AST9051SP - Statistical Methods in Computational Cosmology

Metin San

April 2024

Project description

This document is the final report for the project in the course AST9051SP - Statistical Methods in Computational Cosmology. In this project, we have explored the process of fitting two non-linear parameters in a model through linear and Bayesian methods. This document is accompanied by a Jupyter notebook where we solve a subset of the exercises in the Data analysis recipe: Fitting a model to data[1]. The source code repository containing the code used to produce the results in this document is available at `https://www.github.com/metinsa/`.

Introduction

We will now attempt to fit the two non-linear parameters (α, γ) in the state-of-the-art COBE/DIRBE zodiacal light model[2]. Zodiacal light (ZL) is scattered and re-emitted sunlight by interplanetary dust in the Solar system. In the following, we will fit this model to a data stream containing ZL simulated with ZodiPy[3, 4] and white noise. For simplicity, the simulated ZL only contains the smooth component (diffuse cloud component) of the zodiacal cloud. This data stream represents the observed emission by the COBE/DIRBE experiment at 12μm during DIRBE observational days 50-59. A small subsection of the simulated data is shown in Figure 1. The true parameter values used to generate this data are $\theta_{\text{true}} = (\alpha_{\text{true}}, \gamma_{\text{true}}) = (1.337, 0.942)$.

The data model is given as

$$d_t = z_t + n, \tag{1}$$

where z_t is the simulated ZL timestream, and $n \sim \mathcal{N}(0, 1)$. The ZL z_t is proportional to the line-of-sight integral over the number density of interplanetary dust (IPD) grains n multiplied by the Planck function B_λ

$$z_t \propto \int n(s) B_\lambda ds, \tag{2}$$

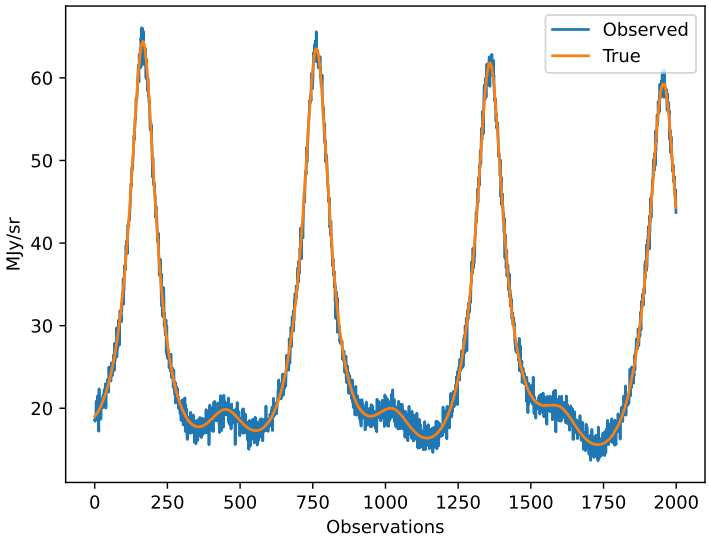

Figure 1: A small subsection of the data worked with in this project. The orange curve is the simulated z_t timestream with ZodiPy, and the blue curve is data stream d where we have added noise.

where s is the line-of-sight. The two parameters we want to fit is contained in the expression of the heliocentric ecliptic IPD number density $n(x, y, z)$, which is given as

$$n(x, y, z) = n_0 r^{-\alpha} e^{-\beta g^{\gamma}}, \tag{3}$$

where n_0 is the density of the IPD at 1 AU, r is the radial distance from the origin of the IPD, and α and γ, the parameters we wish to fit, are shape parameters describing the radial and vertical distribution of the dust. Furthermore we have

$$g = \begin{cases} \zeta^2/2\mu & \text{for } \zeta < \mu \\ \zeta - \mu/2 & \text{for } \zeta \geq \mu, \end{cases} \tag{4}$$

where $\zeta = |Z|/r$ and Z is the height above the symmetry plane of the dust. When fitting our parameters we will have to fit evaluate Eq. (2). For this purpose we use a modified version of the ZodiPy code available at the repository link at the top of this document.

