Computer Modelling: Metropolis Diagnostics

Hwayeon Kang

March 2025

Abstract

In this project, I developed a computational model to estimate cosmological parameters from supernova data using Bayesian inference and Markov Chain Monte Carlo (MCMC) techniques. I began by constructing a Cosmology class and applying numerical integration methods to evaluate cosmological distances. A likelihood function was then used to match theoretical predictions to observational data, and a three-dimensional likelihood grid was constructed to visualize parameter dependencies. I applied the Metropolis-Hastings algorithm to sample the posterior distribution and analyze marginal distributions of parameters. To ensure robustness, I conducted convergence tests using multiple chains and assessed them with both acceptance rates and the Gelman-Rubin statistic. The results showed consistent acceptance rates around 0.178 and convergence diagnostics well below standard thresholds. Overall, the study demonstrates that our sampling strategy yields stable, reliable parameter estimates, providing an effective and efficient framework for computational cosmology.

Contents

1 Introduction

In our *Computer Modelling* course, we have investigated the expansion of the universe through computational techniques. The project began with the development of a `Cosmology` class in Python to model the relationship between a galaxy's redshift z and its distance $D(z)$, defined by the integral:

$$D(z) = \frac{c}{H_0} \int_0^z [\Omega_m(1+z')^3 + \Omega_k(1+z')^2 + \Omega_\Lambda]^{-\frac{1}{2}} dz'$$

where H_0 is the Hubble constant, Ω_m is the matter density parameter, Ω_Λ is the dark energy density parameter, and Ω_k is the curvature density parameter.

To evaluate this integral numerically, we implemented several quadrature methods—namely, the rectangle rule, trapezoid rule, and Simpson's rule—each balancing computational efficiency and accuracy. We then integrated observational supernova data and used Gaussian likelihood functions and numerical optimization techniques to estimate the best-fit cosmological parameters.

In Unit 5, we extended our analysis to include parameter uncertainties. We constructed likelihood grids and applied the Metropolis-Hastings algorithm, a form of Markov Chain Monte Carlo (MCMC), to explore the parameter space more effectively. This allowed us to derive strong estimates and confidence intervals for the cosmological parameters.

In the final unit, Unit 6, we will consolidate our work into a comprehensive report. This report will not only present our findings but also extend the analysis through a mini-project E, offering a deeper exploration of the methods and their implications for modern cosmology.

2 Unit 5 Results

2.1 Likelihood Grid

To construct a three-dimensional likelihood grid for a full cosmological model from Unit 4, I developed the function *compute_likelihood_grid*. This function initializes a 3D grid by iterating through 10 sample points for each parameter, computing the likelihood using the *Likelihood* class from *unit4.py*, and normalizing the results.

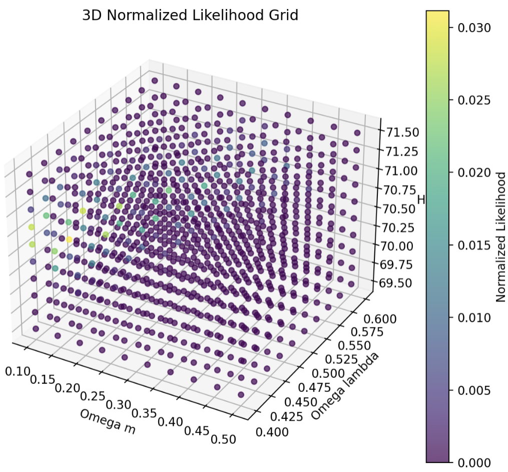

Figure 1: 3D Normalized Likelihood Grid

The function *plot_3d_likelihood* then visualizes this grid in three dimensions using Matplotlib's *scatter* function. Each point in the grid is color-coded based on its likelihood value, providing an intuitive representation of the likelihood landscape. In this resulting grid, each point corresponds to a specific parameter set, with color intensity indicating the normalized likelihood value. Lighter colors represent higher likelihoods, while darker colors indicate lower likelihoods. This visualization helps identify the most probable parameter combinations, showing regions of high likelihood in the parameter space.

