Analyzing Loop Quantum Cosmology of Bianchi I, II, and IX Space with Numerical Methods

by

Timothy Ernest Blackmore

BSc. Mathematics-Physics (Hon), University of New Brunswick, 2021

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

In the Graduate Academic Unit of Physics

Supervisor(s):	Edward Wilson-Ewing, PhD, Mathematics
Examining Board:	Abdelhaq Hamza, PhD, Physics, Chair
	Dennis Tokaryk, PhD, Physics
	Viqar Husain, PhD, Mathematics

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

April, 2022

Abstract

In previous literature effective Hamiltonian equations have been generated capturing lead order loop quantum gravity effects for the anisotropic vacuum Bianchi I, II, and IX spaces. Additionally, analytical transition rules have been derived for these spaces governing their rates of expansion and contraction. In this paper, these systems are evolved numerically using high-order Runge-Kutta methods and the transition rules are both tested for these Bianchi spaces and used to categorize loop quantum gravity bounces in Bianchi IX. It is found that these transition rules can predict how the Kasner exponents behave in Bianchi II if neither the potential effects and quantum effects become significant simultaneously. Additionally, the loop quantum gravity bounces in Bianchi IX were able to be sorted into four separate classifications based on these transition rules.

Dedication

I dedicate this work to my friend Steven Shantz. Your influence was felt by many.

Acknowledgements

I would like to acknowledge my supervisor Dr. Edward Wilson-Ewing for your large amount of support in completing this masters thesis. Also, I would like to acknowledge my friends William Selby, Sebastian Richard, and Devin Morin who made doing this degree all the more enjoyable.

Table of Contents

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

a	Mean Scale Factor: $a = (a_1 a_2 a_3)^{1/3}$
a_i	Directional Scale Factor
C_H	Hamiltonian
c_i	Connection Variables Conjugate to Area Variables
d_i	Variable Used in Canonical Variable Transformation: $\bar{\mu}_i d_i = \pi + \bar{\mu}_i c_i$
Δ	Smallest Non-Zero Eigenvalue of the Area Operator in Loop Quantum Gravity
G	Gravitational Constant
γ	Barbero-Immirzi Parameter
h	Step Size
k_i	Kasner Exponent: $k_i = (a/3a_i)(\dot{a}_i/\dot{a})$
$\bar{k}_i$	Predicted Kasner Exponent
ℓ_{Pl}	Planck Length
p_i	Area Variables: $p_i = a_j a_k$
t	Coordinate Frame Time
τ	Proper Time
U_I	Bianchi I Potential
U_{II}	Bianchi II Potential
U_{IX}	Bianchi IX Potential
$U(p_i)$	General Bianchi Potential

Chapter 1

Introduction

1.1 Incompatibility of General Relativity and Quantum Mechanics

General Relativity (GR) and Quantum Mechanics (QM) are incompatible. This is because GR is a deterministic theory with a dynamical evolving spacetime and QM is a probabilistic theory with a fixed spacetime. In general, this is not an issue since both theories are very effective in their own domains, with GR describing the physics of the very large and QM describing the physics of the very small. However, this becomes problematic at large matter densities where the effects from both are expected to be significant. One possible resolution to this problem is to quantize GR using Loop Quantum Gravity (LQG). Then Loop Quantum Cosmology (LQC) which contains effects from LQG can be applied onto different spacetimes and its effects studied. This has been done on the homogeneous isotropic Friedmann–Lemaître–Robertson–Walker (FLRW) spacetime, commonly used in GR to model the expanding universe. It was found that an LQG bounce replaced the Big Bang singularity [1].

1.2 Outline of Project

In this paper LQC effects are going to be applied to the anisotropic vacuum Bianchi I, Bianchi II, and Bianchi IX spacetimes. These spacetimes become increasingly complex in the order that they were written. Most of the focus will be on Bianchi II and Bianchi IX with LQC effects since their dynamics are the most interesting. However, Bianchi I, II, and IX without LQC effects and Bianchi I with LQC effects will be worth examining to better understand Bianchi II and IX with LQC effects. Many numerical solutions of Bianchi II and IX will be found and will be characterized in as much detail as possible. We will return to all of these spacetimes and study them in more depth later in the paper.

Chapter 2

Background

2.1 The Metric and the Universe

The metric is a tool used to measure the distances between two points in space/spacetime. All observers will agree with this distance irrespective of their velocity or position in the space/spacetime in which the metric is being employed. We are examining models of universes which are increasing or decreasing in scale. The metric will reflect these changes in scale and hence is useful to examine.

2.1.1 The Euclidean Metric

The Euclidean metric is given below

$$ds^2 = dx^2 + dy^2 + dz^2. \tag{2.1}$$

This means if point 1 is located at coordinates (x_1, y_1, z_1) and point 2 is located at coordinates (x_2, y_2, z_2), then the distance between both points, s, is

$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \tag{2.2}$$

The Euclidean metric is another way of stating the Pythagorean Theorem.

2.1.1.1 The Scaled Isotropic Euclidean Metric

Now, let us introduce an isotropic scale factor dependent on time into our system, $a(\tau)$. The Euclidean metric with the scale factor is

$$ds^2 = a(\tau)^2 dx^2 + a(\tau)^2 dy^2 + a(\tau)^2 dz^2 = a(\tau)^2 (dx^2 + dy^2 + dz^2). \tag{2.3}$$

Likewise, the distance between two points is

$$s = a(\tau)\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \tag{2.4}$$

Notice that if $a(\tau)$ increases as time increases then the distance between points will increase, and vice versa. Also, note that scaled Euclidean space is still infinite as long as $a(\tau)$ is not equal to zero. If the scale factor decreases one might be tempted to say the space got smaller, however this is not precise. A better expression is as the scale factor becomes smaller, the distance between points becomes smaller.

2.1.1.2 The Scaled Anisotropic Euclidean Metric

For the anisotropic scaled Euclidean metric there is a different directional scale factor corresponding to each spatial direction

$$ds^2 = a_1(\tau)^2 dx^2 + a_2(\tau)^2 dy^2 + a_3(\tau)^2 dz^2. \tag{2.5}$$

Likewise, the distance between two points is

$$s = \sqrt{a_1(\tau)^2 (x_2 - x_1)^2 + a_2(\tau)^2 (y_2 - y_1)^2 + a_3(\tau)^2 (z_2 - z_1)^2}. \tag{2.6}$$

Now, each spatial direction has a measure of scale which can change over time. These

directional scale factors can vary independently. However, there is still a need for a mean scale factor which corresponds to the isotropic scale factor. To find this we multiply the individual scale factors together and then take the cube root

$$a = (a_1 a_2 a_3)^{1/3}. \tag{2.7}$$

This is known as the mean scale factor and will be used throughout this paper.

2.1.2 The Minkowski Metric

The Minkowski metric measures four dimensional spacetimes instead of three dimensional space. The additional time component was necessary since the Euclidean distance between points in space was found to be relative to the velocity of the observer, as found in the discovery of special relativity. The Minkowski metric given below measures distances through spacetime which are observer independent

$$ds^2 = -d\tau^2 + dx^2 + dy^2 + dz^2. \tag{2.8}$$

Note that τ is referring to the proper time. The distance between two points in spacetime with coordinates (τ_1, x_1, y_1, z_1) and (τ_2, x_2, y_2, z_2) is then

$$s = \sqrt{-(\tau_2 - \tau_1)^2 + (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \tag{2.9}$$

2.1.3 The FLRW Metric and the LQG Bounce

The FLRW spacetime is used as a model to describe a simple expanding universe in general relativity. The FLRW metric has a scale factor for its spatial components and effects from special relativity as is reflected in its metric

$$ds^2 = -d\tau^2 + a(\tau)^2(dx^2 + dy^2 + dz^2). \tag{2.10}$$

In the relevant FLRW models $a(\tau)$ is increasing as time increases. If the models are evolved backwards in time eventually the scale factor $a(\tau) \to 0$. However, it has been shown that when LQC effects are applied to relevant FLRW models an LQG bounce can be recovered. This is where $a(\tau)$ reaches some minimum value instead of going to zero. This would indicate that instead of the Big Bang there would be a Big Bounce with an epoch where the universe was contracting before the bounce [1]. A pictorial example of an LQG bounce is shown in Figure 2.1. As will be discussed later in this paper, a similar Big Bounce will be recovered in the Bianchi spaces when LQC effects are applied.

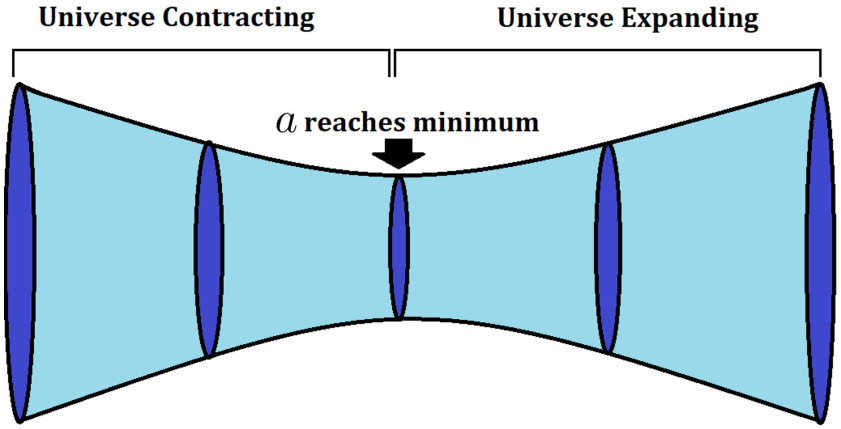

Figure 2.1: This is an LQG bounce on an FLRW spacetime where time is increasing in the right direction. First the universe is contracting, the scale factor a reaches a minimum value, and then the universe begins expanding.

2.2 Bianchi Spaces

The Bianchi spaces are a group of 4 dimensional spacetimes that are anisotropic and homogeneous and contain effects from special relativity. They are the simplest generalizations of the FLRW spacetimes. The early universe may have been anisotropic and so the Bianchi spaces serve as a useful toy model for the exploration of LQC effects on an anisotropic spacetime. It will be found when LQC effects are applied

that there will be an LQG bounce where the mean scale factor $a(\tau)$ given by (2.7) reaches some minimum value.

2.2.1 Classical Bianchi I

The Bianchi I classical metric is a well known metric in GR

$$ds^2 = -d\tau^2 + a_1^2 dx^2 + a_2^2 dy^2 + a_3^2 dz^2 \tag{2.11}$$

where

$$a_1 = (\tau - \tau_0)^{k_1}, \quad a_2 = (\tau - \tau_0)^{k_2}, \text{ and } a_3 = (\tau - \tau_0)^{k_3}. \tag{2.12}$$

Here τ_0 is a constant of integration fixed by initial conditions. The k_i values are constants called the Kasner exponents and govern the rate of change of their corresponding a_i values. Using the ADM formulation of General Relativity equations of motion can be found describing this system in vacuum [2]. These equations of motion are equivalent to Einstein's equations when the stress energy tensor is equal to zero.

$$\begin{aligned} \frac{dp_1}{dt} &= \frac{p_1}{\gamma}(p_2 c_2 + p_3 c_3) \\ \frac{dc_1}{dt} &= -\frac{c_1}{\gamma}(p_2 c_2 + p_3 c_3). \end{aligned} \tag{2.13}$$

Here t is some coordinate time and γ is the Barbero-Immirzi parameter. The p_i terms are area variables and are related to the directional scale factors of the system a_i, by

$$p_i = a_j a_k \tag{2.14}$$

Furthermore, the c_i terms are the connection variables. The remaining equations of motion for p_2, p_3, c_2, and c_3 can be found by using cyclic permutation. Since $p_i = a_j a_k$ we can find an expression for $(\dot{a}_i/a_i)$

$$\ln a_i = \frac{1}{2}(\ln p_j + \ln p_k - \ln p_i) \rightarrow \frac{\dot{a}_i}{a_i} = \frac{1}{2}\left(\frac{\dot{p}_j}{p_j} + \frac{\dot{p}_k}{p_k} - \frac{\dot{p}_i}{p_i}\right). \tag{2.15}$$

Using (2.13) this simplifies to

$$\frac{\dot{a}_i}{a_i} = \frac{p_i c_i}{\gamma}. \tag{2.16}$$

Each $p_i c_i$ term is constant by using the product rule and equation (2.13)

$$\frac{d}{dt}\left(\frac{p_i c_i}{\gamma}\right) = \frac{1}{\gamma}(\dot{p}_i c_i + p_i \dot{c}_i) = \frac{1}{\gamma^2}(p_i c_i (p_j c_j + p_k c_k) - p_i c_i (p_j c_j + p_k c_k)) = 0. \tag{2.17}$$

This means the solution of the differential equation (2.15) is

$$a_i = A_i \exp\left(\frac{p_i c_i}{\gamma} t\right) \tag{2.18}$$

where A_i is some constant based on the initial conditions. For this system the lapse to be equal to the volume [11]

$$N = a_1 a_2 a_3 \tag{2.19}$$

which relates the coordinate time t and proper time τ by $N dt = d\tau$. Hence,

$$A_1 A_2 A_3 \exp\left(t \Sigma_i p_i c_i\right) = \Sigma_i p_i c_i (\tau - \tau_0), \quad \tau = \frac{1}{\Sigma_i p_i c_i} \ln\left(\frac{\Sigma_i p_i c_i}{A_1 A_2 A_3}(\tau - \tau_0)\right) \tag{2.20}$$

Then by use of (2.18)

$$a_i = B_i(\tau - \tau_0)^{k_i}, \quad \text{where} \quad B_i = A_i\left(\frac{\Sigma_i p_i c_i}{A_1 A_2 A_3}\right)^{k_i} \quad \text{and also} \quad k_i = \frac{p_i c_i}{\gamma \Sigma_i p_i c_i}. \tag{2.21}$$

Here k_i is the Kasner exponent as per the metric given in (2.11). While the current form of k_i given in (2.21) is acceptable it is only appropriate in the classical case. We want to write the Kasner exponents in terms of our scale factors in a form such that they have a usable definition when LQC effects are applied.

First using (2.18) and the mean scale factor (2.7) we find

$$\frac{a}{\dot{a}} = \frac{3}{\gamma \Sigma_i p_i c_i}. \tag{2.22}$$

This can be substituted along with (2.16) into the Kasner expression given in (2.21) to find

$$k_i = \frac{a}{3a_i}\left(\frac{\dot{a}_i}{\dot{a}}\right). \tag{2.23}$$

With this definition we can calculate the Kasner exponents for both classical and non-classical Bianchi I, II, and IX spaces. Lastly the constraints are recovered [11]

$$\begin{aligned} &\text{(i)} \quad \sum_i k_i = 1, \\ &\text{(ii)} \quad \sum_i k_i^2 = 1. \end{aligned} \tag{2.24}$$

2.3 The General Bianchi Hamiltonian

We have examined the equations of motion for classical Bianchi I in depth. Now we wish have the ability to examine the equations of motion for Bianchi I, II, and IX both with classical and with LQC effects. To do this we will need to generate the

Hamiltonian for all of these systems. This has been done in previous literature but will be summarized here since this will be done repeatedly throughout this paper. The Hamiltonian for the Bianchi spacetimes can be written both classically and non-classically in the following form [2, 11].

$$C_H = (\text{“Kinetic Terms”}) + (\text{“Potential Terms”}) \tag{2.25}$$

While not exactly the same the “Kinetic” terms correspond to kinetic terms in a standard Hamiltonian and the “Potential” terms correspond the the potential terms in a standard Hamiltonian. From now on we will just refer to “Kinetic” as kinetic and “Potential” as potential.

2.3.1 Classical Hamiltonian vs. Effective Hamiltonian

The classical Hamiltonian formulation of Bianchi spaces is an exact solution from the ADM formulation of GR. Here the Kinetic Terms from expression (2.25) are always represented as

$$\text{Kinetic Terms} = -\frac{1}{8\pi G\gamma^2}(p_1p_2c_1c_2 + p_1p_3c_1c_3 + p_2p_3c_2c_3) \tag{2.26}$$

with p_i and c_i being the previously seen working variables. For this Hamiltonian formulation the p_i variables will serve as our momentum variables and the c_i variables will serve as our position coordinates. As given in previous literature, when LQC effects are applied its leading order effects can be represented with the following modification to the Kinetic Terms [2, 10]

$$\text{Kinetic Terms} = -\frac{p_1p_2p_3}{8\pi G\gamma^2\Delta}[\sin\bar{\mu}_1c_1\sin\bar{\mu}_2c_2 + \sin\bar{\mu}_2c_2\sin\bar{\mu}_3c_3 + \sin\bar{\mu}_3c_3\sin\bar{\mu}_1c_1] \tag{2.27}$$

where

$$\bar{\mu}_i = \sqrt{\frac{p_i \Delta}{p_j p_k}}. \tag{2.28}$$

Note that Δ is the smallest non-zero eigenvalue of the area operator in LQG. Also, by taking the Sine small angle approximation $|\bar{\mu}_i c_i| \ll 1$ expression (2.27) turns into expression (2.26). So, all of the LQC effects come from the Sine expressions. Additionally, we can force the Sine small angle approximation in expression (2.27) by taking $\Delta \to 0$. So, another way of thinking of Δ is as the variable which determines how strongly LQC effects are felt. This paper is about Bianchi spaces with LQC effects, however the classical version of the kinetic terms will be seen in multiple derivations and explanations later in the paper when in regions where LQC effects are negligible.

2.3.2 Bianchi I, II, and IX Potential

The Bianchi spacetime potentials make up the potential terms in expression (2.25)

$$\text{Potential Terms} = U_i. \tag{2.29}$$

The potentials for Bianchi I, II, and IX respectively are [11]

$$\begin{aligned} &U_I = 0, \quad U_{II}(p_i) = \frac{p_2^2 p_3^2}{p_1^2 32\pi G}, \\ &U_{IX}(p_i) = \frac{1}{32\pi G}\left(\frac{p_2^2 p_3^2}{p_1^2} + \frac{p_1^2 p_3^2}{p_2^2} + \frac{p_1^2 p_2^2}{p_3^2}\right) - \frac{1}{16\pi G}(p_1^2 + p_2^2 + p_3^2). \end{aligned} \tag{2.30}$$

So by substituting expressions from (2.26), (2.27), and (2.30) into expression (2.25) we can generate the classical or non-classical effective Hamiltonian for Bianchi I, II, and IX.

2.3.3 Constants and Equations of Motion

2.3.3.1 Constants

In the above expressions (2.26), (2.27), and (2.30), G is the gravitational constant and as previously stated γ is the Barbero-Immirzi Parameter. Also, the Δ term which is the smallest non-zero eigenvalue of the area operator is related to γ by $\Delta = 4\sqrt{3}\pi\gamma\ell_{Pl}^2$ where ℓ_{Pl} is the Plank length. For this thesis we scale the units for G, γ, and ℓ_{Pl} such that $G = 1$, $\gamma = 1$, $\ell_{Pl} = 1$, and hence $\Delta = 4\sqrt{3}\pi$.

2.3.3.2 Generating Equations of Motion

We can substitute any of our Hamiltonians into the Hamiltonian equations shown below to generate six coupled differential equations which we can evaluate to evolve the system over time.

$$\begin{aligned} \frac{dp_i}{dt} &= -8\pi G\gamma \frac{\partial C_H}{\partial c_i}, \\ \frac{dc_i}{dt} &= 8\pi G\gamma \frac{\partial C_H}{\partial p_i}. \end{aligned} \tag{2.31}$$

We can check this by finding the equations of motion for classical Bianchi I. To do this we substitute the classical Bianchi I Hamiltonian into (2.31). We should recover the equations of motion given in expression (2.13). Also, note there is a Hamiltonian constraint $C_H = 0$ which is constant due to energy conservation. The fact that C_H is equal to zero specifically is is due to the additional input of diffeomorphism invariance, which is an important symmetry of general relativity. This Hamiltonian constraint will be useful when performing error calculations later.

2.4 Numerical Methods and Considerations

2.4.1 Runge-Kutta and Languages Used

We used high order Runge-Kutta methods in order to numerically solve the Hamiltonian equations (2.31) for Bianchi I, II and IX spaces with classical and LQC effects. Unless otherwise specified, the sixth order Runge-Kutta method called Butcher-1 was used with the following tableau as seen in Table 2.1 [8].

0							
1/3	1/3						
2/3	0	2/3					
1/3	1/12	1/3	-1/12				
1/2	-1/16	9/8	-3/16	-3/8			
1/2	0	9/8	-3/8	-3/4	1/2		
1	9/44	-9/11	63/44	18/11	0	-16/11	
	11/120	0	27/40	27/40	-4/15	-4/15	11/120

Table 2.1: Butcher-1, Sixth Order Runge-Kutta Method

The language C++ was used to apply the Runge-Kutta method and export the results to a CSV file and MATLAB was used to import that CSV file and display the results. We chose C++ since it is a compiled efficient open source language. Also, for some select calculations in non-classical Bianchi IX Julia was used instead of C++, since we were doing variable-precision calculations.

2.4.2 Performing Calculations

When applying the Runge-Kutta method you only recover the p_i and c_i values with a corresponding time, t_i. However, we were often displaying other values including but not limited to the directional scale factors and the Kasner exponents. Instead of passing in the p_i, c_i and t_i values into MATLAB and then calculating these other values in MATLAB, all displayed values were calculated in our C++/Julia program

and then exported into the CSV file. All MATLAB did was display the final results which was done to avoid rounding error and keep all the calculations in one language. This was particularly important when performing the variable-precision calculations in Julia.

2.4.3 Managing CSV File Size

Since we were doing very high precision calculations the CSV file size quickly became unmanageable. Each step has several calculated values to be displayed. Numerical solutions could generate CSV files of 50+ gigabytes which was unsustainable since many calculations were performed. Also, calculations were inefficient since more time was spent writing the CSV files than actually performing the Runge-Kutta calculations. Because of this, only the values of 1 out of every 1000 steps were outputted into the CSV file to be displayed for the high precision calculations. Also, while it was necessary to calculate the variables p_i, c_i, and t_i for each one of those 1000 steps we only had to calculate some displayed values including Kasner exponents for the one step that was exported. To avoid running out of memory in C++/Julia, the variables generated from the current step overwrite the variables generated during the previous step. The only record of the calculation is from the CSV file which is updated line by line during the calculation. An additional benefit of this method is if the calculation failed all the information in the CSV file up to the point of failure is recovered.

Another consideration is that to calculate the Kasner exponents using expression (2.23) the derivative of a and a_i is required. To calculate these values in our C++/Julia code we used the simplest scheme for a numerical derivative where j refers to the specific step.

$$\dot{a}_j \approx (a_j - a_{j-1})/h, \quad \text{and} \quad \dot{a}_{ij} \approx (a_{ij} - a_{ij-1})/h \tag{2.32}$$

where h is the step size. So the a and a_i values from the previous step are preserved for the entirety of the current step before being erased.

2.4.4 Examining Classical Bianchi I Numerically

While the Bianchi I classical Hamiltonian can be solved for analytically as mentioned previously, it is worth doing a numerical run for Bianchi I since we are going to be doing numerical runs for the remainder of this paper. This will also be an opportunity to understand how Kasner exponents and scale factors behave without LQC effects and without potential effects. To generate Figure 2.2 we solved the differential equations given by (2.13) for c_1 and p_1 and the corresponding differential equations for c_2, c_3, p_2, and p_3 numerically. Notes, we are using natural units so our coordinate time, t, will be in Planck units.

As we can see the Kasner exponents, k_i, are constant and the logarithm of the scale factors, $\ln a_i$, are increasing or decreasing at a constant rate. We can show this is what we expect analytically. From expression (2.17) we know the $p_i c_i$ terms will be constant which we can substitute into the classical expression for Kasner exponents given by (2.21). Hence the Kasner exponents themselves will be constant. Also, by equation (2.18) the $\ln a_i$ terms will increase or decrease linearly and by using (2.7) we see that the logarithm of the mean scale factor, $\ln a$, also will increase or decrease linearly. In Figure (2.2) the Kasner exponents are $k_i = -0.23892$, $k_2 = 0.91565$, and $k_3 = 0.32327$ which satisfy relationships (2.24).

While we are moving away from Bianchi I classical to more complicated spaces, understanding how Kasner exponents and scale factors act in Bianchi I classical spacetime is very important. This is because in regions where LQC and potential effects are negligible, Kasner exponents and scale factors will behave as they do here.

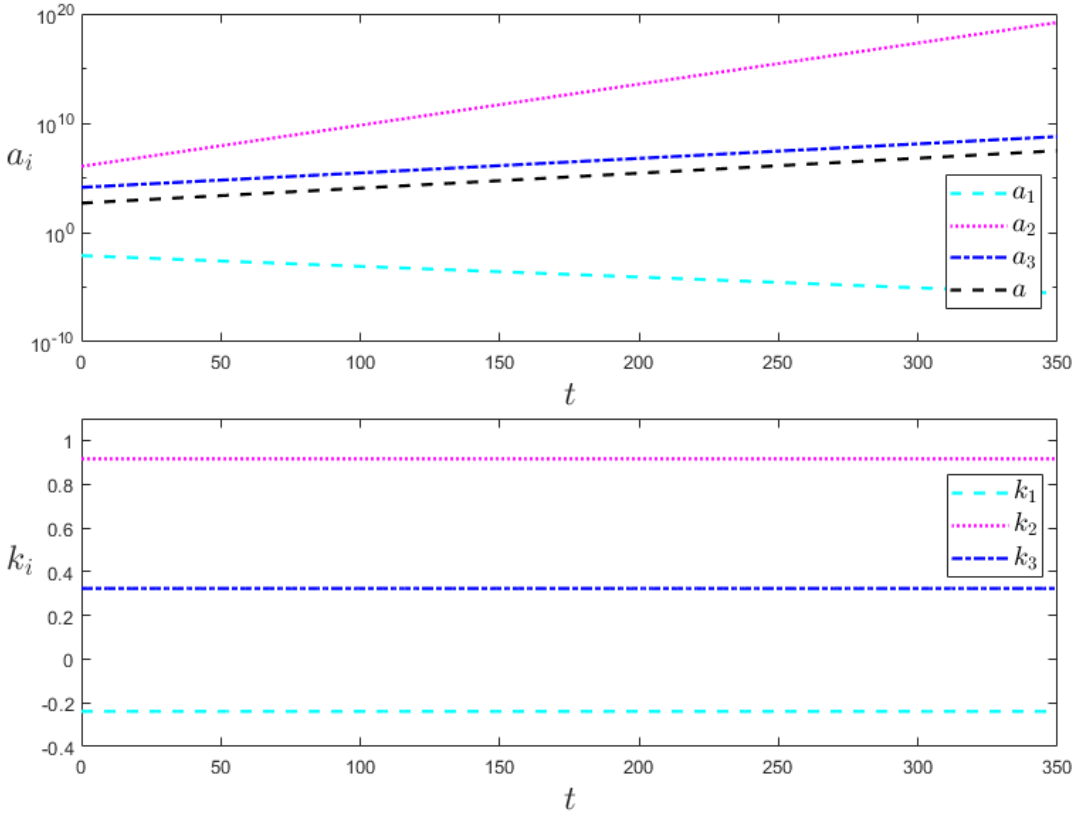

Figure 2.2: The top subplot shows the direction scale factors plotted against time which are increasing or decreasing exponentially. The bottom subplot shows the Kasner exponents plotted against time which are constant. Note that the Kasner exponents are $k_i = -0.23892$, $k_2 = 0.91565$, and $k_3 = 0.32327$ which satisfy the relationships (2.24). Also, t is in Planck units.

2.5 Previous Literature

In previous literature the cosmologies of Bianchi I, II, and IX with LQC effects have already been researched numerically [7, 4, 5]. These papers had a large focus on different parameters including the shear and Hubble rates. They also considered spaces with stiff matter, radiation, dust in addition to vacuum, with the focus being on the different types of LQG bounces.

In this paper we will mostly be examining scale factors, a_i, and Kasner exponents, k_i, always in vacuum. As seen above, in classic Bianchi I the Kasner exponents remain constant. However, when LQC effects are applied to the Bianchi spaces the

Kasner exponents will undergo a shift during an LQG bounce governed by the LQG bounce rule (3.2), as will be discussed in more detail. This LQG transition rule has been mentioned in previous literature in numerical studies but was not studied in much detail [4]. Likewise, the potentials given by (2.30) for Bianchi II or IX become significant, a Kasner transition occurs where the Kasner exponents again shift to new values. The shift in these Kasner exponents are governed by explicit transition rules derived in previous literature for Bianchi II (4.4) and Bianchi IX (5.3) as will be discussed in more depth. However, while changes in Kasner exponents are briefly discussed in the Bianchi II and IX papers, there is no relation to these transition rules and little focus is put on the these changes [4, 5]. A Kasner transition is discussed in previous literature, but is referring to what we refer to as the LQG bounce rule [7]. Also, in this specific paper they did not use the LQG bounce rule (3.2) for their "Kasner transition", but instead did analysis on which type of LQG bounces were possible by examining Kasner exponents and categorizing LQG bounces [7].

This project focuses on characterising Bianchi I, II, and IX by using the explicit LQG bounce rule in conjunction with the Kasner transition rules to better understand these spaces. Additionally, there is a greater focus on numerical techniques and a brief exploration of potential chaos in LQC Bianchi IX.

Chapter 3

Bianchi I

3.1 The Effective Hamiltonian

By substituting (2.27) and U_I from (2.30) into expression (2.25) we can generate the effective Hamiltonian for Bianchi I with LQC effects

$$C_H = -\frac{p_1 p_2 p_3}{8\pi G\gamma^2\Delta}[\sin\bar{\mu}_1 c_1 \sin\bar{\mu}_2 c_2 + \sin\bar{\mu}_2 c_2 \sin\bar{\mu}_3 c_3 + \sin\bar{\mu}_3 c_3 \sin\bar{\mu}_1 c_1]. \tag{3.1}$$

Again, we will substitute this effective Hamiltonian into (2.31) and numerically solve the resulting coupled differential equations to evolve the system. As we can see since $U_I = 0$ the only contribution to the evolution of the system comes from the LQC kinetic terms, and no contribution comes from the potential terms.

3.2 The LQG Bounce

The LQC effects become most prominent when the mean scale factor, a, which characterizes the size of our system reaches a minimum. Here gravity becomes momentarily repulsive and our system undergoes an LQG bounce. While Bianchi spaces

cannot describe in full the early universe there is a direct physical interpretation of this phenomenon which has been recreated in other spacetimes. This is The Big Bounce which is a potential model for early universe. Note that before and after the LQG bounce the Kasner exponents are constant in the regions resembling classical epochs just as we would expect. It is only when LQC effects become prominent that the Kasner exponents begin to noticeably shift. In previous literature Kasner transition rules have been been found to predict the change in Kasner exponents during a bounce [2].

$$\bar{k}_i = \frac{2}{3} - k_i. \tag{3.2}$$

Here the k_i values refer to the initial Kasner exponents before the LQG bounce and $\bar{k}_i$ refer to the final Kasner exponents after the LQG bounce. In Figure 3.1 seen below we run a calculation and test this Kasner transition rule on all of the Kasner exponents. As you can see in Figure 3.1, the transition rule (3.2) is able to predict the shift in Kasner exponents almost exactly.

3.3 Approximating Error

The Hamiltonian constraint $C_H = 0$ can be used to approximate the error of a run. If we define the expression

$$C_{Hi} = -\frac{p_1 p_2 p_3}{8\pi G\gamma^2\Delta}[\sin(\bar{\mu}_j c_j)\sin(\bar{\mu}_k c_k)] \tag{3.3}$$

then we can calculate the relative error of for non-classical runs with the expression

$$\text{Relative Error} = \frac{|C_H|}{\sqrt{C_{H1}^2 + C_{H2}^2 + C_{H3}^2 + U_i^2}}. \tag{3.4}$$

For all the runs except the variable precision runs done in Julia, double floats are used

for all constants and variables. Double floats give between 15-16 digits of precision in decimal so the minimum relative error we expect to see is between 1.0×10^{-15} and 1.0×10^{-16}. The relative error is displayed in the bottom subplot in Figure 3.1.