We start by building some intuition over how these two parameters affect the timestream by evaluating the model over a range of these parameters. The results are shown in Fig. 2. We observe that these parameters mostly affect the shape of the dips and not so much the peaks.

In order to compare two models we need a statistical quantity that measures how well our model reproduces the data. For this purpose we will use the χ_2

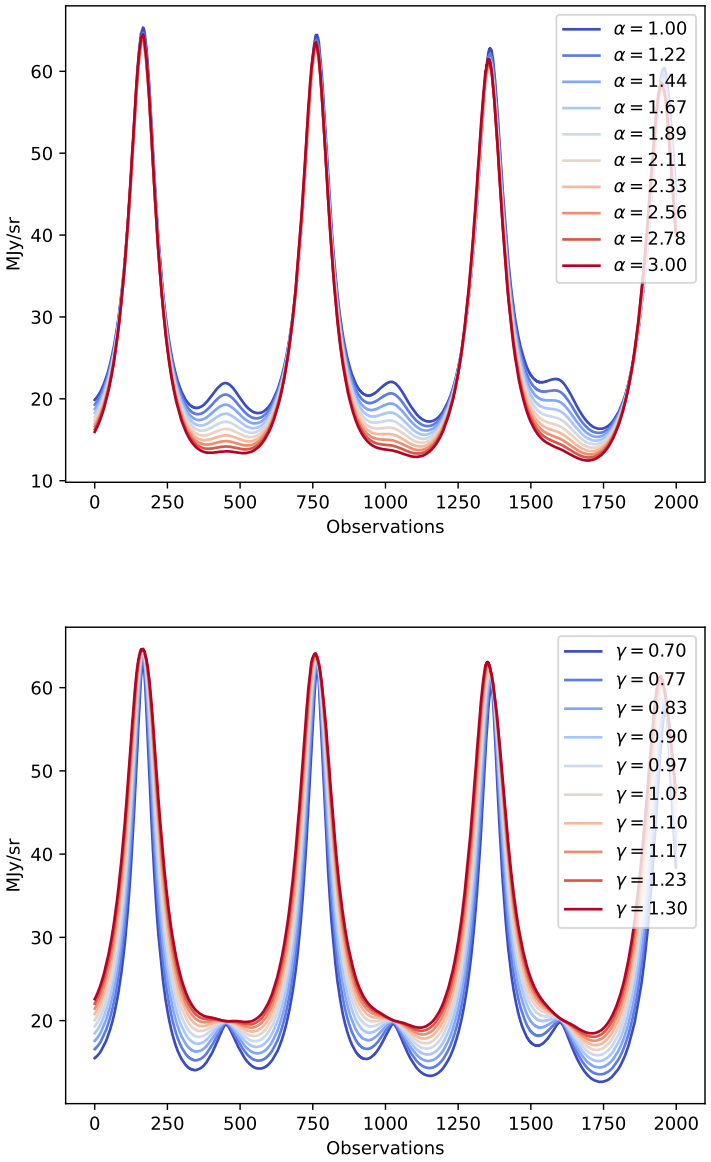

Figure 2: The effect of varying the two parameters α and γ on the timestream, shown in the top and bottom panels respectively.

statistic, given by

$$\chi^2 = \sum_i^N \frac{(y_i - \hat{y}_i)^2}{\sigma_{y,i}^2}, \tag{5}$$

where y_i is the observed data, $\hat{y}_i$ is the prediction made by out model with parameters $\theta = (\alpha, \gamma)$, and $\sigma_{y,i}$ is the uncertainties associated with y_i. Since our y_i values come from simulated data with noise drawn independently from a normal distribution with mean 0 and variance 1 the χ^2 reduces to

$$\chi^2 = \sum_i^N (y_i - \hat{y}_i)^2 . \tag{6}$$

The loglikelihood in our case, which we will use when sampling the parameters, is then simply

$$\mathcal{L} = -\frac{1}{2}\chi^2. \tag{7}$$

Grid search

Since we are only working two parameters it is perfectly possible to do a grid search over the parameter space to find the best fit. This is when we evaluate our model for z_t over a grid of θ values and record the chisq. We note that the volume of the parameter space grows exponentially with the number of parameters, and grid searches are therefore not feasible for high-dimensional parameter spaces. In these cases we can use more advanced methods such as Markov Chain Monte Carlo (MCMC) methods, which we will do in the next section. These methods also lets us characterize the uncertianties in the parameters, unlike the grid method which simply lets us find the maximum likelihoods.

