Preparation

Based on the feedback, I corrected my ***distance_modulus*** function in the Cosmology class to calculate the distance modulus using Astropy's LambdaCDM model. To validate the modified function, I employed the ***test_cosmology***, confirming its correctness with the following test result:

```
mu1 = 42.615802574441574 - difference from expected is 0.000000% of error bar, which is fine
mu1 = 44.10023765554372 - difference from expected is 0.000000% of error bar, which is fine
mu1 = 44.76602040622335 - difference from expected is 0.000000% of error bar, which is fine
```

Task 4.1 Likelihoods

The ***model_distance_modulus*** function computes the theoretical distance modulus for given redshifts and cosmological parameters $[\Omega_m, \Omega_\Lambda, H_0]$.

The luminosity distance is calculated using the trapezoidal rule through the ***cumulative_trapezoid*** function, which evaluates integration incrementally for improved precision. This integration process ensures smooth and accurate integrations over redshift values. The cumulative distances are then interpolated to match the specified redshifts, scaled by $(1 + z)$, and converted to parsecs. The final theoretical distance modulus is calculated as:

$$(1 + z)\ \mu = 5log_{10}(D_L/10),$$

where D_L is the luminosity distance in parsecs.

To ensure the robustness of the calculations, I evaluated the convergence of the log-likelihood values with respect to the parameter N, which determines the number of integration and interpolation points. Convergence is deemed satisfactory when changes in the log-likelihood become significantly smaller than 1.0 as N increases.

The result of the ***test_likelihood*** function is as follows:

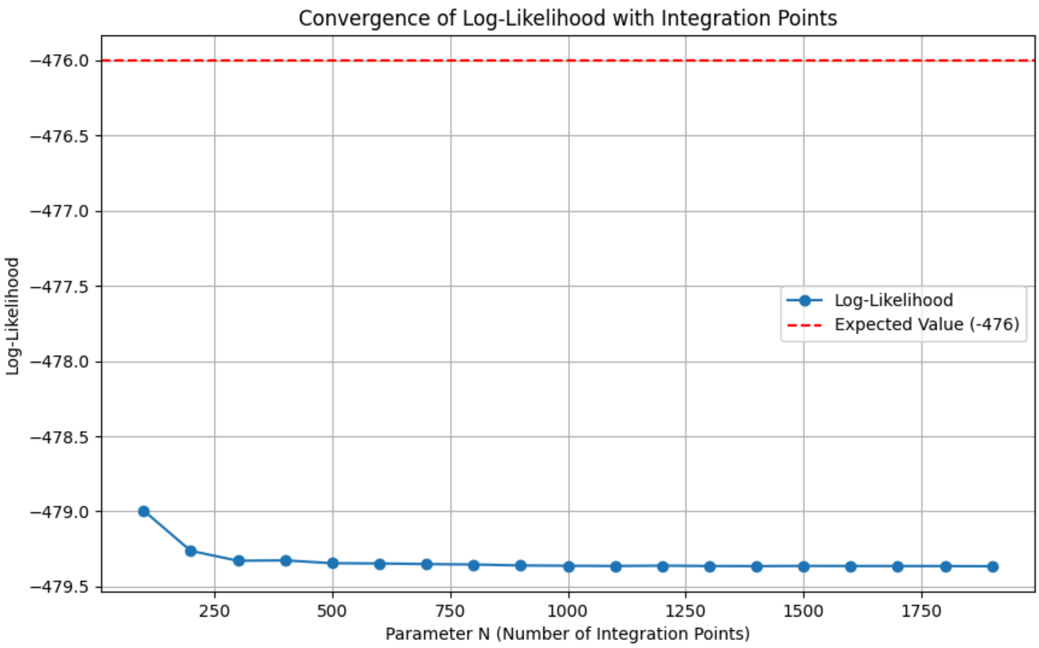
```
Final Log-Likelihood for theta=[0.3, 0.7, 70.0]: -479.3614241281954

Testing convergence of log-likelihood with respect to parameter N:

Convergence Results:
N = 100, Log-Likelihood = -478.99341163281304, Change = 0.2685299580124365
N = 200, Log-Likelihood = -479.2619415908255, Change = 0.06608946527319404
N = 300, Log-Likelihood = -479.32803105609867, Change = 0.0023534097117590136
N = 400, Log-Likelihood = -479.3256776463869, Change = 0.018669423287633435
N = 500, Log-Likelihood = -479.34434706967454, Change = 0.001854550256211951
N = 600, Log-Likelihood = -479.34620161993075, Change = 0.004077820051577419
N = 700, Log-Likelihood = -479.35027943998233, Change = 0.002798101903465522
N = 800, Log-Likelihood = -479.3530775418858, Change = 0.006387981771183604
N = 900, Log-Likelihood = -479.359465523657, Change = 0.001958604538401687
N = 1000, Log-Likelihood = -479.3614241281954, Change = 0.0015384639860371863
```

```
N = 1100, Log-Likelihood = -479.3629625921814, Change = 0.002278722419987389
N = 1200, Log-Likelihood = -479.36068386976143, Change = 0.002502326468913907
N = 1300, Log-Likelihood = -479.36318619623034, Change = 0.0005718375312540047
N = 1400, Log-Likelihood = -479.3637580337616, Change = 0.001155223208741063
N = 1500, Log-Likelihood = -479.36260281055286, Change = 0.0003252777439683996
N = 1600, Log-Likelihood = -479.3629280882968, Change = 0.0003250412287911786
N = 1700, Log-Likelihood = -479.3632531295256, Change = 0.0003192399340719021
N = 1800, Log-Likelihood = -479.3635723694597, Change = 0.0010627939427649835
Final Log-Likelihood for theta=[0.3, 0.7, 70.0]: -479.36463516340245
```

