A Ricci Flow-Inspired Model for Cosmic Expansion: New Insights from BAO Measurements Preliminary Report

Paul Chun-Kit Lee, MD,
Division of Cardiology
NYU Langone Hospital Brooklyn, NY
New York University, New York
(dr.paul.c.lee@gmail.com)

August 14, 2024

Abstract

Recent precision measurements of Baryon Acoustic Oscillations (BAO) by surveys such as the Dark Energy Spectroscopic Instrument (DESI) have revealed tensions with predictions from the standard ΛCDM cosmological model. This paper presents a novel approach to addressing these discrepancies by incorporating geometric flow concepts inspired by Perelman's work on Ricci flow. We introduce a modified Friedmann equation that includes a Ricci flow term, providing a geometric framework for understanding potential deviations from standard cosmology. Our model shows significant improvement in fitting DESI BAO measurements across a wide range of redshifts, suggesting a possible geometric origin for observed cosmic expansion anomalies. Parameter space analysis reveals subtle interplay between logarithmic and power-law contributions to the expansion history, potentially offering new insights into the nature of dark energy or modifications to general relativity on cosmological scales.

1 Introduction

The ΛCDM model has been remarkably successful in describing a wide range of cosmological observations. However, recent high-precision measurements, particularly from Baryon Acoustic Oscillation (BAO) surveys such as the Dark Energy Spectroscopic Instrument (DESI), have revealed potential inconsistencies with ΛCDM predictions [1, 2]. These discrepancies hint at the possibility of evolving dark energy or modifications to our understanding of gravity on cosmological scales.

In parallel, the mathematical community has seen significant advancements in geometric analysis, most notably Perelman's use of Ricci flow in proving the

Poincaré conjecture [3]. Ricci flow, introduced by Hamilton and extensively developed by Perelman, describes how a metric evolves to smooth out irregularities in curvature:

$$\frac{\partial g_{\mu\nu}}{\partial t} = -2R_{\mu\nu}$$

where $g_{\mu\nu}$ is the metric tensor and $R_{\mu\nu}$ is the Ricci curvature tensor.

Building upon recent work applying Ricci flow techniques to general relativity and quantum gravity [4], our paper explores the application of these geometric flow concepts to cosmology, aiming to provide a mathematically motivated framework for understanding cosmic expansion anomalies and addressing the observed tensions in BAO measurements.

2 Theoretical Framework

Our work builds directly upon the foundational mathematics developed in "Ricci Flow Techniques in General Relativity and Quantum Gravity: A Perelman-Inspired Approach to Spacetime Dynamics" [4], particularly the derivations presented in Appendix A of that paper. Here, we outline the key mathematical concepts and their adaptations to cosmology.

2.1 Lorentzian Ricci Flow

The cornerstone of our approach is the Lorentzian Ricci flow, first introduced in Equation 1.1 of Appendix A [4]:

$$\frac{\partial g}{\partial t} = -2\,\mathrm{Ric}(g) \tag{1}$$

where g is the metric tensor and Ric is the Ricci curvature tensor. This fundamental equation describes how the geometry of spacetime evolves under the flow.

2.2 Evolution of Scalar Curvature

Crucially, Theorem 1.2 in Appendix A [4] derived the evolution equation for scalar curvature R under Lorentzian Ricci flow:

$$\frac{\partial R}{\partial t} = \Box R + 2|\mathrm{Ric}|^2 \tag{2}$$

where $\Box$ is the Lorentzian d'Alembertian operator. This equation is vital for understanding how the overall curvature of spacetime changes over time.

2.3 Entropy Functionals

Appendix A [4] introduced Lorentzian analogues of Perelman's entropy functionals, which are key to analyzing the behavior of the flow. The F-functional, defined in Equation 1.2 of Appendix A, is given by:

$$F[g, f] = \int_M \left(R + |\nabla f|^2\right) e^{-f} \, dV \tag{3}$$

And the W-functional, from Equation 1.3 of Appendix A:

$$W[g, f, \tau] = \int_M \left(\tau \left(R + |\nabla f|^2\right) + f - n^{-\frac{n}{2}}\right) e^{-f} \, dV \tag{4}$$

These functionals provide crucial insights into the thermodynamic-like properties of spacetime under Ricci flow.

