

Astr 596: Project 4: Measuring the Hubble Constant and Universe Matter Density Parameter using Markov Chain and Hamiltonian Monte Carlo Simulations

Nodoka Masamune

November 2025

1 Executive Summary

In this project we used Markov Chain Monte Carlo (MCMC) and Hamiltonian Monte Carlo (HMC) algorithms to determine the matter density parameter of the universe, Ω_m and the expansion rate of the universe, h . The matter density parameter Ω_m is the ratio of the current matter density, which considers both dark matter and baryonic matter, and the critical density, which is the density needed for a flat universe. The expansion rate, h , the normalized Hubble Constant, is a dimensional value which can be used to calculate the size and age of the universe.

From MCMC sampling we derived results of $\Omega_m = 0.2968 \pm 0.031$ and $h = 0.7028 \pm 0.0071$. From HMC sampling we derived similar results of $\Omega_m = 0.2971 \pm 0.0309$ and $h = 0.7029 \pm 0.0072$. These agree with the accepted value for $\Omega_m \approx 0.3$ well, and fall right between the two contesting values of $h = 0.674 \pm 0.005$ (Perlmutter et al) and 0.730 ± 0.010 (Riess et al), which are from using early universe cosmic microwave and Planck data, and late universe supernova and SH0ES data respectively. I got a correlation value of -0.617 and -0.598 for Ω_m and h from MCMC and HMC, which again, is expected: you can increase one value and decrease the other and maintain a relatively constant distance modulus, or μ .

2 Methodology

We used the Joint Light-Curve Analysis data, which contains redshift and brightness data values of various type IA supernovae in our MCMC and HMC algorithms to derive Ω_m and h (Betoule et al). Type IA supernovae were used because they are a standard candle, meaning they all have basically the same intrinsic brightness as they result from the same explosion of a white dwarf

reaching the Chandrakeshar limit. We assumed a flat universe, which is consistent with past data from CMB and Planck results, which allowed us to reduce the number of parameters to two, as the dark energy parameter, Ω_Δ , reduces to $1 - \Omega_m$. This reduces dimensionality of the problem and allows for better precision in our results. We used a total of 31 data points.

Our equation relating our parameters, Ω_m and h , with our data values, z and μ was (Hogg):

$$D_L = \frac{c(1+z)}{H_0} \int_0^z \frac{1}{E(z')} dz' \quad (1)$$

$$\text{where: } E(z) = \sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)} \quad (2)$$

$$D_L^* = D_L h \quad (3)$$

$$\mu = 25 - 5 \log_{10}(h) + 5 \log_{10} \frac{D_L^*}{Mpc} \quad (4)$$

We also used Pen's 1999 fitting formula to approximate the integral, which can be computationally costly to calculate (Pen):

$$D_L(z) = \frac{c(1+z)}{H_0} [\eta(1, \Omega_m) - \eta(\frac{1}{1+z}, \Omega_m)] \quad (5)$$

$$\text{where:} \quad (6)$$

$$\eta(a, \Omega_m) = 2\sqrt{s^3 + 1} \left[\frac{1}{a^4} - 0.1540 \frac{s}{a^3} + 0.4304 \frac{s}{a^2} + 0.19097 \frac{s^3}{a} + 0.066941 s^4 \right]^{-1/8} \quad (7)$$

$$\text{and: } s^3 = (1 - \Omega_m)/\Omega_m. \quad (8)$$

Since this calculation would have to be repeated thousands of times while sampling, the far more efficient Pen approximation was used during both MCMC and HMC runs.

These equations were then used to create our likelihood functions, which also contained prior constraints on our values, such that $0 \leq \Omega_m \leq 1.0$, and $0.5 \leq h \leq 0.9$. The log likelihood function calculated the logged likelihood to observe our data given model parameters using this equation:

$$\log \mathcal{L}(\Omega_m, h) = -\frac{1}{2}(\mu_{obs} - \mu_{model})^T C^{-1} (\mu_{obs} - \mu_{model})$$

Finally, the log posterior function combines the logged likelihood and the prior constraints data.

$$\log \mathcal{P}(\Omega_m, h | D) = \log \mathcal{L}(\Omega_m, h) + \log \mathcal{P}(\Omega_m, h)$$

Logged probabilities are used to convert probabilities, which are multiplied, into simpler addition equations.