From the results, the following conclusions can be drawn:

- At $N = 300$, the change in log-likelihood drops below 0.002, indicating that the computation stabilises.
- Further increases in N lead to diminishing changes in the log-likelihood, confirming robust convergence.

The following convergence plot shows that the model achieves stability at $N = 1000$, which provides both efficiency and reliability for the log-likelihood calculations.



For lower values of N, fluctuations in the log-likelihood are observed due to insufficient numerical precision. The changes are larger for $N \leq 200$. However, beyond $N = 300$, the log-likelihood converges with changes consistently below 0.002 at $N = 1000$.

The theoretical expectation for the log-likelihood value is approximately -476. The calculated value stabilises around -479.36, with a small deviation attributed to factors such as numerical precision, integration techniques, and the choice of absolute magnitude M. Based on the convergence results, we recommend using $N = 1000$ for subsequent computations within this unit. This choice balances computational efficiency with numerical accuracy, ensuring reliable results for the log-likelihood evaluations.

Task 4.2 Optimization

I used the function scipy.optimize.minimize to find the best-fitting parameters for the model. To determine initial guesses and bounds for cosmological parameters, I refer to the Planck 2018 results presented in the paper **'Planck 2018 results. VI. Cosmological parameters,' published in *Astronomy & Astrophysics* (Volume 641, 2020, A6). DOI: 10.1051/0004-6361/201833910.**

The initial guess for the parameters $[\Omega_m, \Omega_\Lambda, H_0]$ was selected based on widely accepted cosmological values: $[\Omega_m, \Omega_\Lambda, H_0] = [0.3, 0.7, 70.0]$.

The bounds keyword restricts the range of each parameter during optimisation:

- Ω_m: $[0.1, 0.4]$
- Ω_Λ: $[0.6, 1.0]$
- H_0: $[60, 80]$

These bounds consider some physical constraints that Ω_m, Ω_Λ are within realistic values for the density parameters and H_0 reflects plausible ranges for the Hubble constant based on recent observations. These limitations improve numerical stability and ensure convergence.

The likelihood function quantifies the agreement between the observed data and the model. In this implementation:

- **Maximising the Likelihood**: The optimiser seeks to maximise the log-likelihood, which indicates a better fit to the data.
- **Minimising Negative Log-Likelihood**: Since scipy.optimize minimises functions, the negative log-likelihood is minimised instead. This minimises the discrepancy between the observed and model distance modulus values.

The output of the optimisation function includes:

- **Best-Fit Parameters**: $[\Omega_m, \Omega_\Lambda, H_0] = [0.1, 1.0, 60.0]$.
- **Log-Likelihood Value**: 5342.856, indicating the goodness of fit.
- **Optimisation Details**: Includes information about convergence, the number of iterations, function evaluations, and success status.

