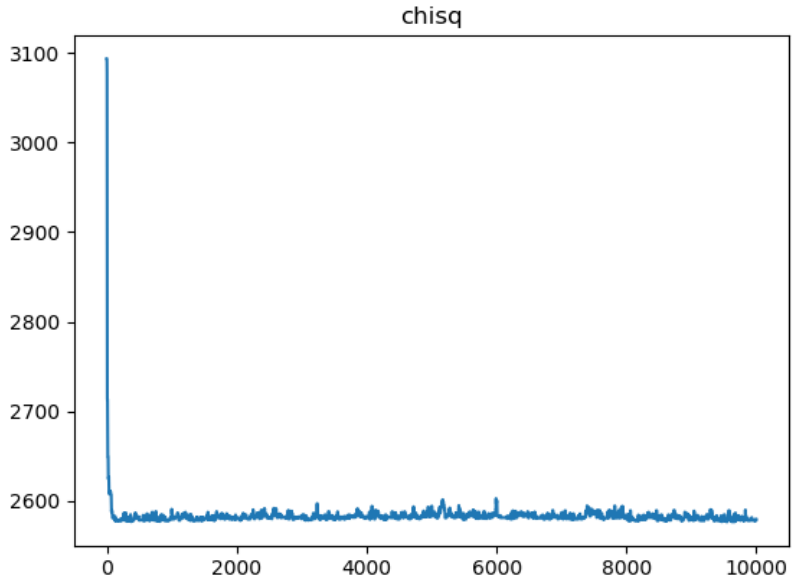

Figure 4: chisq graph for 10000 steps

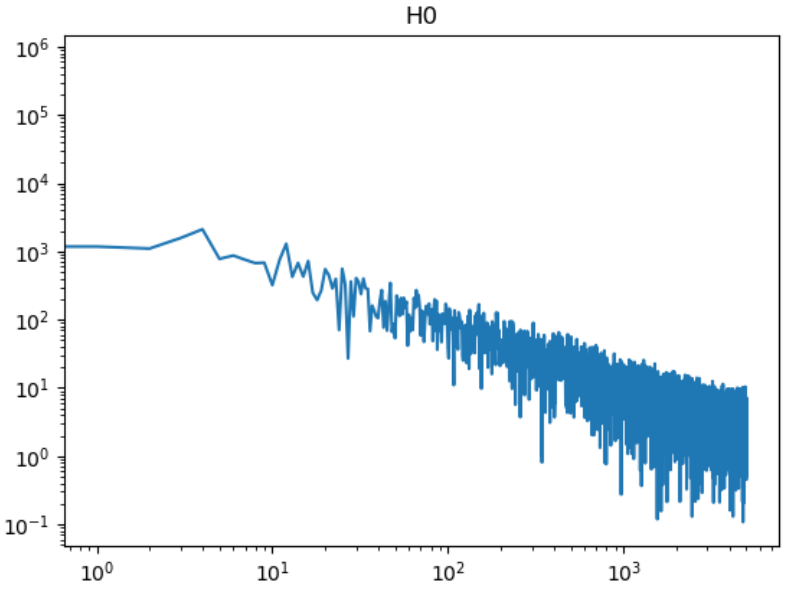

Figure 5: H0 graph for 10000 steps

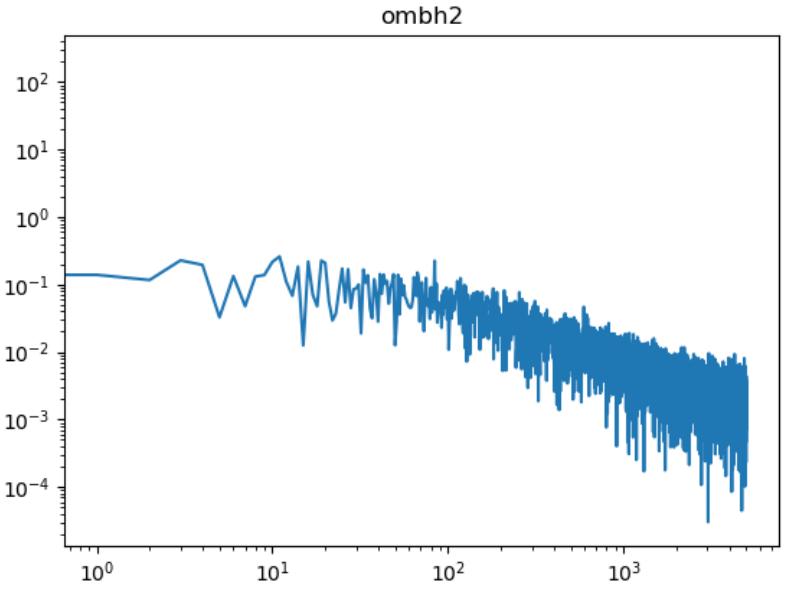

Figure 6: ombh2 graph for 10000 steps

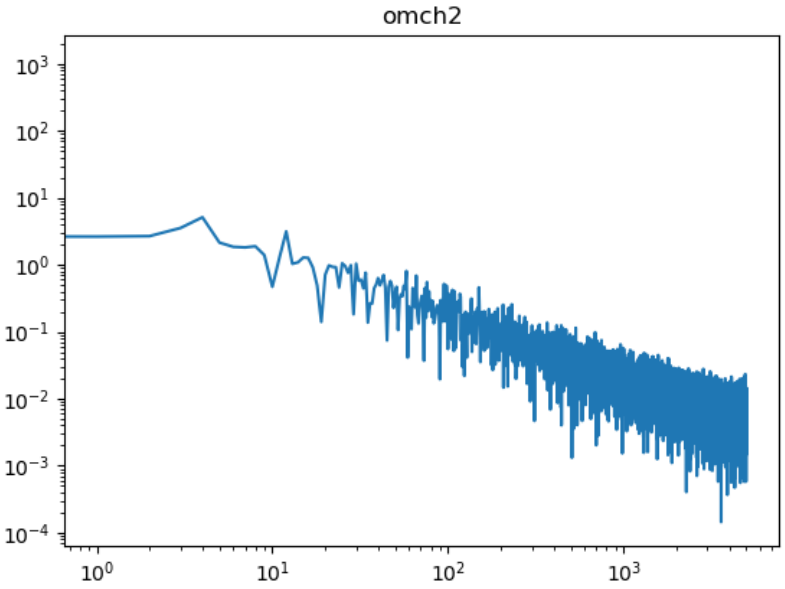

Figure 7: omch2 graph for 10000 steps

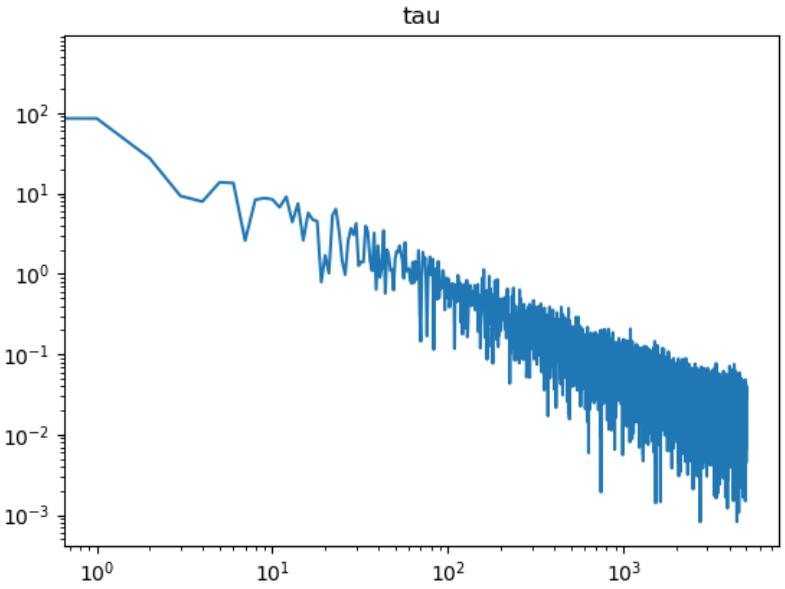

Figure 8: tau graph for 10000 steps

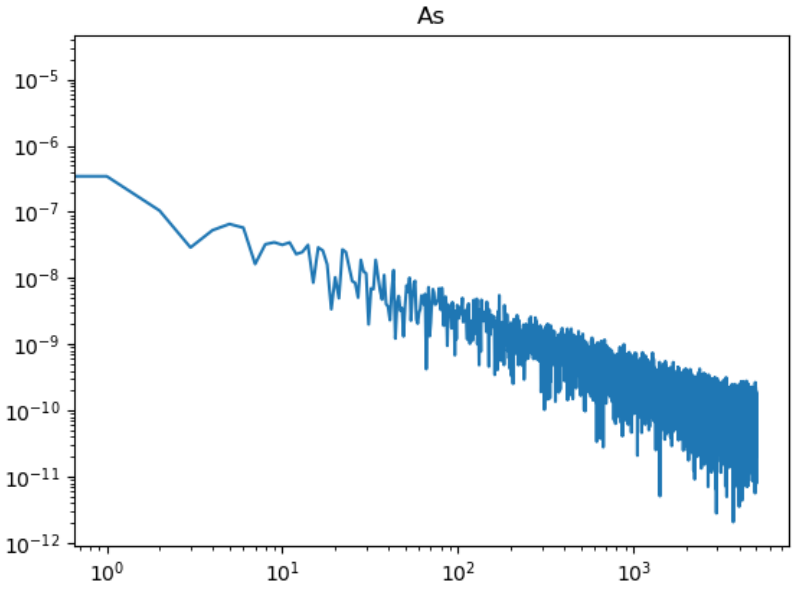

Figure 9: As graph for 10000 steps

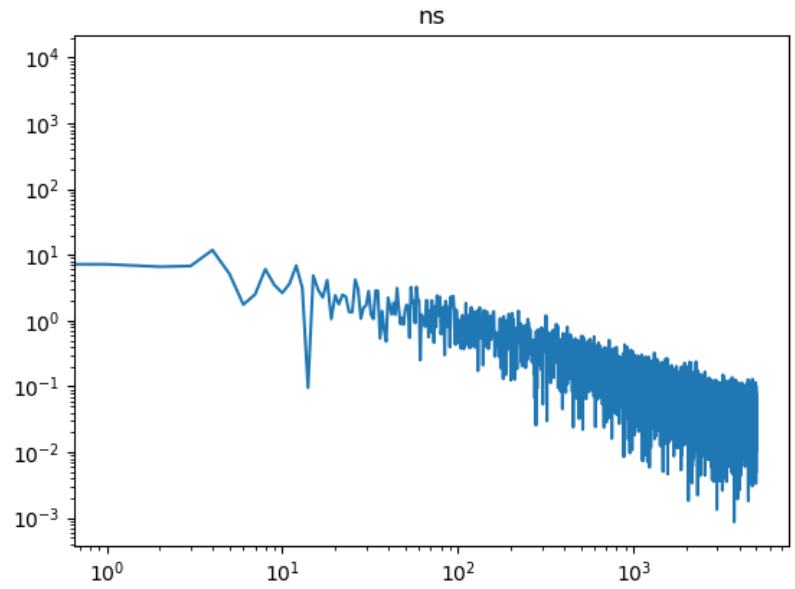

Figure 10: ns graph for 10000 steps

Problem 4

Refer to codes: ps4_Problem4_pt1_cov.py, ps4_Problem4_pt2_chain.py, ps4_Problem4_pt1_importance.py. Here, I fixed 0.054 to re-estimate the parameter covariance matrix, saved as param_covar_tau. Then, repeating the code from