We evaluating the model over a grid of 50×50 points with parameters ranging from $\alpha \in (1, 1.6)$ and $\gamma \in (0.6, 1.2)$. The resulting χ^2 values are shown in in Figure 3. From the eliptical shape of the χ^2 values we can see that the two parameters are indeed correlated, which is also what we see in the model plots in Figure 2.

1 Metropolis-Hastings

The next method we will use to sample these parameters is the Markov Chain Monte Carlo (MCMC) Metropolis-Hastings algorithm. Metropolis-Hastings lets us sample from a generic probability distribution $P(\theta)$ even when we don't know the true underlying distribution. It works by iterativly accepting or rejecting candidate samples from a target distribution $g(\theta) \propto P(\theta)$. The algorithm is described below

1. Select an initial value θ_0
2. for i = 1, ..., n, repeat:

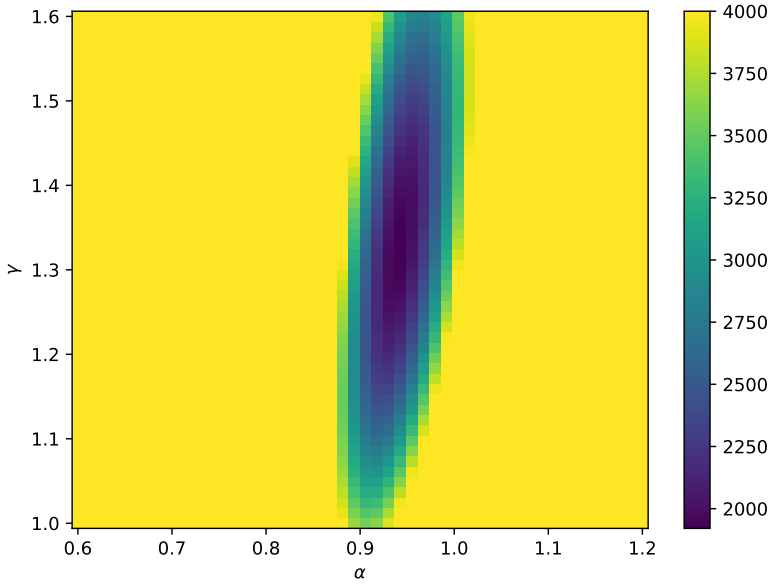

Figure 3: The χ^2 values for the grid search over the parameter space.

(a) Draw a candidate $\theta^* \sim q(\theta^*|\theta_{i-1})$ from the proposal distribution q

(b) Compute the acceptance factor a (usually α but we already used α for the fitted parameter)

$$a = \frac{g(\theta^*)/q(\theta^*|\theta_{i-1})}{g(\theta_{i-1})/q(\theta_{i-1}|\theta^*)} = \frac{g(\theta^*)q(\theta_{i-1}|\theta^*)}{g(\theta_{i-1})q(\theta^*|\theta_{i-1})} \tag{8}$$

(c) if $a \geq 1$: accept θ^* and set $\theta_i \leftarrow \theta^*$
if $0 < a < 1$ accept θ^* and set $\theta_i \leftarrow \theta^*$ with probability a
else: reject θ^* and set $\theta_i \leftarrow \theta_{i-1}$ with probability $1 - a$.