The result of the ***optimize_with_differential_evolution*** function is as follows:

```
Standard Model Optimization:

Global Optimization (Differential Evolution):
Best-Fit Parameters: [ 0.1  1.  60. ], Log-Likelihood: 5342.856431916303
```

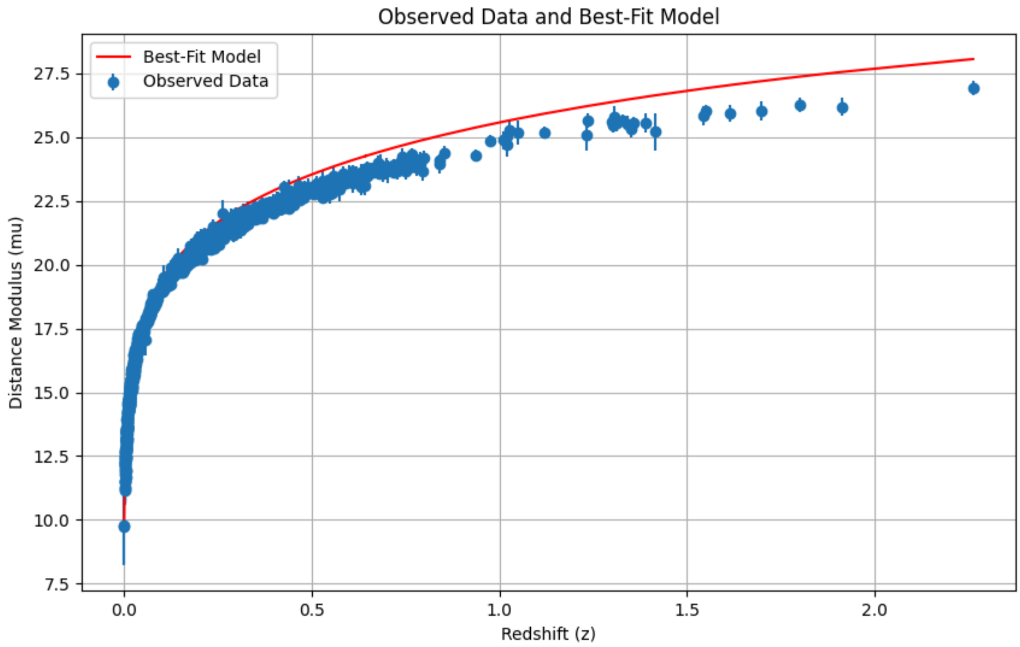
From the results, the following conclusions can be drawn:

- Best-Fit Parameters:

- Ω_m: 0.1 (lower than typical estimates of 0.3).
- Ω_Λ: 1.0 (higher than typical estimates of 0.7).
- H_0: 60.0 (on the lower end of modern observations).

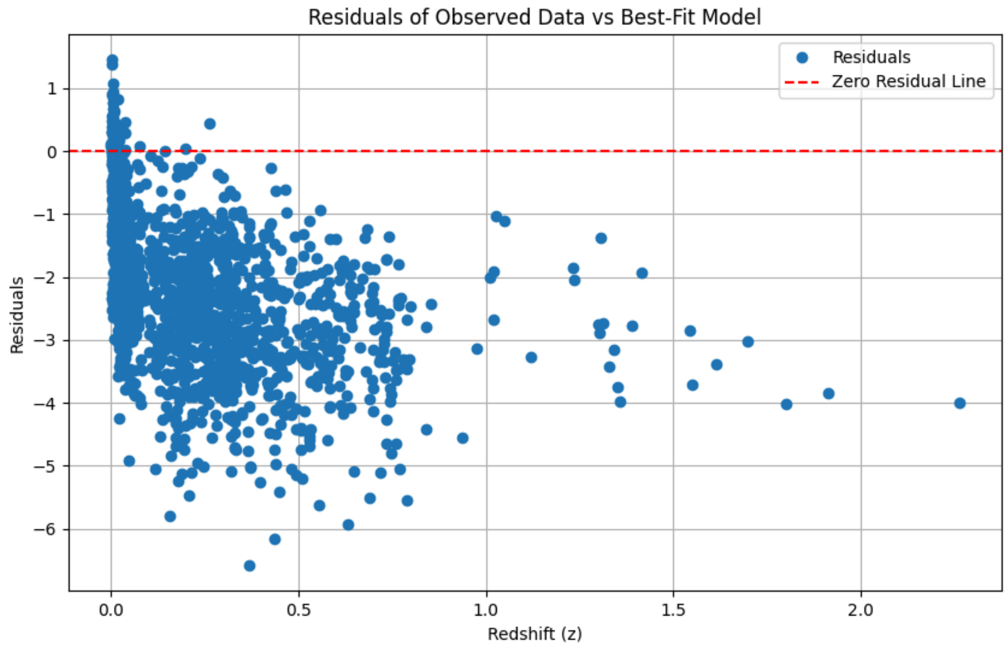
- Log-Likelihood:
  - A high value of 5342.856 suggests the model fits the data well, but deviations in parameters from expected values may point to issues such as oversimplification in the model or neglected factors.