The log posterior function is later called by the MCMC and HMC samplers to explore the possible parameter space of the two parameters.

MCMC sampling begins by randomly generating an initial parameter guess. A new proposal is generated using the initial parameter. The acceptance probability of this proposed parameter is calculated using the logged posterior function, and the proposed parameter is accepted with probability α , where:

$$\alpha = \min \left(1, \frac{\mathcal{P}(\text{proposal}|D)}{\mathcal{P}(\text{initial}|D)} \right).$$

If the move is accepted, the proposal becomes the new parameter. This process is repeated thousands of times, using random walking that naturally spends more time in high probability regions of the parameter spaces. While coding was minimal, MCMC can be highly inefficient, with a far lower acceptance rate than HMC, and greater correlation between steps.

HMC uses a leapfrog algorithm to simulate Hamiltonian equations, and uses the ΔH , or the total ‘energy’ change between the parameter and proposed parameter to reject or accept the proposed parameter. This uses a gradient to propose the next parameter, leading to far more efficient and intelligent exploration of the parameter space. This meant that it converged to a value faster and with greater computational efficiency, as it had a higher acceptance rate.

The difference in autocorrelation between steps in the parameter space can be seen in figure 1 which are plots of autocorrelation for both parameters. The HMC autocorrelation values fell faster and stayed consistently lower, while MCMC autocorrelation values stayed higher and had a greater spread in values.

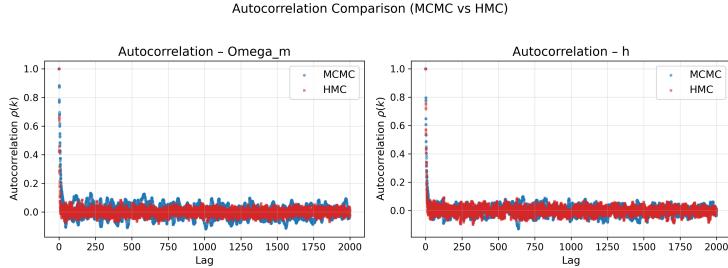


Figure 1: These two plots show the autocorrelation of the two parameters as the sampling progresses for both MCMC and HMC. Note the steeper fall and smaller range of values from HMC compared to MCMC.

3 Results

We ran four chains with 5000 steps for both, with a burn fraction of 0.2. The results of these runs can be seen in the plots below:

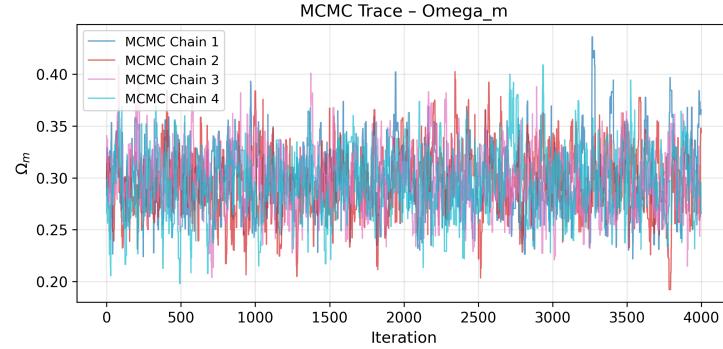


Figure 2: This is the plot of four MCMC chains for Ω_m of length 5000 steps, with 20 percent burn in. The burn in has been removed prior to plotting.

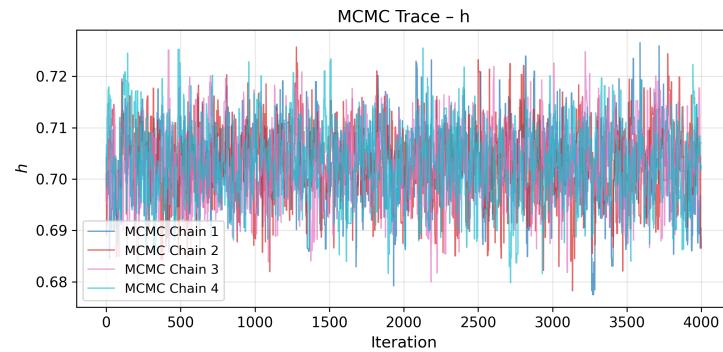


Figure 3: This is the plot of four MCMC chains for h of length 5000 steps, with 20 percent burn in. The burn in has been removed prior to plotting.