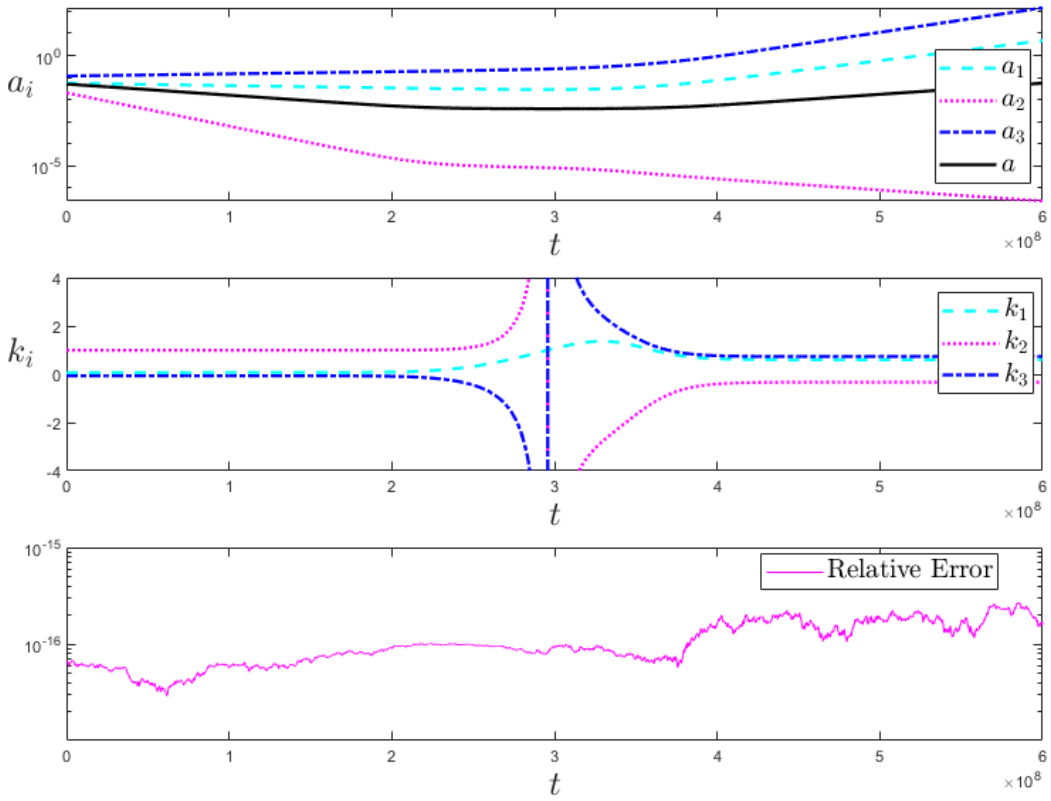

Figure 3.1: Bianchi I with LQC Effects: The top subplot shows the direction scale factors plotted against time. The middle subplot shows the Kasner exponents undergoing their transitions. The bottom subplot shows the error term which should remain close to zero. The relative error term is defined as $C_H/\sqrt{C_{H1}(t_i)^2 + C_{H2}(t_i)^2 + C_{H3}(t_i)^2}$ where C_{Hi} is specified by (3.3). Numerically the initial k_i values were $k_1 = 0.069733$, $k_2 = 0.995437$, and $k_3 = -0.065168$. The predicted final $\bar{k}_i$ values using equation 3.2 were $\bar{k}_1 = 0.596933$, $\bar{k}_2 = -0.328770$, $\bar{k}_3 = 0.731835$ while they were numerically found to be $\bar{k}_1 = 0.596933$, $\bar{k}_2 = -0.328769$, $\bar{k}_3 = 0.731834$.

3.4 Canonical Variable Change

There is a hidden variable change during the calculation shown in Figure 3.1. Before the LQG bounce, the terms $\bar{\mu}_i c_i$ in the expression (3.1) were approximately equal to zero and after the LQG bounce $\bar{\mu}_i c_i \to -\pi$. However, floating point numbers which are being used in the calculation can be much more precise around zero than around $-\pi$. As the $\bar{\mu}_i c_i$ converges to $-\pi$ their precision is limited and the whole calculation eventually fails. So, we implemented the canonical variable transformation from c_i to d_i just after the bounce

$$\bar{\mu}_i d_i = \pi + \bar{\mu}_i c_i. \tag{3.5}$$

Here d_i becomes the new working variable. Furthermore the Hamiltonian (3.1) is invariant when substituting $\bar{\mu}_i d_i$ into the place of $\bar{\mu}_i c_i$. So, to perform the canonical transformation we only need to calculate each d_i component, substitute it into the place of c_i, and then continue the calculation. We wish to do this canonical transformation when the $\bar{\mu}_i c_i$ values are generally halfway between zero and $-\pi$. We decided perform the canonical transformation at the LQG bounce when a is at a minimum. It was necessary to do this numerical procedure in Bianchi II and IX space as well.

3.5 Choosing Initial Conditions

While not explicitly shown we needed to choose initial p_i and c_i values for the calculation shown in Figure 3.1. Here we will go over a general procedure for choosing initial conditions for any Bianchi I LQC calculation. Our initial conditions must satisfy the $C_H = 0$ which we can do by setting $C_H = 0$ in expression (3.1) and rearranging for one of the c_i terms. We do this in the expression below and arbitrarily choose to rearrange for c_1

$$c_1 = \bar{\mu}_1^{-1} \arcsin\left(\left(-\sin\bar{\mu}_2 c_2 \sin\bar{\mu}_3 c_3\right)\left(\sin\bar{\mu}_2 c_2 + \sin\bar{\mu}_3 c_3\right)^{-1}\right). \tag{3.6}$$

If we already have chosen values for c_2, c_3, p_1, p_2, and p_3 this expression can be used to calculate the required value for c_1 to generate an acceptable solution. Additionally, we would like to start the calculation in a region away from LQC effects where $|\bar{\mu}_i c_i| \ll 1$. One way to enforce this is to choose c_2 and c_3 to be very small and the p_i terms to be of the same order of magnitude and such that $p_i \gg 1$.

Chapter 4

Bianchi II

4.1 The Effective Hamiltonian

By substituting (2.27) and U_{II} from (2.30) into expression (2.25) we can generate the effective Hamiltonian for Bianchi II with LQC effects

$$C_H = -\frac{p_1 p_2 p_3}{8\pi G\gamma^2\Delta}[\sin\bar{\mu}_1 c_1 \sin\bar{\mu}_2 c_2 + \sin\bar{\mu}_2 c_2 \sin\bar{\mu}_3 c_3 + \sin\bar{\mu}_3 c_3 \sin\bar{\mu}_1 c_1] + \frac{p_2^2 p_3^2}{p_1^2 32\pi G}. \tag{4.1}$$

As we can see there is now a contribution from the potential terms. By substituting (2.14) into expression $U_{II}(p_i)$ from (2.30) we recover an alternate form of U_{II} in terms of scale factors.

$$U_{II}(a_1) = \frac{a_1^4}{32\pi G} \tag{4.2}$$

As we can see U_{II} is only dependent on a_1 and will reach a maximum when a_1 reaches a maximum.

4.1.1 Bianchi II Classical Kasner Transitions

Since the added U_{II} term is present in both the classic Bianchi II Hamiltonian and the non-classic Bianchi II effective Hamiltonian, it is worth briefly examining the classic Bianchi II Hamiltonian. This is because since Bianchi II classic is is a simpler spacetime and we can isolate the effects of the potential U_{II} term. Then when we move to Bianchi II with LQC effects, we will understand how the U_{II} term affects the solution in regions away from LQC effects. Below is the classic Bianchi II Hamiltonian generated from substituting (2.26) and $U_{II} = a_1^4/32\pi G$ into expression (2.25)

$$C_H = -\frac{1}{8\pi G\gamma^2}(p_1p_2c_1c_2 + p_1p_3c_1c_3 + p_2p_3c_2c_3) + \frac{a_1^4}{32\pi G}. \tag{4.3}$$

Let us assume we choose initial conditions such that the scale factor a_1 is initially increasing. We can visualize the Bianchi II solution as a ball bouncing off of a wall where the wall is the Bianchi II potential $U_{II}(a_1)$ as seen in Figure 4.1.

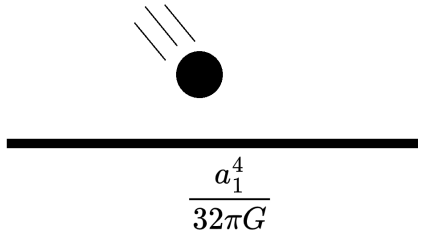

Figure 4.1: The Kasner transition can be modelled as our solution/ball bouncing off of a potential energy wall. In the first stage our solution will look like a classic Bianchi I solution with a_1 increasing. This is analogous to the ball travelling towards the potential wall. In the second stage U_{II} dominates in the Hamiltonian shown in (4.3) and a_1 reaches a maximum. This is analogous to the ball striking the potential energy wall. In the last stage our solution will look like another classic Bianchi I solution with a_1 decreasing. This is analogous to the ball travelling away from the wall.

First the Bianchi II potential is small but increasing, $U_{II}(a_1) \ll 1$, we recover the Bianchi I Hamiltonian whose solution has constant k_i values and exponentially in-

creasing and decreasing a_i values. Then the Bianchi II potential $U_{II}(a_1)$ potential grows very large and the dynamics change. Here the a_1 value reaches a maximum which is analogous to the ball hitting the wall. Then again the Bianchi II potential becomes very small, $U_{II}(a_1) \ll 1$, and decreases as a_1 also decreases. Again, we recover another Bianchi I solution but now the a_i values are exponentially increasing and decreasing at different rates and the k_i values are modified. This whole process is called a Kasner transition and the location of the Kasner transition is taken to be when a_1 reaches a maximum. Please see Figure 4.2 for an example of a numerical solution. From previous literature we can predict how the Kasner exponents will change during this transition [3, 9].

$$\bar{k}_1 = \frac{-k_1}{1+2k_1}, \quad \bar{k}_2 = \frac{k_2+2k_1}{1+2k_1}, \quad \bar{k}_3 = \frac{k_3+2k_1}{1+2k_1} \tag{4.4}$$

As we can see in Figure 4.2 the transition rules (4.4) predict the final Kasner exponents very accurately.

4.2 Choosing Initial Conditions

Now we are returning to Bianchi II space with LQC effects where we expect solutions with both an LQG bounce and Kasner transition(s). Assuming that the LQG bounce and Kasner transition(s) are well separated we can predict changes in the Kasner exponents using the LQG transition rule (3.2) and the Kasner transition rules (4.4) which we are going to test numerically.

Since the dynamics of Bianchi II LQC are richer than Bianchi I LQC, we did many more calculations in Bianchi II LQC. Hence, we are going to discuss the process and methodology of setting up calculations before examining our numerical results. When choosing initial conditions for Bianchi II space with LQC effects we must fulfill the condition $C_H = 0$ (4.1). Similarly to Bianchi I with LQC effects we can do this

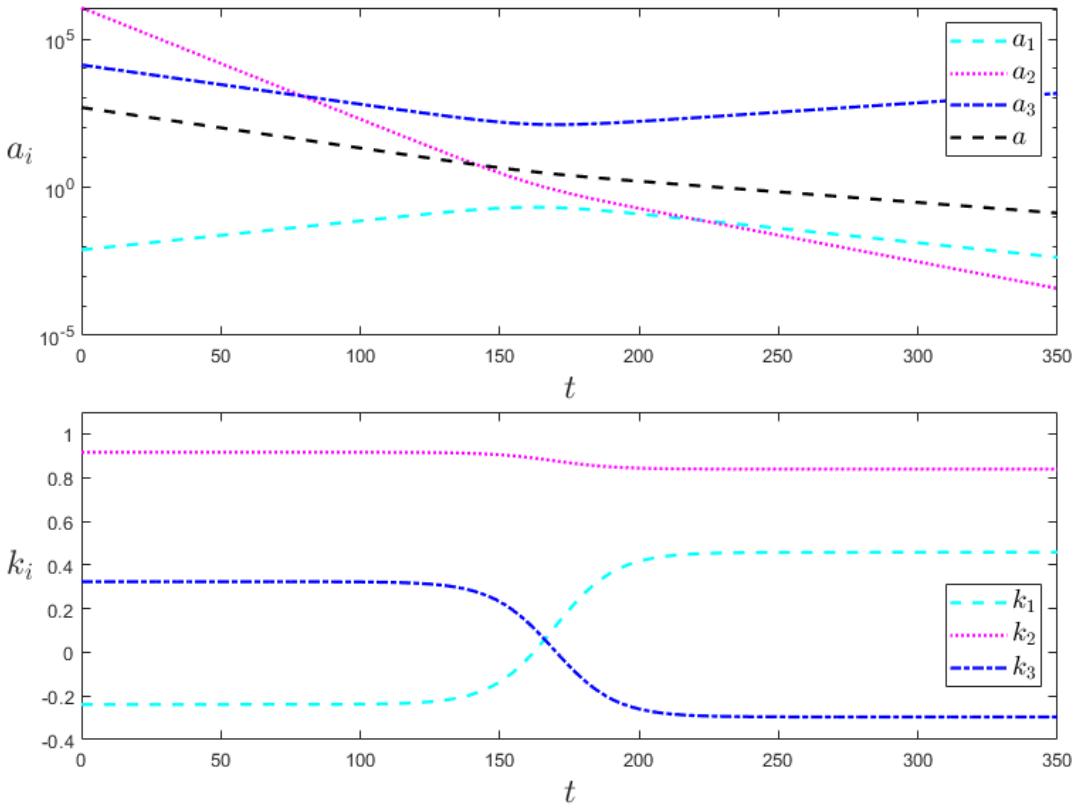

Figure 4.2: This is output from a classical Bianchi II space examining a Kasner transition. The top subplot shows scale factors undergoing the Kasner transition and the bottom subplot shows the same transition in terms of Kasner exponents. Notice that the Kasner transition happens when a_1 reaches its maximum value. Numerically the initial k_i values were $k_1 = -0.238919$, $k_2 = 0.915652$, and $k_3 = 0.323271$. The predicted final k_i values were $\bar{k}_1 = 0.457557$, $\bar{k}_2 = 0.838463$, $\bar{k}_3 = -0.296015$ using (4.4) while they were numerically found to be $\bar{k}_1 = 0.457560$, $\bar{k}_2 = 0.838459$, $\bar{k}_3 = -0.296018$.

by fixing all the p_i and c_i and then solving for one of the c_i values such that $C_H = 0$. Without loss of generality, when we do this for c_1 we recover the expression

$$c_1 = \bar{\mu}_1^{-1} \arcsin\left(\left(\frac{p_2 p_3 \gamma^2 \Delta}{4p_1^3} - \sin\bar{\mu}_2 c_2 \sin\bar{\mu}_3 c_3\right)\left(\sin\bar{\mu}_2 c_2 + \sin\bar{\mu}_3 c_3\right)^{-1}\right). \quad (4.5)$$

Furthermore, just as in Bianchi I we wish to find solutions which begin away from LQC effects. Additionally, we also want a solution that starts away from Kasner transitions. By choosing each term $|\bar{\mu}_i c_i| \ll 1$ we force the solution to be in a classical epoch. By using (3.3) we can rewrite expression (4.3) as

$$C_H = C_{H1} + C_{H2} + C_{H3} + U_{II}(p_i). \quad (4.6)$$

In order to choose initial conditions such that we are away from Kasner transitions, we want to choose p_i and c_i values such that $|C_{Hi}| \gg U_{II}(p_i)$. By examining the form of the potential U_{II} as seen in equation (2.30), we could easily do this by enforcing $p_1 \gg p_2 p_3$. However, it will be helpful to us later to have chosen p_i values and the c_i values to be the same order of magnitude as each other. To make this more precise we can choose p_i and c_i values such that

$$p_1 \sim p_2 \sim p_3 \sim p \quad (4.7)$$

and

$$|c_1| \sim |c_2| \sim |c_3| \sim c. \quad (4.8)$$

Then $|C_{Hi}| \gg U_{II}(p_i)$ implies that

$$\frac{p^3}{8\pi G \gamma^2 \Delta} \sin^2\left(c\sqrt{\frac{\Delta}{p}}\right) \gg \frac{p^2}{32\pi G}. \quad (4.9)$$

Since we set $|\bar{\mu}_i c_i| \ll 1$ we must choose $c\sqrt{\frac{\Delta}{p}} \ll 1$ and hence $p \gg c^2\Delta$. As mentioned previously, by choice of units we set the Planck length $\ell_{Pl} = 1$, $\gamma = 1$ and hence $\Delta = 4\sqrt{3}\pi$. Now we can apply the small angle approximation to equation (4.9) to recover

$$p^2c^2 \gg \frac{p^2}{4} \tag{4.10}$$

which implies

$$1 \ll c \ll p. \tag{4.11}$$

This means one way to recover acceptable initial conditions is to choose initial c_i values much greater than 1 and initial p_i values much greater than our initial c_i values.

4.3 Choosing Initial k_i Values

Now we are able to generate valid initial conditions. However, we also want to be able to choose our initial k_i values. From previous literature we know one requirement of k_1 is it must be in the range $-1/3 < k_1 < 0$ for the system to be before an LQG bounce and all Kasner transitions [11]. Using expressions (2.24) we can derive the expressions for initial k_2 and k_3

$$k_2 = \frac{1 - k_1}{2} + \frac{\sqrt{1 + 2k_1 - 3k_1^2}}{2} \tag{4.12}$$

and

$$k_3 = \frac{1 - k_1}{2} - \frac{\sqrt{1 + 2k_1 - 3k_1^2}}{2}. \tag{4.13}$$

The expression (2.24) is only true classically so the initial k_i values are only a good

approximation before the LQC effects become significant. By choosing a value of k_1 we are fixing the values of k_2 and k_3. Hence, we only need a method to set k_1 to have full control over all initial Kasner exponents. Since we are working in the classical regime before the LQG bounce we can use the classical form for Kasner exponents given by expression (2.21) to create an expression for k_1. By rearranging for c_1 we find

$$c_1 = \frac{k_1}{p_1(1-k_1)}(p_2c_2 + p_3c_3). \tag{4.14}$$

Thus, to choose an initial k_1 value, we input it into equation (4.14) along with the other previously chosen p_i and c_i values to determine the new c_1 value. However, in general this will make it so $C_H \neq 0$. Then we can use a modified version of equation (4.5) to set c_3 such that $C_H = 0$. Again, this will make it such that the initial k_1 is no longer our desired value. We can repeatedly implement equations (4.14) and (4.5) in sequence such that c_1 and c_3 settle to their required values so that both expressions are satisfied. This can be done manually or with a while/for loop.

4.4 Modifying Location of LQG bounce

We have developed a method for changing initial Kasner values. Now we wish to develop a method to choose where the LQG bounce occurs. Already, the initial p_i and c_i values were chosen such that

$$p_1 \sim p_2 \sim p_3 \sim p \tag{4.15}$$

and

$$|c_1| \sim |c_2| \sim |c_3| \sim c. \tag{4.16}$$

This allows us to say

$$\mu_i c_i \sim c\left(\sqrt{\frac{\Delta}{p}}\right) \tag{4.17}$$

The LQC effects become more prominent as the magnitude of $\mu_i c_i$ becomes larger, in general the LQC effects will be less prominent when the p_i values are increased and the c_i values are decreased. By choosing larger p_i values and smaller c_i values we simply postpone the LQG bounce. However, we wish to keep the same initial Kasner values and and the condition $C_H = 0$ true. In order to do this we need to keep the potential U_{II} in the C_H expression (4.1) constant and each $p_i c_i$ term constant. To keep the potential U_{II} constant we must satisfy the condition

$$\frac{p_{2n}p_{3n}}{p_{1n}} = \frac{(p_2 + d_2)(p_3 + d_2)}{(p_1 + d_1)} = \frac{p_2 p_3}{p_1} \tag{4.18}$$

where the new p values are taken as $p_{1n} = p_1 + d_1$, $p_{2n} = p_2 + d_2$, and $p_{3n} = p_3 + d_2$. Since we want to choose the new p_i values to be larger to postpone the LQG bounce the term d_1 is chosen to be a positive number and then d_2 is solved for. Since all the p_{in} are now fixed we can then find the new c_{in} values such that $p_{in}c_{in} = p_i c_i$.

4.5 Testing Transition Rules when LQG Bounce and Kasner Transition(s) are Well Separated

There are two cases which describe all possible well separated Bianchi II solutions with LQC effects. These cases were found using analytical methods in previous literature [11]. In both cases it was found that k_1 must initially be in the range $-1/3 < k_1 < 0$. The first case is when an LQG bounce occurs before any Kasner transitions. It was found when this occurs there must be one Kasner transition after the LQG bounce. The second case is when there is one Kasner transition before the

LQG bounce. In the sub-case where initially $-1/3 < k_1 < -2/7$, there will first be a Kasner transition and then the LQG bounce with no further Kasner transitions. In the sub-case where initially $-2/7 < k_1 < 0$, there will be first be a Kasner transition, then an LQG bounce, and then one final Kasner transition.

For each case and sub-case, we can take our initial Kasner exponents and use transition rules (4.4) and (3.2) to predict what the final Kasner exponents should be compare these to numerical results.

4.5.1 Case 1: Testing Transition Rules when LQG Bounce occurs before Kasner Transition

Let us first examine the first case in which an LQG bounce occurs before any Kasner transition. A numerical calculation is shown in Figure 4.3 where initially $k_1 = -0.320427$.

This is just an example of one run. In fact, we performed 15 separate runs with different initial k_1 values ranging from $-1/3$ to 0. The final numerical Kasner exponent values versus the initial k_1 values are plotted in Figure 4.4.

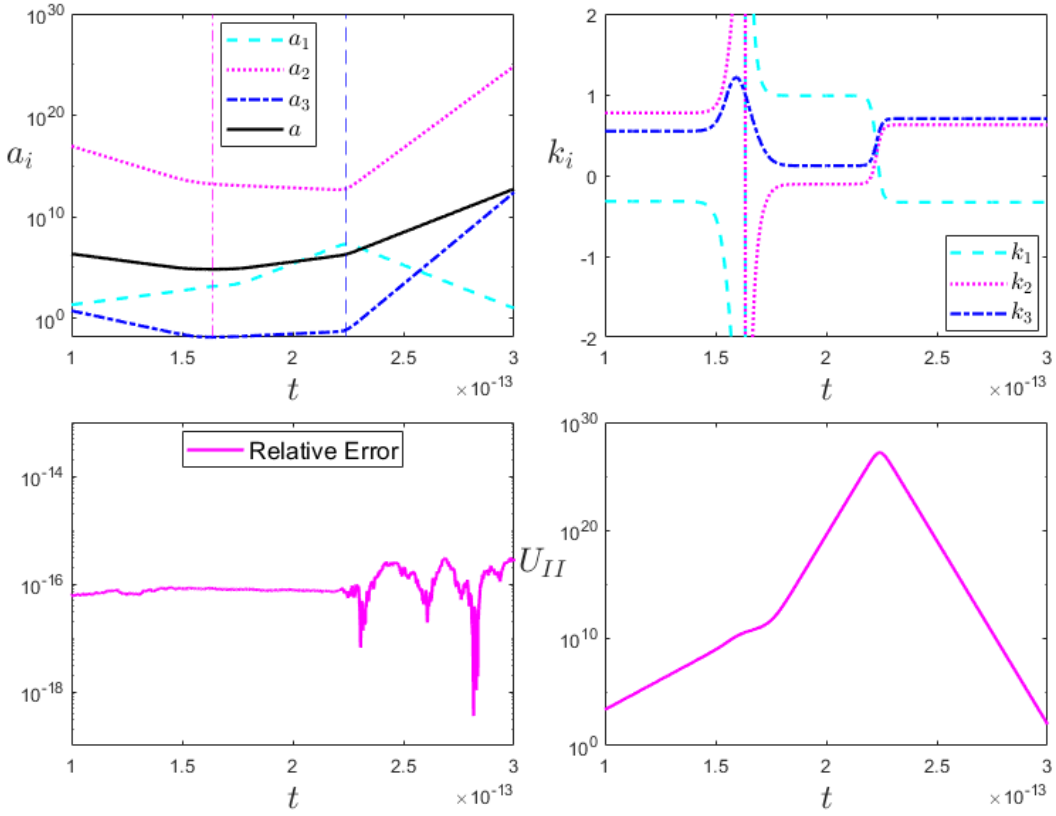

Figure 4.3: In the top left subplot the vertical dashed/dotted magenta line indicates the location of the LQG bounce and the vertical blue dashed line indicates the location of the Kasner transition. The relative error term is defined as $C_H/\sqrt{C_{H1}(t_i)^2 + C_{H2}(t_i)^2 + C_{H3}(t_i)^2 + U_{II}(t_i)^2}$ where C_{Hi} is specified by (3.3). Here initially $k_1 = -0.320427$, $k_2 = 0.773262$, $k_3 = 0.547172$. The predicted final Kasner exponents are $\bar{k}_1 = -0.331887$, $\bar{k}_2 = 0.627933$, $\bar{k}_3 = 0.703951$. The final numerical Kasner exponents are $\bar{k}_1 = -0.331894$, $\bar{k}_2 = 0.627930$, $\bar{k}_3 = 0.703944$. As we can see there is close agreement between theory and the numerics.

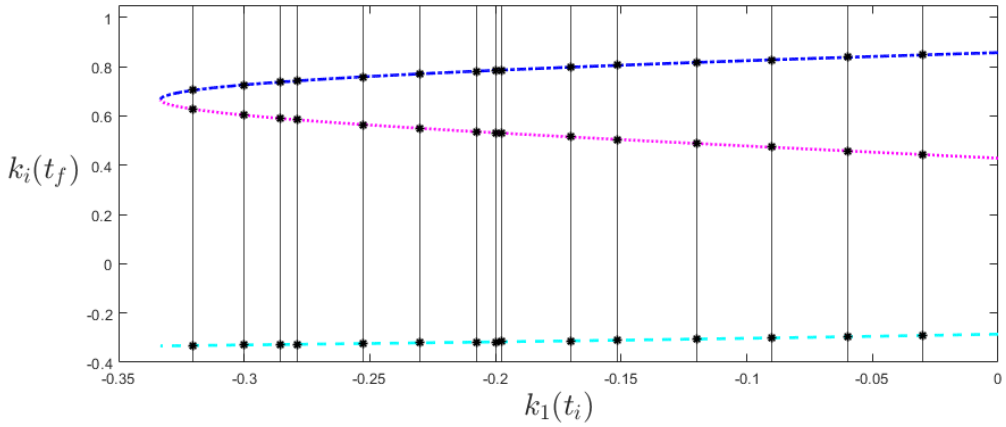

Figure 4.4: Here we are plotting the final k_i values versus the corresponding initial k_1 values for the case where the LQG bounce comes before the Kasner transition. The horizontal curved lines in the background are the theoretical final k_i values. The bottom dashed cyan line is $k_1(t_f)$, the middle dotted magenta line is $k_2(t_f)$, and the top blue dashed/dotted line is $k_3(t_f)$. The black circles are numerically calculated final k_i values. Each vertical black line represents a numerical run and connects the three corresponding black circles for that run. In total 15 runs were completed and the numerics agreed closely with the theory.

4.5.2 Case 2: Testing Transition Rules when LQG Bounce occurs after Kasner Transition(s)

Here we are examining the second case in which there is a Kasner transition before the LQG transition. First we will examine the sub-case where initially $-1/3 < k_1 < -2/7$. We performed such a calculation which can be seen in Figure 4.5 where initially $k_1 = -0.320427$. As we can see, there is no further Kasner transitions after the LQG bounce as expected. Note that this calculation shown in Figure 4.5 has the same initial Kasner exponents as the calculation shown in Figure 4.3. We generated the plot shown in Figure 4.5 by postponing the LQC effects so they happened after the Kasner transition.

Now let us examine the sub-case where initially $-2/7 < k_1 < 0$. A numerical calculation is shown in Figure 4.6 where we chose initially $k_1 = -0.120001$. Here we can see that there is a second Kasner transition after the LQG bounce just as

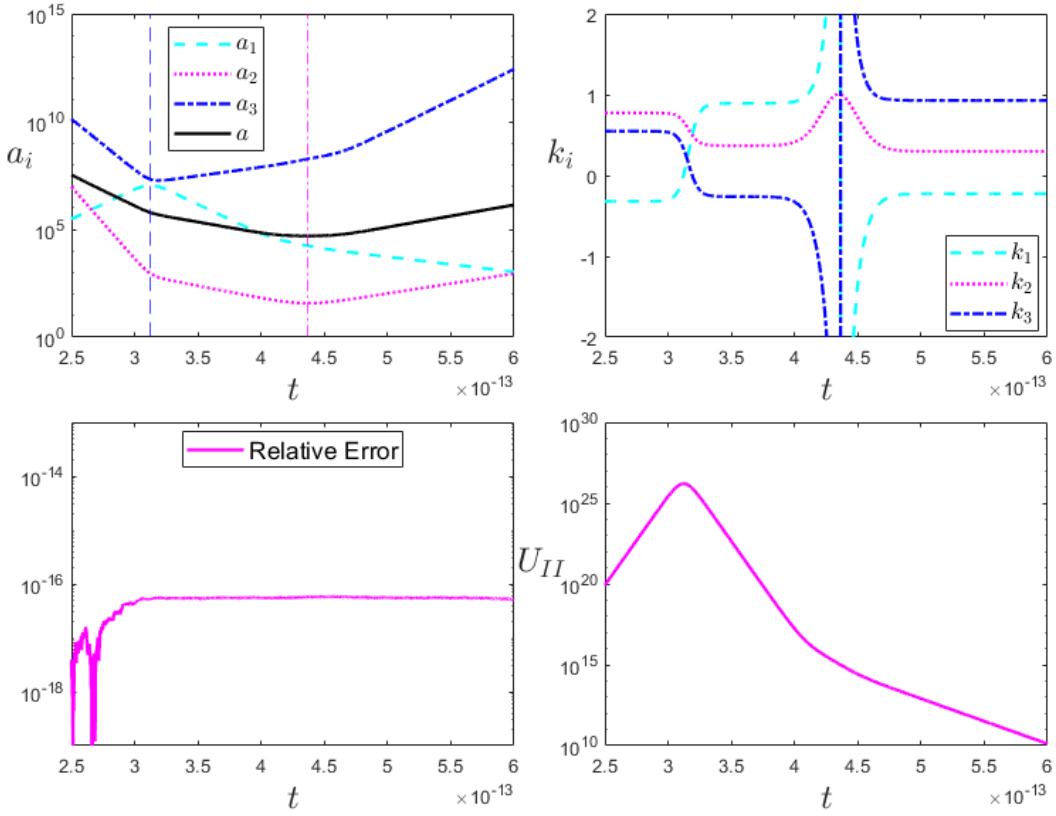

Figure 4.5: In the top left subplot the magenta vertical dashed line indicates the location of the LQG bounce and the blue vertical dashed line indicates the location of the Kasner transition. The C_H error term is defined as $C_H/\sqrt{C_{H1}(t_i)^2 + C_{H2}(t_i)^2 + C_{H3}(t_i)^2 + U_{II}(t_i)^2}$ where C_{Hi} is specified by (3.3). Here initially $k_1 = -0.320427$, $k_2 = 0.773262$, $k_3 = 0.547172$. The predicted final Kasner exponents are $\bar{k}_1 = -0.225527$, $\bar{k}_2 = 0.297992$, $\bar{k}_3 = 0.927517$. The final numerical Kasner exponents are $\bar{k}_1 = -0.225544$, $\bar{k}_2 = 0.298010$, $\bar{k}_3 = 0.927533$. As we can see there is close agreement between theory and the numerics.

expected. We have shown two cases in which there is a Kasner transition before the LQG bounce. In total we performed 15 separate runs with different initial k_1 values ranging from $-1/3$ to 0. The final theoretical Kasner exponents are plotted against the initial k_1 values as can be seen in Figure 4.7.

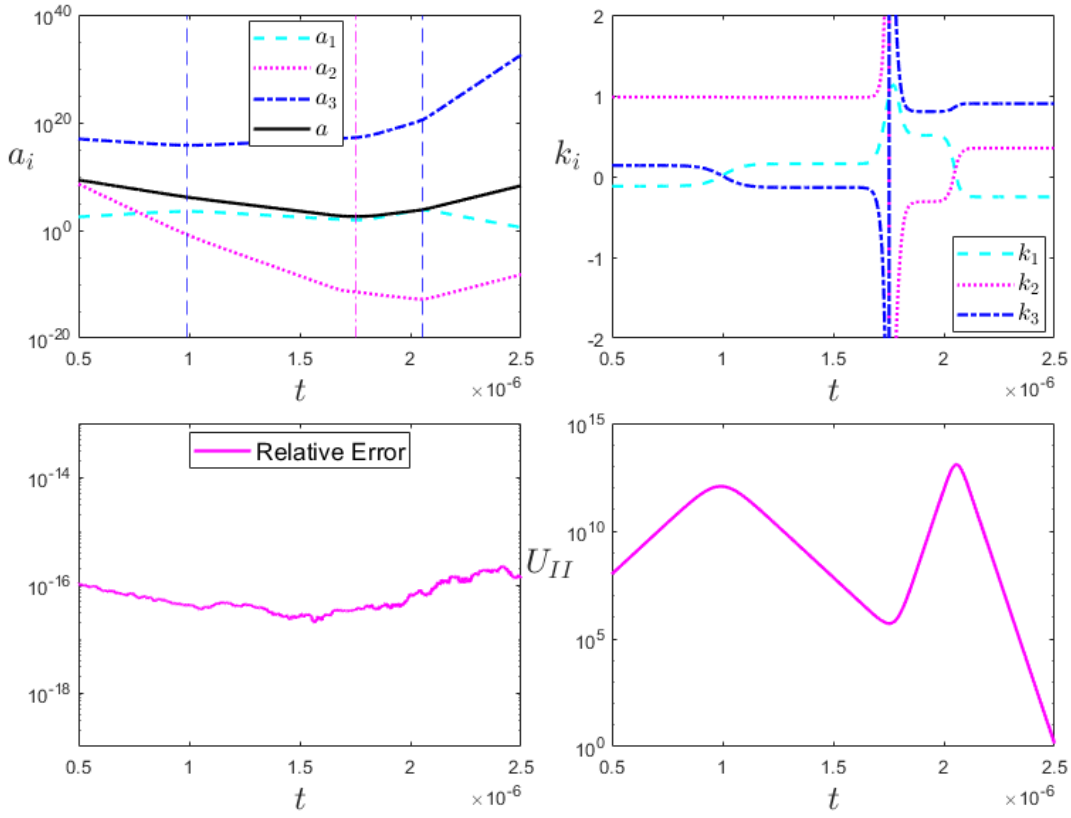

Figure 4.6: In the top left subplot the magenta vertical dashed line indicates the location of the LQG bounce and the blue vertical dashed lines indicates the location of the Kasner transitions. The relative error term is defined as $C_H/\sqrt{C_{H1}(t_i)^2 + C_{H2}(t_i)^2 + C_{H3}(t_i)^2 + U_{II}(t_i)^2}$ where C_{Hi} is specified by (3.3). Here initially $k_1 = -0.119999$, $k_2 = 0.983335$, $k_3 = 0.136679$. The predicted final Kasner exponents are $\bar{k}_1 = -0.252174$, $\bar{k}_2 = 0.350000$, $\bar{k}_3 = 0.902164$. The final numerical Kasner exponents are $\bar{k}_1 = -0.252179$, $\bar{k}_2 = 0.350008$, $\bar{k}_3 = 0.902148$. As we can see there is close agreement between theory and the numerics.