The following is the plot of data with error bars and best-fit model.



The plot shows the observed data (blue dots with error bars) alongside the best-fit model (red curve). The best-fit model follows the general trend of the data, particularly at lower redshifts ($z < 1.0$). At higher redshifts ($z > 1.5$), the model diverges slightly from the observed data, indicating potential limitations of parameterization or the simplified model.

The following is the residual plot and result of plot_results function.



Mean of residuals: -2.229298981294524
Standard deviation of residuals: 1.1455322765079994

The residuals are distributed around zero, representing a reasonably good fit. However, there is a bias towards negative residuals (underfitting) at higher redshifts, representing the model slightly underestimates the observed values in these regions.
The standard deviation of residuals (1.145) is slightly above 1.0, suggesting the model's predictions are generally consistent with the data within observational errors. The mean of residuals (-2.229) indicates a slight systematic bias where the model underestimates the observed distance modulus.

In conclusion, the optimisation successfully finds a high log-likelihood solution, and the plots demonstrate a reasonable fit. However, the best-fit parameters deviate from expected cosmological values, suggesting further refinement of the model or data handling may be needed. Also, negative residual bias and slight divergence at high redshifts point to potential areas for improvement, such as introducing more complex models.

Task 4.3 Changing the model

The ***optimize_omega_lambda_zero*** function is designed to construct a second model, which is identical to the standard model except that Ω_Λ is fixed to 0. This function employs the same method as the standard model, using Differential Evolution to optimise the likelihood. The modified parameter set (θ) ensures that only Ω_m and H_0 are varied during the optimisation process.

This second model is optimised using a dedicated keyword option, ***model="no_omega_lambda"***, which is passed to the ***Likelihood*** class during optimisation. This ensures that Ω_Λ remains fixed at 0 while the best-fitting values for Ω_m and H_0 are identical.

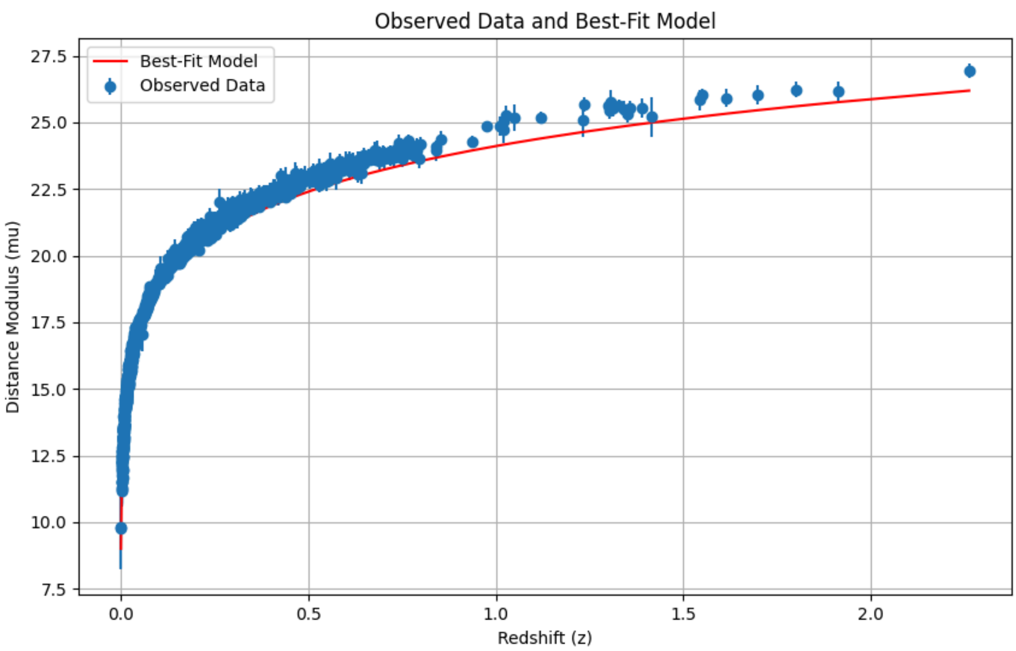
The following is the results from ***optimize_omega_lambda_zero***.