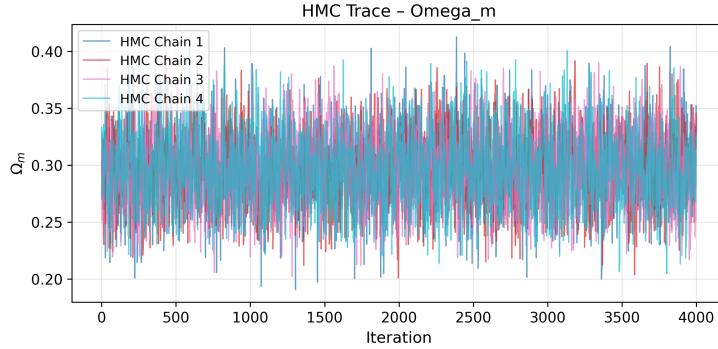


Figure 4: This is the plot of four HMC chains for Ω_m of length 5000 steps, with 20 percent burn in. The burn in has been removed prior to plotting.

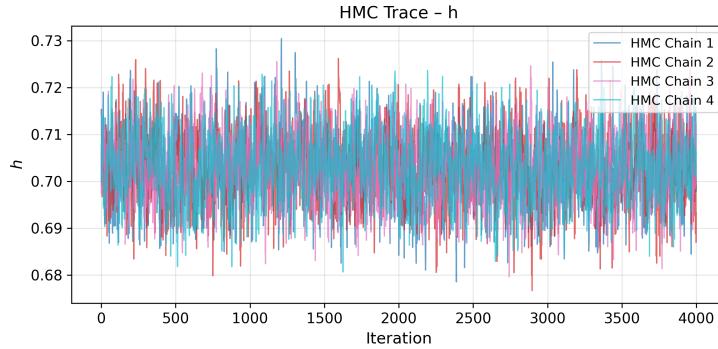


Figure 5: This is the plot of four HMC chains for h of length 5000 steps, with 20 percent burn in. The burn in has been removed prior to plotting.

We derived $\Omega_m = 0.2968 \pm 0.031$ and $h = 0.7028 \pm 0.0071$ from MCMC sampling. From HMC sampling we derived similar results of $\Omega_m = 0.2971 \pm 0.0309$ and $h = 0.7029 \pm 0.0072$. The summary of these runs can be seen in tables 1 and 2 below.

Table 1: **MCMC Posterior Parameter Summary.** Mean values, standard deviations, credible intervals (16th, 50th, and 84th percentiles), and correlation coefficient between Ω_m and h for MCMC chains.

Parameter	Mean	Std. Dev.	16th	50th	84th	Correlation(Ω_m, h)
Ω_m	0.2968	0.0310	0.2668	0.2958	0.3271	
h	0.7028	0.0071	0.6958	0.7030	0.7097	-0.598

Table 2: **HMC Posterior Parameter Summary.** Mean values, standard deviations, credible intervals (16th, 50th, and 84th percentiles), and correlation coefficient between Ω_m and h for HMC chains.

Parameter	Mean	Std. Dev.	16th	50th	84th	Correlation(Ω_m, h)
Ω_m	0.2971	0.0309	0.2659	0.2961	0.3286	
h	0.7029	0.0072	0.6956	0.7029	0.7101	-0.617

We also graphed corner plots of both parameters comparing the results of MCMC and HMC, which is shown in the figure below.