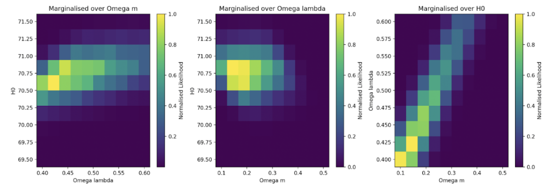

Figure 2: 2D Marginalized Likelihoods

The grids above represent the 2D marginalized likelihood distributions, where each subplot shows the likelihood distribution after integrating over one parameter. The three panels correspond to marginalized likelihoods in Ω_m, Ω_Λ, and H_0, respectively. Brighter regions indicate higher likelihoods, revealing the most probable regions in the two-dimensional parameter spaces.

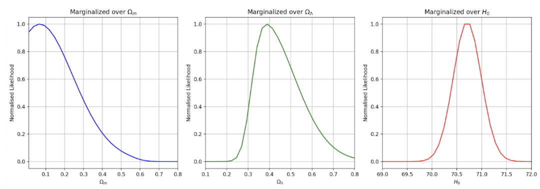

Figure 3: 1D Marginalized Likelihoods

The above 1D marginalized likelihoods are obtained by integrating over two parameters. The three plotted curves represent the likelihood distributions for Ω_m, Ω_Λ, and H_0. The peak positions indicate the most probable values, while the curve spread reflects the uncertainty in each parameter estimate. Notably, the sharp peak in H_0 shows strong constraints on this parameter, while the broader distributions for Ω_m and Ω_Λ indicate greater flexibility in their values.

Next, I performed the same process using Matplotlib's *imshow* function instead.

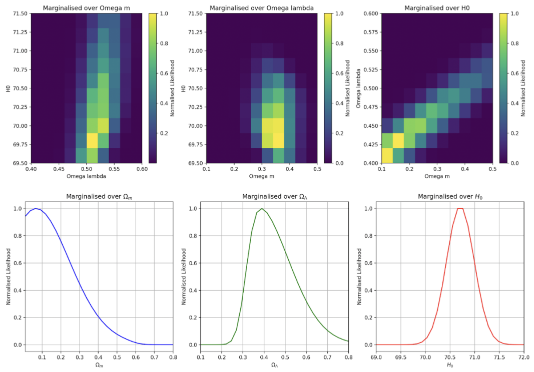

Figure 4: 2D & 1D grids using the imshow function

As seen in the above grids, the results using *imshow* are nearly identical to those obtained with *pcolormesh*.

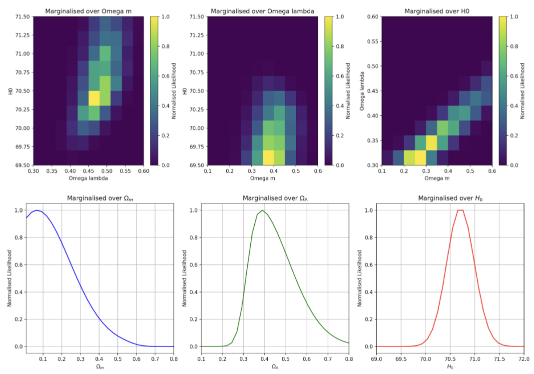

Figure 5: Optimal Parameter Range

Upon further investigation of the most suitable parameter range, I found that the range $\Omega_m = [0.1, 0.6]$, $\Omega_\Lambda = [0.3, 0.6]$, $H_0 = [69.5, 71.5]$ provides the most well-defined likelihood distributions, as shown above. This aligns with my initial expectation from the 1D marginalized likelihood grids, where H_0 exhibits strong constraints, requiring a much narrower range than other parameters.

To examine the impact of sample size on computation time and likelihood grids, I tested five different numbers of sample points: 10, 20, 30, and 50, while keeping the same parameter range, $\Omega_m = [0.1, 0.6]$, $\Omega_\Lambda = [0.3, 0.6]$, $H_0 = [69.5, 71.5]$. Considering the trade-off between grid resolution and computational cost, using 30 sample points—which required 56.25 seconds of computation—appears to offer the optimal balance between detail and efficiency.