We will draw the candidate θ^* from a normal distribution $\mathcal{N}(\theta_{i-1}, \text{const})$. Since the normal distribution is symmetric about its mean, the ratio of the proposal distributions $q(\theta^*|\theta_{i-1})/q(\theta_{i-1}|\theta^*)$ cancels out, leaving us with the so called random walk Metropolis Hasting with

$$a = \frac{g(\theta^*)}{g(\theta_{i-1})}. \tag{9}$$

The target distribution in our case is

$$g(\theta) = \text{const} \exp\left[-\frac{1}{2}\chi^2\right], \tag{10}$$

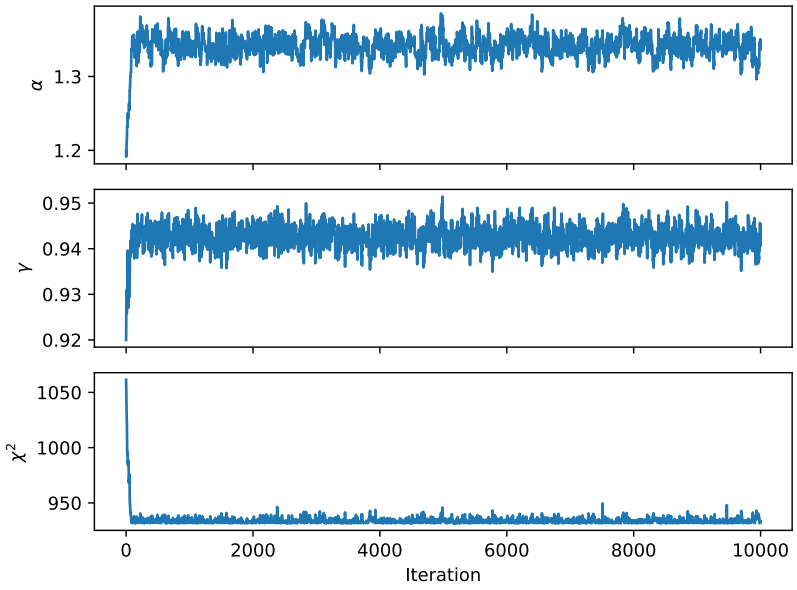

Figure 4: Trace plot showing 10000 samples (including the burn-in phase) of the parameters α, γ, and the χ^2 metric, using the Metrpolis-Hasting algorithm.

and

$$a = \exp\left[\chi^{*2} - \chi^2_{i-1}\right], \tag{11}$$

where we get the χ^2 by evaluating the ZodiPy model for the candidate θ values similarly to in the grid search case.

We select initial values $\theta_0 = (\alpha_0, \gamma_0) = (1.2, 0.92)$. Additionally, we need to select stepsizes Δ_θ which we use to draw the candidate samples. We pick these to be $\Delta_\theta = (\Delta_\alpha, \Delta_\gamma) = (0.01, 0.005)$, motived by the values from the χ^2 grid. We let the Metropolis-Hastings algorithm run for 10000 iterations, and make three trace plots: 1) A trace plot of α, γ, and χ^2 for all 10000 samples, 2) A trace plot of the same parameters but with the first 1000 samples removed (burn-in phase), and 3) A corner plot showing the correlation between the two parameters. The results are shown in Figures 4, 5, and 6.

We can find the maximum likelihoods of the parameters by finding the sample with the smallest χ^2 value

$$\theta_{\mathrm{ML}} = \theta[\mathrm{argmin}(\chi^2)]. \tag{12}$$

However, there are other quicker ways to estimate the maximum likelihoods. The main benefit of the MCMC method is that it lets us estimate the uncertainties in the parameters, which is otherwise traditional difficult to estimate. We can compute the mean and standard deviation of the parameters from the