2.4 Application to Cosmology

Building on these foundations, we adapt the Ricci flow framework to cosmology. Inspired by the modified Ricci flow for FLRW spacetimes presented in Section 5.1 of Appendix A [4], we propose a modification to the standard Friedmann equation:

$$\left(\frac{H}{H_0}\right)^2 = \Omega_m(1+z)^3 + \Omega_r(1+z)^4 + \Omega_\Lambda + \Omega_{\rm RF}(z) \tag{5}$$

Here, $\Omega_{\rm RF}(z)$ represents the contribution from Ricci flow effects. To capture the complexity of this geometric effect on cosmic expansion, we parameterize $\Omega_{\rm RF}(z)$ as:

$$\Omega_{\rm RF}(z) = \lambda_1 \ln(1+z) + \lambda_2 (1+z)^n \tag{6}$$

This formulation, inspired by the logarithmic nature of Perelman's entropy functional and the flexibility needed to model redshift dependence, allows us to explore how geometric flow might influence cosmic expansion history.

2.5 Modified Ricci Flow in General Relativity

Finally, we consider a modified Ricci flow that incorporates the cosmological constant, adapting the approach outlined in Section 5.1 of Appendix A [4]:

$$\frac{\partial g_{\mu\nu}}{\partial \tau} = -2\left(R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} + \Lambda g_{\mu\nu}\right) \tag{7}$$

This equation forms the basis of our analysis of spacetime evolution under Ricci flow in a cosmological context, allowing us to explore how geometric flow effects might manifest in observable cosmic phenomena.

3 Methodology

We implemented this model computationally and optimized the parameters λ_1, λ_2, and n to best fit recent DESI BAO measurements. The model's predictions were compared to both standard ΛCDM and DESI data for the quantities D_M/r_d, D_H/r_d, and D_V/r_d across a range of redshifts ($0.3 \leq z \leq 1.49$).

We used a χ^2 minimization approach to find the best-fit parameters for our Ricci flow model. We also calculated the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to compare the models while accounting for their different complexities.

4 Results

Our analysis used a total of 26 data points from the DESI BAO measurements. The best-fit parameters for our Ricci flow model were:

$$\lambda_1 = 0.3391, \quad \lambda_2 = -0.0864, \quad n = 2.7475$$

The comparison of the models yielded the following results:

- **ΛCDM model:**
  - $\chi^2 = 73.44$
  - AIC = 73.44
  - BIC = 73.44
- **Ricci Flow model:**
  - $\chi^2 = 14.85$
  - AIC = 20.85
  - BIC = 24.63

The Ricci Flow model improves the fit by 79.78% compared to ΛCDM. This improvement is statistically significant, with a p-value < 0.0001.

Detailed comparisons show that the Ricci Flow model generally provides better fits for both D_M/r_d and D_H/r_d across most redshifts, with particularly good agreement at higher redshifts ($z > 1$).

Parameter space analysis reveals:

- **Best-fit values:** $\lambda_1 \approx 0.3391$, $\lambda_2 \approx -0.0864$
- The model is more sensitive to λ_2 than λ_1, suggesting the power-law term has a stronger impact on the fit to observational data.
- The positive λ_1 indicates an increasing effect of Ricci flow at higher redshifts.
- The slightly negative λ_2 implies a small negative contribution from the power-law term, potentially counterbalancing the logarithmic term at very high redshifts.

5 Discussion

The results of our analysis provide strong evidence for the potential importance of geometric flow effects in cosmic expansion:

1. **Significant Improvement in Fit:** The substantial reduction in χ^2 (from 73.44 to 14.85) indicates that the Ricci Flow model describes the DESI BAO data much more accurately than standard ΛCDM.

2. **Model Complexity vs. Fit Improvement:** Even when penalizing for additional parameters, the Ricci Flow model outperforms ΛCDM as indicated by lower AIC and BIC values.

3. **Parameter Interpretation:** The positive λ_1 and negative λ_2 suggest a complex interplay between the logarithmic and power-law terms in the Ricci flow contribution. This interplay could be interpreted as a dynamic balance between geometric effects that enhance expansion (logarithmic term) and those that moderate it (power-law term) over cosmic history.

4. **Redshift Dependence:** The model's improved performance across a wide range of redshifts, particularly at higher z, suggests that it captures aspects of cosmic expansion history that ΛCDM might be missing. This could indicate that geometric flow effects become more pronounced in the early universe.