2.2 Metropolis

I used the Metropolis algorithm to analyze the supernova dataset.

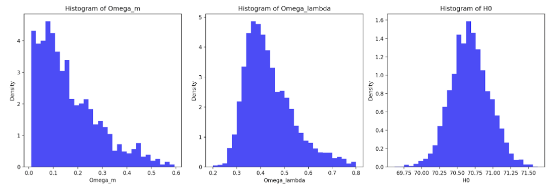

Figure 6: 1D Histogram

The marginal distributions of Ω_m, Ω_Λ, and H_0 are visualized using histograms as above. Each histogram shows the frequency density of the sampled values. 10,000 samples are used to plot the histograms.

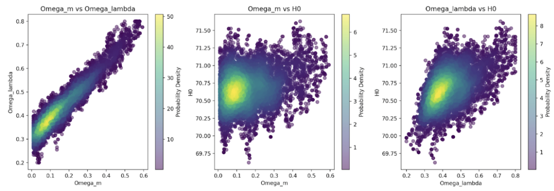

Figure 7: 2D Scatter Density Plots

The relationships between pairs of parameters are shown above using scatter plots, where color represents probability density. The density is computed using Kernel Density Estimation (KDE).

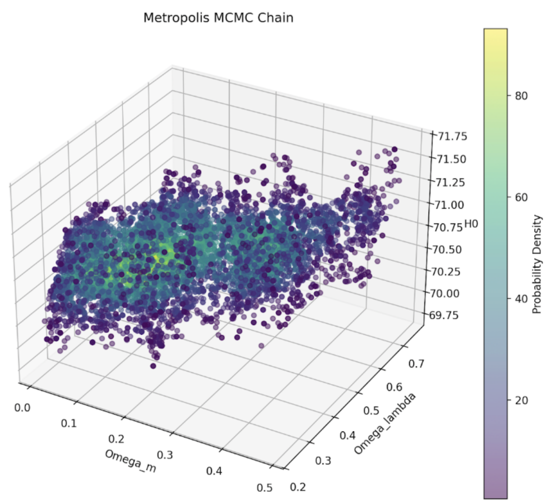

Figure 8: 3D Scatter Plot

The full distribution in Ω_m–Ω_Λ–H_0 space is visualized with a 3D scatter plot, where the color of each point represents probability density.

Compared to the likelihood grid, which provides an exact evaluation over a fixed parameter space, the Metropolis algorithm offers a sample-based estimate of the posterior distribution. This makes it more flexible but also more sensitive to initial conditions and step sizes.

3 Convergence Tests

To assess the convergence of our Metropolis-Hastings sampling process, I conducted convergence tests by running multiple independent chains and analyzing their behavior using both acceptance rates and the Gelman-Rubin statistic. I initiated ten Metropolis chains with consistent step sizes and initial conditions, each producing 6000 samples. The average acceptance rate across chains was approximately 0.178, indicating reasonable exploration of the parameter space.

To verify convergence, I computed the Gelman-Rubin statistic $(R - 1)$ for each cosmological parameter as I progressively increased the number of chains and the number of samples. The final $R - 1$ values for Ω_m, Ω_Λ, and H_0 were all below 0.02, which is well within the commonly accepted convergence threshold. Furthermore, plots of $R - 1$ against the number of chains and samples, which will be further discussed in the next section, confirmed that the statistic stabilized quickly, suggesting robust convergence. These results indicate that our posterior estimates are reliable and not significantly influenced by the length or number of chains used.

4 Mini-Project E

4.1 Acceptance Rate of the Metropolis Algorithm

To evaluate the efficiency of the Metropolis-Hastings sampling process, I computed the acceptance rate across multiple independent chains. Each chain was initialized with the same starting point and step sizes, running for 6000 samples. The final acceptance rates ranged from approximately 0.171 to 0.195, with an average acceptance rate of 0.178 across all chains. This value falls indicate that the chains are making sufficiently frequent accepted moves while still thoroughly exploring the parameter space[1]. This balance is essential to ensure that the Markov chains do not get stuck in local regions and that the posterior distribution is sampled efficiently.