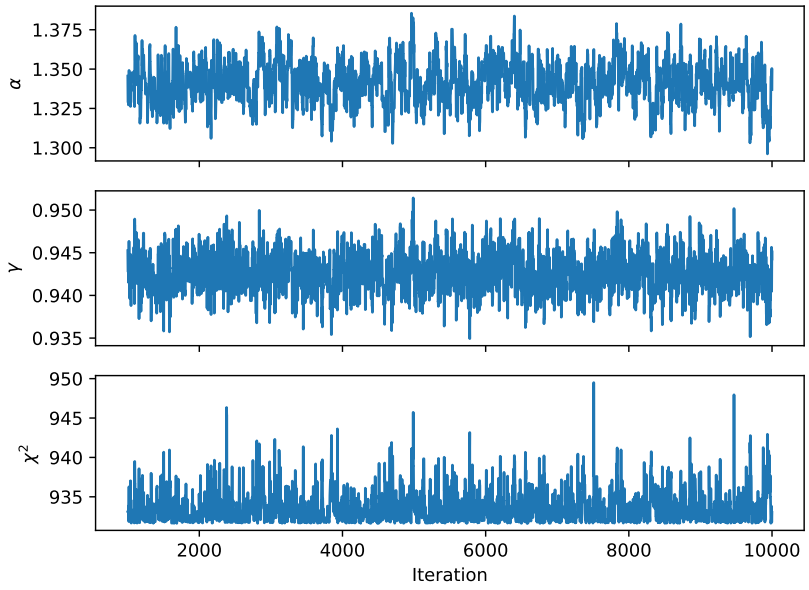

Figure 5: Trace plot showing 9000 samples of the parameters α, γ, and the χ^2 metric, using the Metrpolis-Hasting algorithm.

chain as follows

$$\mu_\theta = \frac{1}{N}\sum_i^N \theta_i, \tag{13}$$

$$\sigma_\theta^2 = \frac{1}{N}\sum_i^N (x_\theta - \mu_\theta)^2. \tag{14}$$

We find that for 9000 samples with an accept rate of 0.3405%, the best fit parameters using Metropolis-Hastings are

$$\alpha = 1.3420 \pm 0.0155$$
$$\gamma = 0.9426 \pm 0.0024.$$

2 Hamiltonian Monte Carlo

Although the Metropolis-Hastings seems to do a good job at sampling the parameters in our case, it can be very inefficient in exploring the full parameter space and may sometimes lead to slow convergence. Another sampling method, which required much fewer samples to converge is the Hamiltonian Monte Carlo method (HMC). HMC is very similar to Metropolis-Hastings in that we propose

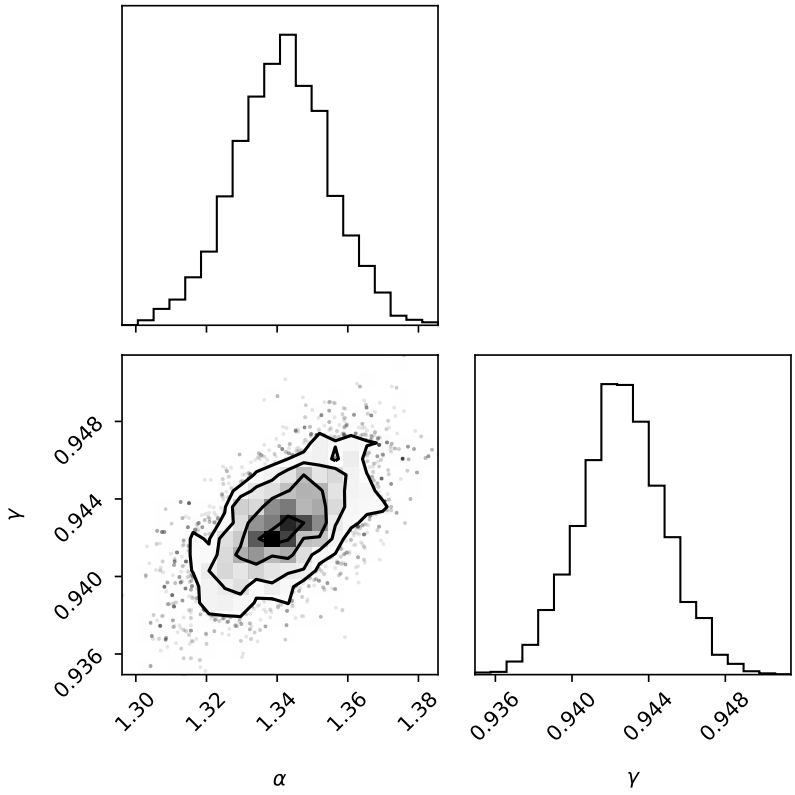

Figure 6: Corner plot showing the 1- and 2-D histograms and the parameters α, γ, obtained from post burn-in Metropolis Hastings samples.