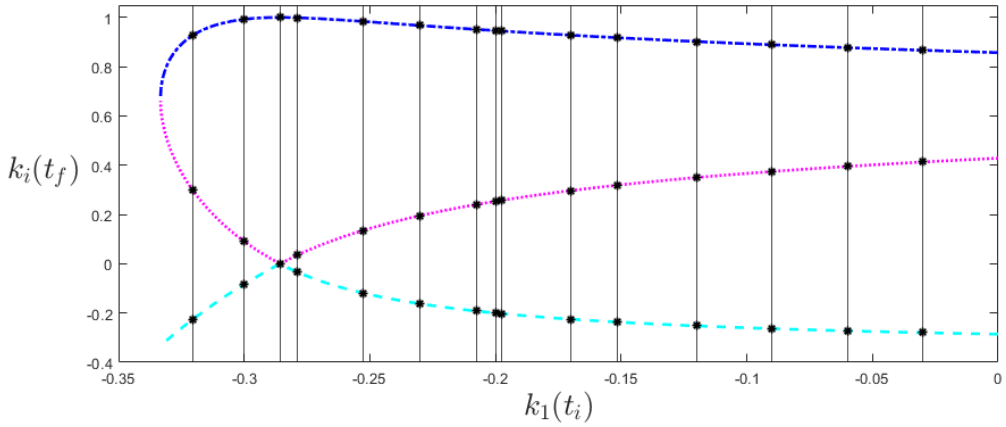

Figure 4.7: Here we are plotting the final k_i values versus the corresponding initial k_1 values for the case where the LQG bounce comes after the Kasner transition. The horizontal curved lines in the background are the theoretical final k_i values. The bottom dashed cyan line is $k_1(t_f)$, the middle dotted magenta line is $k_2(t_f)$, and the top blue dashed/dotted line is $k_3(t_f)$. The black circles are numerically calculated final k_i values. Each vertical black line represents a numerical run and connects the three corresponding black circles for that run. In total 15 runs were completed and the numerics agreed closely with the theory. Note that the initial k_1 values in this plot are the same as in Figure 4.4.

4.6 Testing Transition Rules when LQG Bounce and Kasner Transition(s) are Not Well Separated

So far we have been assuming the LQG bounce and Kasner transitions have been well separated. However, sometimes both the Bianchi II potential and LQC effects become dominant close to each other or at the same time. In this case we expect the LQG bounce to overlap with the Kasner transition(s). An numerical example is shown in Figure 4.8 where the LQG bounce is overlapping both Kasner transitions. Here we can no longer use the transition rules to predict final k_i values since the LQG bounce and Kasner transitions are no longer well separated. We wish to investigate how the final theoretical k_i values in Figure 4.4 morph into the final theoretical k_i

values in Figure 4.7 as the LQG bounce is moved through the Kasner transition.

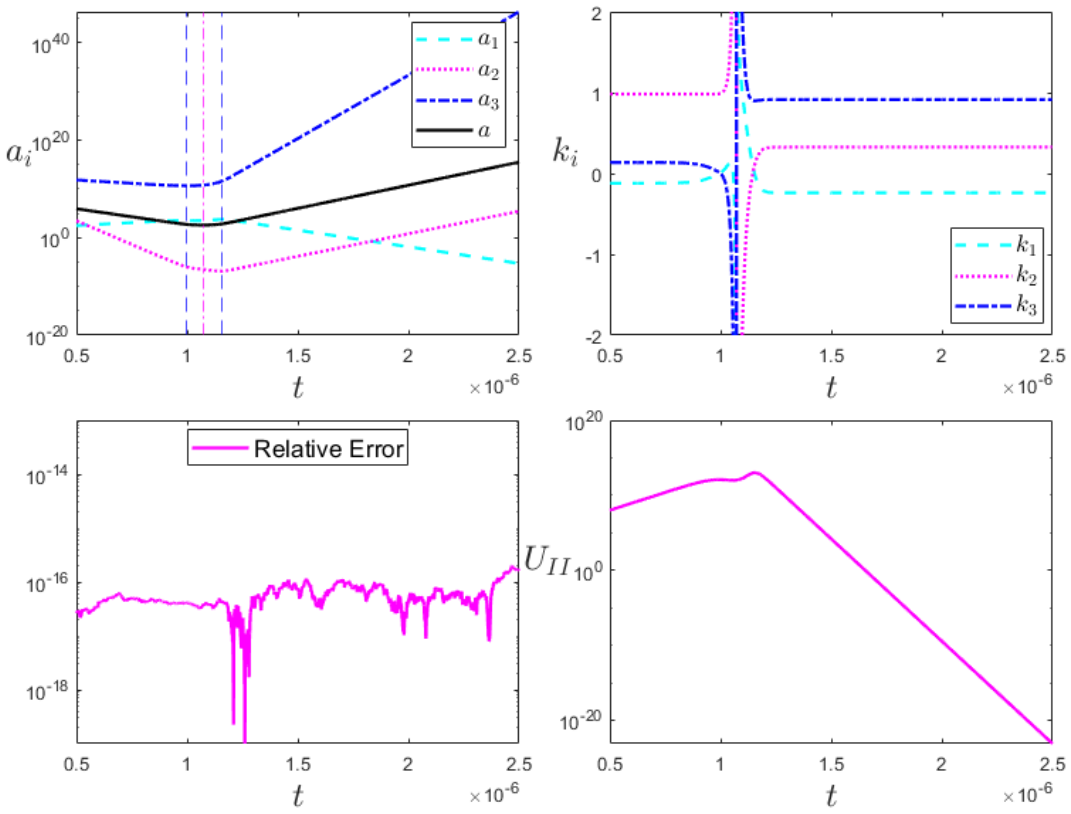

Figure 4.8: In the top left subplot the vertical magenta dashed line indicates the location of the LQG bounce and the vertical blue dashed/dotted lines indicate the locations of the Kasner transitions. The relative error term is defined as $C_H/\sqrt{C_{H1}(t_i)^2 + C_{H2}(t_i)^2 + C_{H3}(t_i)^2 + U_{II}(t_i)^2}$ where C_{Hi} is specified by (3.3). This shows a run where the LQG bounce and Kasner transitions are overlapping. As we can see, the Kasner transition to the right is dominant as it is taller.

To accomplish this we must first define a term we will call Δt. We define Δt to be the time between the LQG bounce whose position is denoted by the minimum of the mean scale factor and the dominant Kasner transition denoted by the maximum of a_1 in the dominant Kasner transition. If there is only one Kasner transition it is taken to be the dominant Kasner transition. If there are two Kasner transitions, the dominant Kasner transition will have a taller spike in potential. When initially $-2/7 < k_1 < 0$, as the LQG bounce is moved through the Kasner transition the

potential of the dominant Kasner transition will appear to birth a smaller non-dominant spike in potential when the second Kasner transition is formed. In Figure 4.8, in the subplot "U_{II} vs. t", we can see that the Kasner transition to the right is the dominant Kasner transition.

We performed a series of runs where the LQG bounce was initially before the Kasner transition for four different initial k_1 values. Then we nudged the LQG bounce to the right using what was developed in section 4.4. For each nudge we recorded the final k_i values and the Δt value. We then created a figure of subplots of final k_i values versus Δt which can be seen in Figure 4.9.

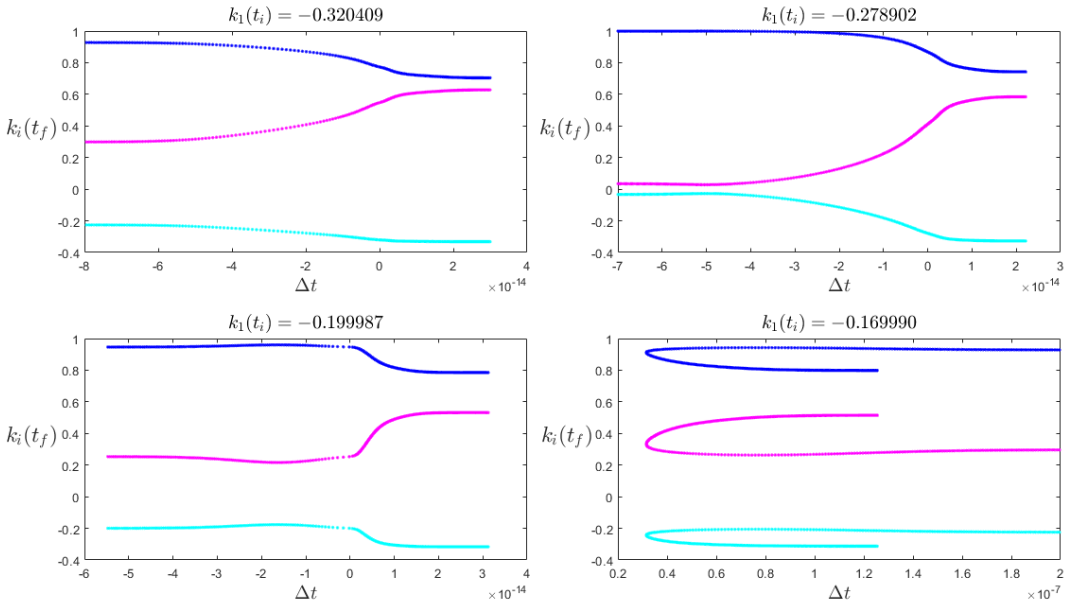

Figure 4.9: Here are four plots of Final k_i values versus Δt. The initial k_i values corresponding to each plot are given in each subplot's title. For each subplot the bottom horizontal cyan line is $k_1(t_f)$, the middle magenta line is $k_2(t_f)$, and the top blue line is $k_3(t_f)$. Each dot in the lines are an individual $k_i(t_f)$ value.

The most notable subplot in Figure 4.9 is "$k_1(t_i) = -0.169990$". Here Δt remains positive. This is due to the Kasner bounce to the right instead of the left being dominant.

It appears that when k_1 is approximately in the range $-1/3 < k_1 < -1/5$, Δt will

be negative for part of the plot. Likewise, when Δt is approximately in the range $-1/5 < k_1 < 0$, Δt remains positive. We can find this critical point $k_1 = -1/5$ analytically by setting both Bianchi II potential spikes to be equal in height to each other, so neither are dominant. To accomplish this we fix the slope of $\ln a_1$ to be equal an opposite before and after the LQG bounce. In the classical regime the k_1 value will control how quickly the scale factor a_1 increases or decreases exponentially. This can be seen by substituting the classical expression for k_i (2.21) into the classical expression for a_i (2.18). Also, from the updated expression for k_i given in (2.23), since a and a_1 will both be decreasing before the LQG bounce and increasing after the LQG bounce we expect k_1 to have the same sign before and after the bounce. So to fix the slope of $\ln a_1$ to be equal an opposite before and after the LQG bounce we use the expression

$$\tilde{k}_1 = \bar{k}_1 \tag{4.19}$$

where $\tilde{k}_1$ is the value of k_1 before the LQG bounce and $\bar{k}_1$ is the value of k_1 after the LQG bounce. Then using the LQG bounce rule (3.2) we recover

$$\tilde{k}_1 = \frac{1}{3}. \tag{4.20}$$

Finally since there is a Kasner transition before the LQG bounce we can use the Kasner transition rule for k_1 (4.4) to calculate the initial k_1 value before the first Kasner transition

$$\begin{aligned} \frac{-k_1}{1+2k_1} &= \frac{1}{3} \\ \rightarrow k_1 &= -\frac{1}{5}. \end{aligned} \tag{4.21}$$

4.6.1 A Discussion of Time Scale

The numerical values of Δt in Figure 4.9 are not significant when specifying if an LQG bounce and a Kasner transition are well separated. However, the shapes of the $k_i(t_f)$ vs. Δt plots seem to be significant. To show this we chose a completely new set of p_i and c_i values with varying Δt values while attempting to set $k_i(t_i) = -0.320409$ to match the top left subplot in Figure 4.9. The results of this exercise can be seen in Figure 4.10.

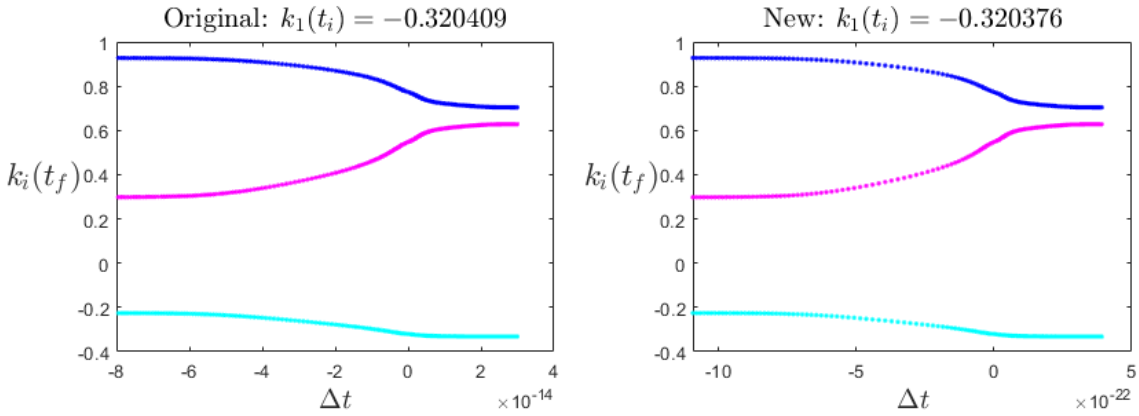

Figure 4.10: In the subplot on the right, completely new p_i, and c_i values were chosen randomly while attempting to hold $k_1(t_i)$ constant. The new $k_1(t_i) = -0.320376$ is very close to the original value on the left.

Notice that while the time shapes of both plots look similar, the time scale of Δt is very different. One possibility of what determines the time scale of the system is the phase space of the entire system which is drastically different for the original and the new p_i, c_i values. By rescaling Δt in the right subplot of Figure 4.10 and overlaying the graphs it is clear how similar both solutions as seen in Figure 4.11.

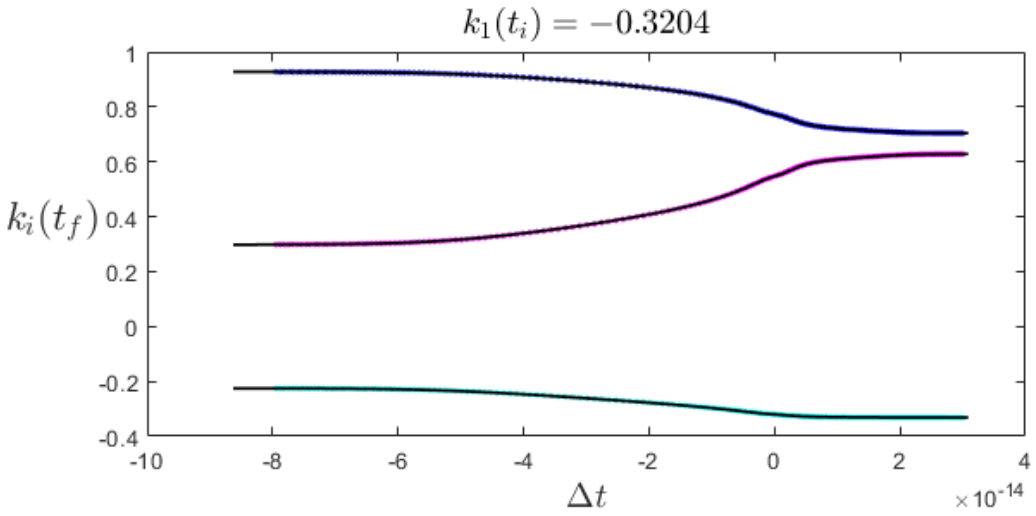

Figure 4.11: Here Δt in the right subplot of Figure 4.10 was rescaled, and overlaid on the left subplot of Figure 4.10. The new rescaled calculation was replaced by a black line.

4.7 Relative Kasner Transition Location

Finally, when there are two Kasner transitions let us define Δt_1 to be the difference in time between the LQG bounce and the first Kasner transition. Likewise, let us define Δt_2 to be the difference in time between the LQG bounce and the second Kasner transition. Now we can plot graphs for any group of runs with k_1 initially in the region $-2/7 < k_1 < 0$. This was done as can be seen in Figure 4.12 for several runs with differing initial k_1 values. As seen when the Kasner transitions are well separated from the LQG bounce, the relationship between Δt_2 and Δt_1 is linear which is expected from two observations. Firstly, the maximum Bianchi II potential value corresponding to each Kasner transition appears to remains approximately constant as the LQG bounce is moved to the right. It is the minimum potential value which becomes smaller as the LQG bounce is delayed. Secondly, by transition rules the k_i values will remain the same before and after the LQG bounce as the LQG bounce is nudged to the right. Hence, the slope of $\ln a_1$ will remain the same before and after the LQG bounce as the LQG bounce is nudged to the right.

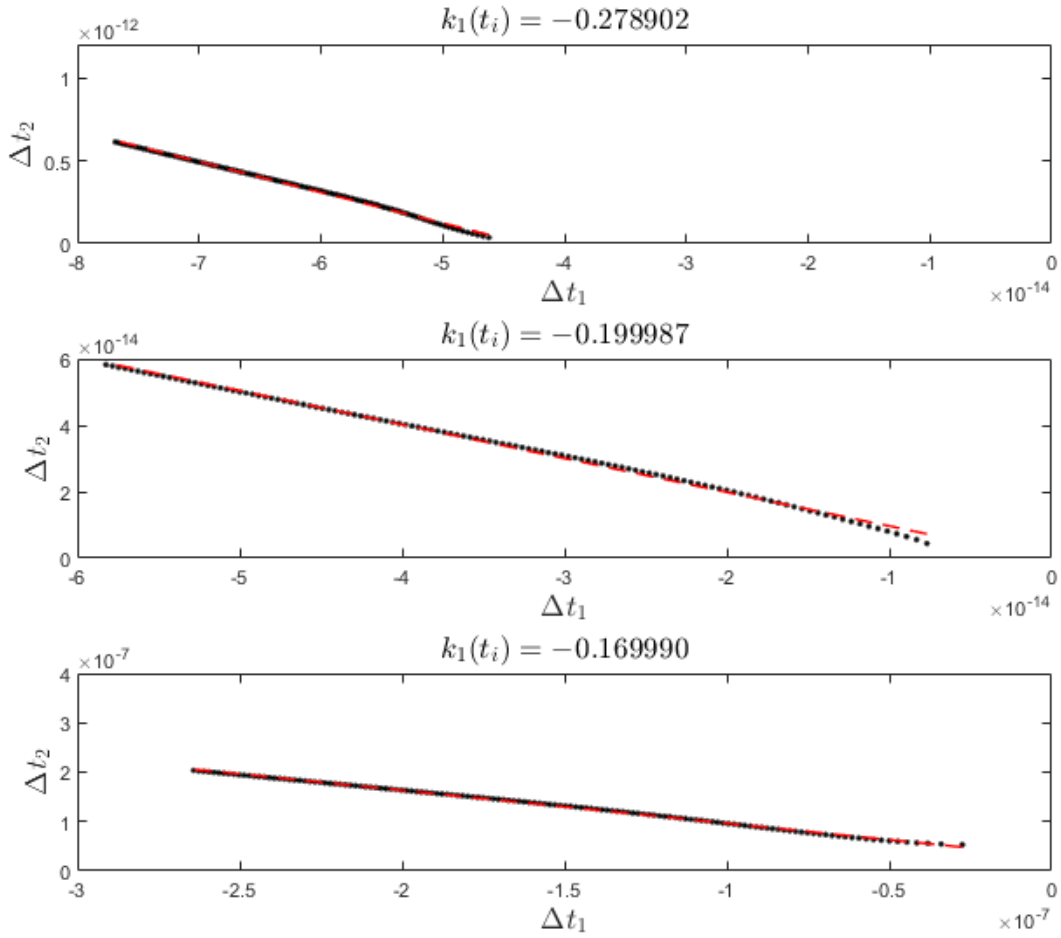

Figure 4.12: These are plots of Δt_2 vs. Δt_1 for several groups of runs. The difference in time between the LQG bounce and the first Kasner transition is Δt_1. The difference in time between the LQG bounce and the second Kasner transition is Δt_2. Note that there is a linear relationship between Δt_2 and Δt_1 away from the intercept.

Chapter 5

Bianchi IX

5.1 The Effective Hamiltonian

By substituting U_{IX} from (2.30) and (2.27) into expression (2.25) we can generate the effective Hamiltonian for Bianchi IX with LQC effects

$$\begin{aligned} C_H = &- \frac{p_1 p_2 p_3}{8\pi G\gamma^2\Delta}[\sin\bar{\mu}_1 c_1 \sin\bar{\mu}_2 c_2 + \sin\bar{\mu}_2 c_2 \sin\bar{\mu}_3 c_3 + \sin\bar{\mu}_3 c_3 \sin\bar{\mu}_1 c_1] \\ &+ \frac{1}{32\pi G}\left(\frac{p_2^2 p_3^2}{p_1^2} + \frac{p_1^2 p_3^2}{p_2^2} + \frac{p_1^2 p_2^2}{p_3^2}\right) - \frac{1}{16\pi G}(p_1^2 + p_2^2 + p_3^2). \end{aligned} \tag{5.1}$$

We can make the potential term more intuitive by substituting (2.14) into expression $U_{IX}(p_i)$ to recover an alternate form of U_{IX} in terms of scale factors.

$$U_{IX}(a_i) = \frac{1}{32\pi G}(a_1^4 + a_2^4 + a_3^4) - \frac{1}{16\pi G}(a_2^2 a_3^2 + a_1^2 a_3^2 + a_1^2 a_2^2) \tag{5.2}$$

As we can see U_{IX} contains three terms which resemble the Bianchi II potential of the form $a_i^4/32\pi G$ and three cross terms of the form $a_i^2 a_j^2/32\pi G$.

5.1.1 Kasner Transitions in Classical Bianchi IX

Just as in Bianchi II, it is worth examining the classic Bianchi IX solution since it will isolate the Bianchi IX potential effects allowing us to examine them in more detail. These potential effects will also appear in the LQC Bianchi IX solutions and will act similarly when the LQC effects are negligible. The Bianchi IX potential is much more complicated than the Bianchi II potential but there are similarities in how it affects the dynamics of our solution. Similarly to Figure 4.1 we can model our solution as a ball bouncing off of potential energy walls as shown in Figure 5.1.

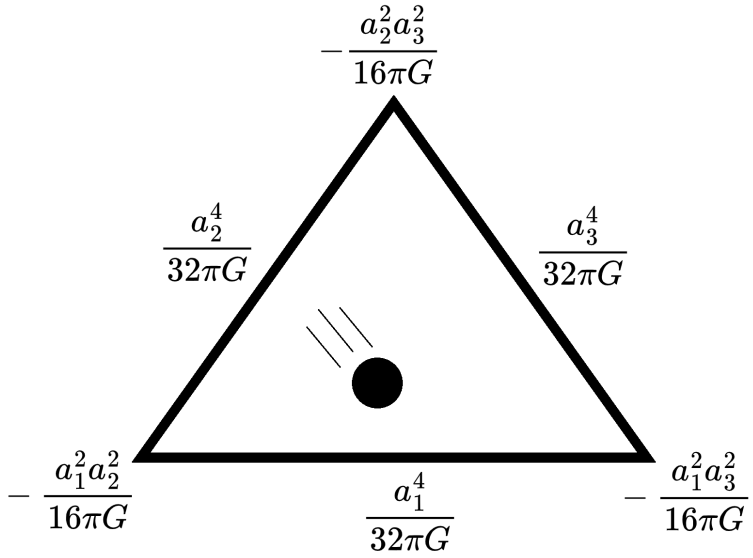

Figure 5.1: The classical Bianchi IX solution can be modelled as a ball bouncing off of potential energy walls. If one and only one scale factor a_i becomes significant there will be a Kasner transition where the solution/ball will bounce off of the corresponding energy wall. The modified Kasner transition rules given by (5.3) can be used to predict the shift in Kasner exponents. If two scale factors a_i and a_j become significant at the same time, the solution/ball will strike a corner and the resulting shift in Kasner exponents cannot be predicted.

As mentioned previously, we have three terms in the form of the Bianchi II potential which are of the form $a_i^4/32\pi G$. If one and only one of our scale factors, a_i, become large U_{IX} will resemble U_{II} when a_1 is large and we will have a Kasner transition. This is analogous to our ball in Figure 5.1 striking one of the potential energy walls.

We can predict this Kasner transition with the modified transition rules given in (5.3). In the case that a_1 becomes large then i=1, j=2, and k=3 in (5.3). The Kasner transition rules for when a_2 and a_3 becomes large can be generated by cyclic permutation [11].

$$\bar{k}_i = \frac{-k_i}{1+2k_i}, \quad \bar{k}_j = \frac{k_j+2k_i}{1+2k_i}, \quad \bar{k}_k = \frac{k_k+2k_i}{1+2k_i} \tag{5.3}$$

However, if two individual scale factors a_i and a_j become large simultaneously then our cross terms of the form $a_i^2 a_j^2/32\pi G$ in the Bianchi IX potential (5.2) become significant. This is analogous to the ball striking the corresponding corner in Figure 5.1. When this happens no simple transition rule can predict how the dynamics in our system will change.

5.1.2 Recollapses and LQG Bounces

As shown previously (2.7) the definition of the mean scale factor is $a = (a_1 a_2 a_3)^{1/3}$. Hence, in classical Bianchi IX, if the mean scale factor, a, is increasing then some of the individual scale factors, a_i, must be increasing as well. There will be more and more Kasner transitions until all scale factors a_1, a_2, and a_3 become significant. Then the cross terms in the Bianchi IX potential (5.2) will dominate and the dynamics will change. It is found numerically that the mean scale factor, a, will reach a maximum value here which we will call recollapse for the remainder of this paper. After the recollapse the mean scale factor, a, will be decreasing and there will be fewer and fewer Kasner transitions per unit time. Please see Figure 5.2 to see a numerical example of a recollapse in classical Bianchi IX.

When we apply LQC effects by solving the LQC Bianchi IX equations of motion instead of the classical Bianchi IX equations of motion, we can better understand the dynamics of the system by examining the mean scale factor, a. When a begins increasing we eventually have a recollapse and a reaches some maximum value. Then

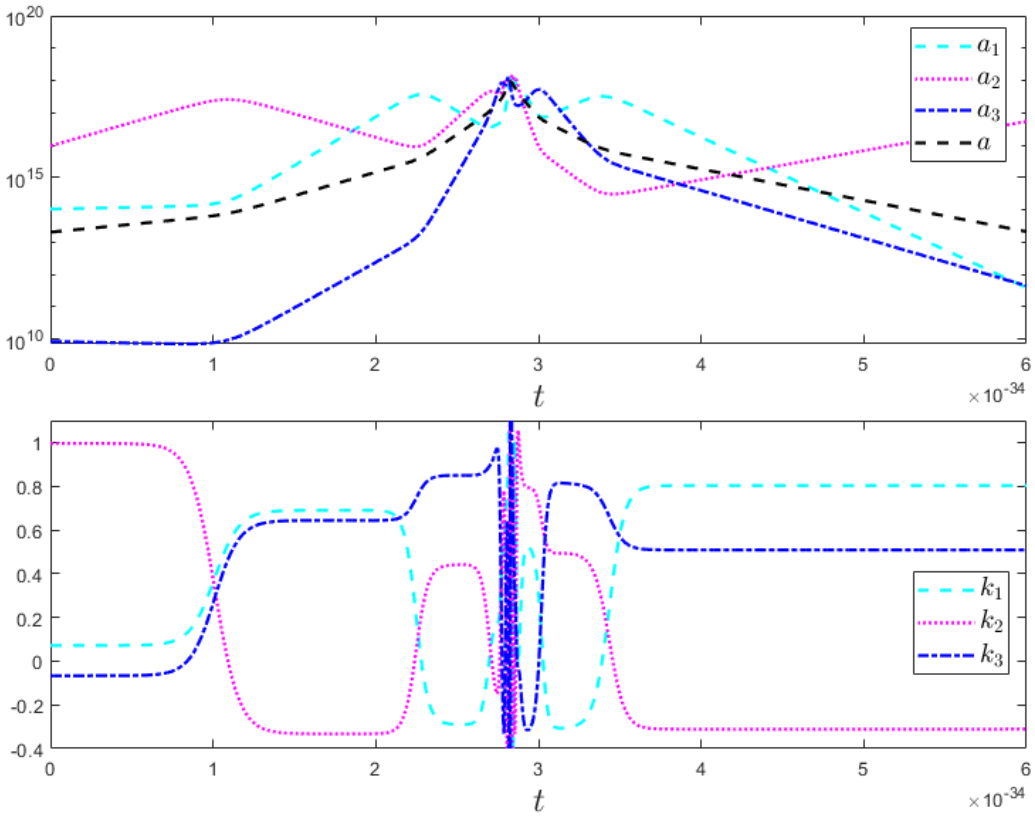

Figure 5.2: This is an example of a recollapse in classical Bianchi IX. As we can see in the top subplot the mean scale factor, a, increases, reaches a maximum value, and then decreases. As can be seen in the bottom subplot there are increasingly many Kasner transitions leading up to the recollapse. During the recollapse the dynamics appear tumultuous and are currently impossible to predict analytically. After the recollapse there are increasingly fewer Kasner transitions per unit time.

a begins to decrease and LQC effects become significant resulting in an LQG bounce where a reaches some minimum value. After this a increases once again resulting in a recollapse. We expect this cycle to continue on forever and is shown pictorially in Figure 5.3. This same cyclic infinite result of recollapses and bounces was is well established in previous literature.

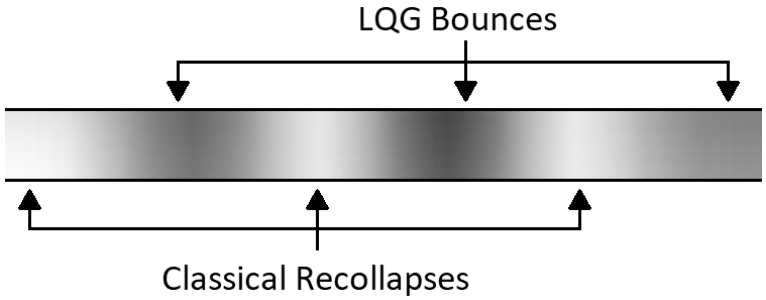

Figure 5.3: This diagrams shows how in LQC Bianchi IX we expect there to be infinitely many bounces and recollapses. The lighter shading indicates a reaching a maximum and the darker shading indicates a reaching a minimum.

5.2 Example of a Run

An example of a Bianchi IX LQC calculation is shown in Figure 5.4. As can be seen in the a_i vs. t subplot, the a continuously reaches a maximum at a recollapse and a minimum at an LQG bounce. Note that the density of Kasner transitions is at the lowest when approaching the LQG bounce and at the highest near the recollapses just as expected. It appears the exact solution would go on forever, however it becomes very difficult to do long calculations with many recollapses numerically. This is because from experience in setting up these calculations, the recollapses represent points of instability in the calculation where the error can grow substantially.