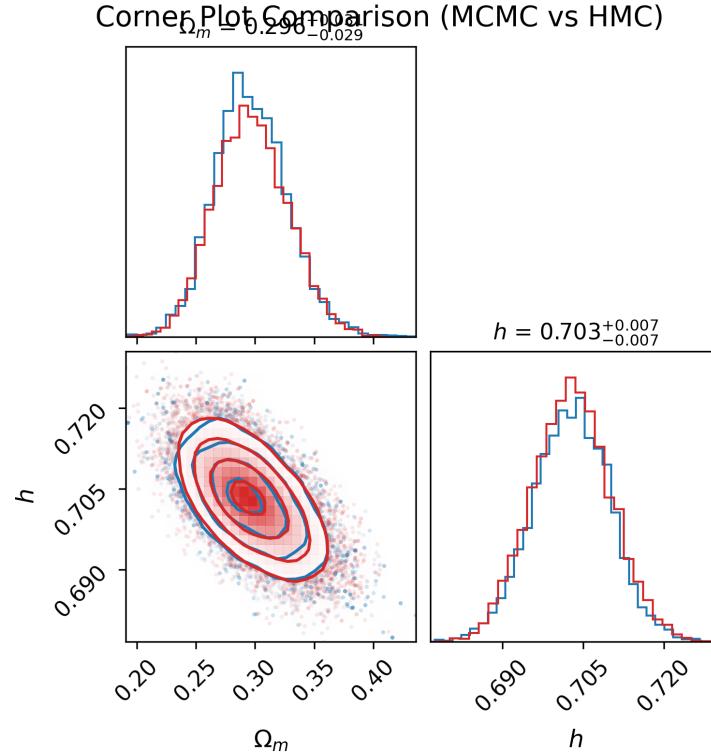


Figure 6: These are the corner plots comparing results from MCMC and HMC. Both methods resulted in similar conclusions, with similar centers and spreads.

These show that both the MCMC and HMC algorithms gave very similar results with near identical spreads.

4 Validation

Validation was done at every step of the project. First, we ensured that both equations relating our parameters and our data were implemented correctly, and checked which one ran faster. We also confirmed that the approximation consistently gave values within 0.4 percent of the actual calculation. The results of these tests can be seen in the table below.

Table 3: **Comparison of Distance Modulus Integrations.** Evaluation of the theoretical distance modulus $\mu(z)$ at $z = 0.5$ using both the integral and Pen approximations. Differences $\Delta\mu$ and relative percentage deviations are shown, along with the relative computational speed factor.

Boundary	$\mu_{\text{int}}(0.5)$	$\mu_{\text{pen}}(0.5)$	$\Delta\mu$ target (mag)	Max $\Delta\mu$ (%) over $z \in [0, 1.3]$	Pen faster (\times)
Zero	42.263	42.269	0.003	0.018	5.33

As you can see, both the integration and approximation gave the correct numerical μ values, and the maximum percent difference between the two methods within the tested z values of 0 to 1.3 was 0.018

Validation tests for MCMC included ensuring that sigma values for proposing were tuned correctly to ensure an acceptance rate between 0.2 and 0.5, and that the Gelman–Rubin value, \hat{R} , which quantifies the ‘mixing’ between chains is less than 1.1. The results of this can be seen in table 4 below.

Table 4: **MCMC Chain Diagnostics.** Effective sample sizes (ESS), autocorrelation times (τ), and acceptance rates for four Metropolis–Hastings chains. The final row reports Gelman–Rubin convergence statistics (\hat{R}) for each parameter.

Chain	Init Ω_m	Init h	ESS $_{\Omega_m}$	τ_{Ω_m}	ESS $_h$	τ_h	Acceptance
1	0.306	0.730	353.3	14.15	494.9	10.10	0.216
2	0.824	0.508	120.3	41.55	136.3	36.68	0.220
3	0.684	0.951	282.2	17.72	261.8	19.10	0.228
4	0.931	0.821	90.0	55.57	306.4	16.32	0.230
\hat{R}			1.002		1.003		

The Gelman–Rubin value was also calculated for truncated sections of the chains showing that both the Ω_m and h values were converging properly within the steps we were running. The figure below shows plotted values of both parameters.

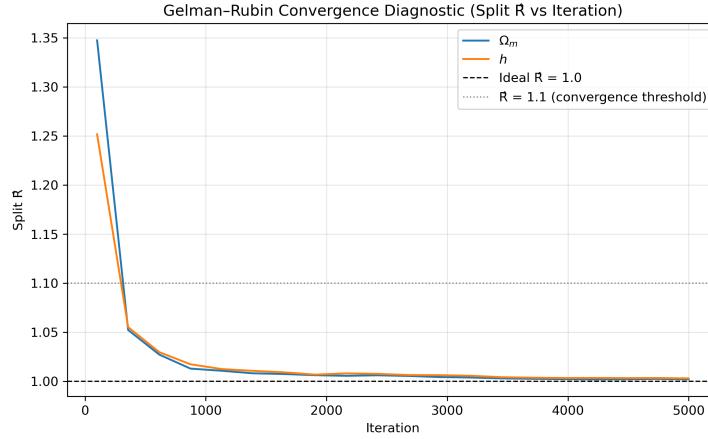


Figure 7: This plot shows the truncated Gelman Rubin value for MCMC runs, showing that the chain has converged within the 5000 steps.