problem 3 but with a tau constraint where we only accept steps within, $\tau = 0.054 \pm 0.0074$. And then applying importance sampling on the chain outputted from problem 3 with expected parameters that include our constraint, we can compare the two methods. Note, the expected parameters were taken from the final parameters of the chain with the constraint and the standard deviation as the error.

From the output we see that the chisq is a bit lower using the constraint and running the full chain, instead of importance sampling. But the outputted parameters are still relatively close, so it saves a lot more time to use importance sampling and get a decent estimate of the parameters.

```
Michelle@VeronicaMars MINGW64 ~/Documents/Git/PHYS_512/phys512_hw/problem_sets/ps4 (main)
$ C:/Users/Michelle/Anaconda3/python.exe c:/Users/Michelle/Documents/Git/PHYS_512/phys512_hw/pr
sets/ps4/ps4_Problem4_pt3_importance.py
importance sampled parameter  0  has mean  68.56336119456125
importance sampled parameter  1  has mean  0.022715457908193747
importance sampled parameter  2  has mean  0.11902142484502032
importance sampled parameter  3  has mean  0.05278319186761518
importance sampled parameter  4  has mean  2.1128846033318646e-09
importance sampled parameter  5  has mean  0.953701064943146
new_params: [6.85633612e+01 2.27154579e-02 1.19021425e-01 5.27831919e-02
 2.11288460e-09 9.53701065e-01]
chisq:  2635.921737998917
chain with tau constraint:  [6.77289307e+01 2.22585502e-02 1.18653955e-01 5.36803313e-02
 2.08867970e-09 9.71007925e-01]
chisq:  2577.485811215338
(base)
```