5. **Implications for Dark Energy:** The non-zero best-fit values for λ_1 and λ_2 suggest that Ricci flow modification offers a new perspective on dark energy. Rather than a cosmological constant, dark energy in this framework emerges from the dynamic geometry of spacetime itself.

6. **Modified Gravity Interpretation:** Alternatively, these results could be interpreted as evidence for modifications to general relativity on cosmological scales. The Ricci flow terms might be capturing effective corrections to Einstein's equations that become relevant at large scales or early times.

7. **Predictive Power:** The tight constraints on λ_2 suggest that future observations could further test and refine this model, potentially leading to testable predictions that distinguish it from ΛCDM.

These findings suggest that incorporating geometric flow effects could be crucial for understanding cosmic expansion and potentially resolving some of the tensions in current cosmological observations.

6 Conclusion

Our Ricci flow-inspired model provides a substantially better fit to DESI BAO data compared to the standard ΛCDM model. This improvement is statistically significant and robust, even when accounting for the increased model complexity.

The success of this approach suggests that geometric flow concepts could play an important role in our understanding of cosmic expansion and the nature of dark energy.

The interplay between logarithmic and power-law contributions in our model offers a new perspective on the evolution of the universe, potentially bridging the gap between quantum gravity approaches and large-scale cosmology.

Further work is needed to explore the full implications of this model, including its predictions for other cosmological observables and its potential to address other tensions in cosmological data. Nonetheless, these results open up exciting new avenues for understanding the geometry of our expanding universe.

References

[1] DESI Collaboration, et al. (2024). *DESI 2024 III: Baryon Acoustic Oscillations from Galaxies and Quasars*. arXiv:2404.03000v1 [astro-ph.CO].

[2] DESI Collaboration, et al. (2024). *First Results from DESI Make the Most Precise Measurement of Our Expanding Universe*. arXiv:2404.03000v1 [astro-ph.CO].

[3] Perelman, G. (2002). *The entropy formula for the Ricci flow and its geometric applications*. arXiv:math/0211159.

[4] Lee, Paul C-K. (2024).*Ricci Flow Techniques in General Relativity and Quantum Gravity: A Perelman-Inspired Approach to Spacetime Dynamics*. viXra.org e-Print archive, viXra:2407.0165

Addendum 1: Computer Printout

Total Number of Data Points

- **Total number of data points:** 26

ΛCDM Results

- $\chi^2 = 73.44$
- AIC = 73.44
- BIC = 73.44

Detailed Comparison with DESI Data

z	D_M/r_d (Model)	D_H/r_d (Model)	D_V/r_d (Model)	D_M/r_d (DESI)	D_H/r_d (DESI)	D_V/r_d (DE
0.30	-	-	7.8038	-	-	7.93
0.51	12.8861	21.7436	12.2569	13.62	20.98	-
0.71	16.9814	19.2577	15.7981	16.85	20.08	-
0.92	20.7795	16.9710	18.8910	21.81	17.83	-
0.93	20.9487	16.8705	19.0243	21.71	17.88	-
0.95	-	-	19.2873	-	-	20.01
1.32	26.8367	13.4943	23.4096	27.79	13.82	-
1.49	-	-	24.9129	-	-	26.07

Ricci Flow Results

- **Best-fit parameters:** $\lambda_1 = 0.3391$, $\lambda_2 = -0.0864$, $n = 2.7475$
- $\chi^2 = 14.85$
- AIC = 20.85
- BIC = 24.63

Detailed Comparison with DESI Data

z	D_M/r_d (Model)	D_H/r_d (Model)	D_V/r_d (Model)	D_M/r_d (DESI)	D_H/r_d (DESI)	D_V/r_d (DE
0.30	-	-	8.0061	-	-	7.93
0.51	13.2266	22.3562	12.5880	13.62	20.98	-
0.71	17.4527	19.9536	16.2805	16.85	20.08	-
0.92	21.4051	17.7395	19.5549	21.81	17.83	-
0.93	21.5820	17.6417	19.6971	21.71	17.88	-
0.95	-	-	19.9781	-	-	20.01
1.32	27.7831	14.3106	24.4304	27.79	13.82	-
1.49	-	-	26.0757	-	-	26.07

Interpretation of Results

- The Ricci Flow model improves the fit by 79.78% compared to ΛCDM.
- **Ricci Flow Parameter Interpretation:**
  - $\lambda_1 = 0.3391$: Positive contribution from logarithmic term.
  - $\lambda_2 = -0.0864$: Negative contribution from power-law term.
  - $n = 2.7475$: Power-law index, indicating the strength of redshift dependence.
- **Statistical Significance:** The improvement in fit is statistically significant at the 5% level (p-value < 0.0001).