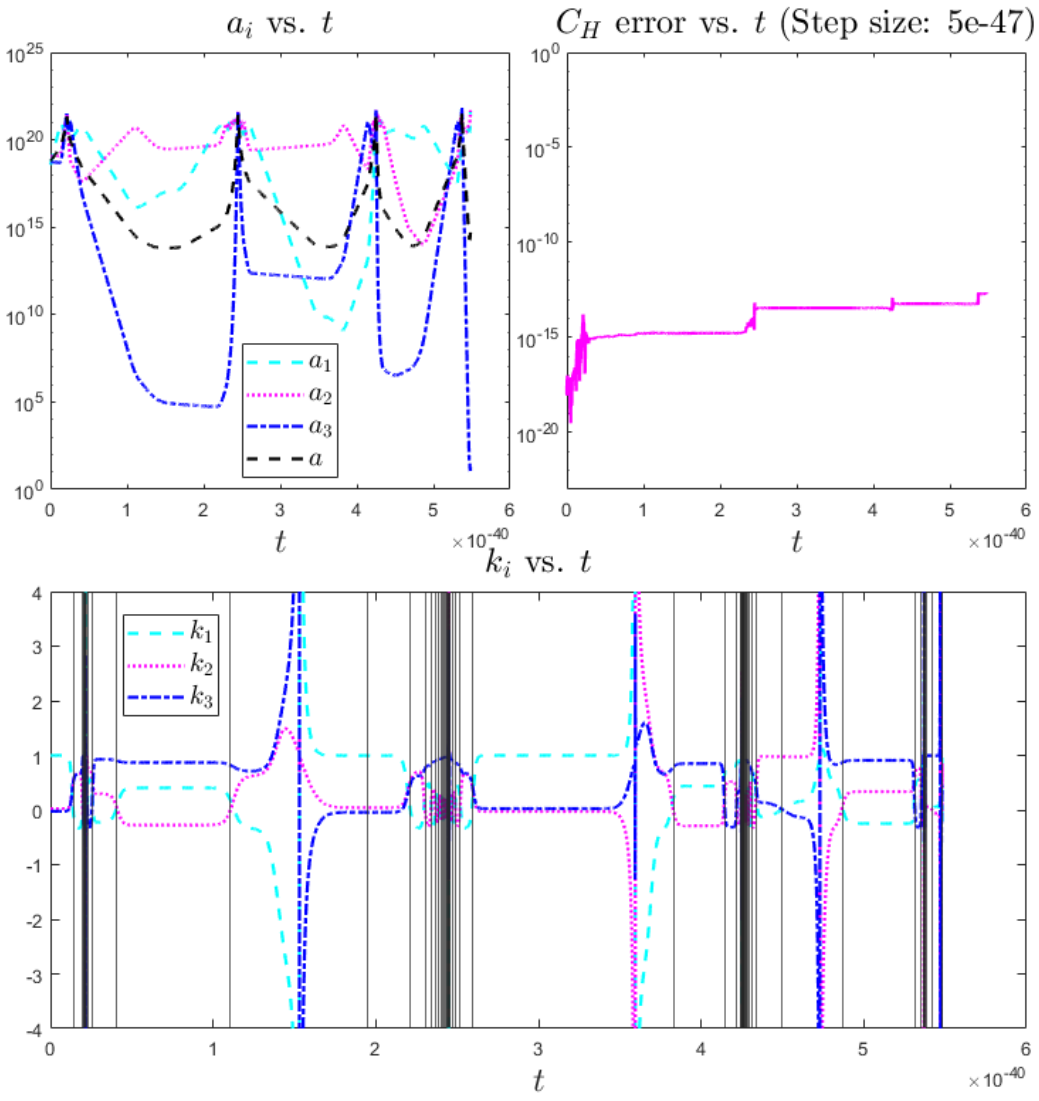

Figure 5.4: This is an example of an LQC Bianchi IX calculation. As can be seen in the top left subplot, "a_i vs. t", the mean scale factor oscillates from having a minimum value at an LQG bounce and a maximum value at a classical recollapse. As can be seen by the low relative C_H error in the "C_H error vs. t", plot this is a very precise calculation. The recollapses are very unstable as can be seen by the jumps in C_H error around the recollapse points. Note that in the "k_i vs. t" subplot, vertical black lines have been added at each Kasner transition for the reader's convenience.

5.3 Analysis Methodology and Initial Conditions

In LQC Bianchi II, there were two distinct cases that needed to be tested which were the LQG bounce happening before any Kasner transitions and the LQG bounce happening after a Kasner transition. For both cases, many runs were done with different initial k_1 values chosen in the range (-1/3,0). However, as we have seen so far this is not the case in Bianchi IX. There are many types of Kasner transitions, many types of LQG bounces, and many types of recollapses in a single run. While there may be ways to classify different runs, there is no distinct method that is the most intuitive which rises above the others. Additionally, it is much more common for a run in LQC Bianchi IX to fail often from an increase in error around the unstable recollapse points.

Hence, the method of analysis has changed when analyzing LQC Bianchi IX in comparison to LQC Bianchi II. Instead of carefully choosing our initial conditions to generate expected results, we are not more concerned with generating many different solutions to examine individually. As discussed previous, in order for a solution to be valid the condition $C_H = 0$ must be met. Using the Hamiltonian (5.1) we can do this simply by choosing values of p_1, p_2, p_3, c_2, c_3, and then choosing c_1 with the expression

$$c_1 = \bar{\mu}_1^{-1} \arcsin\left(\left(\gamma^2\Delta\left(\frac{1}{4}\left(\frac{p_2p_3}{p_1^3}+\frac{p_1p_3}{p_2^3}+\frac{p_1p_2}{p_3^3}\right)-\frac{1}{2}\left(\frac{p_1}{p_2p_3}+\frac{p_2}{p_1p_3}+\frac{p_3}{p_1p_2}\right)\right)\right.\right.$$
$$\left.\left.-\sin\bar{\mu}_2c_2\sin\bar{\mu}_3c_3\right)\left(\sin\bar{\mu}_2c_2+\sin\bar{\mu}_3c_3\right)^{-1}\right). \tag{5.4}$$

Now, while not absolutely necessary it is often preferable to begin a solution away from LQC effects and Bianchi IX potential effects. The argument works the same here as it did in Bianchi II in section 4.2 of this paper. First we choose the initial p_i

and c_i values to be the same order of magnitude as each other

$$p_1 \sim p_2 \sim p_3 \sim p \tag{5.5}$$

and

$$|c_1| \sim |c_2| \sim |c_3| \sim c. \tag{5.6}$$

Then to enforce our solution beginning away from LQG bounces and Kasner transitions we enforce $|C_{Hi}| \gg U_{II}(p_i)$ and $|\bar{\mu}_i c_i| \ll 1$ which again results us choosing initial p_i and c_i values such that $1 \ll c \ll p$. So in practice, when generating a run we choose our initial p_i and c_i values to be of the same order of magnitude of each other and such that $1 \ll c \ll p$ is satisfied. Then we calculate a new value of c_1 using expression (5.4) hoping the new c_1 value also satisfies $1 \ll c \ll p$. Finally, we apply the Runge-Kutta method to see if the run correctly evolves. If it fails we keep choosing new initial p_i and c_i values until we find a run that correctly evolves.

5.4 A Series of 12 Runs

In total there were twelve LQC Bianchi IX runs that we examining in greater depth. Each one of the 12 run solutions along with their corresponding initial conditions are given in Appendix A. Note that our example given in Figure 5.4 is actually Figure A.11 in Appendix A. It was attempted to provide a wide range of runs with different characteristics. However, these were chosen manually so unconscious bias may have affected the runs chosen to be displayed. Some runs have approximately the same amount of time between recollapses, some the recollapse time is decreasing, and others increasing. There is also a wide range of LQG bounces and Kasner transitions in the sample. Each calculation was ended while the error was still relatively small and the solution was valid.

5.4.1 Binning LQG Bounces in the 12 Runs

As previously mentioned, the recollapse points appear to be unstable and impossible to predict final Kasner exponents given initial Kasner exponents. However, there is hope in predicting final Kasner exponents after an LQG bounce. For each of these twelve runs we classified every LQG bounce to be in one of four bins as shown in Figure 5.5.

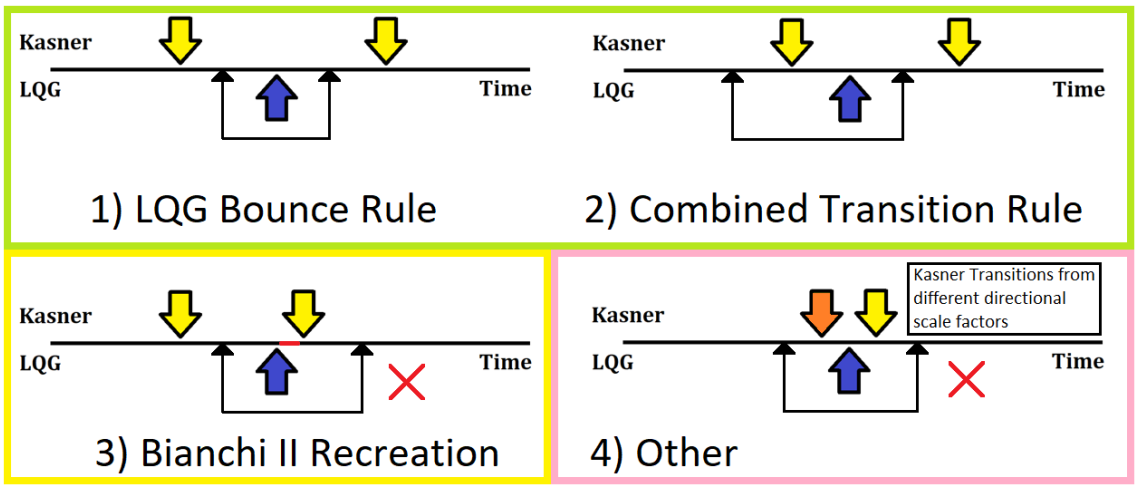

Figure 5.5: This is the criteria for what bin the LQG bounce goes into. Bin 1, If the LQG bounce rule can be used to predict Kasner exponents after the bounce within 1%. Bin 2, if some combined transition rule can be used to predict Kasner exponents after the bounce within 1%. Bin 3, if the Kasner exponents after a bounce can be predicted by recreating the system within Bianchi II numerically within 1%. Bin 4, any other LQG bounce.

Let us go through these bins in greater depth and give an example of the corresponding type of LQG bounce. Each example of an LQG bounce will be taken from one of the the twelve runs in Appendix A. The first bin is is called "LQG Bounce Rule". This is where we can directly apply the LQG bounce rule (3.2) using the Kasner exponents directly before the LQG bounce to predict the Kasner exponents directly after the LQG bounce. An example of an LQG bounce can be taken from the first LQG bounce in Figure A.1 as shown in Figure 5.6.

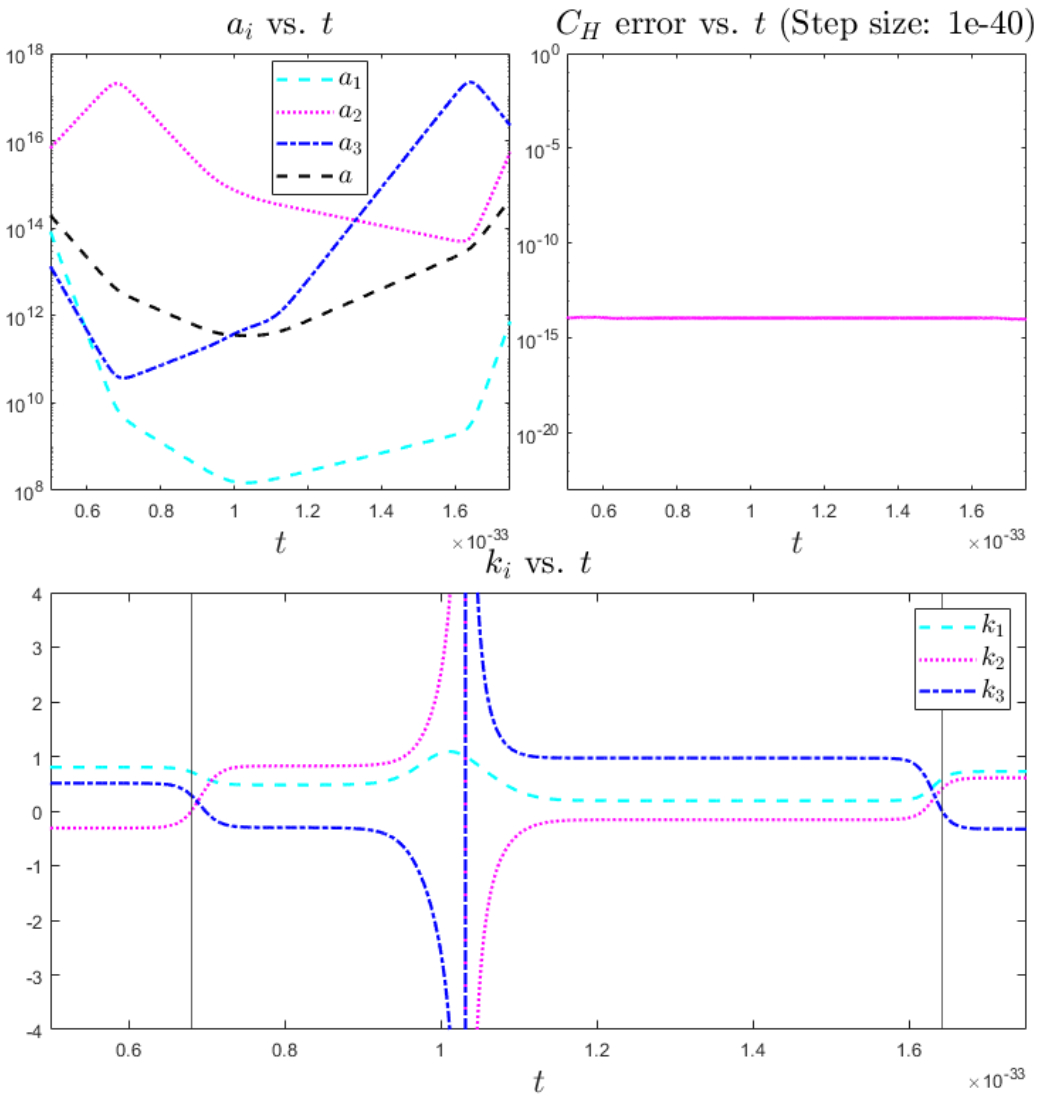

Figure 5.6: This is a section of the numerical solution shown in Figure A.1 showing the first LQG bounce. Here the Kasner transitions are well separated from the LQG bounce. This allows the transition rule (3.2) to be used to predict the $\bar{k}_i$ values directly after the LQG bounce using the k_i values directly before the LQG bounce.

For testing the LQG bounce rule on Figure 5.6 the k_i values were taken at time: 8.055000e-34 and the $\bar{k}_i$ values were taken at time: 1.427000e-33. The results of testing the LQG bounce rule is found in Table 5.1.

	k_1	k_2	k_3
Numerical	0.477779	0.824875	-0.302652
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.188888	-0.158208	0.969319
Numerical	0.189073	-0.158208	0.969134
Percent Error	0.097822%	0.000287%	0.019109%

Table 5.1: Bin 1 LQG Bounce Rule Results

We define the percent error in Table 5.1 to be

$$\text{Percent Error} = \Big|\frac{\text{Numerical } \bar{k}_i - \text{Predicted } \bar{k}_i}{\text{Predicted } \bar{k}_i}\Big|. \tag{5.7}$$

The largest percent error is lower than 1% for each final $\bar{k}_i$ value when applying transition rule (3.2), and hence this LQG bounce is put into the first bin. Also note that the k_i values and $\bar{k}_i$ values are taken when the expression $\sum_i k_i$ is at its flattest in their corresponding regions since we expect the expressions (2.24) to hold in regions that can be approximated by classical Bianchi I.

Now let us examine which LQG bounces go into the second bin named "combined Transition Rule" as shown in Figure 5.5. An LQG bounce is placed into the second bin when it cannot be placed into the first bin, and some Kasner exponents after the LQG bounce can be predicted using a combination of the LQG bounce transition rule (3.2) and the Kasner transition rules (5.3). This often happens when a Kasner transition is close to an LQG bounce but not completely overlapping it. An example of this is shown in Figure 5.7. Here both the LQG bounce rules (3.2) and the Kasner transition rules (5.3) are used in conjunction to predict some $\bar{k}_i$ values after the LQG bounce.

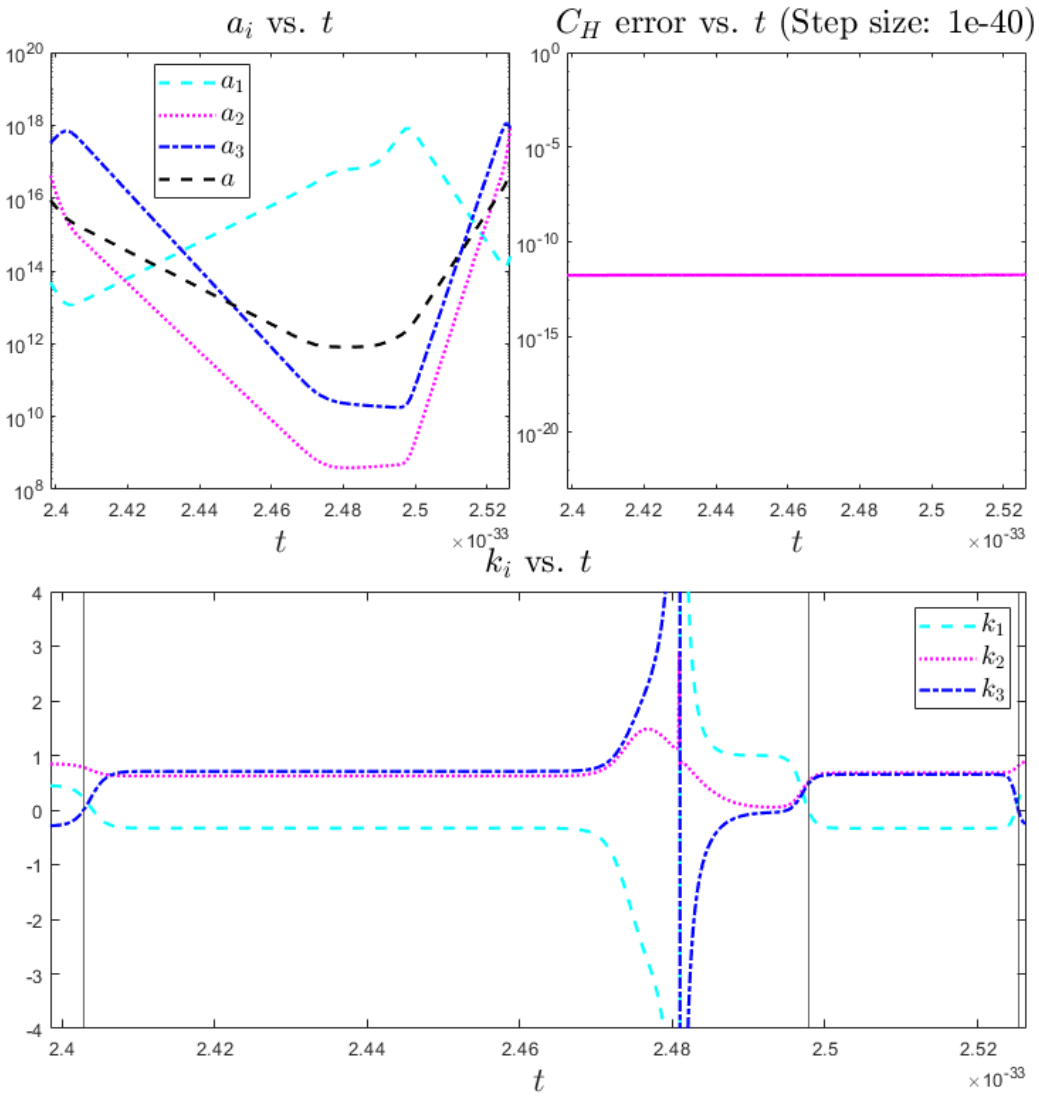

Figure 5.7: This is a section of the numerical solution shown in Figure A.1 showing the fourth LQG bounce. Here there is a Kasner transition right after the LQG bounce preventing only (3.2) being used to predict Kasner exponents after the bounce. The only way to predict final Kasner exponents $\bar{k}_i$ within 1% error is to first use the (3.2) transition rule to get over the LQG bounce, and then use (5.3) to get over the Kasner transition. Using both transition rules together creates a combined transition rule.

For this combined transition rule the k_i values are taken at time 2.432800e-33 and $\bar{k}_i$ values are taken at time: 2.515700e-33.

	k_1	k_2	k_3
Numerical	-0.331438	0.622267	0.709183
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.333123	0.681064	0.652055
Numerical	-0.333129	0.681248	0.651846
Percent Error	0.001814%	0.027052%	0.031980%

Table 5.2: Bin 2 Combined Transition Rule Results

Since the percent error is lower that 1% for all the $\bar{k}_i$ values this LQG bounce is placed into the second bin.

Now, let us examine which LQG bounces go into the third bin which is called "Bianchi II Recreation" according to Figure 5.5. Here no combination of analytical transition rules given by (3.2) and (5.3) can be used to predict any Kasner exponents after the LQG bounce within 1% error. However, the system has one prominent directional scale factor causing a Kasner transition near the LQG bounce so it can be recreated numerically in LQC Bianchi II. This numerical recreation can be used to predict some final Kasner exponents after the LQG bounce within 1%. An example of LQG bounce in Bianchi IX that fits into the third bin is shown in Figure 5.8. As we can see the direction scale factor a_3 becomes very prominent and causes a Kasner transition which partly overlaps the LQG bounce. We can recreate this Kasner transition and LQG bounce in Bianchi II by defining the Bianchi II potential in terms of the a_3 scale factor, $U_{II} = a_3^4/32\pi G$.

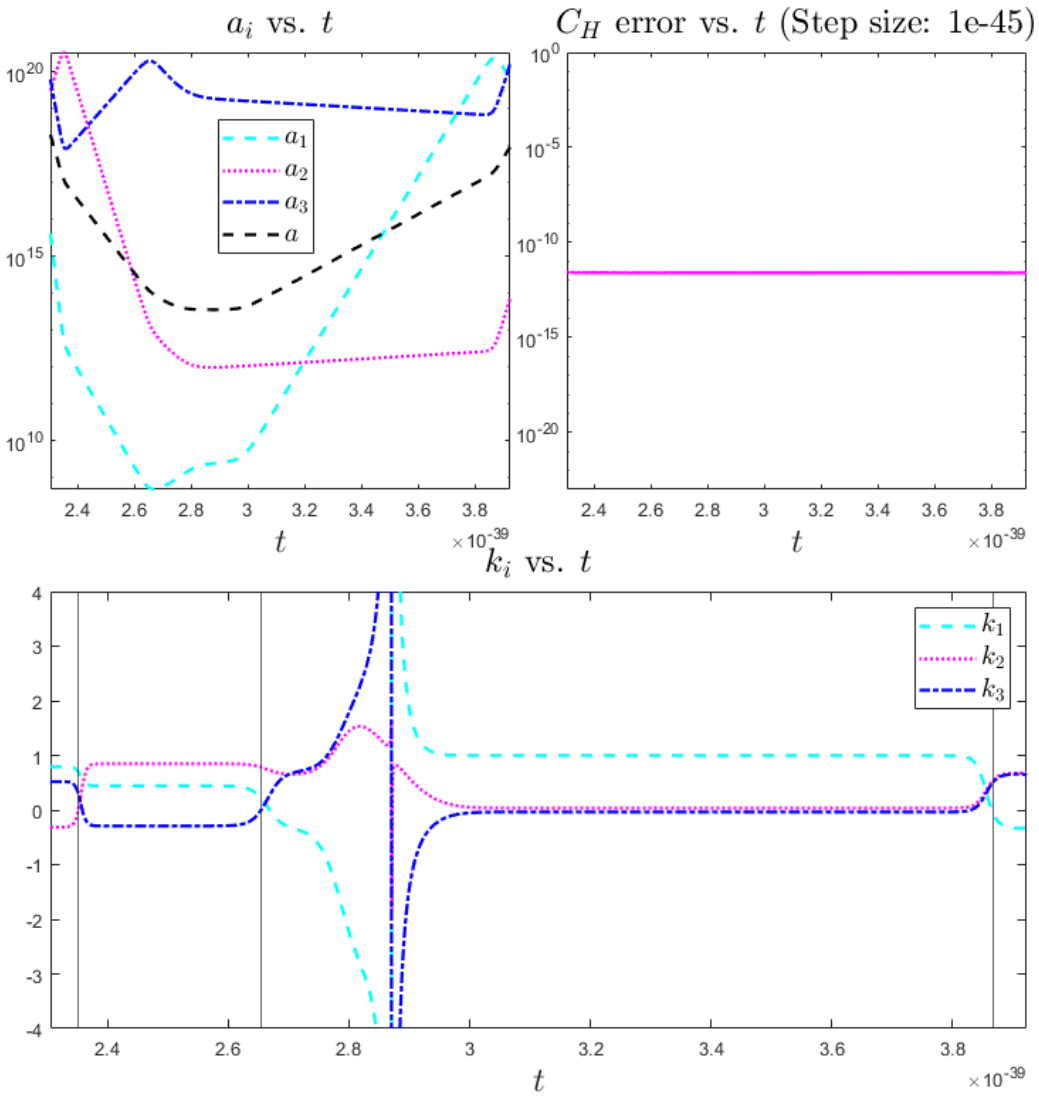

Figure 5.8: This is a section of the numerical solution shown in Figure A.4 showing the second LQG bounce. Here there is a Kasner transition occurs so close to the LQG bounce it prevents any combination of (3.2) and (5.3) transition rules to predict any Kasner exponents after the bounce. The only way to predict final Kasner exponents $\bar{k}_i$ within 1% error without using the LQC Bianchi IX numerical solution is to recreate the the Kasner transition and LQG bounce in LQC Bianchi II. This is possible since the potential of only one directional scale factor, a_3, is prominent.

Here are the results from applying a combined transition rules in Figure 5.8 where the initial k_i values were taken at time 2.436000e-39 and the final $\bar{k}_i$ values taken at time: 3.133000e-39.

	k_1	k_2	k_3
Numerical	0.443669	0.847569	-0.291216
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.998978	0.031710	-0.030743
Numerical	0.998671	0.036800	-0.035481
Percent Error	0.030643%	16.052690%	15.414408%

Table 5.3: Bin 3 Combined Transition Rule Results

As seen in the results, the percent error is above 1% for k_2 and k_3 so this LQG bounce cannot be put into the second bin.

The LQC Bianchi recreation can be seen in Figure 5.9. The initial conditions were chosen by taking the p_i and c_i values where $\sum_i k_i$ was the flattest in between 2.4×10^{-39} and 2.6×10^{-39} in Figure 5.8. Then the LQC Bianchi II Hamiltonian equations were solved where the Bianchi II potential was defined as $U_{II} = a_3^4/32\pi G$. Note that the C_H error is much higher in 5.9 as opposed to 5.8. This is because U_{II} and U_{IX} while similar when a_3 is prominent, are not exactly the same. This larger C_H error in the Bianchi II recreation is acceptable though since it was only constructed to calculate the final $\bar{k}_i$ values which it does quite well.

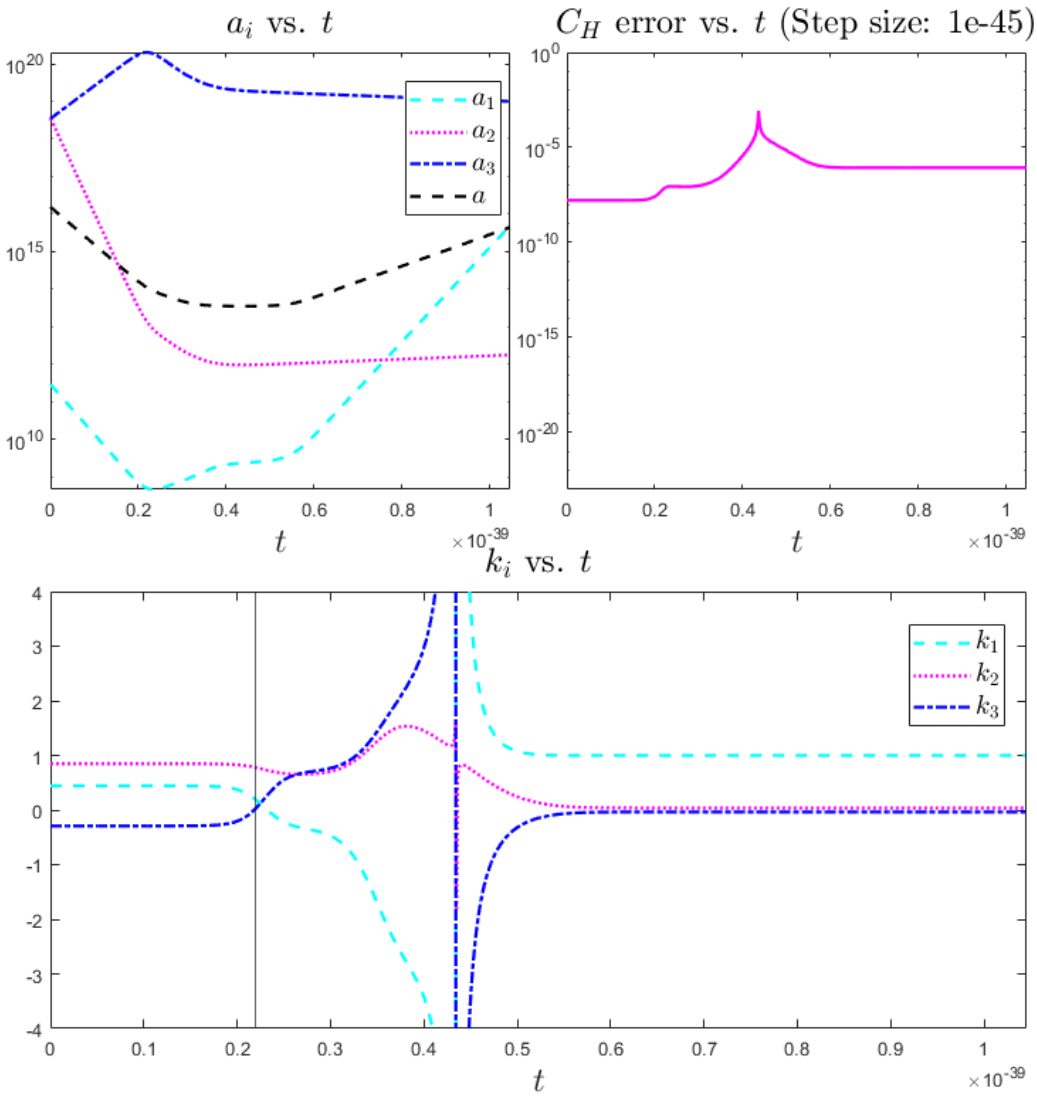

Figure 5.9: This an LQC Bianchi II recreation of the Kasner transition and LQG bounce shown in Figure 5.8. Here the value of c_1 was modified slightly using a version of (4.5) with an updated potential to set $C_H = 0$. This recreation was only possible since only one directional scale factor, a_3, is prominent. The final $\bar{k}_i$ values are almost identical as those in Figure 5.8. Note the C_H error is mach larger here than in Figure 5.8 due to small differences between U_{II} and U_{IX} when a_3 is prominant.

Here are the results from applying a combined transition rules in Figure 5.8 where the initial k_i values were taken at time 2.000000e-42 the final $\bar{k}_i$ values taken at time: 6.970000e-40.

	k_1	k_2	k_3
Numerical	0.443669	0.847569	-0.291216
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.998978	0.031710	-0.030743
Numerical	0.998671	0.036800	-0.035481
Percent Error	0.030637%	16.052594%	15.414496%

Table 5.4: Bin 3 Combined Transition Rule on Bianchi II Recreation Results

As we can see, the results are almost identical for the LQC Bianchi II recreation as was for the original LQC Bianchi IX solution. Most importantly, the final numerical $\bar{k}_i$ values are essentially identical for both, allowing the Bianchi II recreation to predict the final $\bar{k}_i$ values in the original Bianchi IX run to a percent error much lower than 1%.

Lastly, we have the fourth bin where none of the above techniques predict final Kasner exponents after an LQG bounce. Generally this happens when two separate scale factors become prominent. An example of this is shown in Figure 5.10. In this case both the scale factor a_1 and a_3 are large and there is no moment after the LQG bounce where the scale factors settle to constant values. the only way to predict final Kasner exponents after the LQG bounce here is to run our numerical LQC Bianchi IX solution that we generated for this paper.

The results for all the LQG bounces in all of the 12 runs displayed in Appendix A is shown in Table 5.11. It is noteworthy that only just over half of the LQG bounces can be examined using analytical methods. Additionally, note that approximately a quarter of the LQG bounces fell into the other category where the final $\bar{k}_i$ values can only be found using the numerical LQC Bianchi IX solutions.