The HMC algorithm was validated by ensuring that tuning was achieved such that the energy between parameter proposals, ΔH , was relatively conserved and that the acceptance rate was between 60 and 90 percent. Tuning in this instance means adjusting the ϵ and L values, which control the step size and number of steps in each leapfrog integration step. ΔH values were plotted against step and as a histogram, which can be seen in figures 8 and 9 respectively.

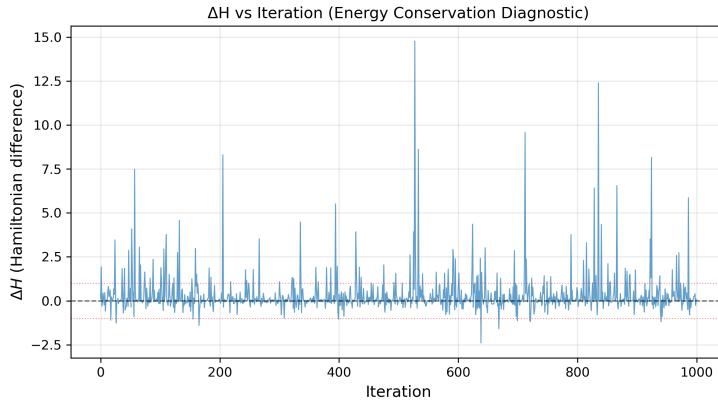


Figure 8: This is the plot of ΔH vs iteration from a single HMC run. This graph shows that generally energy is conserved between steps. The peaks represent instances of rejection.

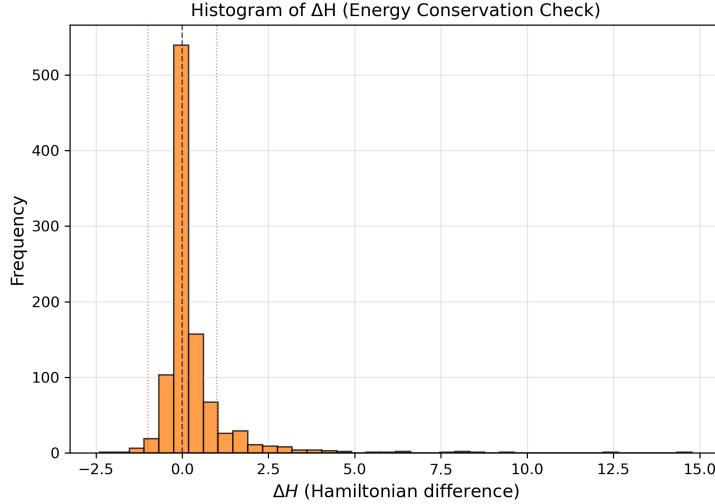


Figure 9: This is a histogram of all Δh values. Note the high peak around zero, once again showing that energy is generally conserved.

While figure 8 shows random high peaks, they represent instances where the proposal was rejected, and are expected. The histogram in figure 9 shows that the ΔH values are strongly centered around 0, showing that energy is relatively conserved between steps, and that the HMC is properly tuned. While in the final analysis, the acceptance rate of each HMC chain was outputted. These can be seen in table 5.

Table 5: **HMC Chain Acceptance Rates.** Acceptance fractions for four independent Hamiltonian Monte Carlo (HMC) chains, tuned for stable sampling performance.

Chain	1	2	3	4
Acceptance rate	0.727	0.772	0.676	0.773

5 Extension

I chose to do extension 4, which reweights the posterior samples from a MCMC run using a Gaussian prior that comes from data from the Planck CMB study, which puts stricter constraints on the Hubble Constant. This Gaussian takes on the form:

$$h \sim \mathcal{N}(0.674, 0.005^2). \quad (9)$$

This process reweights the posterior results with how consistent they are with the Planck CMB data.

This allows us to combine various results: our MCMC chain and information from the Planck CMB studies, without having to rerun our MCMC chain, which is computationally expensive. The results of this can be seen in figures 10 and 11, below.