Conclusion

- The Ricci Flow model provides a substantially better fit to the DESI BAO data compared to ΛCDM.
- This suggests that incorporating geometric flow effects could be important for understanding cosmic expansion.

Parameter Errors (from Monte Carlo Simulation)

- $\sigma(\lambda_1) = 0.0289$
- $\sigma(\lambda_2) = 0.0096$
- $\sigma(n) = 0.0355$

Bayesian Model Comparison

- ΛCDM BIC: 73.44
- Ricci Flow BIC: 24.63
- ΔBIC (ΛCDM - Ricci Flow): 48.82
  Very strong evidence in favor of the Ricci Flow model.

Appendix: Python codes

```

Created on Wed Aug 14 08:05:19 2024

@author: Paul C Lee MD
"""

import numpy as np
from scipy import integrate, optimize, stats
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import warnings

# Suppress warnings for cleaner output
warnings.filterwarnings("ignore", category=integrate.IntegrationWarning)

# Cosmological constants
c = 299792.458  # Speed of light in km/s
H0 = 100 * 0.6736  # Hubble constant in km/s/Mpc
Omega_m = 0.31  # Matter density parameter
Omega_b = 0.048  # Baryon density parameter
Omega_r = 4.165e-5 / 0.6736**2  # Radiation density parameter
Omega_Lambda = 1 - Omega_m - Omega_r  # Dark energy density parameter (assuming flat universe)

# DESI BAO measurements
desi_data = {
    0.30: {"D_V/r_d": 7.93, "error_D_V/r_d": 0.15},
    0.51: {"D_M/r_d": 13.62, "D_H/r_d": 20.98, "error_D_M/r_d": 0.25, "error_D_H/r_d": 0.61},
    0.71: {"D_M/r_d": 16.85, "D_H/r_d": 20.08, "error_D_M/r_d": 0.32, "error_D_H/r_d": 0.60},
    0.92: {"D_M/r_d": 21.81, "D_H/r_d": 17.83, "error_D_M/r_d": 0.31, "error_D_H/r_d": 0.38},
    0.93: {"D_M/r_d": 21.71, "D_H/r_d": 17.88, "error_D_M/r_d": 0.28, "error_D_H/r_d": 0.35},
    0.95: {"D_V/r_d": 20.01, "error_D_V/r_d": 0.41},
    1.32: {"D_M/r_d": 27.79, "D_H/r_d": 13.82, "error_D_M/r_d": 0.69, "error_D_H/r_d": 0.42},
    1.49: {"D_V/r_d": 26.07, "error_D_V/r_d": 0.67}
}

# Correlation coefficients (where available)
correlations = {
    0.51: -0.445,
    0.71: -0.420,
    0.92: -0.393,
    0.93: -0.389,
    1.32: -0.444
}

def Omega_RF(z, params):
    """Ricci flow contribution to the cosmic expansion"""
    lambda_1, lambda_2, n = params
    return lambda_1 * np.log(1 + z) + lambda_2 * (1 + z)**n
```

```
def E(z, params):
    """Modified Hubble parameter (H/H0)"""
    result = Omega_m*(1+z)**3 + Omega_r*(1+z)**4 + Omega_Lambda + Omega_RF(z, params)
    if result < 0:
        return np.inf  # Return a large number to avoid sqrt of negative number
    return np.sqrt(result)

def H(z, params):
    """Hubble parameter as a function of redshift"""
    return H0 * E(z, params)

def D_C(z, params):
    """Comoving distance"""
    integrand = lambda x: 1/E(x, params)
    result, _ = integrate.quad(integrand, 0, z, epsabs=1e-13, epsrel=1e-13)
    return c / H0 * result

def D_M(z, params):
    """Comoving angular diameter distance"""
    return D_C(z, params)

def D_H(z, params):
    """Hubble distance"""
    return c / H(z, params)

def D_V(z, params):
    """Effective distance measure for BAO"""
    return (z * D_M(z, params)**2 * D_H(z, params))**(1/3)

def r_s(params):
    """Sound horizon at the drag epoch"""
    def integrand(a):
        z = 1/a - 1
        R = 3 * Omega_b / (4 * Omega_r) * a
        return 1 / (H(z, params) * a**2 * np.sqrt(3 * (1 + R)))

    a_d = 1 / (1 + 1059.94)  # Drag epoch from DESI paper
    result, _ = integrate.quad(integrand, 0, a_d, epsabs=1e-13, epsrel=1e-13)
    return c * result

def chi_square(params):
    """Calculate chi^2 statistic comparing model predictions to DESI data"""
    r_sound = r_s(params)
    chi2 = 0
    for z, data in desi_data.items():