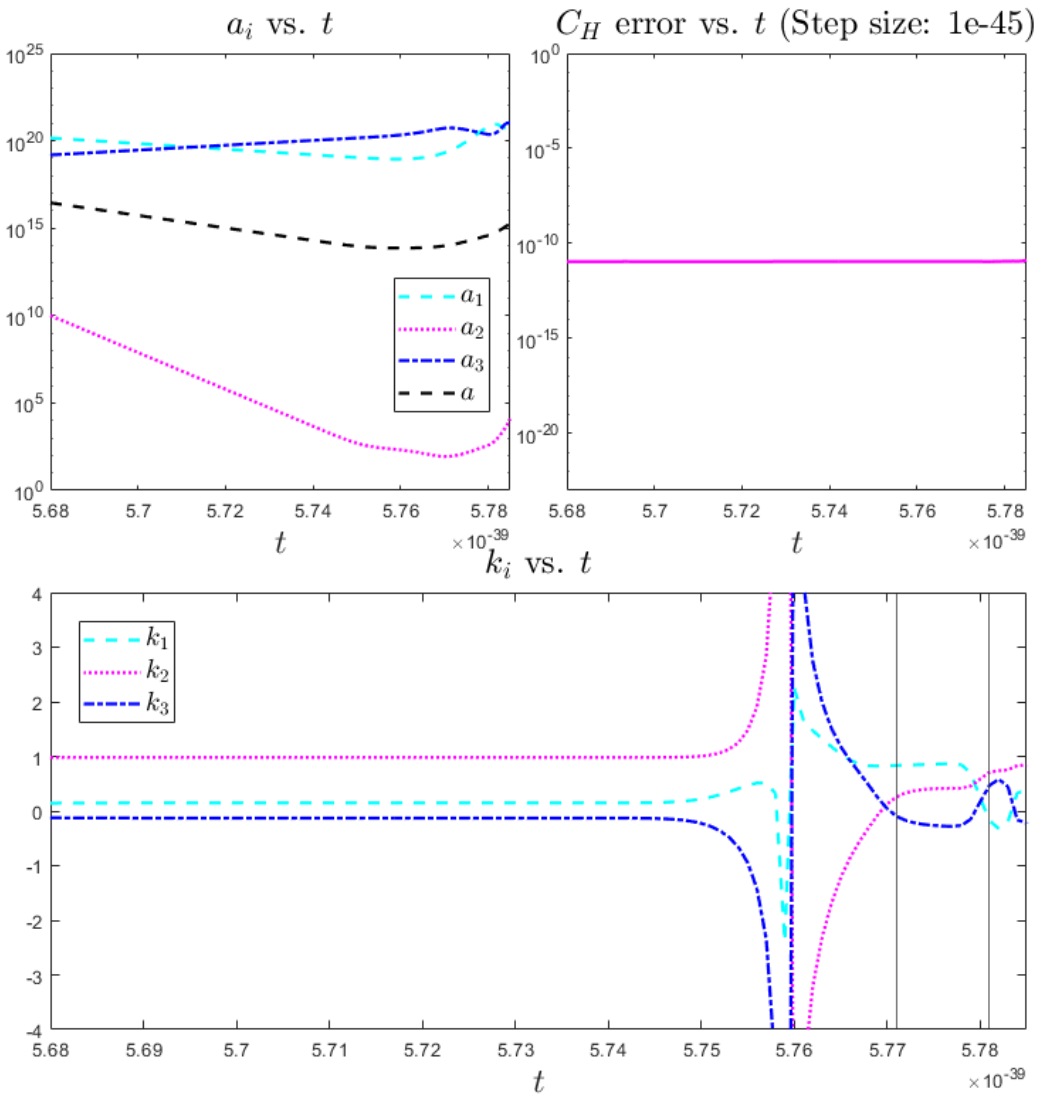

Figure 5.10: This is a section of the numerical solution shown in Figure A.4 showing the fourth LQG bounce. Both a_1 and a_3 are prominent after the LQG bounce causing the Kasner exponents to never settle to constant values. Neither combinations of transition rules or a Bianchi II recreation can be used to predict any final Kasner exponents after the LQG bounce.

Run	1) LQG Bounce Rule	2) Combined Transition Rule	3) Bianchi II Recreation	4) Other
1	1	1	0	3
2	3	2	0	0
3	1	2	0	0
4	1	0	1	2
5	1	2	0	0
6	0	1	1	1
7	1	0	0	2
8	2	1	0	1
9	2	1	0	1
10	0	1	1	1
11	0	1	2	1
12	2	0	3	0
Sum	14	12	8	12

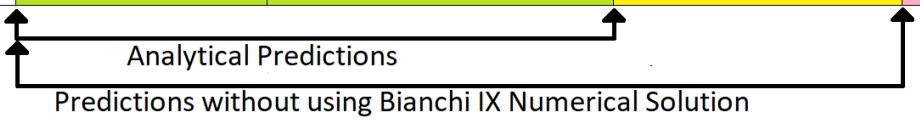

Figure 5.11: Here all the types of LQG bounces are shown for all 12 runs displayed in Appendix A.

5.4.2 Chaos

It has been shown analytically in previous literature that classic Bianchi IX is chaotic [11]. Using our numerical tools developed for this paper we can begin preliminary explorations of understanding whether LQC Bianchi IX is chaotic. To accomplish this, calculations were set up in which two of the initial p_i or c_i values from a previous original run were modified slightly such that $C_H = 0$ was still true. Then these systems were evolved and and calculations ended when while the C_H error was still small and the run was valid. These altered solutions were then compared to their original Bianchi IX runs to see if the difference grew over time. An example of such a comparison plot is shown in Figure (5.12) where the initial conditions for the calculation in Figure A.3 were slightly modified to create an altered Bianchi IX solution.

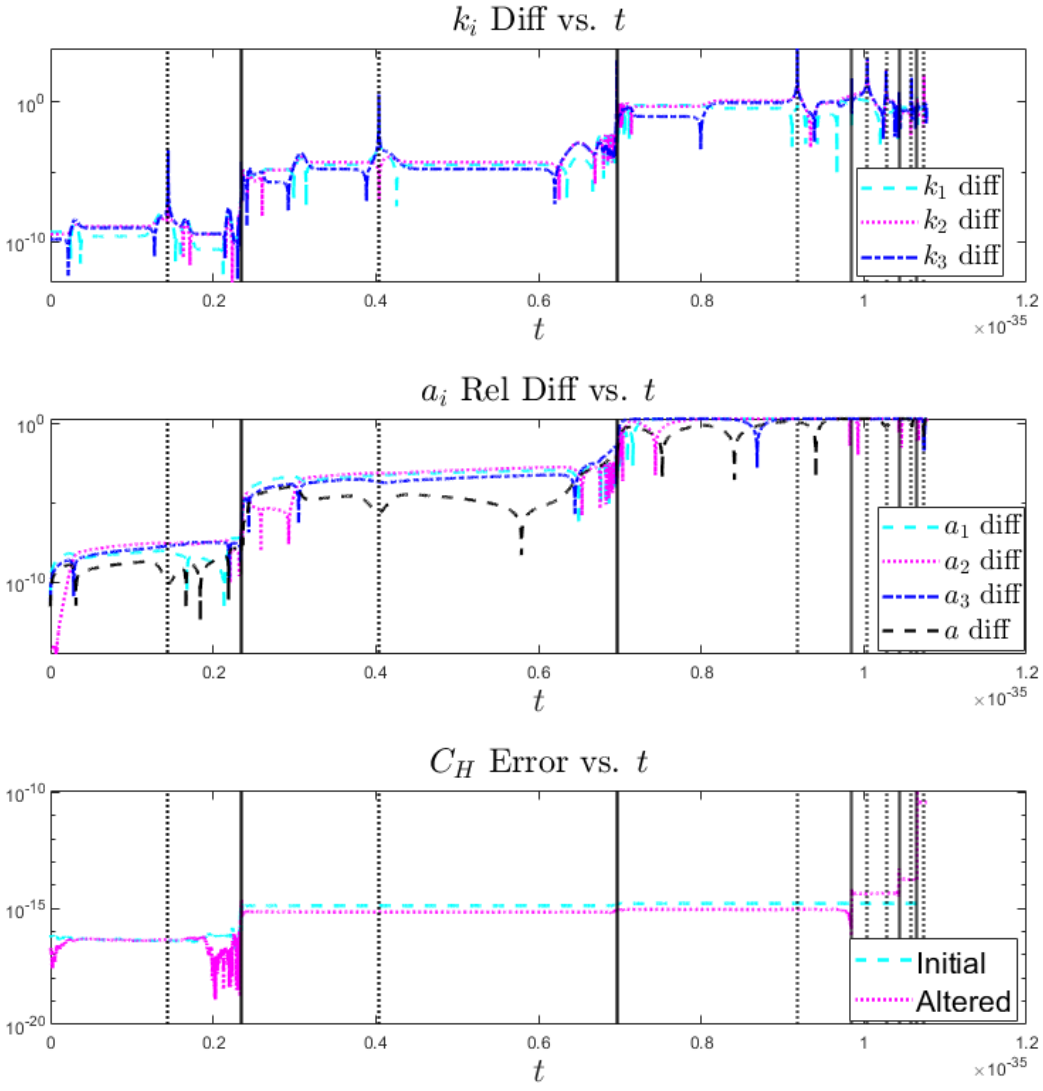

Figure 5.12: This is the LQC Bianchi IX- 3 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The bottom subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

This process was done for all of the 12 LQC Bianchi IX runs and are displayed in Appendix B. As can be seen by the plots, the error tends to grow during recollapses instead of LQG bounces as expected. Each one of the altered solutions eventually looked nothing like the original solution while the error was still acceptably small. This suggests that LQC Bianchi IX is chaotic just like classic Bianchi IX [6].

5.5 High Precision Calculation

During the LQC Bianchi IX calculations it was found after going through three recollapses solutions would diverge no matter how close their initial conditions were. This was true for any step size no matter how small it was made. Since lowering the step size did not seem to improve calculations instead we set up a calculation with more precise variables. To explore chaos we wanted two solutions with similar initial conditions to make it through more than three recollapses before diverging. Additionally, we wanted to have a very high precision run with a very low relative C_H error to test its Kasner transitions.

Previously, C++ was used for all of the Runge-Kutta calculations. The data types chosen for the p_i, c_i, and t values have been double floats which are comprised of 64 bits each. We wrote a program in the language Julia that relies on variable precision for these values which we set to 128 bits. Julia was selected since its variable precision library did not require a plug-in and was found to be significantly faster than both Matlab and Python. Even still, from testing the Julia code was found to be 49 times slower than comparable non-variable precision C++ code.

Since Julia was so much slower the classic fourth order Runge-Kutta method which required fewer calculations per step was used with the Butcher tableau shown below in Table 5.5.

Additionally, the step size was decreased manually by trial and error around recol-

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Table 5.5: Classic Fourth Order Runge-Kutta Method

lapse points. We would find a numerically solution, see where the C_H error grew around the recollapse point, and decrease the step size in that region. This was done repeatedly until we had a very precise solution where the relative C_H error remained very low. Other variable step size procedures were tested but doing it manually had the best results. The manually chosen variable step size is shown in Figure 5.13 corresponding to a high precision calculation is shown in Figure 5.14.

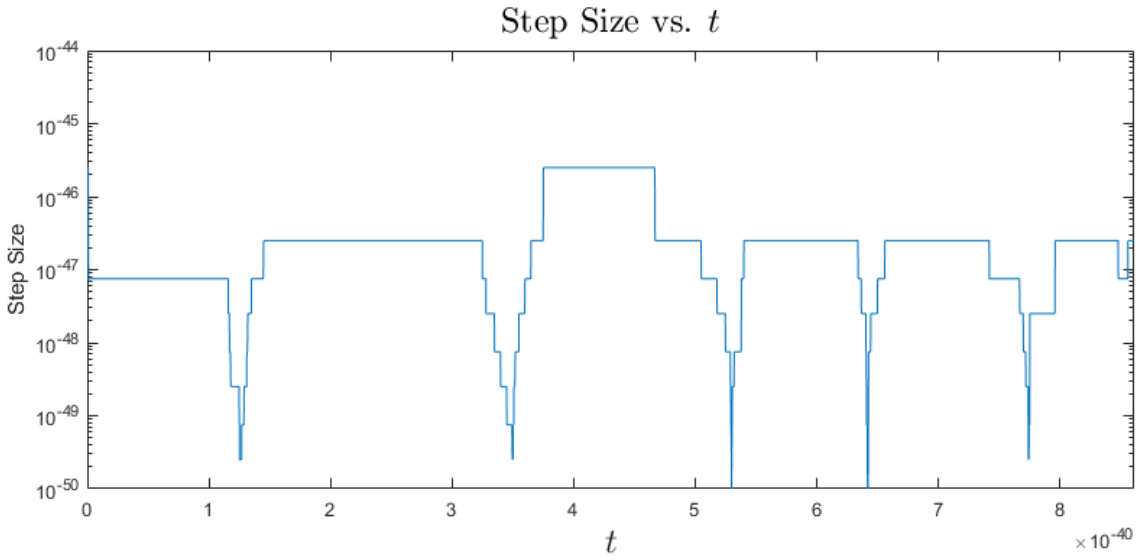

Figure 5.13: This is the variable step size vs. time plot corresponding to the calculation shown in Figure 5.14. The step size is the smallest around the recollapse points to prevent error growing and the calculation failing.

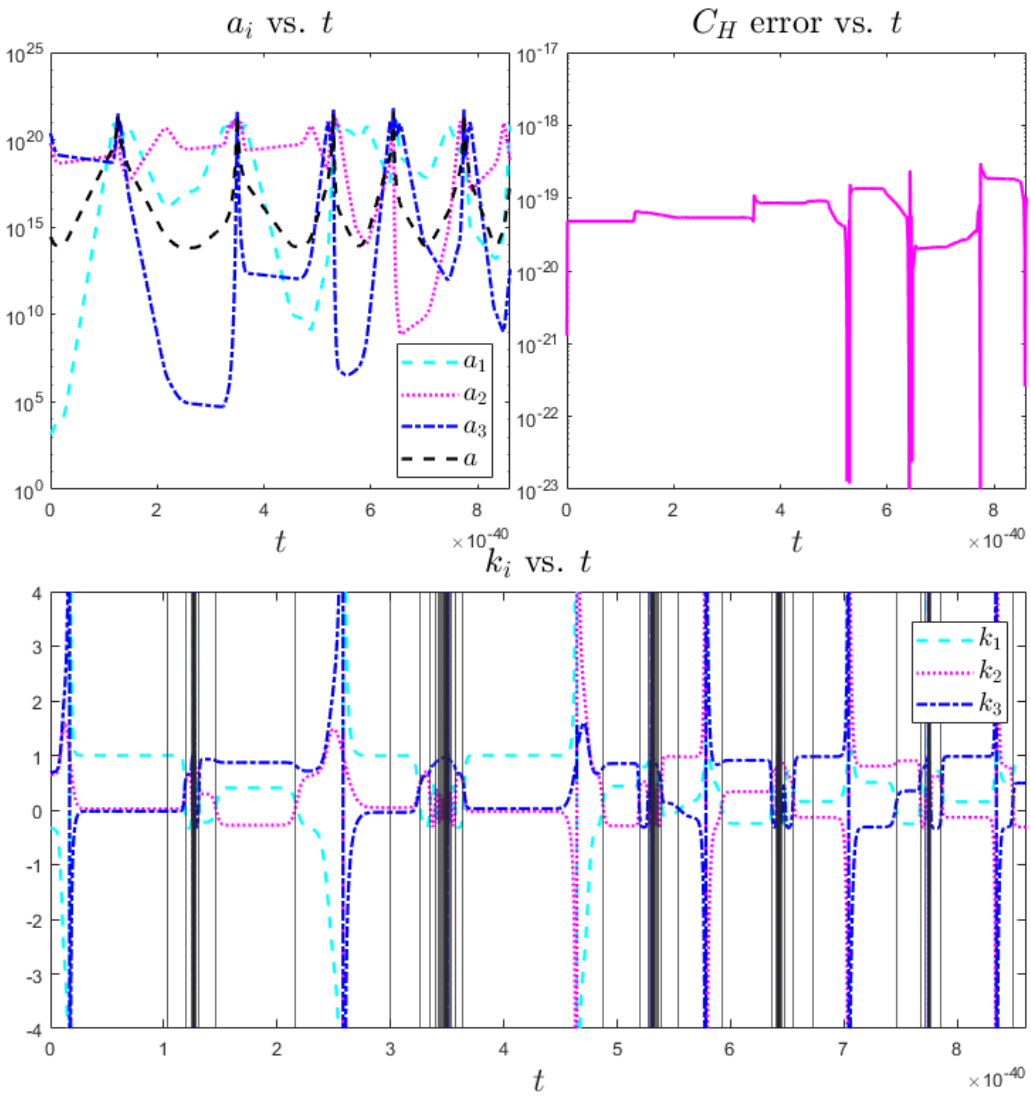

Figure 5.14: Here is a high precision run done in Julia. The same initial conditions were used here as in Figure A.11, except they were chosen earlier. Since this was a high precision run the relative C_H error in the top right subplot was able to be kept around 1×10^{-19} which is 4-5 orders of magnitude better than what can theoretically be achieved in C++ with double float variables. This entire calculation took just less than 5 days.

5.5.1 Chaos

Again, just as in Section 5.4.2 chaos is explored by comparing the calculation shown in Figure 5.14 to another calculation with slightly altered initial conditions. The benefit of doing this with a higher precision calculation calculation is the initial conditions from the altered solution can be made even closer to the original solution. In the previous lower precision calculations, the c_i or p_i were double floats and hence in decimal its sixteenth digit was its last potential significant digit. For this high precision calculation we will modify a c_i value past the sixteenth digit. The initial conditions can be used to generate Figure 5.14 are shown below.

```
p1_0 = 1.57549289819556600057171354507545553267 8e+40
p2_0 = 3.32166618657297193856030290329280362266 6e+23
p3_0 = 8.80411091747172606443843910622085442727 6e+22
c1_0 = 8.78999171391004219378777089180295758196
c2_0 = -8.18024647973045594586004203225537898241 2e+17
c3_0 = -3.27618267012579825205474328538964190226 2e+18
```

For the slightly altered run the initial conditions were kept the same, except for c_1 which was slightly modified in values including and past its twentieth digit as seen below.

```
c1_0 = 8.78999171391004219339864705796079164 62
```

Note that unlike previous altered solutions, only one c_i value was altered. This was possible since c_1 was altered so slightly that any error introduced to the statement $C_H = 0$ was negligible. The altered solution is shown in Figure 5.15. As can be seen by comparing with the original solution shown in Figure 5.14, the altered solution makes it through 4 recollapses before diverging from the original solution. This would suggest that LQC Bianchi IX is chaotic, however much more work needs to

be done in this area. To determine whether LQC Bianchi IX is chaotic, Lyapunov exponents cannot be used in GR since they are coordinate dependent. Any chaotic test has to be be coordinate independent such as the Painlevé test, but this was not conclusive for classic Bianchi IX [6]. A coordinate independent fractal method was used to show classic Bianchi IX was chaotic [6]. Some similar method may need to be employed to show whether LQC Bianchi IX is also chaotic. However, the numerical work done in this paper is a good starting point to examine the potential chaos in LQC Bianchi IX.

The step size profile is similar to that shown in Figure 5.13, but it had to be modified after the fourth recollapse when the solution diverged. The comparison plot comparing the original calculation shown in Figure 5.14 to the altered solution shown in Figure 5.15 is given in Figure 5.16. Here is another a very clear example of the error growing during the recollapse points as oppose to the LQG bounces.

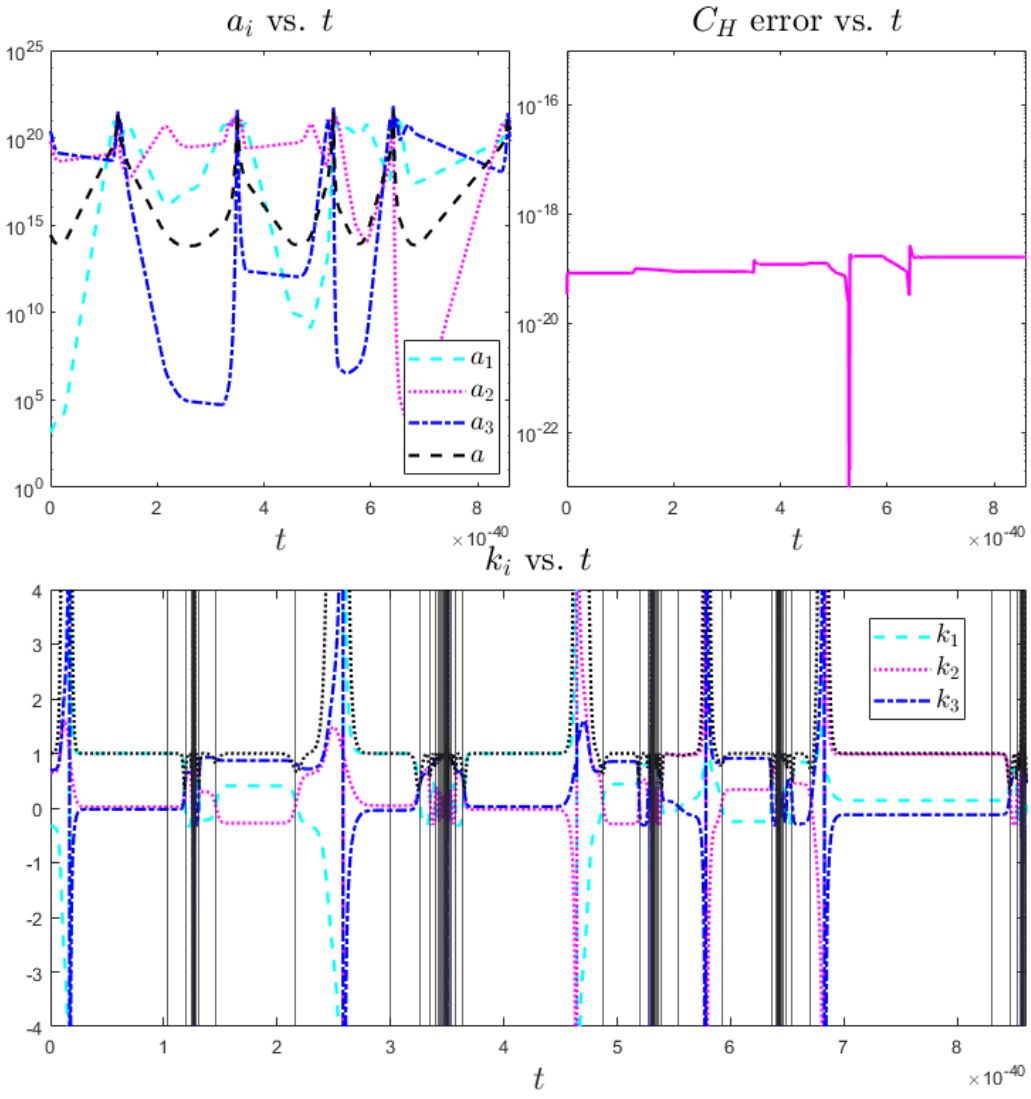

Figure 5.15: Here is the altered high precision run done in Julia. The same initial conditions were used here as in Figure A.11, except they were chosen earlier. Since this was a high precision run the relative C_H error in the top right subplot was able to be kept around 1×10^{-19} which is 4-5 orders of magnitude better than what can theoretically be achieved in C++ with double float variables. This entire calculation took just less than 6 days.

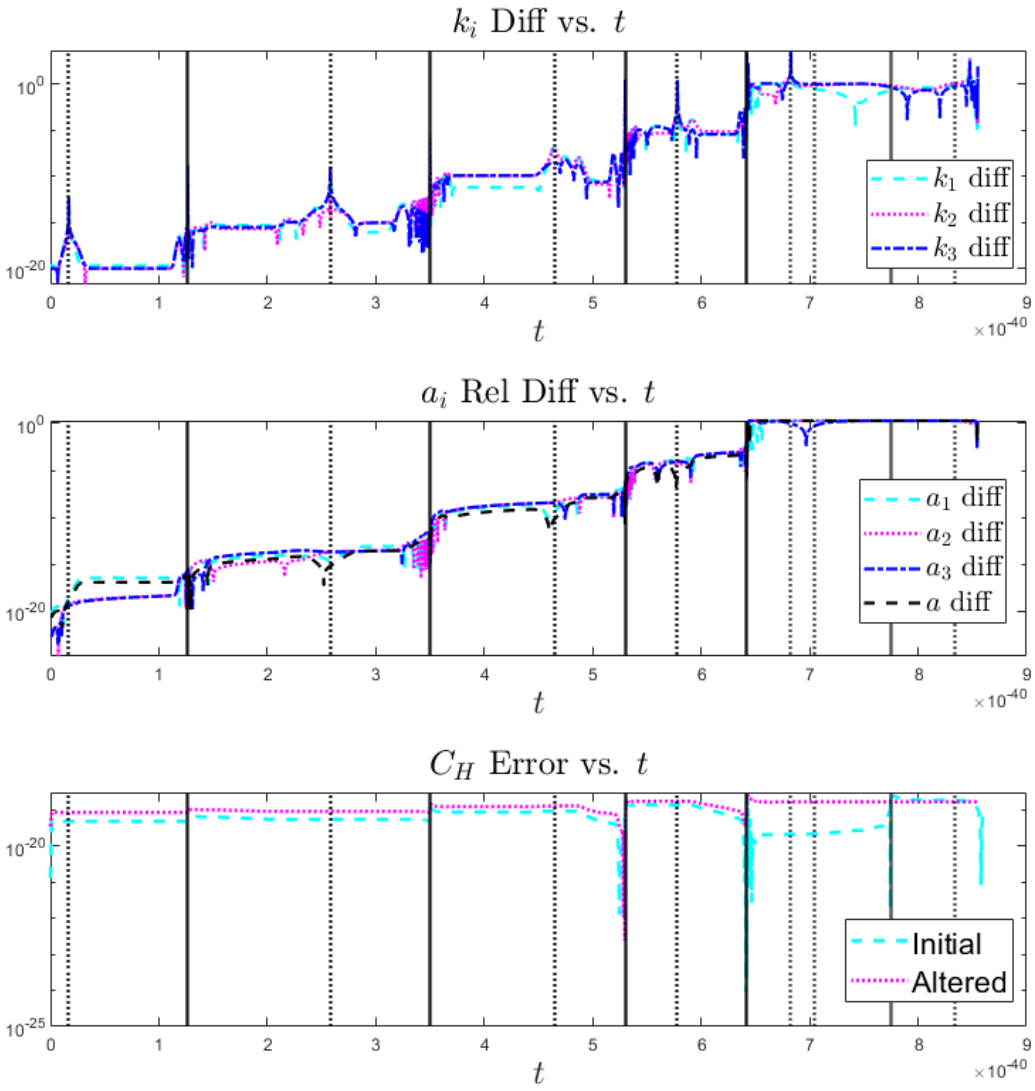

Figure 5.16: This is the High Precision LQC Bianchi IX comparison plot comparing calculations Figure 5.14 and Figure 5.15. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

5.5.2 Examining Kasner Transitions

In addition to chaos, we can use this high precision calculation shown in Figure 5.14 to examine the effectiveness of the Bianchi IX transition rules given by (5.3). There are five recollapse points total in the calculation shown in Figure 5.14 and hence four time periods sandwiched between two recollapse points. We calculated the the predicted $\bar{k}_i$ values and compared them to the numerical $\bar{k}_i$ values for every Kasner transition in these four time periods. Additionally, since calculations in LQC Bianchi IX are reversible we applied the transition rules in reverse to calculate predicted k_i values and compare them to the numerical k_i values. The modified transition rules we used to do this are given below

$$k_i = \frac{-\bar{k}_i}{1+2\bar{k}_i}, \quad k_j = \frac{\bar{k}_j + 2\bar{k}_i}{1+2\bar{k}_i}, \quad k_k = \frac{\bar{k}_k + 2\bar{k}_i}{1+2\bar{k}_i}. \tag{5.8}$$

So each Kasner was tested regularly with rules (5.3) and labelled "Kasner Transition" and backwards with rules (5.8) and labelled "Backwards Kasner Transition". Below is an example of the testing the five Kasner transitions between the first recollapse point and the LQG bounce sandwiched between the first two recollapse points. All 57 Kasner transitions between the first and fourth recollapse points were tested in the same fashion and the results can be found in Appendix C.

1.

Maximum a_2: k_i values taken at time: 1.268779e-40 and $\bar{k}_i$ values taken at time: 1.273328e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.816016	-0.469332	0.653316
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-1.999594	7.651797	-4.652203
Numerical	-0.198277	0.992546	0.205732
Percent Error	90.084113%	87.028594%	104.422247%

Table 5.6: Testing Forward Kasner Transition Rules- 1

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.198277	0.992546	0.205732
	k_1	k_2	k_3
Predicted	0.598579	-0.332501	0.733922
Numerical	0.816016	-0.469332	0.653316
Percent Error	36.325533%	41.152059%	10.982932%

Table 5.7: Testing Backward Kasner Transition Rules- 1

2.

Maximum a_1: k_i values taken at time: 1.273328e-40 and $\bar{k}_i$ values taken at time: 1.288318e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.198277	0.992546	0.205732
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.328576	0.987647	-0.316223
Numerical	0.584474	0.743083	-0.327557
Percent Error	77.881042%	24.762305%	3.584234%

Table 5.8: Testing Forward Kasner Transition Rules- 2

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.584474	0.743083	-0.327557
	k_1	k_2	k_3
Predicted	-0.269473	0.881548	0.387926
Numerical	-0.198277	0.992546	0.205732
Percent Error	26.420429%	12.591274%	46.966197%

Table 5.9: Testing Backward Kasner Transition Rules- 2

3.

Maximum a_3: k_i values taken at time: 1.288318e-40 and $\bar{k}_i$ values taken at time: 1.342532e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.584474	0.743083	-0.327557
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.204820	0.255067	0.949753
Numerical	-0.224059	0.295298	0.928762
Percent Error	9.393258%	15.772590%	2.210156%

Table 5.10: Testing Forward Kasner Transition Rules- 3

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.224059	0.295298	0.928762
	k_1	k_2	k_3
Predicted	0.571636	0.753387	-0.325023
Numerical	0.584474	0.743083	-0.327557
Percent Error	2.245772%	1.367732%	0.779480%

Table 5.11: Testing Backward Kasner Transition Rules- 3

4.

Maximum a_1: k_i values taken at time: 1.342532e-40 and $\bar{k}_i$ values taken at time: 1.744989e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.224059	0.295298	0.928762
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.405991	-0.276907	0.870917
Numerical	0.407005	-0.277316	0.870314
Percent Error	0.249771%	0.147511%	0.069281%

Table 5.12: Testing Forward Kasner Transition Rules- 4

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.407005	-0.277316	0.870314
	k_1	k_2	k_3
Predicted	-0.224368	0.295861	0.928509
Numerical	-0.224059	0.295298	0.928762
Percent Error	0.137501%	0.190189%	0.027245%

Table 5.13: Testing Backward Kasner Transition Rules- 4

5.

Maximum a_2: k_i values taken at time: 1.744989e-40 and $\bar{k}_i$ values taken at time: 2.284489e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.407005	-0.277316	0.870314
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.331471	0.622666	0.708812
Numerical	-0.333699	0.619812	0.713888
Percent Error	0.672234%	0.458301%	0.716134%

Table 5.14: Testing Forward Kasner Transition Rules- 5

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.333699	0.619812	0.713888
	k_1	k_2	k_3
Predicted	0.404499	-0.276748	0.872250
Numerical	0.407005	-0.277316	0.870314
Percent Error	0.619554%	0.205052%	0.221953%

Table 5.15: Testing Backward Kasner Transition Rules- 5

There are two significant things we wish to highlight in the above data. Firstly, the percent error for each Kasner transition tends to go down as as we move from the first recollapse to the LQG bounce between the first and second recollapse points. This is logical since the cross terms in the U_{IX} potential tend to be less significant the lower the mean scale factor, a, is. Hence, Bianchi IX resembles Bianchi II more closely which is where the Kasner transition rules were derived.

Secondly, the percent error where we test the Kasner transition rules in reverse tend to be lower than when we test them forwards. We hypothesize this this is because the Bianchi IX universe is in a state of contraction in the above calculations. In fact,

when the universe is in a state of expansion the forward Kasner transition rules tend to have lower percent error than the backwards Kasner Transition rules. This can be confirmed from examining the percent errors in the Kasner transitions between the LQG bounce sandwiched between the first two recollapse points and the second recollapse point as shown in Appendix C. To see this algebraically for the Kasner exponent corresponding to the scale factor associated with the Kasner transition, let us say that a Kasner transition occurs corresponding to a maximum of the scale factor a_i. Then the scale factor a_i must be increasing before the Kasner transition. As shown in previous literature, if a_i is increasing in an contracting universe then $k_i < 0$. Likewise, if a_i is increasing in an expanding universe then $k_i > 0$ [11]. Let us define the error in both directions

$$|\text{Predicted } \bar{k}_i - \text{Numerical } \bar{k}_i| = \bar{\epsilon} \quad \text{and} \quad |\text{Predicted } k_i - \text{Numerical } k_i| = \epsilon. \tag{5.9}$$

Now, let us make the approximation that the Kasner transition rules are exact and it is the initial k_i and $\bar{k}_i$ values that are inputted with some error labelled ϵ_e. By doing this we can implement the error propagation formula. Then, by taking the derivatives of the $\bar{k}_i$ and k_i expressions from the rules (5.3) and (5.8) respectively, we can find the sensitivity relating ϵ_e individually to $\bar{\epsilon}$ and ϵ as shown below

$$\text{Forward: } \bar{\epsilon} = \frac{\epsilon_e}{(1+2k_i)^2} \quad \text{and} \quad \text{Backward: } \epsilon = \frac{\epsilon_e}{(1+2\bar{k}_i)^2} \tag{5.10}$$

Hence, in a contracting universe in general we expect $\bar{\epsilon} > \epsilon$ and in an expanding universe $\bar{\epsilon} < \epsilon$, which is reflected in the results. We can extend this logic to k_j and k_k. Since the transition rules for k_j and k_k are identical, without loss of generality we can just examine k_j. Let us define

$$|\text{Predicted } \bar{k}_j - \text{Numerical } \bar{k}_j| = \bar{\epsilon}_j \quad \text{and} \quad |\text{Predicted } k_j - \text{Numerical } k_j| = \epsilon_j. \tag{5.11}$$

We again assume that the transition rules from (5.3) and (5.8) for k_j and $\bar{k}_j$ are exact and the initial k_j and $\bar{k}_j$ values have an error of ϵ_{ej}. we implement the error propagation formula by taking partial derivatives of the expressions to find the corresponding sensitivities and using addition in quadrature. The corresponding error propagation formulas both forward and backward respectively are

$$\begin{aligned} \text{Forward: } \bar{\epsilon}_j &= \sqrt{\left(\frac{2}{(1+2k_i)^2}\right)^2 \epsilon_e^2 + \left(\frac{1}{1+2k_i}\right)^2 \epsilon_{ej}^2} \quad \text{and} \\ \text{Backward: } \epsilon_j &= \sqrt{\left(\frac{2}{(1+2\bar{k}_i)^2}\right)^2 \epsilon_e^2 + \left(\frac{1}{1+2\bar{k}_i}\right)^2 \epsilon_{ej}^2} \end{aligned} \tag{5.12}$$

In both expressions above, assuming $\epsilon_{ej} \sim \epsilon_e$ the sensitivity corresponding to ϵ_e is dominant due to the extra square term in the denominator. This also indicates that in a contracting universe we expect $\bar{\epsilon}_j > \epsilon_j$ and in an expanding universe $\bar{\epsilon}_j < \epsilon_j$ to be generally true. Exceptions do arise in some calculations where errors cancel.