a new sample and either accept or reject with some probability. But rather than taking random steps in the parameter space, we instead use Hamiltonian dynamics to find the new proposals. This allows us to move to distant states with high probability of acceptance due to the energy conserving properties of the Hamiltonian. The Hamiltonian is given as

$$H(q,p) = K(p) + V(q), \tag{15}$$

where $K(p,q)$ is the kinetic energy which depends on the momentum p,

$$K(p) = \frac{1}{2}p^2, \tag{16}$$

and the $V(q)$ is the potential energy. The trick to using the Hamiltonian in a MCMC context is that we set the position equal to the parameters we wish to fit $q \equiv \theta$, and set the potential V to the negative log likelihood of our target distribution

$$V(\theta) = \log \mathcal{L}(\theta) = -\frac{1}{2}\chi^2. \tag{17}$$

This allows us find new parameter samples by solving the equations of motion and moving in parameter space along equal energy curves. At any given position θ_i, we can draw a random value for the momentum p, and integrate the Hamiltonian to move along equal energy curves in phase space to pick a new value for parameter value θ_{i+1} with much higher acceptance probability than in the random walk Metropolis-Hastings scenario. The accept rate for MHC is analogous to Metropolis-Hastings

$$a = \min\left(1, \exp\left(H(\theta_i, p_i) - H(\theta_{i+1}, p_{i+1})\right)\right). \tag{18}$$

A volume conserving numerical integration technique is required to to integrate the Hamiltonian. The most commonly used algorithm is the Leap frog algorithm, which is similar to Euler's method, but instead of taking a full step at each iteration, it first takes half a step in momentum $p_{i+1/2}$ and then uses that to find q_{i+1} and p_{i+1} in a volume or energy conserving way:

$$p_{i+1/2} = p_i + \frac{\epsilon}{2}\nabla_\theta V(\theta_i) \tag{19}$$

$$\theta_{i+1} = \theta_i + p_{i+1/2}\frac{\epsilon}{2} \tag{20}$$

$$p_{i+1} = p_{i+1/2} + \frac{\epsilon}{2}\nabla_\theta V(\theta_{i+1}), \tag{21}$$

$$\tag{22}$$

where ϵ is the a step size and $\nabla_\theta V(\theta)$ is the gradient of the potential energy, or in our case, the log-likelihood. This is where we see one of the downside of HMC - it requires us to know the $\nabla_\theta V(\theta)$, the gradient of the log likelihood

$$\nabla_\theta = \sum \left[y - \hat{y}(\theta)\right] \frac{\partial \hat{y}(\theta)}{\partial \theta}. \tag{23}$$

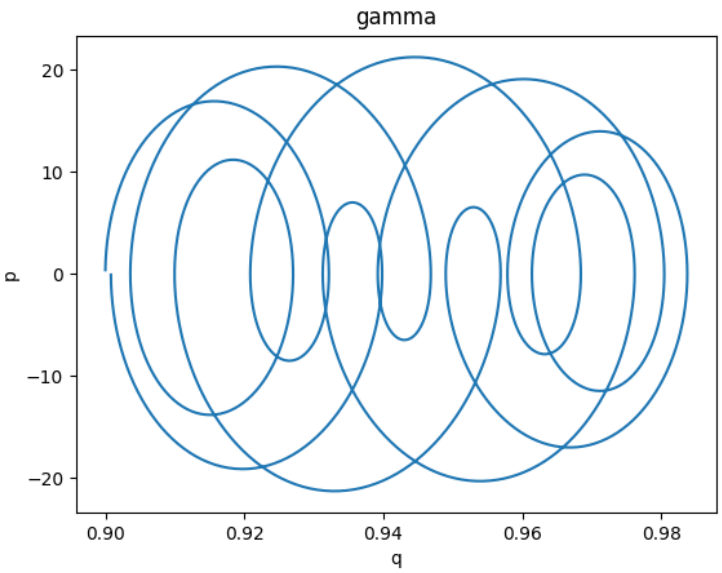

Figure 7: Illustrating an equal energy curve for the parameter γ.