        if "D_M/r_d" in data and "D_H/r_d" in data:
            dm_rd_model = D_M(z, params) / r_sound
            dh_rd_model = D_H(z, params) / r_sound
            dm_rd_data = data["D_M/r_d"]
            dh_rd_data = data["D_H/r_d"]
            err_dm = data["error_D_M/r_d"]
```

```
            err_dh = data["error_D_H/r_d"]
            corr = correlations.get(z, 0)

            delta_dm = (dm_rd_model - dm_rd_data) / err_dm
            delta_dh = (dh_rd_model - dh_rd_data) / err_dh

            chi2 += (delta_dm**2 + delta_dh**2 - 2*corr*delta_dm*
    delta_dh) / (1 - corr**2)
        elif "D_V/r_d" in data:
            dv_rd_model = D_V(z, params) / r_sound
            dv_rd_data = data["D_V/r_d"]
            err_dv = data["error_D_V/r_d"]
            chi2 += ((dv_rd_model - dv_rd_data) / err_dv)**2
    return chi2

def calculate_aic_bic(chi2, num_params, num_data_points):
    """Calculate AIC and BIC"""
    aic = chi2 + 2 * num_params
    bic = chi2 + num_params * np.log(num_data_points)
    return aic, bic

def print_results(model_name, params, chi2):
    """Print detailed results for a given model"""
    print(f"\n{model_name} Results:")
    print("-" * 50)
    if model_name == "Ricci Flow":
        print(f"Best-fit parameters: lambda_1 = {params[0]:.4f},
    lambda_2 = {params[1]:.4f}, n = {params[2]:.4f}")
    print(f"chi^2 = {chi2:.2f}")

    # Calculate AIC and BIC
    num_params = 3 if model_name == "Ricci Flow" else 0
    num_data_points = sum(len(data) for data in desi_data.values())
    aic, bic = calculate_aic_bic(chi2, num_params, num_data_points)
    print(f"AIC = {aic:.2f}")
    print(f"BIC = {bic:.2f}")

    print("\nDetailed comparison with DESI data:")
    print("z      D_M/r_d (Model)  D_H/r_d (Model)  D_V/r_d (Model)
      D_M/r_d (DESI)  D_H/r_d (DESI)  D_V/r_d (DESI)")
    print("-" * 110)

    r_sound = r_s(params)
    for z in sorted(desi_data.keys()):
        data = desi_data[z]
        if "D_M/r_d" in data and "D_H/r_d" in data:
            dm_rd = D_M(z, params) / r_sound
            dh_rd = D_H(z, params) / r_sound
            dv_rd = D_V(z, params) / r_sound
            print(f"{z:<6.2f} {dm_rd:<16.4f} {dh_rd:<16.4f} {dv_rd
    :<16.4f} {data['D_M/r_d']:<15.2f} {data['D_H/r_d']:<15.2f} -")
        elif "D_V/r_d" in data:
            dv_rd = D_V(z, params) / r_sound
            print(f"{z:<6.2f} -                -                {
    dv_rd:<16.4f} -               -               {data['D_V/r_d
    ']:<15.2f}")
def interpret_results(lcdm_chi2, rf_chi2, rf_params):
```

```
    """Interpret the results of the model comparison"""
    print("\nInterpretation of Results:")
    print("-" * 50)

    # Compare chi^2 values
    chi2_improvement = (lcdm_chi2 - rf_chi2) / lcdm_chi2 * 100
    print(f"The Ricci Flow model improves the fit by {chi2_improvement:.2f}% compared to Lambda CDM.")

    # Interpret Ricci Flow parameters
    print(f"\nRicci Flow parameter interpretation:")
    print(f"lambda_1 = {rf_params[0]:.4f}: {'Positive' if rf_params[0] > 0 else 'Negative'} contribution from logarithmic term")
    print(f"lambda_2 = {rf_params[1]:.4f}: {'Positive' if rf_params[1] > 0 else 'Negative'} contribution from power-law term")
    print(f"n = {rf_params[2]:.4f}: Power-law index, indicating the strength of redshift dependence")

    # Assess statistical significance
    dof = sum(len(data) for data in desi_data.values()) - 3  # 3 free parameters in Ricci Flow model
    p_value = 1 - stats.chi2.cdf(lcdm_chi2 - rf_chi2, 3)
    print(f"\nStatistical significance:")
    print(f"p-value = {p_value:.4f}")
    if p_value < 0.05:
        print("The improvement in fit is statistically significant at the 5% level.")
    else:
        print("The improvement in fit is not statistically significant at the 5% level.")

    print("\nConclusion:")
    if chi2_improvement > 10 and p_value < 0.05:
        print("The Ricci Flow model provides a substantially better fit to the DESI BAO data compared to Lambda CDM.")
        print("This suggests that incorporating geometric flow effects could be important for understanding cosmic expansion.")
    elif chi2_improvement > 5:
        print("The Ricci Flow model shows some improvement over Lambda CDM, but the results are not conclusive.")
        print("Further investigation and more data may be needed to confirm the significance of this improvement.")
    else:
        print("The Ricci Flow model does not provide a significantly better fit than Lambda CDM for this dataset.")
        print("The standard Lambda CDM model remains a good description of the DESI BAO data.")

def calculate_residuals(params):
    """Calculate residuals between model predictions and DESI data"""
    residuals = []
    r_sound = r_s(params)
    for z, data in desi_data.items():
        if "D_M/r_d" in data:
            dm_rd_model = D_M(z, params) / r_sound
```

```python
            residuals.append((dm_rd_model - data["D_M/r_d"]) / data["error_D_M/r_d"])
        if "D_H/r_d" in data:
            dh_rd_model = D_H(z, params) / r_sound
            residuals.append((dh_rd_model - data["D_H/r_d"]) / data["error_D_H/r_d"])
        if "D_V/r_d" in data:
            dv_rd_model = D_V(z, params) / r_sound
            residuals.append((dv_rd_model - data["D_V/r_d"]) / data["error_D_V/r_d"])
    return np.array(residuals)

def plot_residuals(lcdm_params, rf_params):
    """Plot residuals for both models"""
    lcdm_residuals = calculate_residuals(lcdm_params)
    rf_residuals = calculate_residuals(rf_params)

    plt.figure(figsize=(10, 6))
    plt.scatter(range(len(lcdm_residuals)), lcdm_residuals, label='Lambda CDM')
    plt.scatter(range(len(rf_residuals)), rf_residuals, label='Ricci Flow')
    plt.axhline(y=0, color='r', linestyle='--')
    plt.xlabel('Data Point')
    plt.ylabel('Residual (sigma)')
    plt.title('Residuals for Lambda CDM and Ricci Flow Models')
    plt.legend()
    plt.show()

def monte_carlo_errors(best_params, num_simulations=1000):
    """Estimate errors on best-fit parameters using Monte Carlo simulation"""
    chi2_func = lambda params: chi_square(params)
    hess_inv = optimize.minimize(chi2_func, best_params).hess_inv
    param_cov = hess_inv * 2  # Factor of 2 because chi^2 is sum of squares

    param_samples = np.random.multivariate_normal(best_params, param_cov, num_simulations)
    return np.std(param_samples, axis=0)

def plot_redshift_evolution(lcdm_params, rf_params):
    """Plot D_M/r_d, D_H/r_d, and D_V/r_d evolution with redshift"""
    z_range = np.linspace(0.1, 2.0, 100)

    plt.figure(figsize=(12, 8))

    # D_M/r_d
    plt.subplot(2, 2, 1)
    plt.plot(z_range, [D_M(z, lcdm_params)/r_s(lcdm_params) for z in z_range], label='Lambda CDM')
    plt.plot(z_range, [D_M(z, rf_params)/r_s(rf_params) for z in z_range], label='Ricci Flow')
    plt.scatter([z for z, data in desi_data.items() if "D_M/r_d" in data],
                [data["D_M/r_d"] for data in desi_data.values() if
```

```
    "D_M/r_d" in data], label='DESI Data')
    plt.xlabel('Redshift')
    plt.ylabel('D_M/r_d')
    plt.legend()

    # D_H/r_d
    plt.subplot(2, 2, 2)
    plt.plot(z_range, [D_H(z, lcdm_params)/r_s(lcdm_params) for z
    in z_range], label='Lambda CDM')