Finally, we wanted to characterize the accuracy of all 57 Kasner transitions shown in Appendix C. The first method was to only look at forward Kasner transitions and no backward Kasner transitions. For each Kasner transition, we took the largest percent error and put it into one of the corresponding bins dependent on its value, [0%, 1%], (1%, 5%], (5%, 15%], (15%, 35%], (35%, ∞). The result of this binning is shown in Figure 5.17.

Additionally, we tried using the backwards Kasner transition rules for when the Bianchi IX universe was contracting and the forwards Kasner transition rules for when the universe was expanding. The same binning procedure was applied and

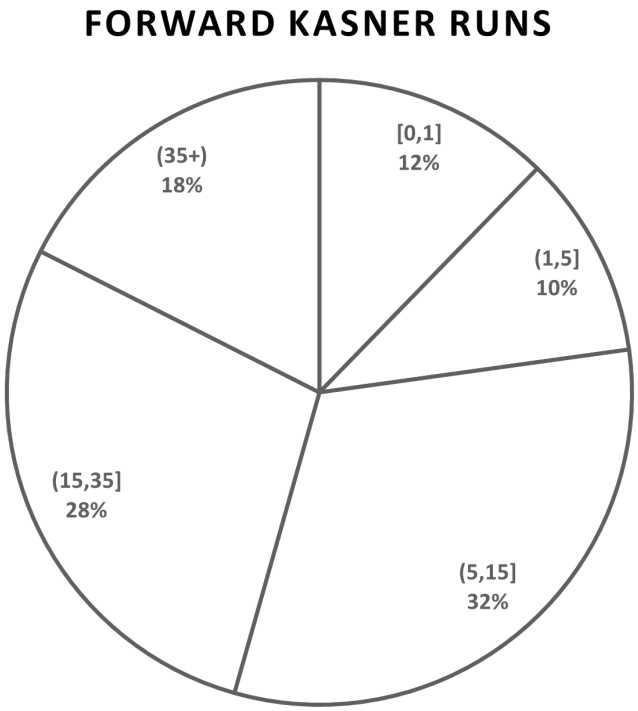

Figure 5.17: Here is the accuracy of the Kasner transitions when only the forward Kasner transition rules are used.

the results are shown in Figure 5.18 which shows the Kasner transitions being more accurate than Figure 5.17 as expected.

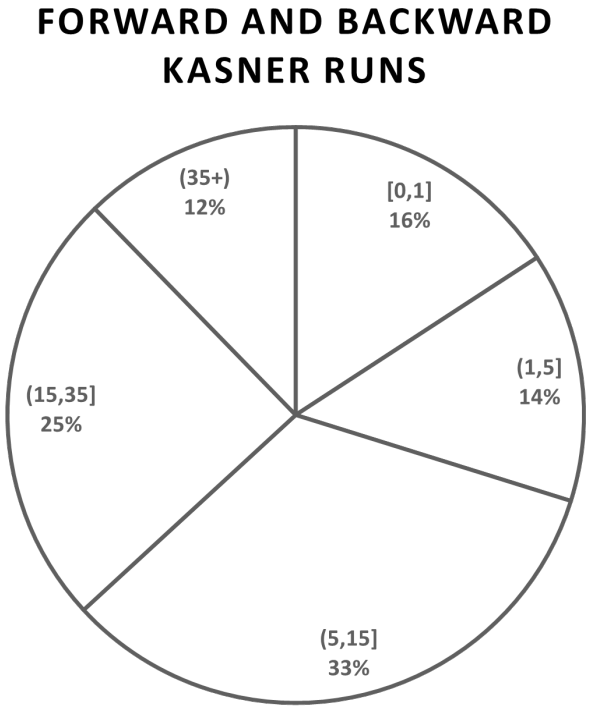

Figure 5.18: Here is the accuracy of the Kasner transitions when both the forward Kasner transition rules and backward Kasner transition rules are used depending on whether the Bianchi IX universe is expanding or contracting respectively. Here the Kasner transition rules seem more accurate than in Figure 5.17 just as expected.

Something to note is how frequent highly inaccurate Kasner transitions were. This tended to happen near the recollapse points where two scale factors would become prominent/intertwined and continuously reach many maximums right after another.

Chapter 6

Conclusion

6.1 LQC Bianchi II Conclusion

Methods were developed to find initial conditions resulting in valid solutions where $C_H = 0$ and where the initial k_i values were chosen. Additionally, a method to move the LQG bounce while holding $C_H = 0$ and the initial k_i values was found. As shown in Figures 4.4 and 4.7, the LQG bounce rule (3.2) and Kasner transition rules (4.4) can be used in conjunction to predict final Kasner exponents when the LQG bounce and Kasner transition(s) are well separated. Unfortunately, as shown in Figure 4.10 the actual time between an LQG bounce and a Kasner transition required for it to be well separated is ambiguous. Although, we can check whether a system is well separated by using our LQC Bianchi II numerical model. Additionally, we can probe systems that are not well separated with our numerical model.

6.1.1 LQC Bianchi IX Conclusion

In LQC Bianchi IX, there were no two simple cases test as was the case in LQC Bianchi II. Instead twelve random numerical solutions were found and analyzed individually. As shown in Figures 5.5 and 5.11, the LQG bounces in each of these

runs was able to be placed into 4 unique bins characterized by the transition rules and the predictive ability of reconstructing the relevant Kasner transition(s) and LQG bounce in LQC Bianchi II. For approximately a quarter of all LQG bounces in the 12 runs, any Kasner exponents after the LQG bounce could not be predicted without using the LQC Bianchi IX numerical solver developed for this paper. Additionally, the numerical calculations suggests that LQC Bianchi IX is chaotic just like classic Bianchi IX. A few high precision calculations were also carried out using multi-variable precision and manually set variable step size in the language Julia. The results from analyzing the high precision runs suggest that LQC Bianchi IX is chaotic. Much more work would have to be done to reach a firm conclusion for this question. Additionally, the Kasner transitions between two recollapses were tested using the Kasner transition rules. The rules were found to be accurate away from both the recollapse points and the LQG bounce, and when only one scale factor, a_i, was dominant.

Bibliography

[1] Abhay Ashtekar, Tomasz Pawlowski, and Parampreet Singh, *Quantum Nature of the Big Bang: Improved dynamics*, Phys. Rev. D **74** (2006), 084003.

[2] Abhay Ashtekar and Edward Wilson-Ewing, *Loop quantum cosmology of Bianchi I models*, Phys. Rev. D **79** (2009), 083535.

[3] V.A. Belinskii, I.M. Khalatnikov, and E.M. Lifshitz, *Oscillatory approach to a singular point in the relativistic cosmology*, Advances in Physics **19** (1970), no. 80, 525–573.

[4] Alejandro Corichi and Edison Montoya, *Effective Dynamics in Bianchi Type II Loop Quantum Cosmology*, Phys. Rev. D **85** (2012), 104052.

[5] ———, *Loop quantum cosmology of Bianchi IX: Effective dynamics*, Class. Quant. Grav. **34** (2017), no. 5, 054001.

[6] Neil J. Cornish and Janna J. Levin, *The Mixmaster universe is chaotic*, Phys. Rev. Lett. **78** (1997), 998–1001.

[7] Brajesh Gupt and Parampreet Singh, *Quantum gravitational Kasner transitions in Bianchi-I spacetime*, Phys. Rev. D **86** (2012), 024034.

[8] Metin Hatun and Fahri Vatansever, *Differential equation solver simulator for Runge-Kutta methods*, Uludağ University Journal of The Faculty of Engineering **21** (2016), no. 1, 145–162.

[9] Charles W. Misner, *Mixmaster universe*, Phys. Rev. Lett. **22** (1969), 1071–1074.

[10] Carlo Rovelli and Edward Wilson-Ewing, *Why are the effective equations of loop quantum cosmology so accurate?*, Phys. Rev. D **90** (2014), no. 2, 023538.

[11] Edward Wilson-Ewing, *The loop quantum cosmology bounce as a Kasner transition*, Class. Quant. Grav. **35** (2018), no. 6, 065005.

Appendix A

The 12 Bianchi IX Runs with LQC Effects

A.1 Initial Conditions

Here are the initial conditions for the 12 Bianchi IX runs.

```
LQC Bianchi IX- 1 Initial Conditions
p1_0 = 8.039999999999999e+25;
p2_0 = 9e23;
p3_0 = 1e30;
c1_0 = 30000000.000000008134767904279735241544;
c2_0 = 3.72232352997251587e+10;
c3_0 = -2.24994857268156738e+03;
h = 1e-40;
t = 2.5455e-33;

LQC Bianchi IX- 2 Initial Conditions
p1_0 = 1e33;
p2_0 = 1e31;
p3_0 = 1e32;
c1_0 = -19999.999999999870318327594218369483237;
c2_0 = -3e+4;
c3_0 = 2.95565295692105883e+03;
h = 1e-43;
t = 3.3468e-36;

LQC Bianchi IX- 3 Initial Conditions
p1_0 = 7e32;
p2_0 = 9e34;
p3_0 = 2e33;
c1_0 = -30000.000000000007168053088236869104398;
c2_0 = -4e+2;
c3_0 = 6.63146731201925195e+03;
h = 0.5e-42;
```

```
t = 1.2016e-35;


LQC Bianchi IX- 4 Initial Conditions
p1_0 = 7e33;
p2_0 = 9e32;
p3_0 = 9e37;
c1_0 = -3000000.000000005277052311112042456738 1;
c2_0 = -4e+4;
c3_0 = 3.36733074272204946e-01;
h = 1e-45;
t = 5.787e-39;


LQC Bianchi IX- 5 Initial Conditions
p1_0 = 1e36;
p2_0 = 2e31;
p3_0 = 3e39;
c1_0 = -79999.999999999980520960678603493077868;
c2_0 = 3e+08;
c3_0 = -2.53378399797385857e+07;
h = 1e-52;
t = 1.2282e-45;


LQC Bianchi IX- 6 Initial Conditions
p1_0 = 4e33;
p2_0 = 7e36;
p3_0 = 3e37;
c1_0 = -3000.000000000197401118453777088126 1618;
c2_0 = 3e+03;
c3_0 = 1.09477483244404061e+03;
h = 5e-47;
t = 2.1676e-39;


LQC Bianchi IX- 7 Initial Conditions
p1_0 = 3.644407087693920e+23;
p2_0 = 4.619481809923470e+22;
p3_0 = 6.508819947785010e+34;
c1_0 = -11330329129524.171780987804676656471287;
c2_0 = 3.726944157209900e+13;
c3_0 = -45.773504837509698;
h = 2e-41;
t = 1.457e-34;


LQC Bianchi IX- 8 Initial Conditions
p1_0 = 9e35;
p2_0 = 8e36;
p3_0 = 5e36;
c1_0 = -8000.000000000000713789375690566162 6527;
c2_0 = 9e+03;
c3_0 = 1.60000138740636453e+03;
h = 4e-47;
t = 3.2676e-40;


LQC Bianchi IX- 9 Initial Conditions
```

```
p1_0 = 3e36;
p2_0 = 2e37;
p3_0 = 3e38;
c1_0 = -699.99999999999844920231170941457203 16;
c2_0 = 8e+03;
c3_0 = 7.11326370614312697e+00;
h = 2e-46;
t = 1.4134e-39;


LQC Bianchi IX- 10 Initial Conditions
p1_0 = 2.026722231822400e+28;
p2_0 = 4.854478714956990e+33;
p3_0 = 5.944594117494380e+22;
c1_0 = 6.704487157259680e+13;
c2_0 = -34788654.737103765863724334053266205 17;
c3_0 = 1.765985877358840e+19;
h = 5e-48;
t = 3.0715e-41;


LQC Bianchi IX- 11 Initial Conditions
p1_0 = 6e37;
p2_0 = 2e+37;
p3_0 = 4e37;
c1_0 = 7000.00000000002835421528593858278542 44;
c2_0 = 5e+02;
c3_0 = -2.44185984334625346e+02;
h = 5e-47;
t = 5.482e-40;


LQC Bianchi IX- 12 Initial Conditions
p1_0 = 4.769976610040370e+29;
p2_0 = 4.454851282085210e+22;
p3_0 = 7.486142946432510e+33;
c1_0 = -1258176952189.760341218777705746478313;
c2_0 = 7.277017823366510e+18;
c3_0 = -9.653259994732720e+07;
h = 5e-47;
t = 7.378e-40;
```

A.2 Numerical Solutions

Here we have the corresponding numerical plots for the Bianchi IX runs.

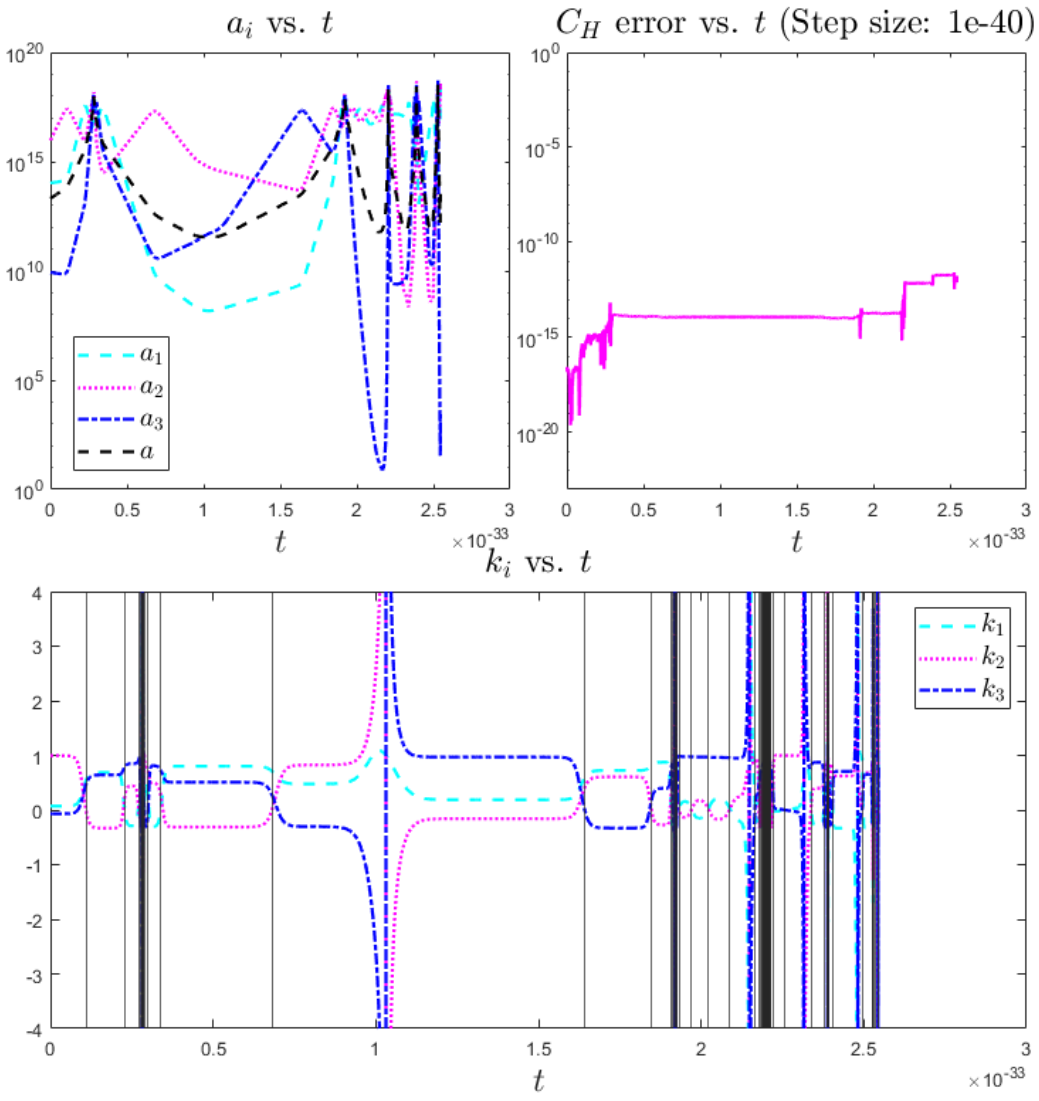

Figure A.1: This is the LQC Bianchi IX- 1 plot.

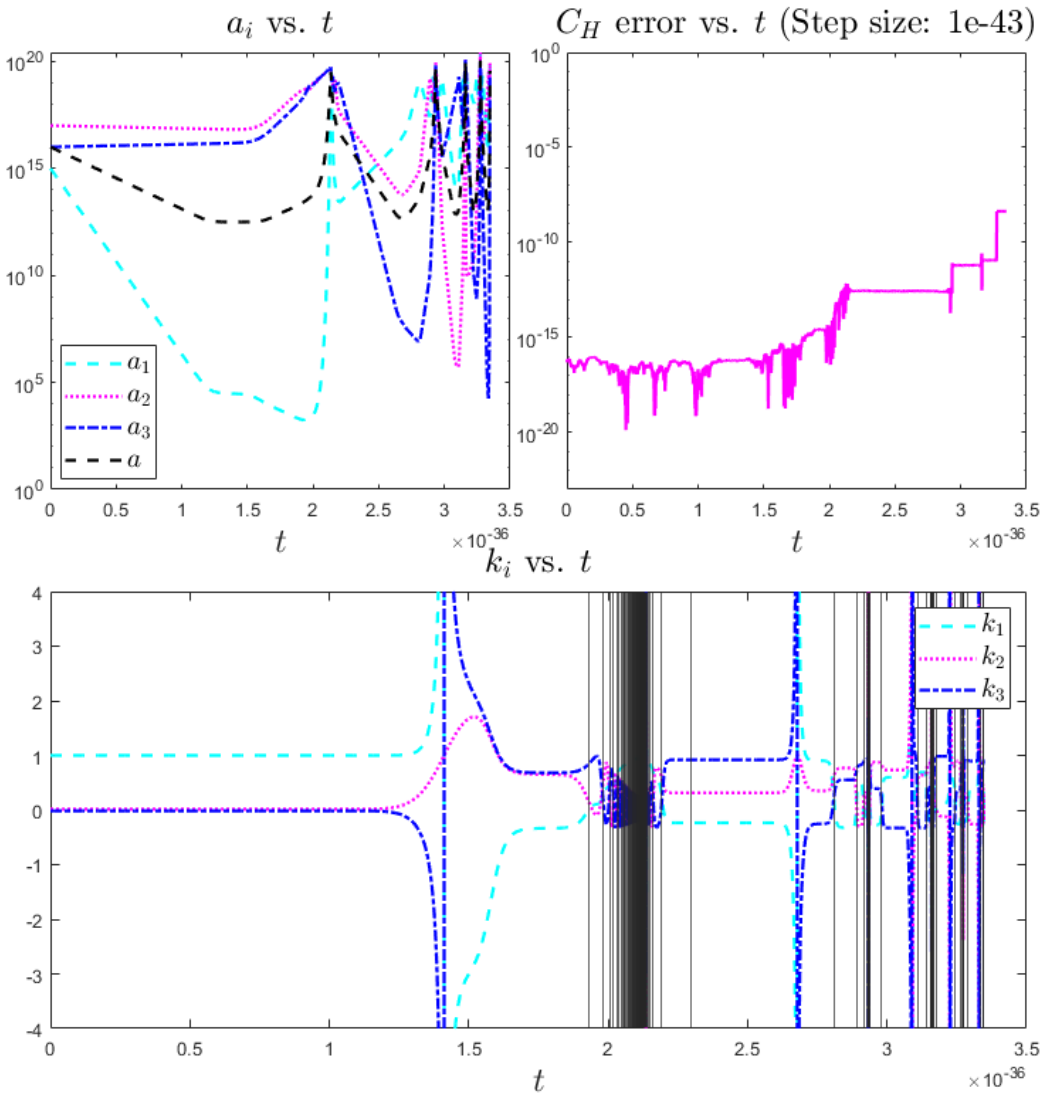

Figure A.2: This is the LQC Bianchi IX- 2 plot.

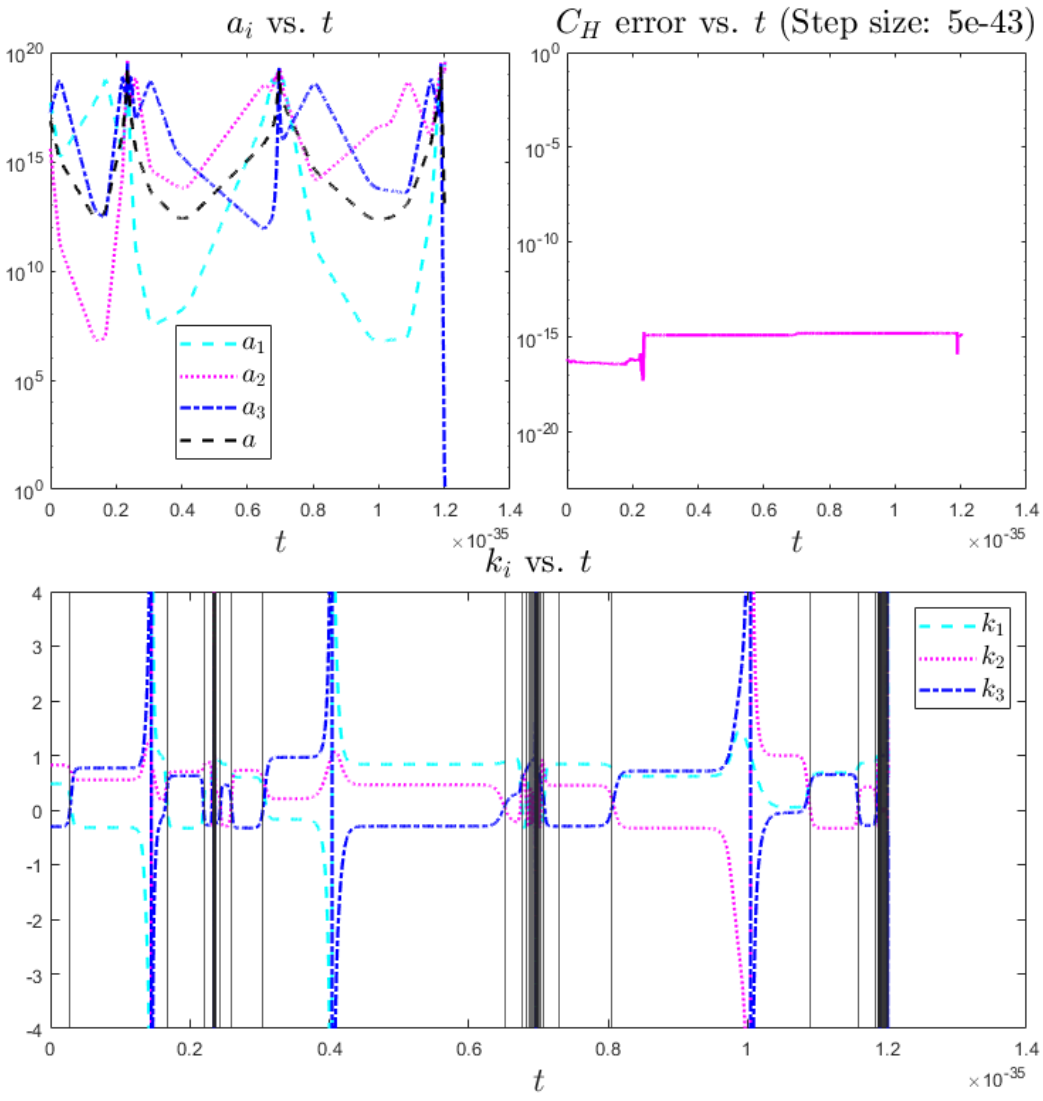

Figure A.3: This is the LQC Bianchi IX- 3 plot.

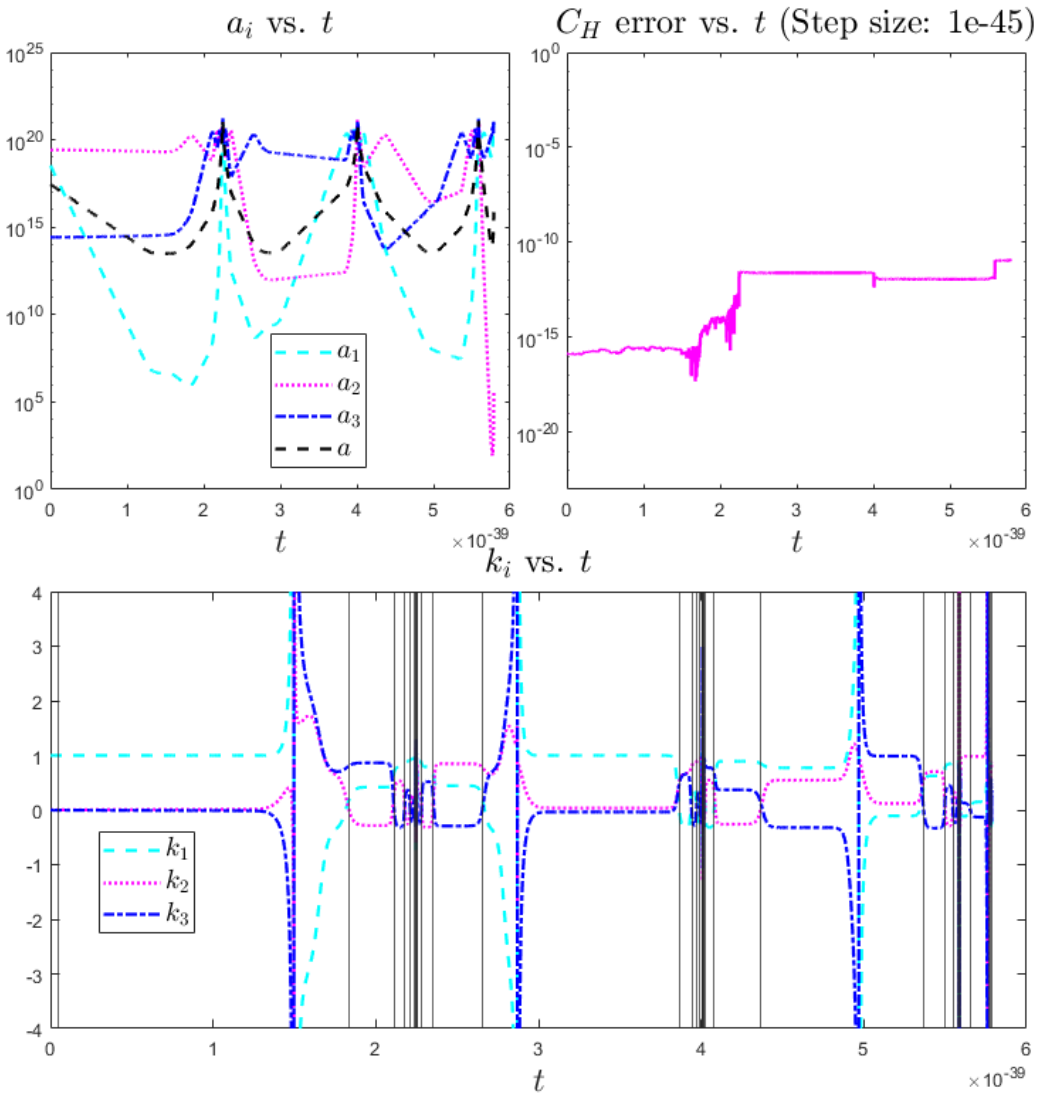

Figure A.4: This is the LQC Bianchi IX- 4 plot.

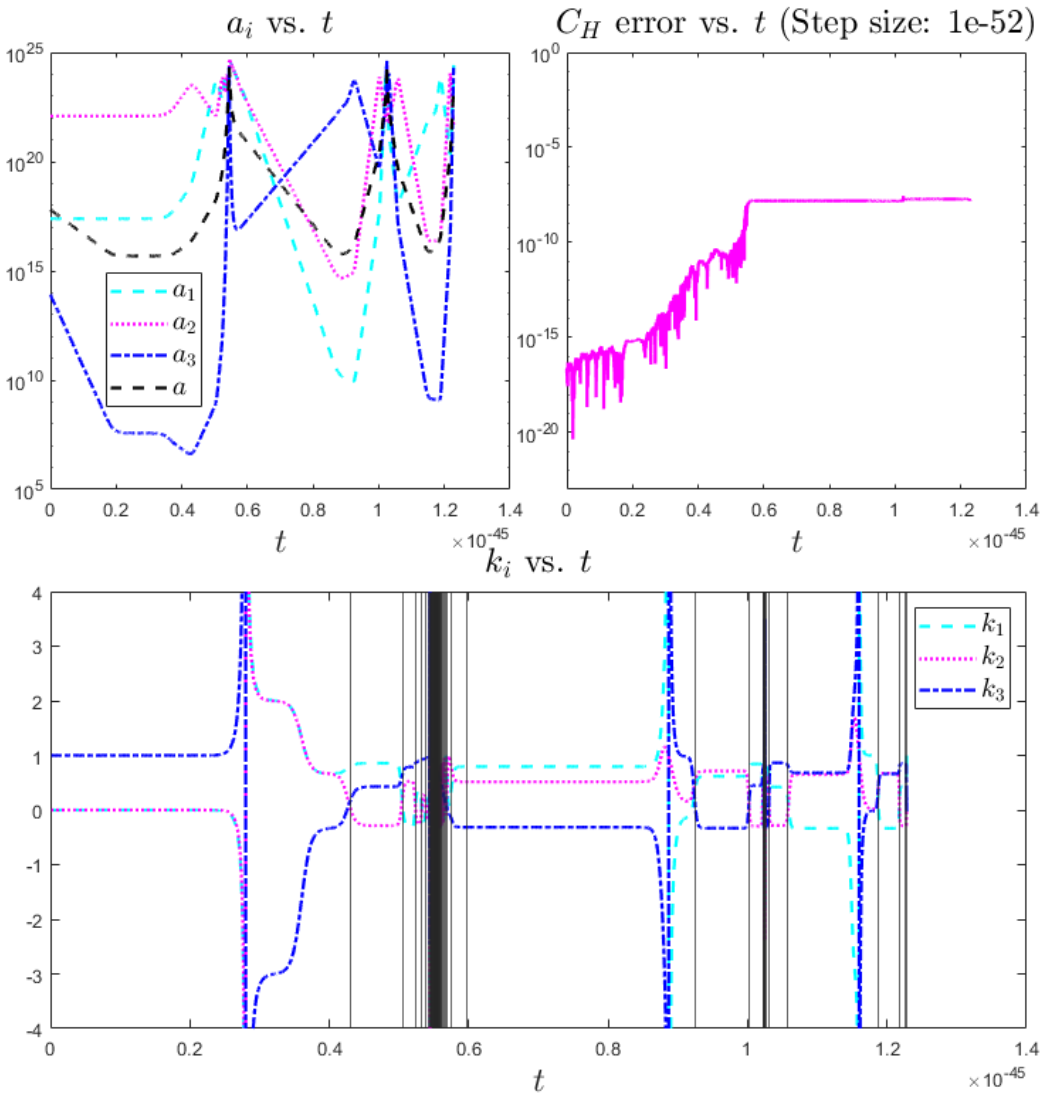

Figure A.5: This is the LQC Bianchi IX- 5 plot.