For us that amounts to two new functions that we need to compute by differentiating Eq (3) with respect to α and γ

$$\frac{\partial n}{\partial \alpha} = -n_0 r^{-\alpha} f \ln r = -n \ln r \tag{24}$$

$$\frac{\partial n}{\partial \gamma} = n_0 r^{-\alpha} f(-\beta g^{\gamma} \ln g) = -\beta g^{\gamma} \ln g \; n \tag{25}$$

Figure 7 illustrates the equal energy phase space contours that we move along for the parameter γ after drawing a random momentum p. When selecting a new sample, we specify the two hyper parameters L, the number of leap-frog "leap"/iterations, and ϵ. For our two parameters, we pick $L = 5$, and $\epsilon = 0.0001$. For each iteration, we need to additionally evaluate ZodiPy L times where we also compute the gradients in Eq. (24, 25). By selecting $L = 10$, $\epsilon = 0.001$, and running for 5000 iterations, with an accept rate of 0.6599% we get the following results seen in Figures 8, 9, and 10. The best fit parameters from these chains are

$$\alpha = 1.3371 \pm 0.0106$$
$$\gamma = 0.9421 \pm 0.0018.$$

The HMC solution perfectly reproduces the underlying true parameters, with smaller uncertainties than those in the Metropolis-Hasting even with only half the samples.

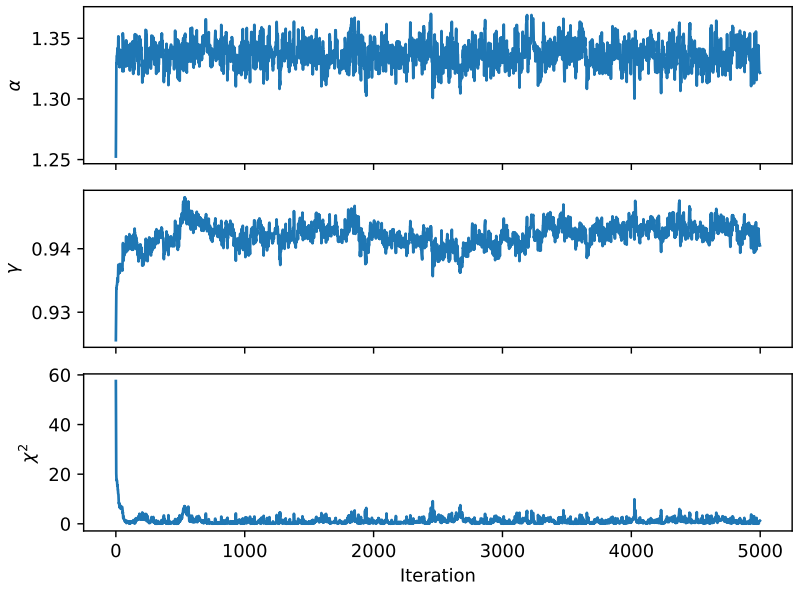

Figure 8: Trace plot showing 10000 samples (including the burn-in phase) of the parameters α, γ, and the χ^2 metric, using the Hamiltonian Monte Carlo algorithm.