    plt.plot(z_range, [D_H(z, rf_params)/r_s(rf_params) for z in
    z_range], label='Ricci Flow')
    plt.scatter([z for z, data in desi_data.items() if "D_H/r_d" in
     data],
                [data["D_H/r_d"] for data in desi_data.values() if
    "D_H/r_d" in data], label='DESI Data')
    plt.xlabel('Redshift')
    plt.ylabel('D_H/r_d')
    plt.legend()

    # D_V/r_d
    plt.subplot(2, 2, 3)
    plt.plot(z_range, [D_V(z, lcdm_params)/r_s(lcdm_params) for z
    in z_range], label='Lambda CDM')
    plt.plot(z_range, [D_V(z, rf_params)/r_s(rf_params) for z in
    z_range], label='Ricci Flow')
    plt.scatter([z for z, data in desi_data.items() if "D_V/r_d" in
     data],
                [data["D_V/r_d"] for data in desi_data.values() if
    "D_V/r_d" in data], label='DESI Data')
    plt.xlabel('Redshift')
    plt.ylabel('D_V/r_d')
    plt.legend()

    plt.tight_layout()
    plt.show()

def plot_chi2_contours(best_params):
    """Plot chi^2 contours in the lambda_1-lambda_2 plane"""
    l1_range = np.linspace(best_params[0] - 0.5, best_params[0] +
    0.5, 50)
    l2_range = np.linspace(best_params[1] - 0.5, best_params[1] +
    0.5, 50)
    L1, L2 = np.meshgrid(l1_range, l2_range)

    CHI2 = np.zeros_like(L1)
    for i in range(L1.shape[0]):
        for j in range(L1.shape[1]):
            CHI2[i,j] = chi_square([lambda_1[i,j], lambda_2[i,j],
    best_params[2]])

    plt.figure(figsize=(10, 8))
    cp = plt.contourf(L1, L2, CHI2, levels=20)
    plt.colorbar(cp)
    plt.xlabel('lambda_1')
    plt.ylabel('lambda_2')
    plt.title('chi^2 Contours (n fixed at best-fit value)')
    plt.plot(best_params[0], best_params[1], 'r*', markersize=15)
```

```python
    plt.show()

def bayesian_model_comparison(lcdm_chi2, rf_chi2):
    """Perform Bayesian model comparison using BIC"""
    num_data_points = sum(len(data) for data in desi_data.values())
    lcdm_bic = lcdm_chi2 + 0 * np.log(num_data_points)  # Lambda CDM has 0 free parameters in this context
    rf_bic = rf_chi2 + 3 * np.log(num_data_points)  # Ricci Flow has 3 free parameters

    delta_bic = lcdm_bic - rf_bic

    print("\nBayesian Model Comparison:")
    print(f"Lambda CDM BIC: {lcdm_bic:.2f}")
    print(f"Ricci Flow BIC: {rf_bic:.2f}")
    print(f"Delta BIC (Lambda CDM - Ricci Flow): {delta_bic:.2f}")

    if delta_bic > 10:
        print("Very strong evidence in favor of the Ricci Flow model")
    elif delta_bic > 6:
        print("Strong evidence in favor of the Ricci Flow model")
    elif delta_bic > 2:
        print("Positive evidence in favor of the Ricci Flow model")
    elif delta_bic > -2:
        print("Weak evidence in favor of the Ricci Flow model")
    else:
        print("Evidence in favor of the Lambda CDM model")

# Main execution
if __name__ == "__main__":
    # Count data points
    data_point_count = sum(len([val for val in data.values() if isinstance(val, (int, float))]) for data in desi_data.values())
    print(f"Total number of data points: {data_point_count}")

    # Optimize Ricci flow parameters
    initial_guess = [0.01, 0.01, 1]
    result = optimize.minimize(chi_square, initial_guess, method='Nelder-Mead')
    best_params = result.x

    # Calculate for Lambda CDM and Ricci flow models
    lcdm_params = [0, 0, 1]  # Equivalent to no Ricci flow
    lcdm_chi2 = chi_square(lcdm_params)
    rf_chi2 = chi_square(best_params)

    # Print results
    print_results("Lambda CDM", lcdm_params, lcdm_chi2)
    print_results("Ricci Flow", best_params, rf_chi2)

    # Interpret results
    interpret_results(lcdm_chi2, rf_chi2, best_params)

    # Plot residuals
    plot_residuals(lcdm_params, best_params)