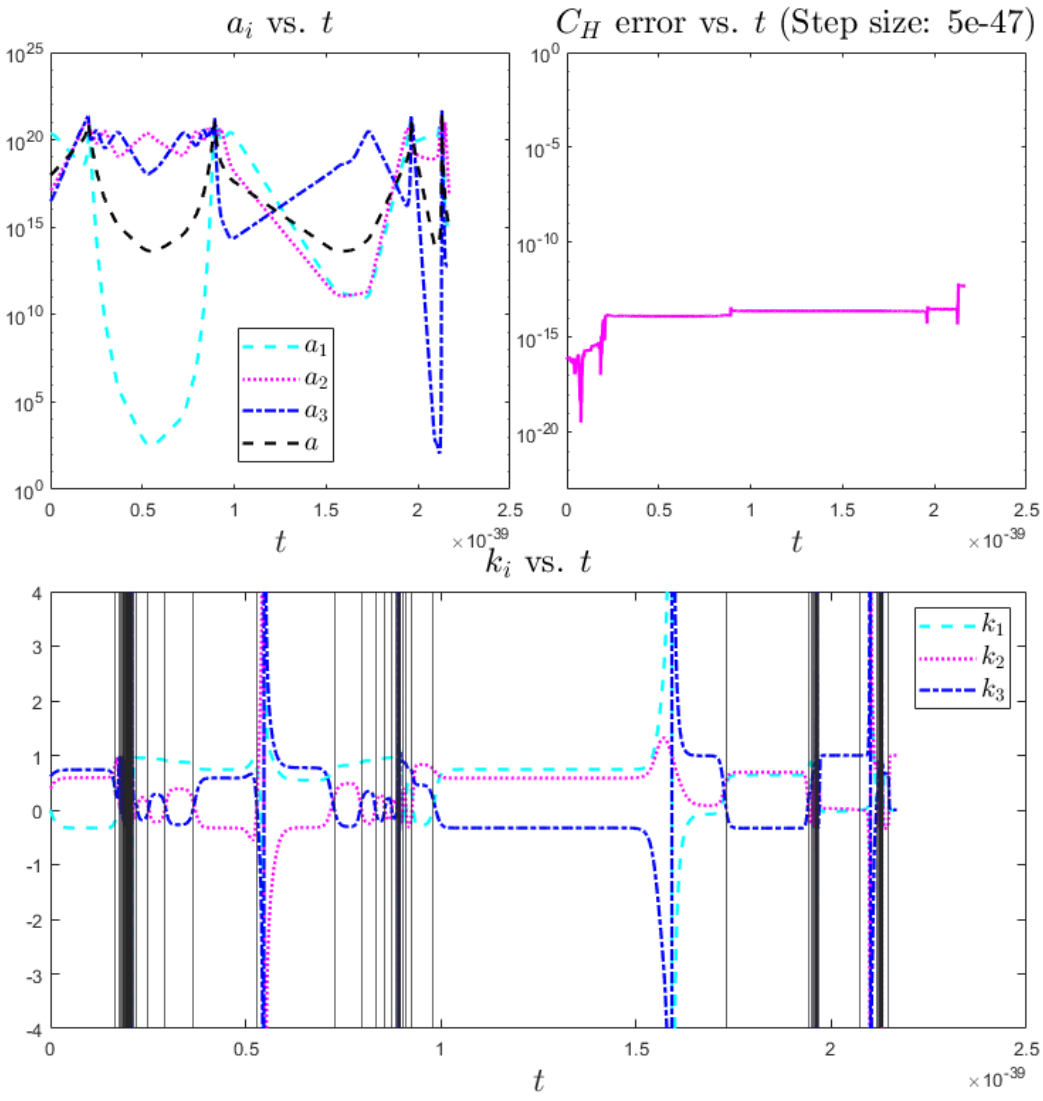

Figure A.6: This is the LQC Bianchi IX- 6 plot.

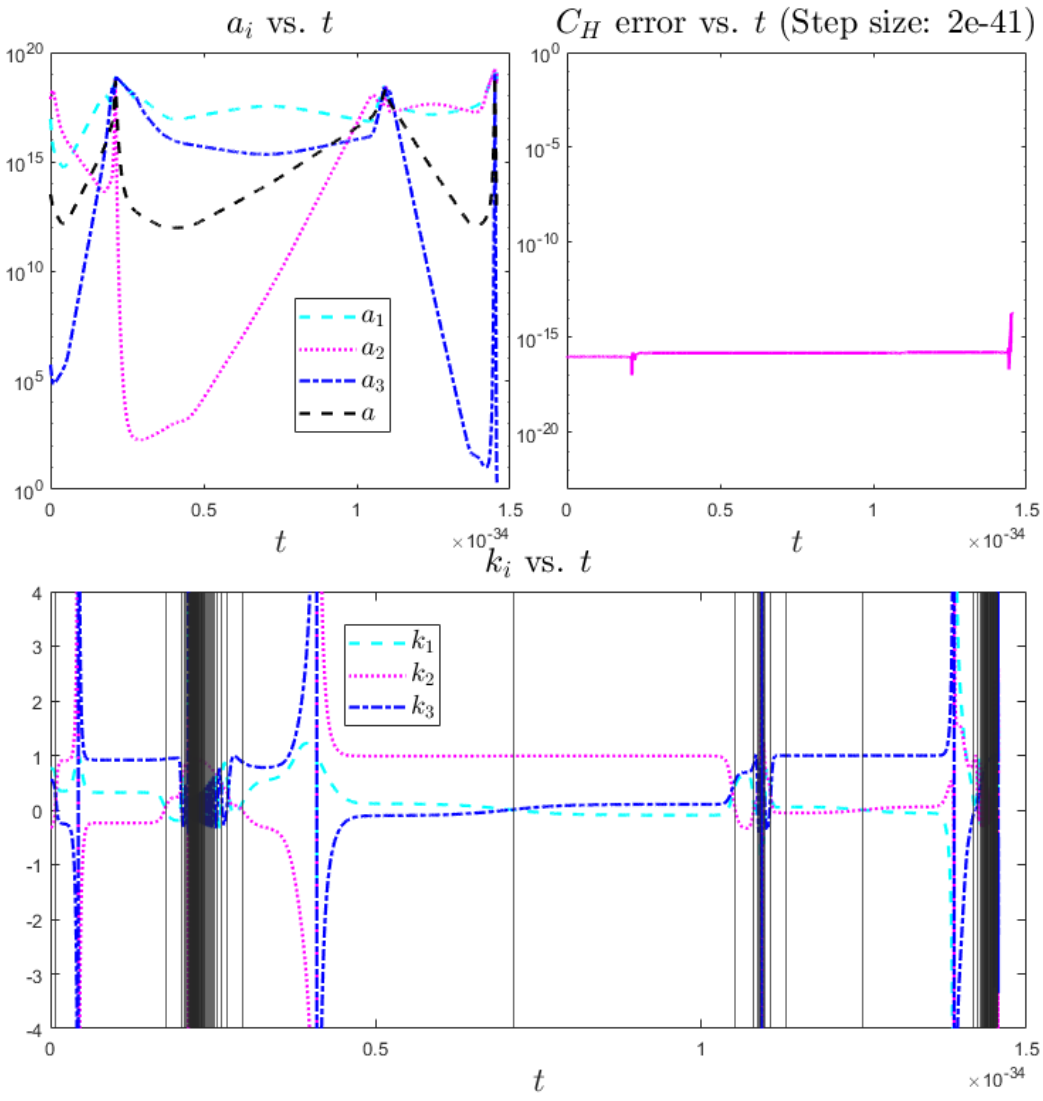

Figure A.7: This is the LQC Bianchi IX- 7 plot.

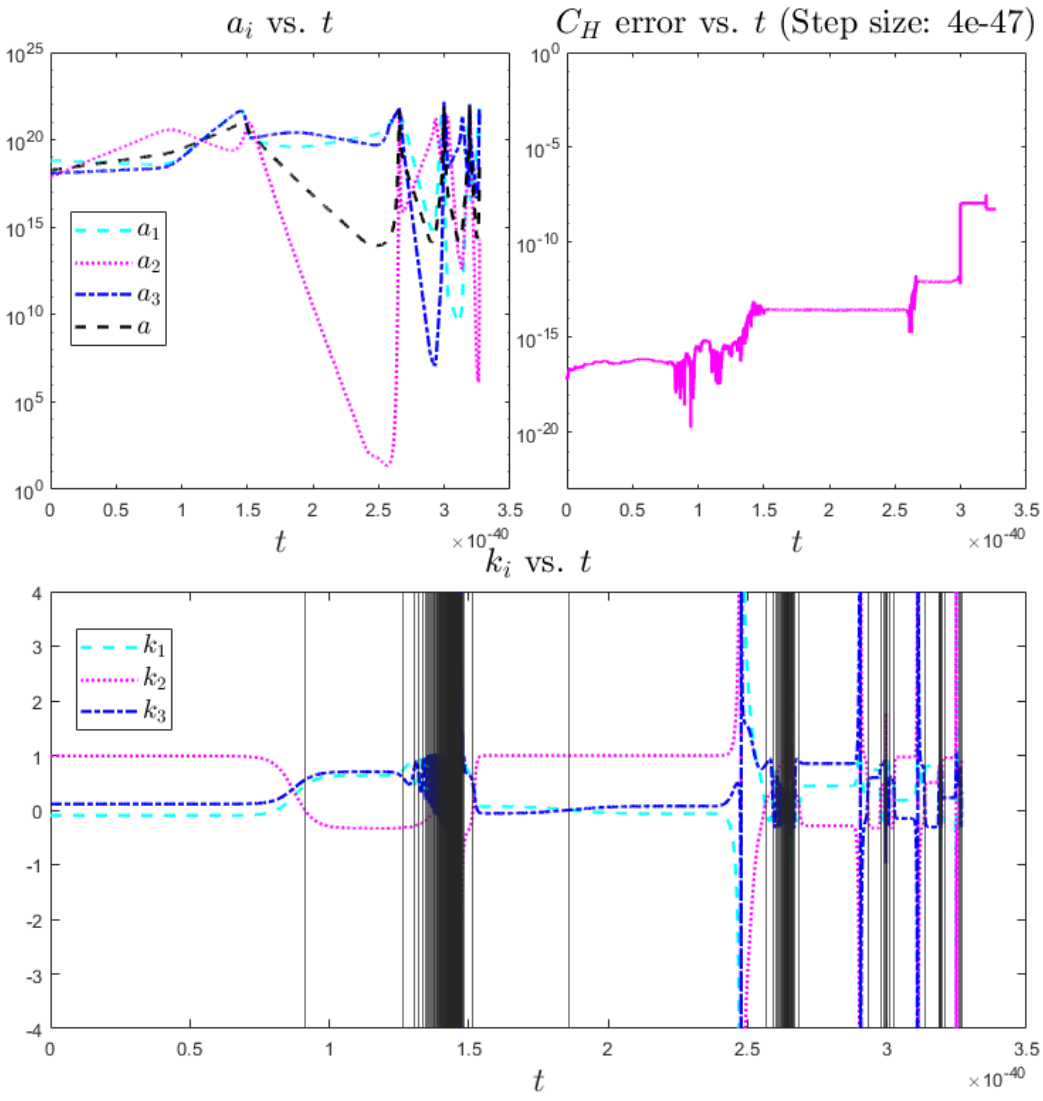

Figure A.8: This is the LQC Bianchi IX- 8 plot

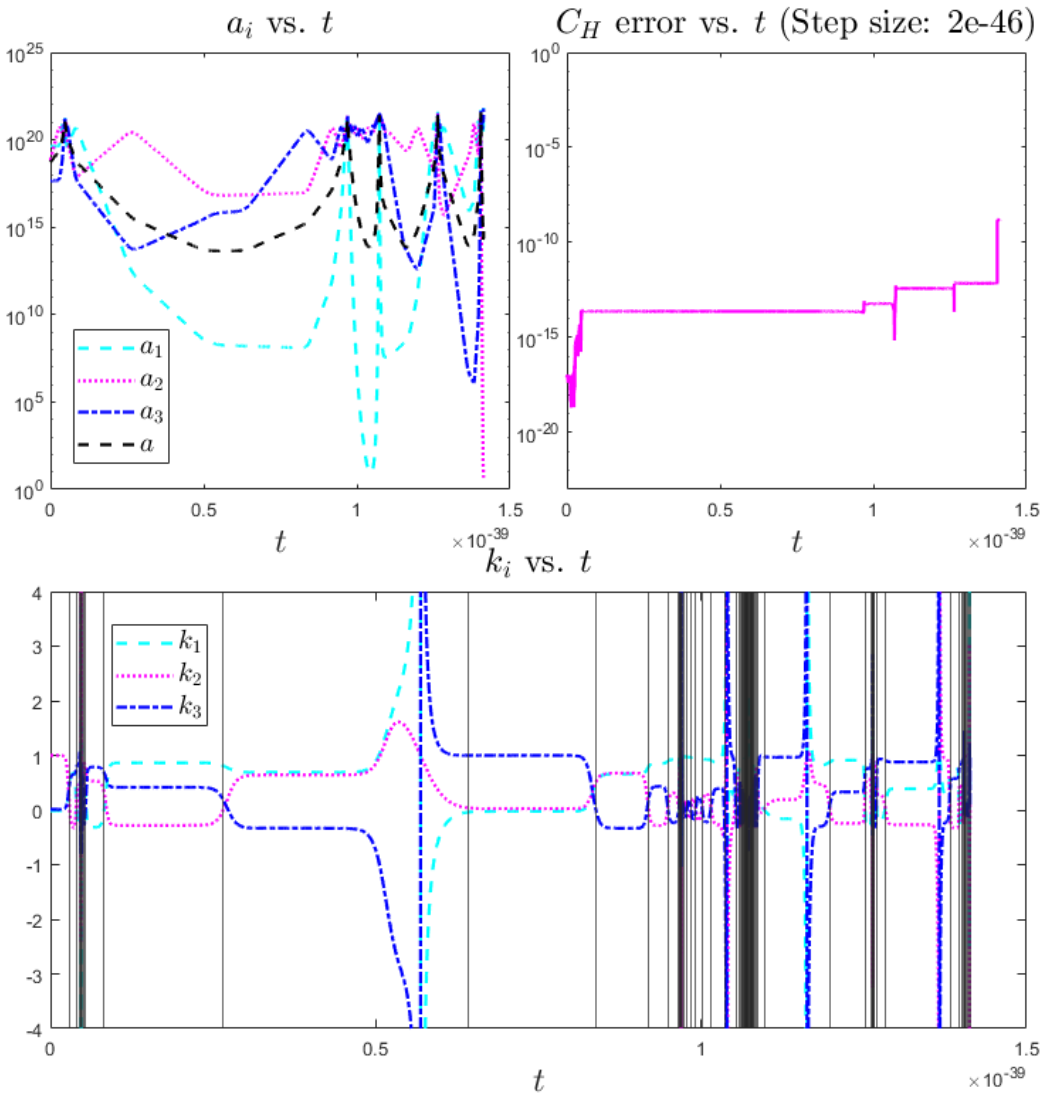

Figure A.9: This is the LQC Bianchi IX- 9 plot

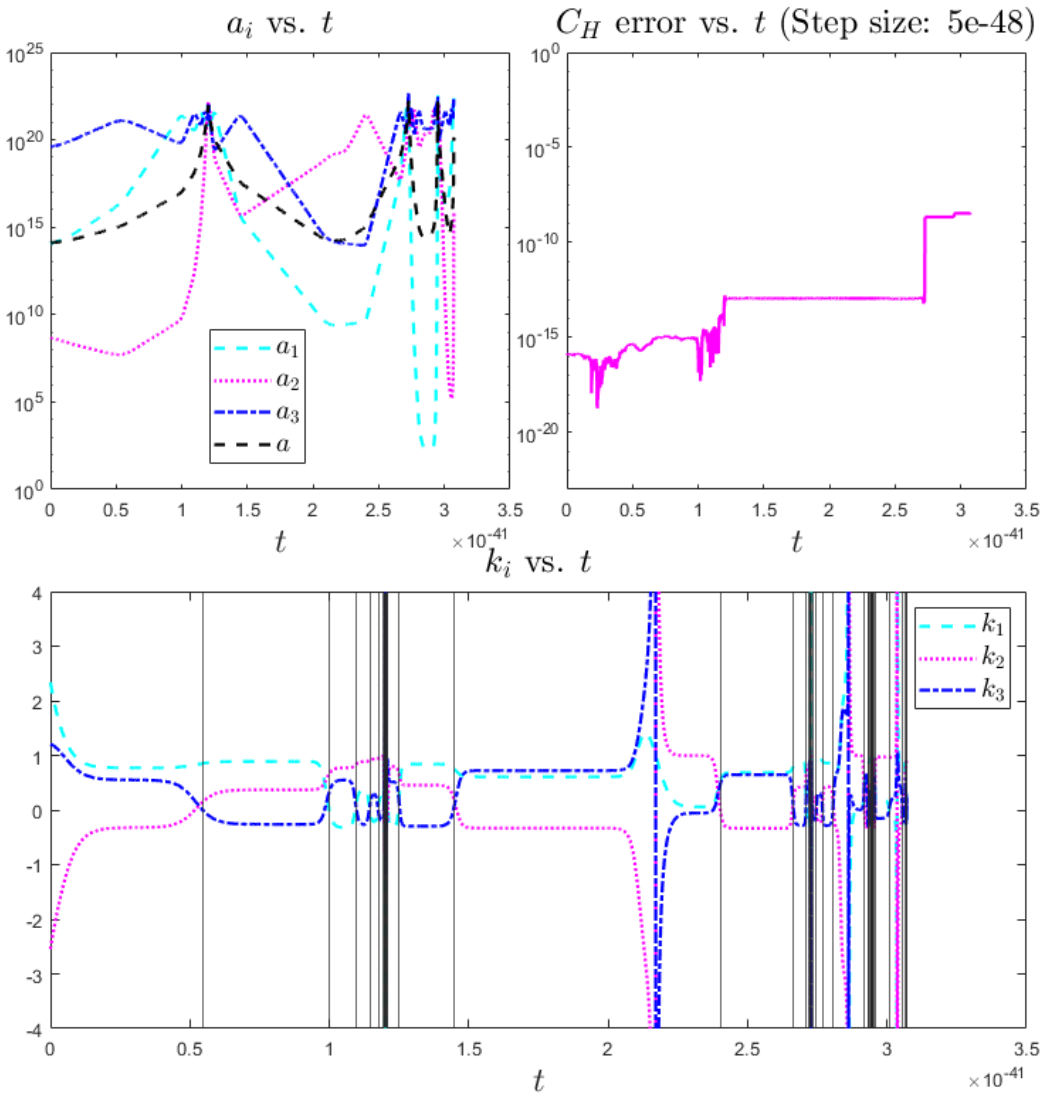

Figure A.10: This is the LQC Bianchi IX- 10 plot

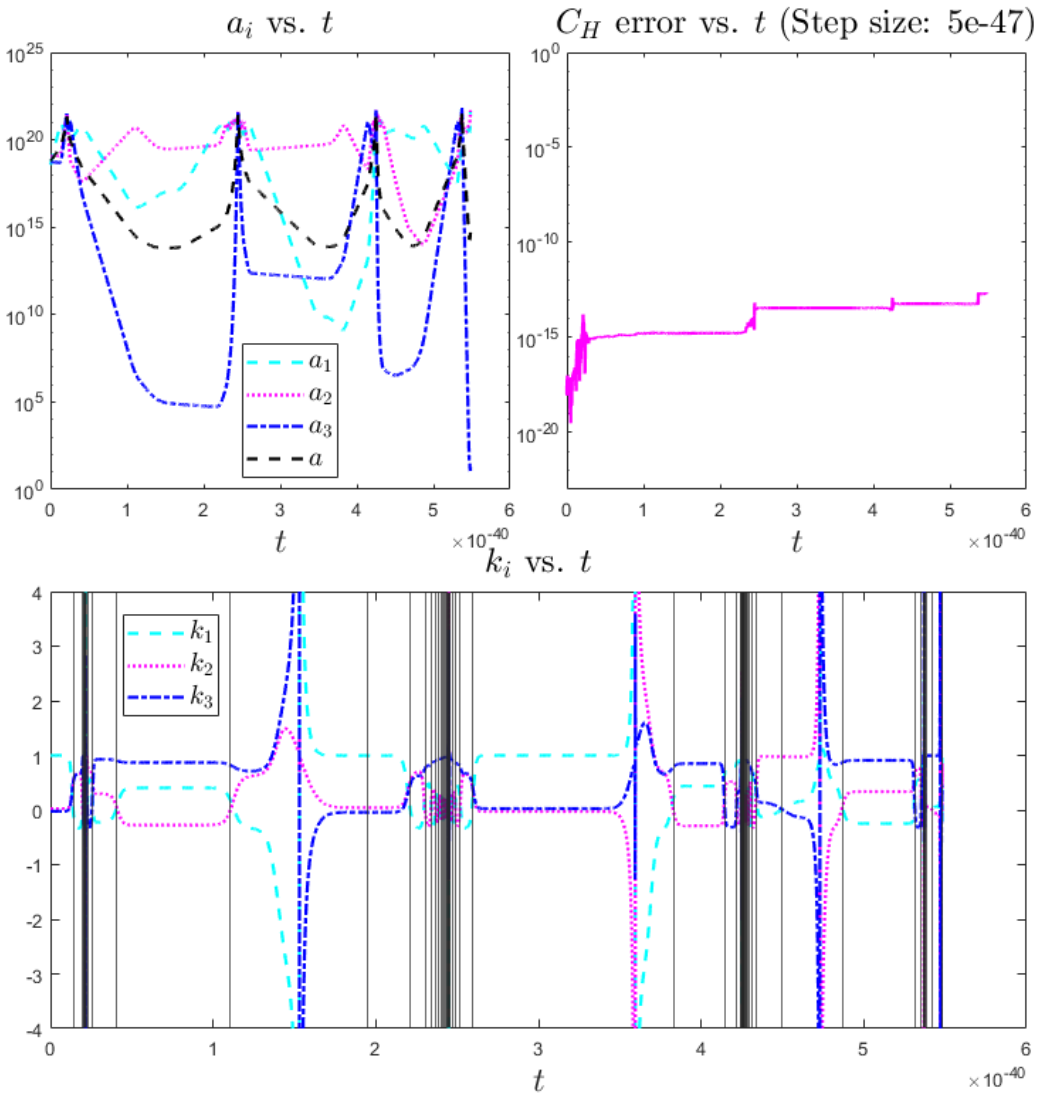

Figure A.11: This is the LQC Bianchi IX- 11 plot

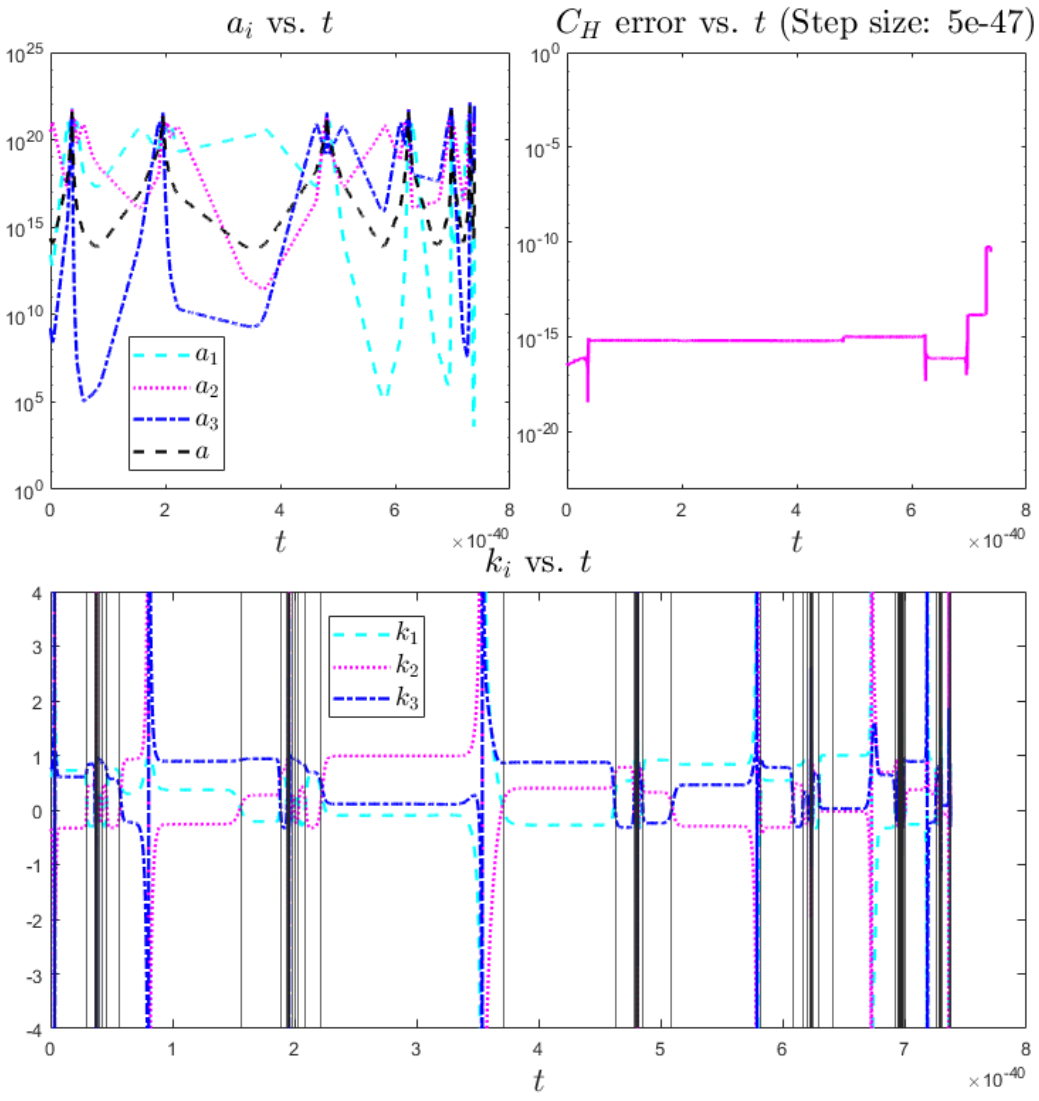

Figure A.12: This is the LQC Bianchi IX- 12 plot

Appendix B

The 12 Altered Bianchi IX Runs with LQC Effects

B.1 Initial Conditions

Here are the initial conditions for the 12 Bianchi IX runs with slightly altered initial conditions.

```
LQC Bianchi IX Altered- 1 Initial Conditions
//1
p1_0 = 8.039999996999999e+25;
p2_0 = 9e23;
p3_0 = 1e30;
c1_0 = 30000000.011193445415417708766479674721;
c2_0 = 3.72232352997251587e+10;
c3_0 = -2.24994857268156738e+03;
h = 1e-40;
t = 2.2749e-33;


LQC Bianchi IX Altered- 2 Initial Conditions
p1_0 = 1.000000003e33;
p2_0 = 1e31;
p3_0 = 1e32;
c1_0 = -19999.999940033021524433100810465697 22;
c2_0 = -3e+4;
c3_0 = 2.95565295692105883e+03;
h = 1e-43;
t = 3.1879e-36;


LQC Bianchi IX Altered- 3 Initial Conditions
p1_0 = 7e32;
p2_0 = 9e34;
p3_0 = 2e33;
c1_0 = -30000.000042975775654501635333811220213;
c2_0 = -4e+2;
c3_0 = 6.63146731801925195e+03;
h = 0.5e-42;
t = 1.0773e-35;
```

```
LQC Bianchi IX Altered- 4 Initial Conditions
p1_0 = 7e33;
p2_0 = 9e32;
p3_0 = 9e37;
c1_0 = -3000000.033249606060966326921555836095 2;
c2_0 = -4e+4;
c3_0 = 3.36733074972204946e-01;
h = 1e-45;
t = 5.787e-39;


LQC Bianchi IX Altered- 5 Initial Conditions
p1_0 = 1e36;
p2_0 = 2e31;
p3_0 = 3e39;
c1_0 = -80000.001752320012201069612167185132725;
c2_0 = 3e+08;
c3_0 = -2.53378393797385857e+07;
h = 1e-52;
t = 1.1923e-45;


LQC Bianchi IX Altered- 6 Initial Conditions
p1_0 = 4e33;
p2_0 = 7e36;
p3_0 = 3e37;
c1_0 = -2998.53825634107818525748139492502747 83;
c2_0 = 3e+03;
c3_0 = 1.0947743324440406 1e+03;
h = 5e-47;
t = 2.1676e-39;


LQC Bianchi IX Altered- 7 Initial Conditions
p1_0 = 3.644407081693920e+23;
p2_0 = 4.619481809923470e+22;
p3_0 = 6.508819947785010e+34;
c1_0 = -11330329147704.516078945501911527089792;
c2_0 = 3.726944157209900e+13;
c3_0 = -45.773504837509698;
h = 2e-41;
t = 1.457e-34;


LQC Bianchi IX Altered- 8 Initial Conditions
p1_0 = 9e35;
p2_0 = 8e36;
p3_0 = 5e36;
c1_0 = -7999.999910000008405449519712488211194 1;
c2_0 = 9e+03;
c3_0 = 1.60000136740636453e+03;
h = 4e-47;
t = 3.0092e-40;


LQC Bianchi IX Altered- 9 Initial Conditions
p1_0 = 3e36;
p2_0 = 2e37;
```

```
p3_0 = 3e38;
c1_0 = -700.000001947771692118706721372967595554;
c2_0 = 8e+03;
c3_0 = 7.11326372614312697e+00;
h = 2e-46;
t = 1.4134e-39;


LQC Bianchi IX Altered- 10 Initial Conditions
p1_0 = 2.026722231822400e+28;
p2_0 = 4.854478714956990e+33;
p3_0 = 5.944594117494380e+22;
c1_0 = 6.704487157259680e+13;
c2_0 = -34788654.665121574744595413760539662295;
c3_0 = 1.765985887358840e+19;
h = 5e-48;
t = 3.0105e-41;


LQC Bianchi IX Altered- 11 Initial Conditions
p1_0 = 6e37;
p2_0 = 2e+37;
p3_0 = 4e37;
c1_0 = 6999.998767346756245839493748888196477 2;
c2_0 = 5e+02;
c3_0 = -2.44185983334625346e+02;
h = 5e-47;
t = 3.28e-40;


LQC Bianchi IX Altered- 12 Initial Conditions
p1_0 = 4.769976610040370e+29;
p2_0 = 4.454851282085210e+22;
p3_0 = 7.486142946432510e+33;
c1_0 = -1258176951757.882540046203382289685 5687;
c2_0 = 7.277017823366510e+18;
c3_0 = -9.653259998732720e+07;
h = 5e-47;
t = 6.835e-40;
```

B.2 Numerical Solutions

Here we have the corresponding numerical comparing the original and slightly altered Bianchi IX runs.