4.2 Gelman-Rubin Statistic

To quantitatively assess convergence, I computed the Gelman-Rubin statistic $(R - 1)$ after running multiple chains. This statistic compares the variance within each chain to the variance between chains for each parameter. A value close to zero indicates convergence. After discarding the initial 1,000 samples (burn-in), the final Gelman-Rubin statistics for the three cosmological parameters were 0.0147 for Ω_m, 0.0155 for Ω_Λ, and 0.0076 for H_0. These values confirm that the chains have converged and are sampling from a common stationary distribution[1].

I visualized the convergence behavior using three sets of plots.

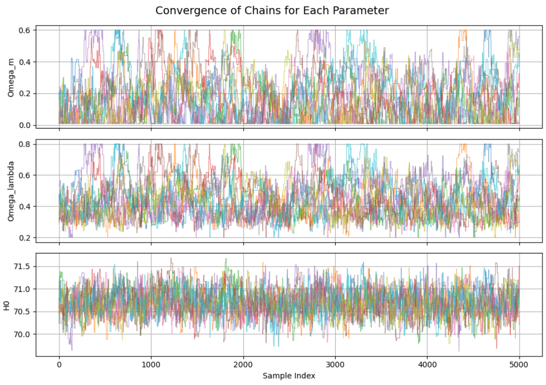

Figure 9: Chain Convergence for Each Parameter

The plot above represents the trace of each chain for Ω_m, Ω_Λ, and H_0, showing how values evolve across sample indices. The chains quickly stabilize and remain within consistent ranges, visually confirming convergence.

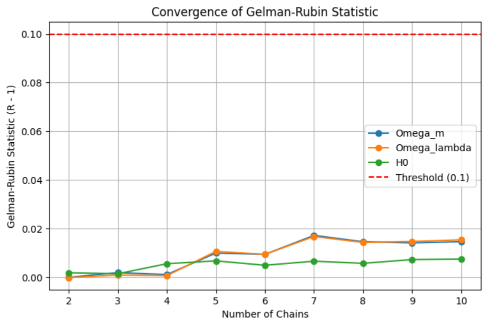

Figure 10: Convergence of Gelman Rubin

This plot illustrates how the Gelman-Rubin statistic decreases with the number of chains. All parameters remain well below the red threshold line, confirming inter-chain agreement.

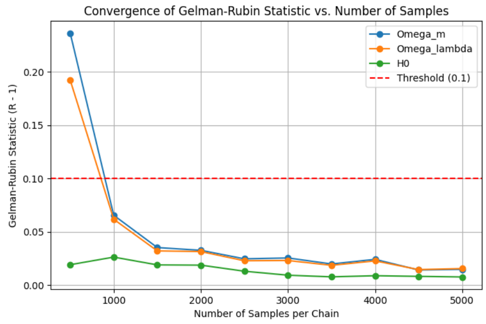

Figure 11: Convergence of Gelman-Rubin vs Sample Number

This plot tracks $(R - 1)$ versus the number of samples per chain, clearly showing convergence after 2000 samples. Together, these plots validate the robustness of our sampling procedure and confirm that the results are not sensitive to chain length beyond a certain point.

5 Conclusions

In this project, I developed a computational framework to analyze cosmological parameters using supernova data through Bayesian inference and Markov Chain Monte Carlo (MCMC) methods. Starting from the construction of a likelihood model, numerical integration, parameter optimization, and uncertainty quantification were progressively implemented. The Metropolis-Hastings algorithm allowed us to efficiently sample from the posterior distribution, and I thoroughly assessed the convergence of our sampling process using both visual diagnostics and quantitative metrics. In particular, the acceptance rates across chains were consistently within an optimal range, and the Gelman-Rubin statistics confirmed strong convergence for all parameters. Our analysis also demonstrated that reliable estimates could be obtained with approximately 2000 samples per chain post burn-in, optimizing the trade-off between computational cost and result accuracy.

References

[1] Joe Zuntz. Computer modelling course — unit 6, 2025. Accessed March 2025.