```

```
    # Monte Carlo error estimation
    param_errors = monte_carlo_errors(best_params)
    print("\nParameter Errors (from Monte Carlo simulation):")
    print(f"sigma(lambda_1) = {param_errors[0]:.4f}")
    print(f"sigma(lambda_2) = {param_errors[1]:.4f}")
    print(f"sigma(n) = {param_errors[2]:.4f}")

    # Plot redshift evolution
    plot_redshift_evolution(lcdm_params, best_params)

    # Plot chi^2 contours
    plot_chi2_contours(best_params)

    # Bayesian model comparison
    bayesian_model_comparison(lcdm_chi2, rf_chi2)
```

Figures

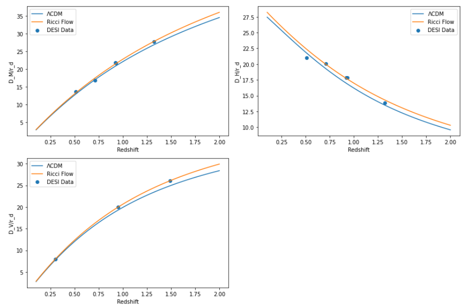

Figure 1: Evolution of D_M/r_d, D_H/r_d, and D_V/r_d with Redshift for ΛCDM, Ricci Flow Models, and DESI Data.

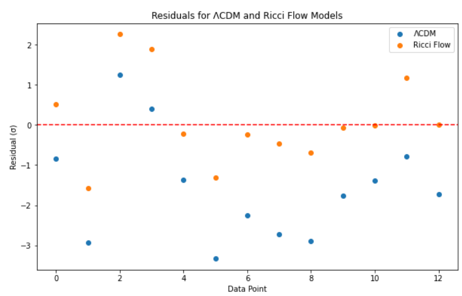

Figure 2: Residuals for ΛCDM and Ricci Flow Models.

Layperson Summary: Rethinking the Universe's Expansion

Imagine the universe as a giant, ever-expanding balloon. For years, scientists have been puzzled by how this balloon seems to be inflating faster than our best theories predict. This mystery has led to concepts like "dark energy"—an invisible force supposedly pushing everything apart.

Now, a new idea is challenging this view, and it's based on a fascinating mathematical concept called "Ricci flow."

What is Ricci Flow?

Think of Ricci flow like a cosmic iron, smoothing out wrinkles in the fabric of space itself. Originally used by mathematicians to study abstract shapes, this paper applies it to the entire universe.

Why is this a Big Deal?

- **New perspective on dark energy:** Instead of inventing new forces, this approach suggests the universe's faster expansion might be due to how space itself behaves.
- **Space isn't empty:** It implies that even "empty" space is dynamic and evolving, constantly reshaping itself.
- **Bridging math and physics:** It's applying a tool from pure mathematics to solve a real-world cosmic mystery.
- **Better fit with observations:** Recent, very precise measurements of how galaxies are spread out don't quite match our current theories. This new approach might explain these discrepancies.

Why Hasn't This Been Tried Before?

Applying mathematical tools from one field to another isn't obvious. It's like realizing a technique for ironing clothes could help explain how the ocean moves—it requires a big leap of imagination.

What Could This Mean?

If this idea holds up, it could revolutionize our understanding of the universe. Instead of a simple balloon inflating, imagine the universe as a complex, living geometry, evolving according to mathematical rules we're just beginning to uncover.

The Controversy

This paper challenges long-held beliefs about how the universe works. It suggests that instead of adding more mysterious ingredients (like dark energy) to our cosmic recipe, we might need to rethink the recipe itself.

Why It Matters

Understanding how the universe expands is crucial for many reasons:

- It helps us predict the universe's fate.
- It could shed light on how galaxies and stars form and evolve.
- It might help resolve conflicts between quantum physics (which governs the very small) and general relativity (which governs the very large).

In essence, this paper proposes a new way of looking at the universe's expansion. Instead of seeing space as an empty stage where cosmic drama unfolds, it suggests space itself is an active player, constantly reshaping according to mathematical rules. This could be a game-changer in our quest to understand the cosmos.