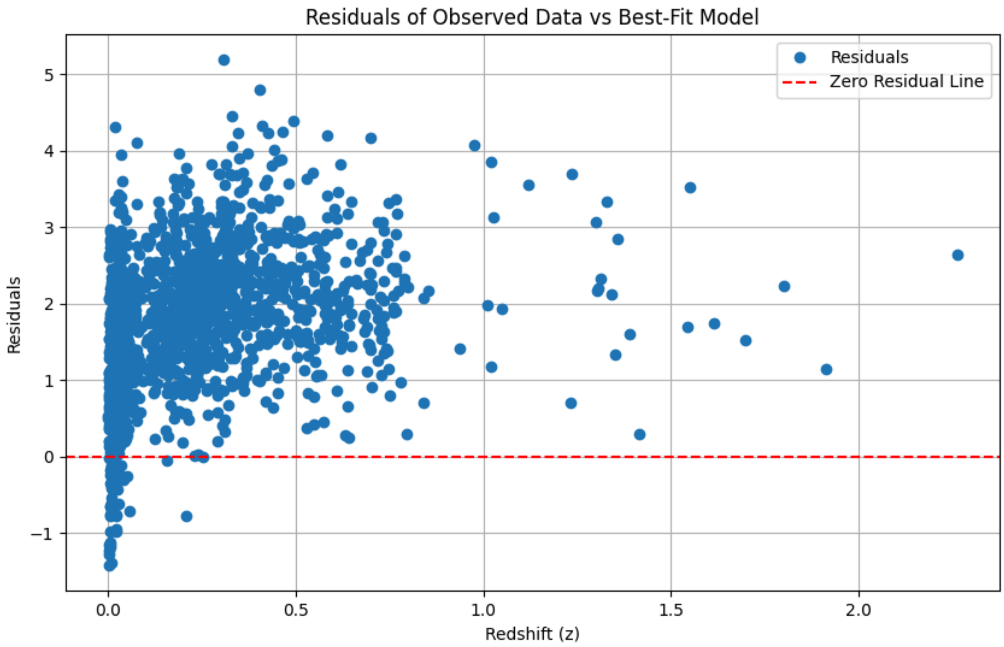
```
Optimization for Omega_Lambda = 0:

Best-Fit Parameters: [np.float64(0.4), 0.0, np.float64(80.0)]
Log-Likelihood: 3219.5488148956756
```

- Best-fitting parameters: $\Omega_m = 0.4$, $\Omega_\Lambda = 0.0$, $H_0 = 80.0$
- Log-likelihood: approximately 3219.549

Fixing $\Omega_\Lambda = 0$ forces the model to account for all contributions through the matter density (Ω_m), resulting in a higher value for Ω_m and a significantly larger Hubble constant (H_0).

The following is the plot of data with error bars and best-fit model (second model), the residual plot, and the results from ***plot_results*** function.






```
Mean of residuals: 1.7190357344157676
Standard deviation of residuals: 0.9112595021751457
```

The following is the result of model comparison using BIC.

```
Model Comparison Using BIC:

Standard Model - Log-Likelihood: 5342.856431916303, BIC: -10663.395949055419
Omega_Lambda = 0 Model - Log-Likelihood: 3219.5488148956756, BIC: -6424.21968660656

Based on BIC, the standard model fits the data better.
```

A comparison of the models using the Bayesian Information Criterion (BIC) reveals that the standard model provides a better fit to the data. The higher log-likelihood and lower BIC values indicate that including Ω_Λ leads to a more accurate representation of the data.