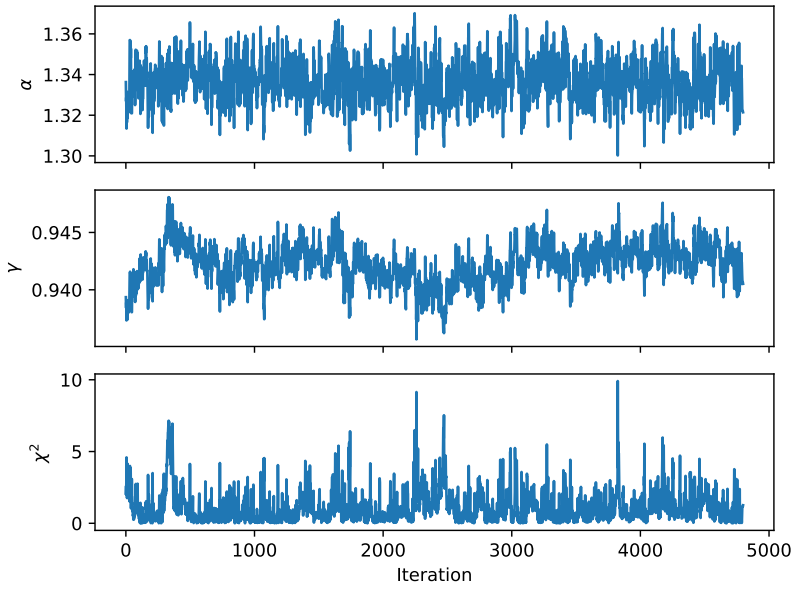

Figure 9: Trace plot showing 9000 samples of the parameters α, γ, and the χ^2 metric, using the Hamiltonian Monte Carlo algorithm.

References

[1] David W. Hogg, Jo Bovy, and Dustin Lang. "Data analysis recipes: Fitting a model to data". In: *arXiv e-prints*, arXiv:1008.4686 (Aug. 2010), arXiv:1008.4686. DOI: 10.48550/arXiv.1008.4686. arXiv: 1008.4686 [astro-ph.IM].

[2] T. Kelsall et al. "The COBE Diffuse Infrared Background Experiment Search for the Cosmic Infrared Background. II. Model of the Interplanetary Dust Cloud". In: *apj* 508.1 (Nov. 1998), pp. 44–73. DOI: 10.1086/306380. arXiv: astro-ph/9806250 [astro-ph].

[3] M. San et al. "COSMOGLOBE: Simulating zodiacal emission with ZodiPy". In: *aap* 666, A107 (Oct. 2022), A107. DOI: 10.1051/0004-6361/202244133. arXiv: 2205.12962 [astro-ph.EP].

[4] Metin San. "ZodiPy: A Python package for zodiacal light simulations". In: *Journal of Open Source Software* 9.96 (2024), p. 6648. DOI: 10.21105/joss.06648. URL: https://doi.org/10.21105/joss.06648.

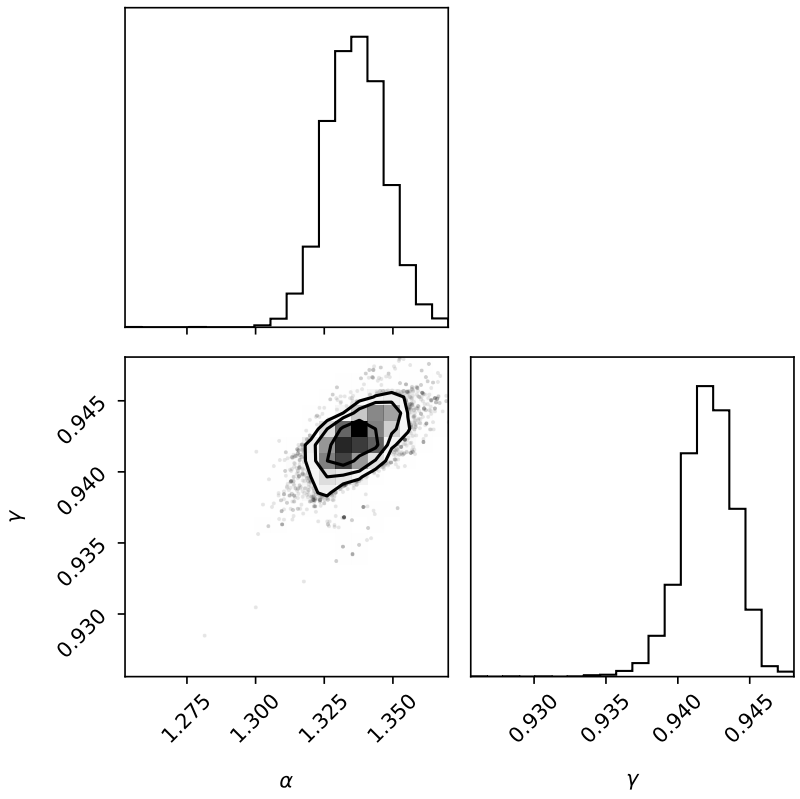

Figure 10: Corner plot showing the 1- and 2-D histograms and the parameters α, γ, obtained from post burn-in Hamiltonian Monte Carlo samples.