Joint Posterior: Before (blue) vs After (orange) Planck Prior

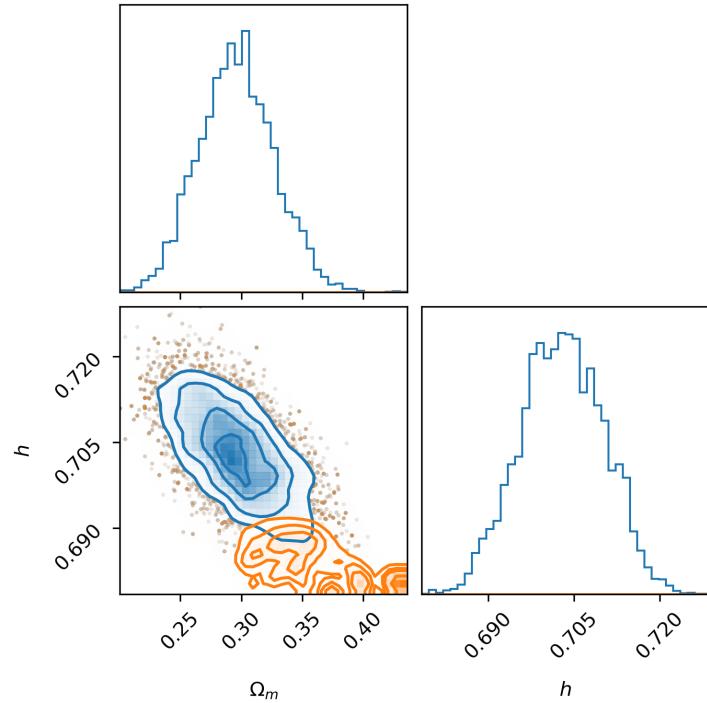


Figure 10: These are the corner plots showing the result of the Planck reweighting. It shows how the value of h has shifted dramatically compared to the original MCMC run.

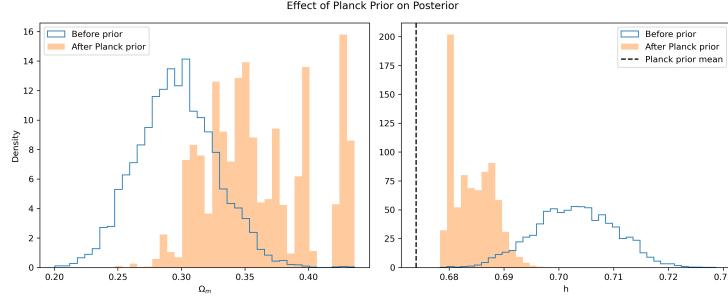


Figure 11: These are histograms of the results from the Planck gaussian reweighting. The histogram width has narrowed significantly in the h posterior, showing how the reweighting has led to a more precise result.

As you can see, the Planck reweight shifted the best-fit parameters and also narrowed the possible value of h considerably. Since Ω_m is correlated with h , it also narrowed Ω_m but not as dramatically as h . This extension showed how various results from independent datasets can be used to derive the parameters.

6 Conclusions

We were successful at using MCMC and HMC sampling to derive Ω_m and h values most consistent with the provided datasets. MCMC and HMC results were consistent with each other and past results.

From MCMC we derived $\Omega_m = 0.2968 \pm 0.031$ and $h = 0.7028 \pm 0.0071$; from HMC sampling we derived similar results of $\Omega_m = 0.2971 \pm 0.0309$ and $h = 0.7029 \pm 0.0072$.

Our h value falls in between the two most agreed upon values of h , which are derived from two different datasets: early universe and late universe.

While HMC is more computationally efficient due to its higher acceptance rate and intelligent posterior exploration compared to MCMC's low acceptance rate and random walk, it still took longer to run chains of the same length as MCMC.

My extension showed that various results and independent datasets can be combined during analysis to get more precise results. Using the Planck CMB Gaussian, my MCMC results for h were shifted lower, had a significantly smaller standard deviation and were therefore more precise.

7 References

- Betoule et al., 2014, ‘Improved cosmological constraints from a joint analysis...’
(JLA sample) (AA 568, A22)
- Hogg, 1999, ‘Distance measures in cosmology’ (arXiv:astro-ph/9905116)
- Pen, 1999, ‘Brief Note: Analytical Fit to the Luminosity Distance for Flat

Cosmologies with a Cosmological Constant' (arXiv:astro-ph/9904172)
Perlmutter et al., 1999, 'Measurements of Omega and Lambda from 42 High-
Redshift Supernovae' (ApJ 517, 565)
Riess et al., 1998, 'Observational Evidence from Supernovae for an Accelerating
Universe...' (AJ 116, 1009)