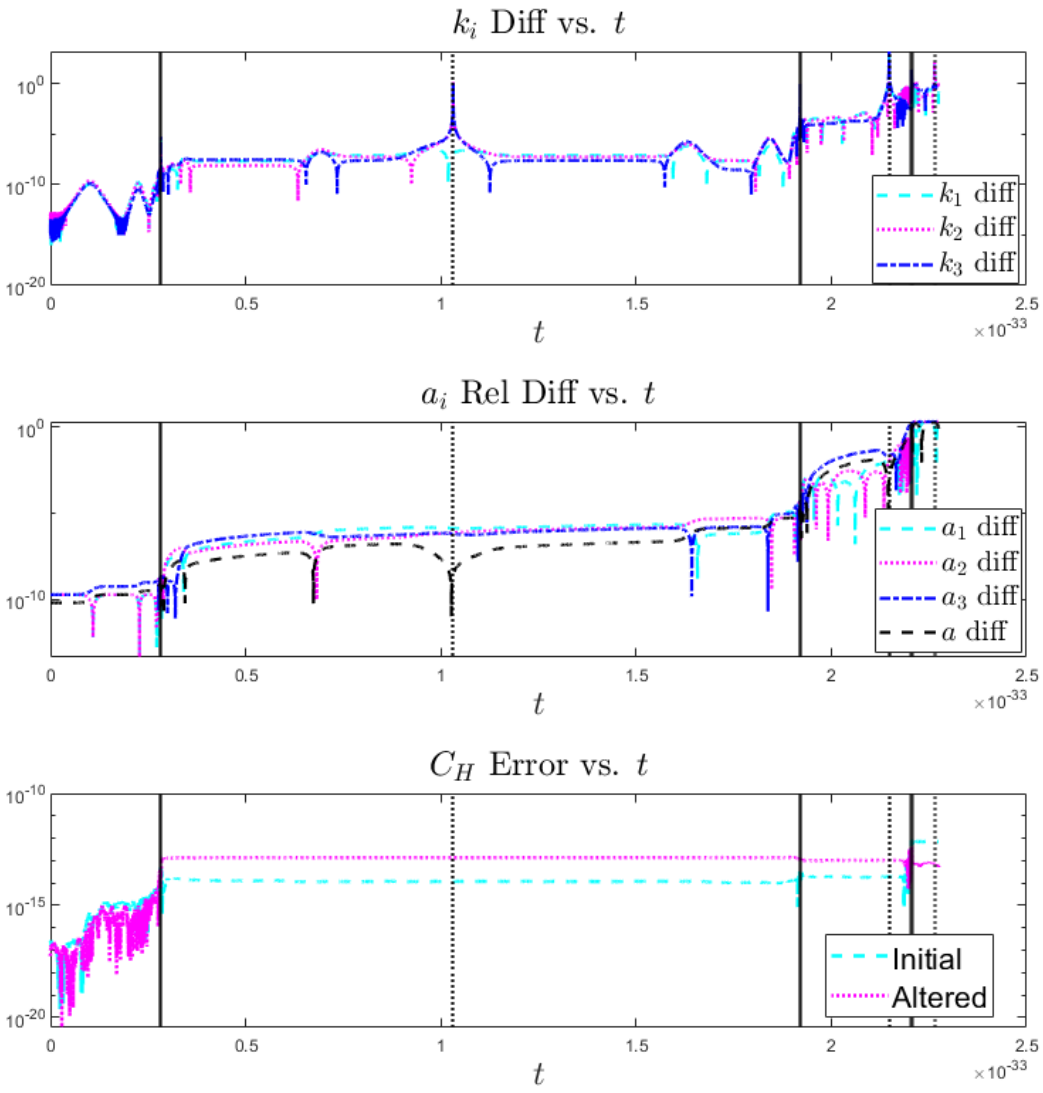

Figure B.1: This is the LQC Bianchi IX- 1 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

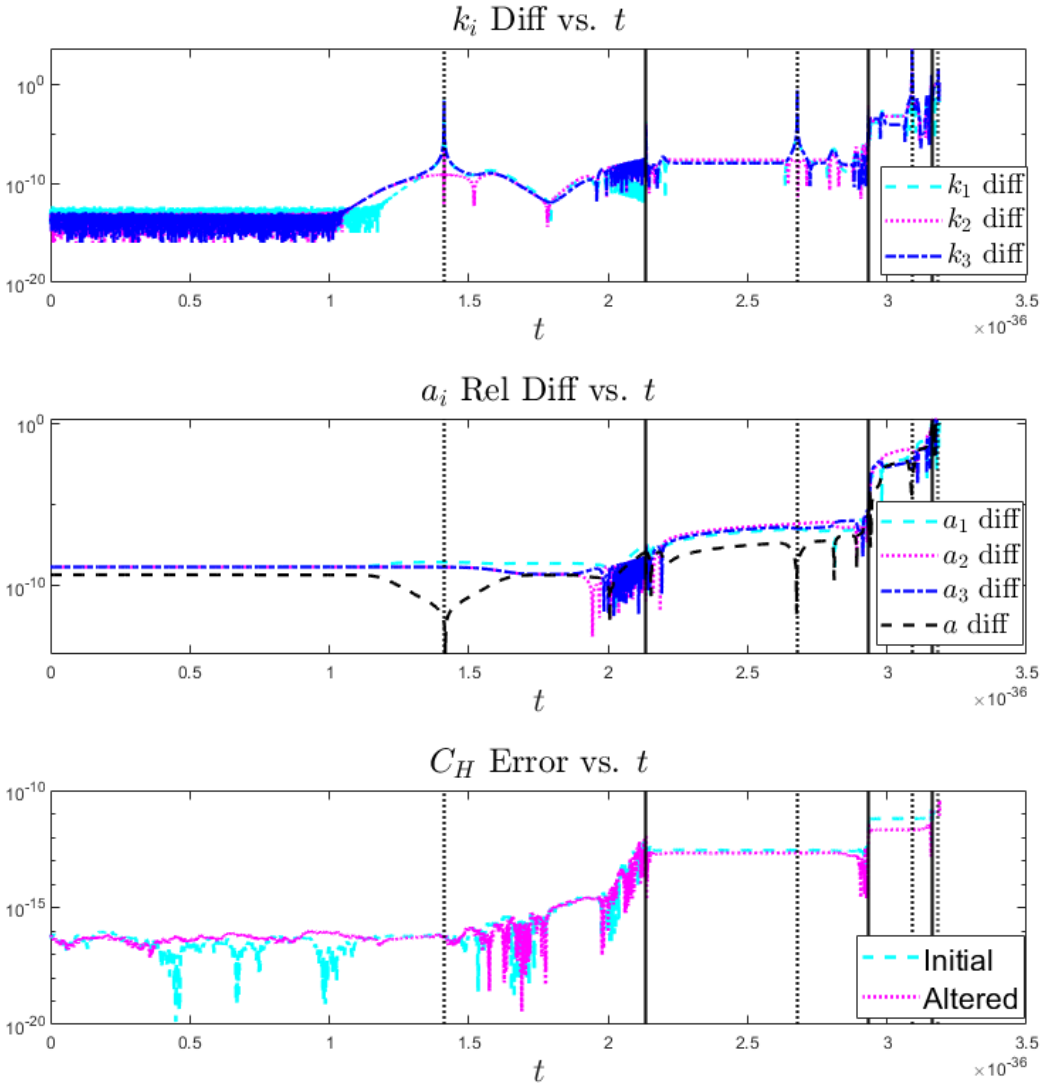

Figure B.2: This is the LQC Bianchi IX- 2 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

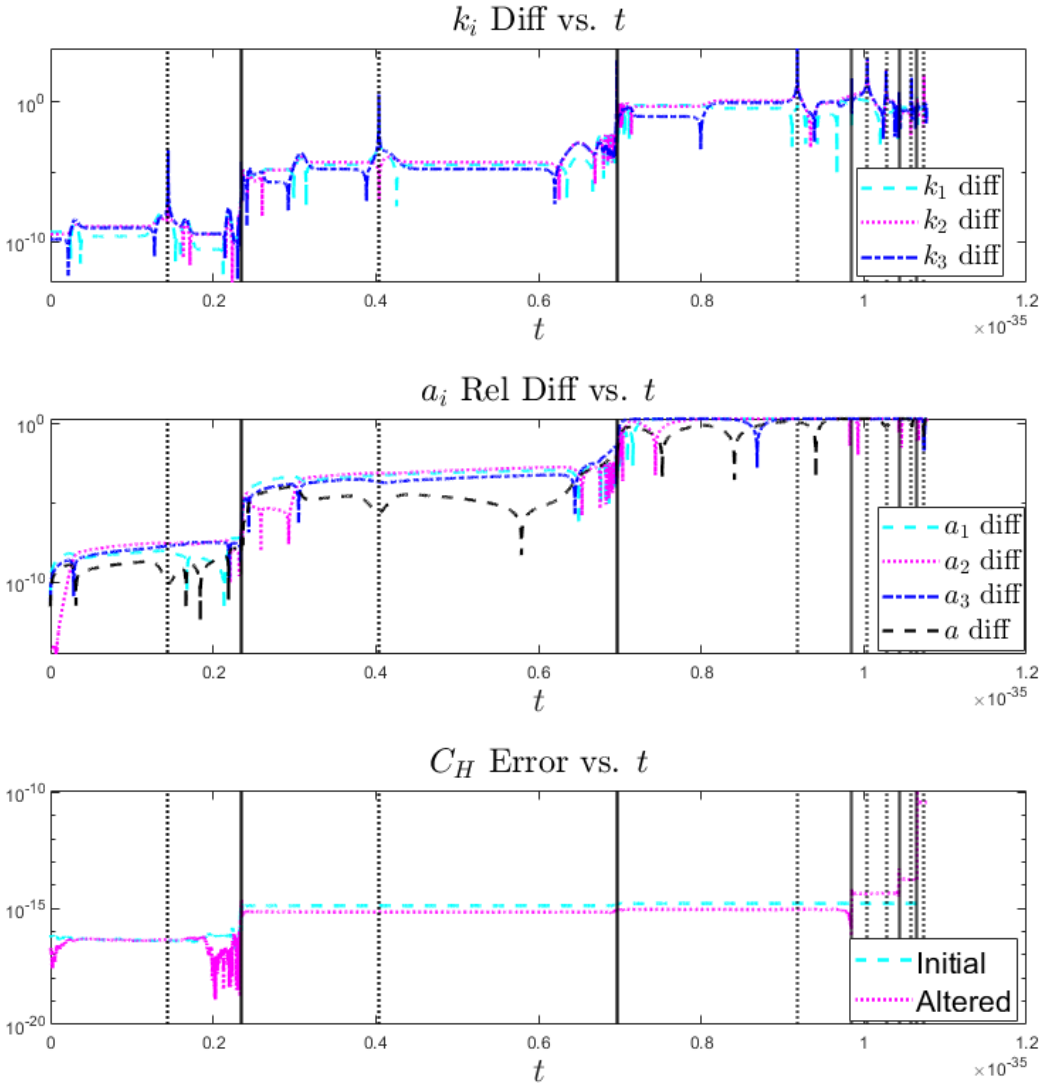

Figure B.3: This is the LQC Bianchi IX- 3 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

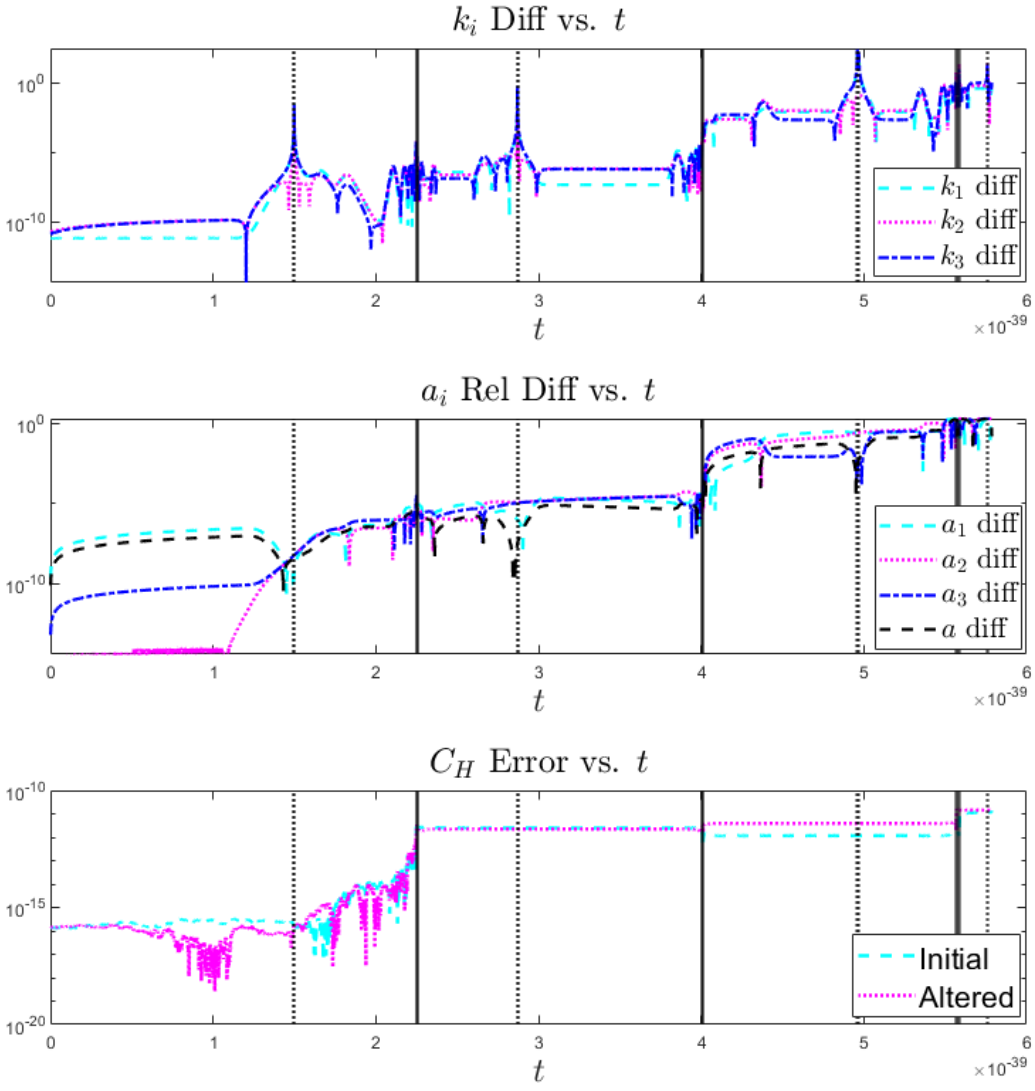

Figure B.4: This is the LQC Bianchi IX- 4 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

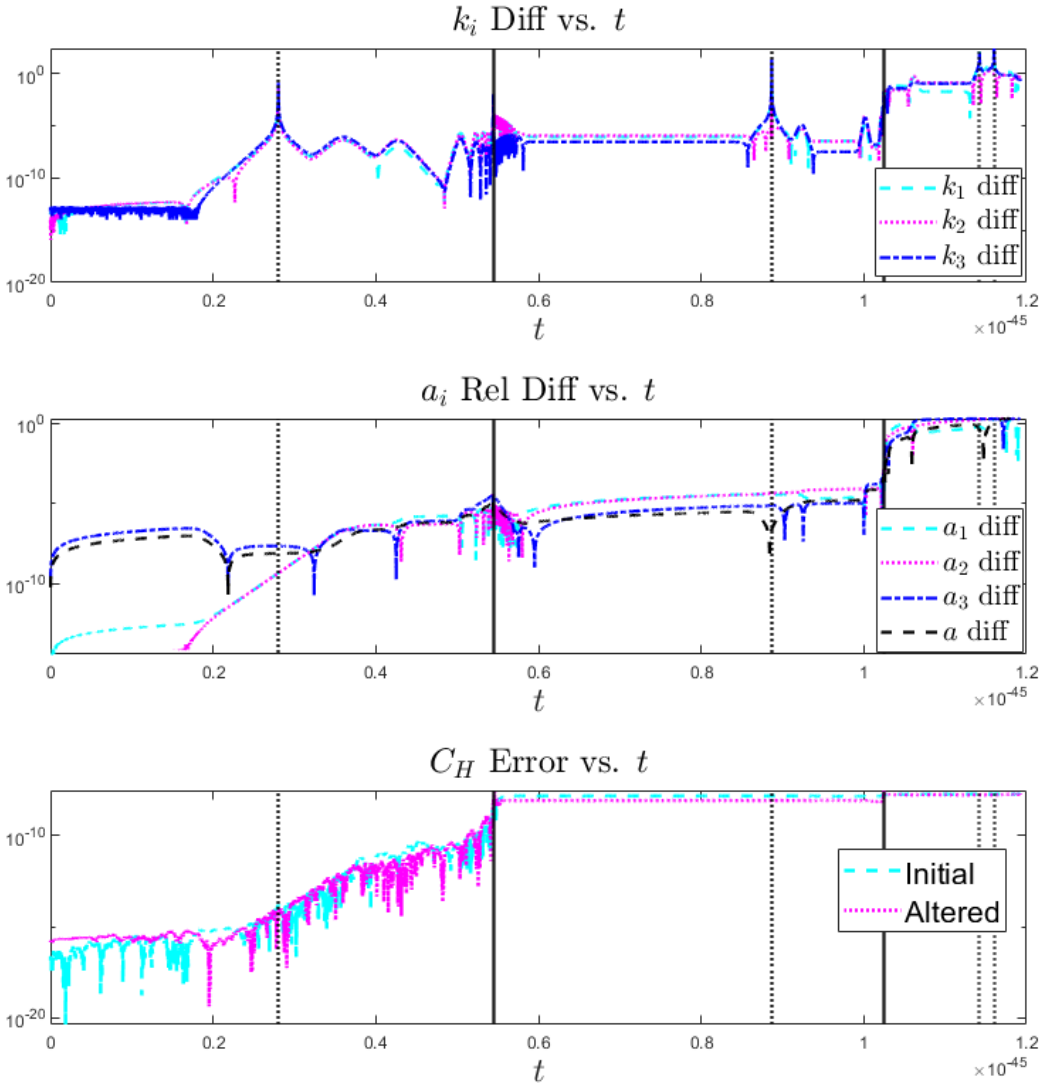

Figure B.5: This is the LQC Bianchi IX- 5 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

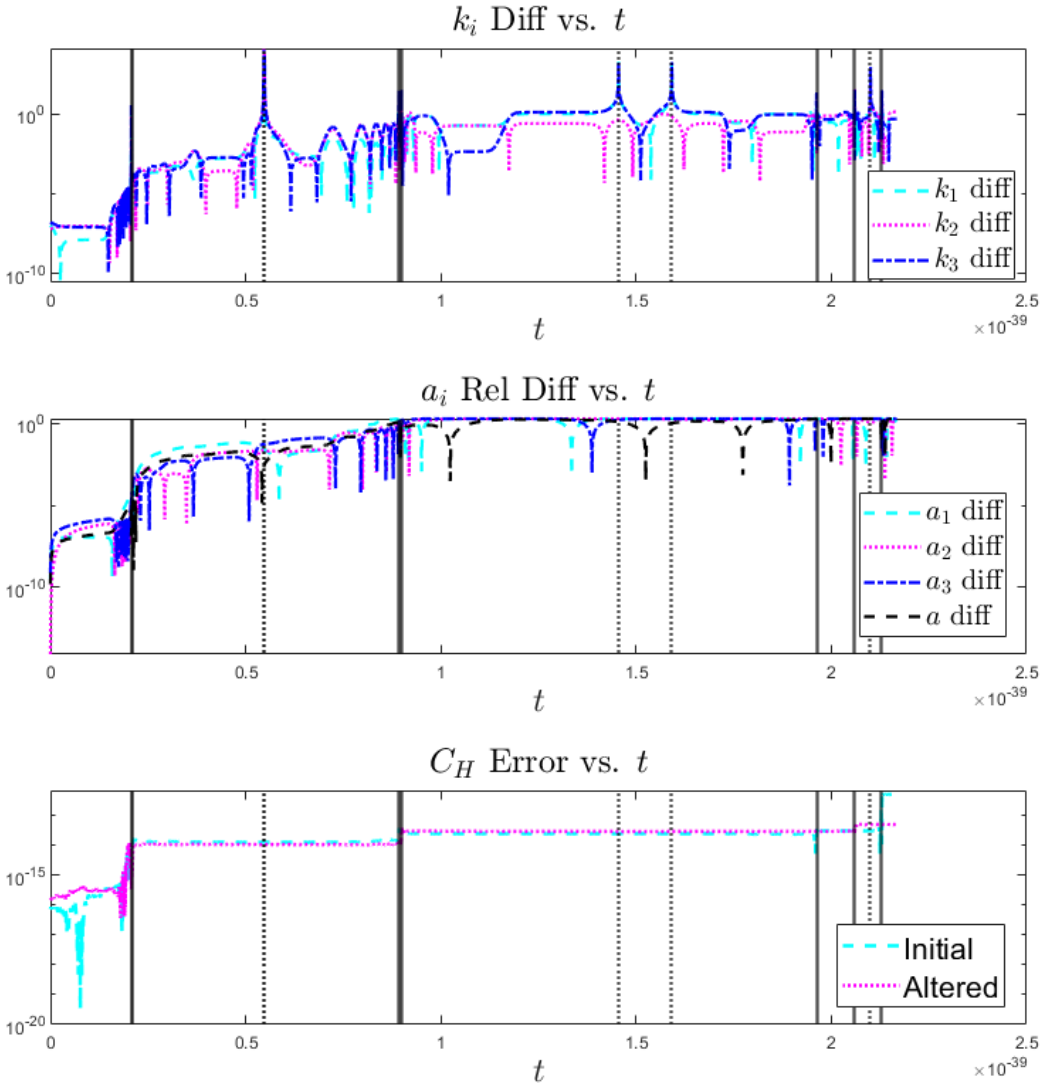

Figure B.6: This is the LQC Bianchi IX- 6 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

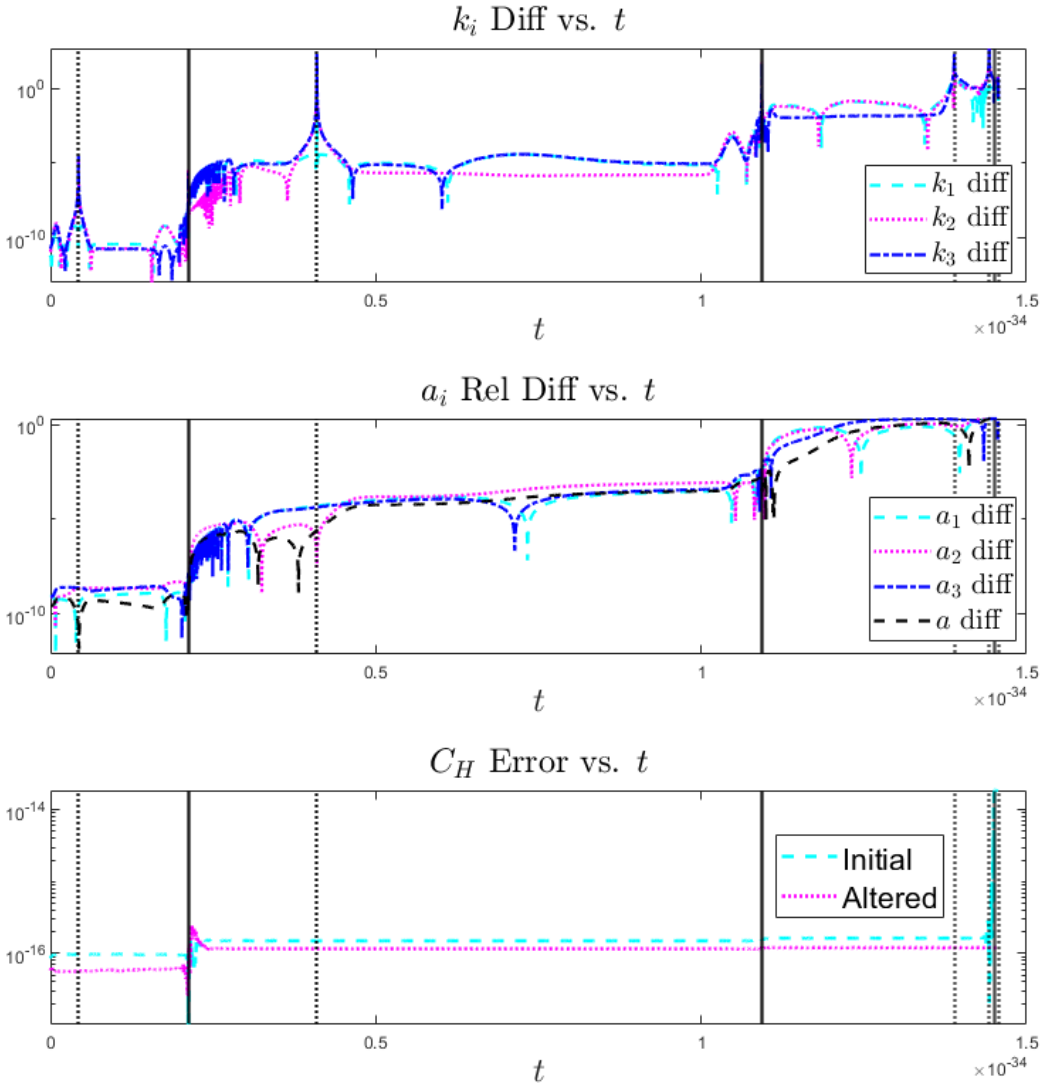

Figure B.7: This is the LQC Bianchi IX- 7 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

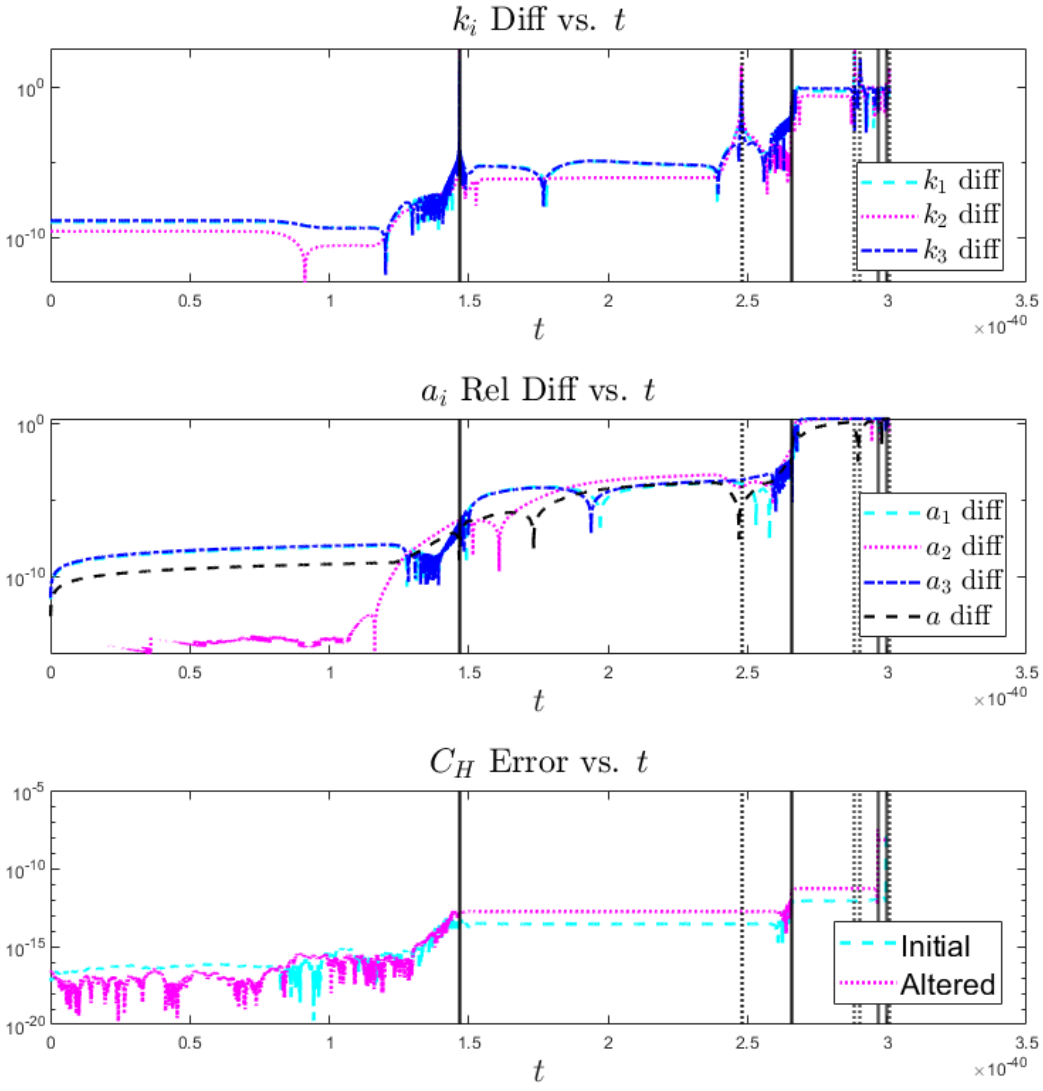

Figure B.8: This is the LQC Bianchi IX- 8 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

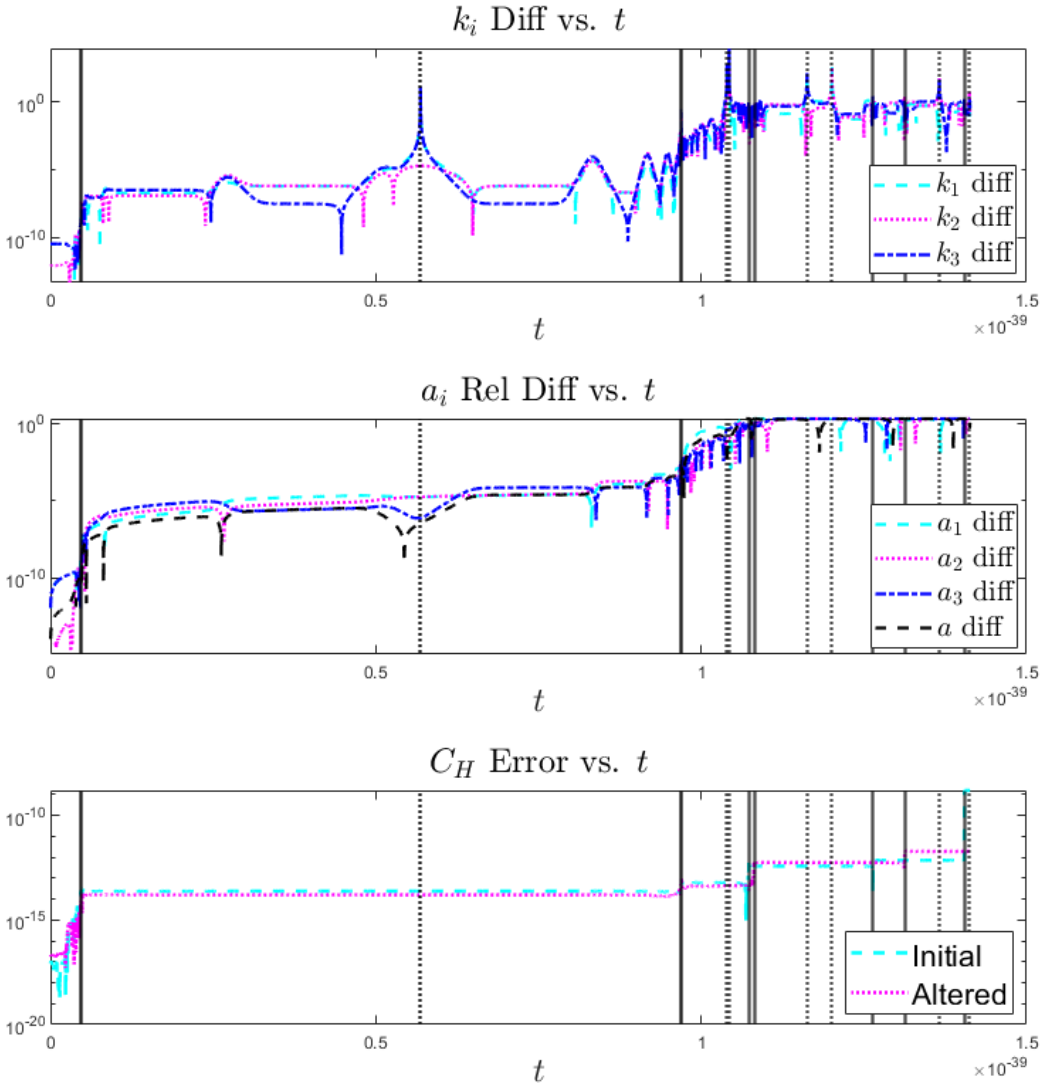

Figure B.9: This is the LQC Bianchi IX- 9 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

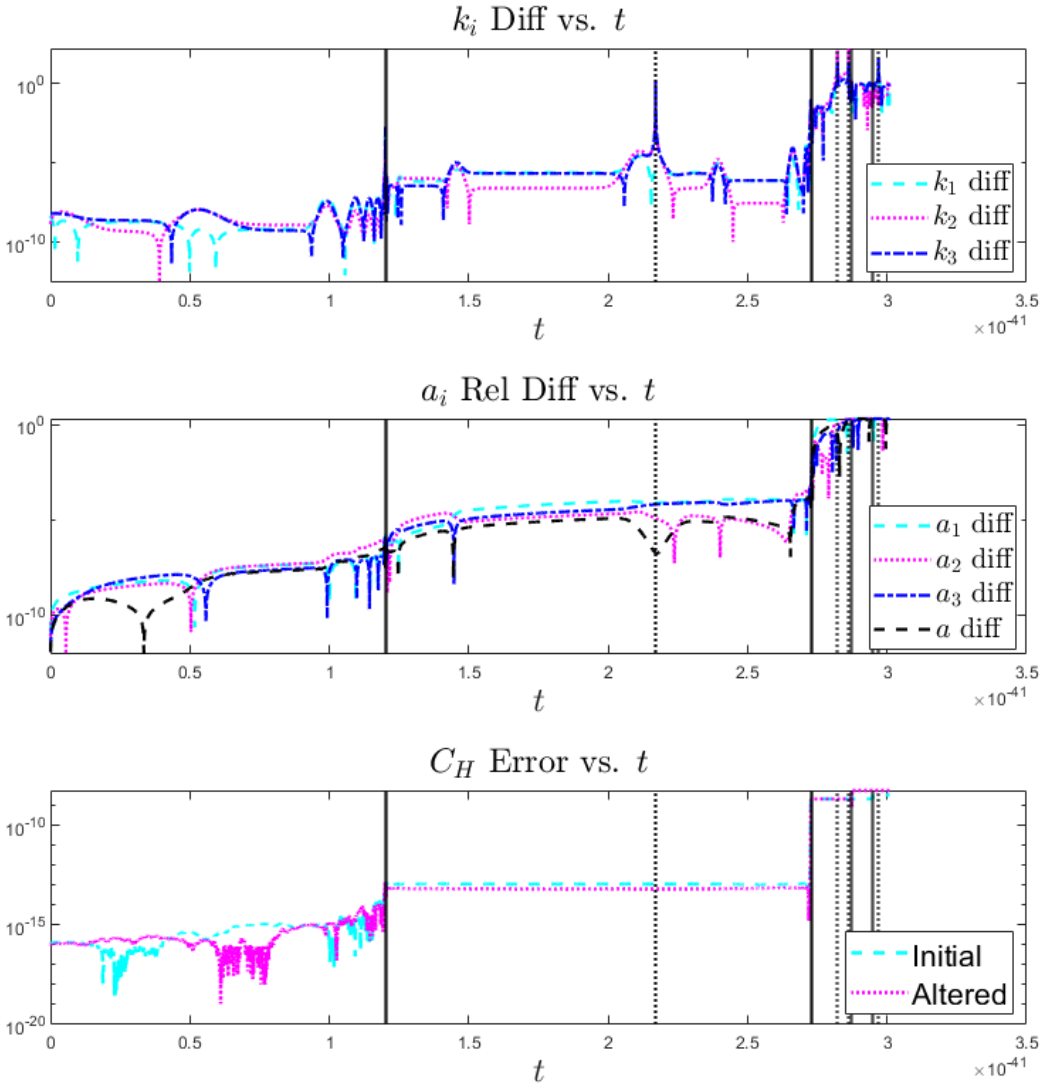

Figure B.10: This is the LQC Bianchi IX- 10 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

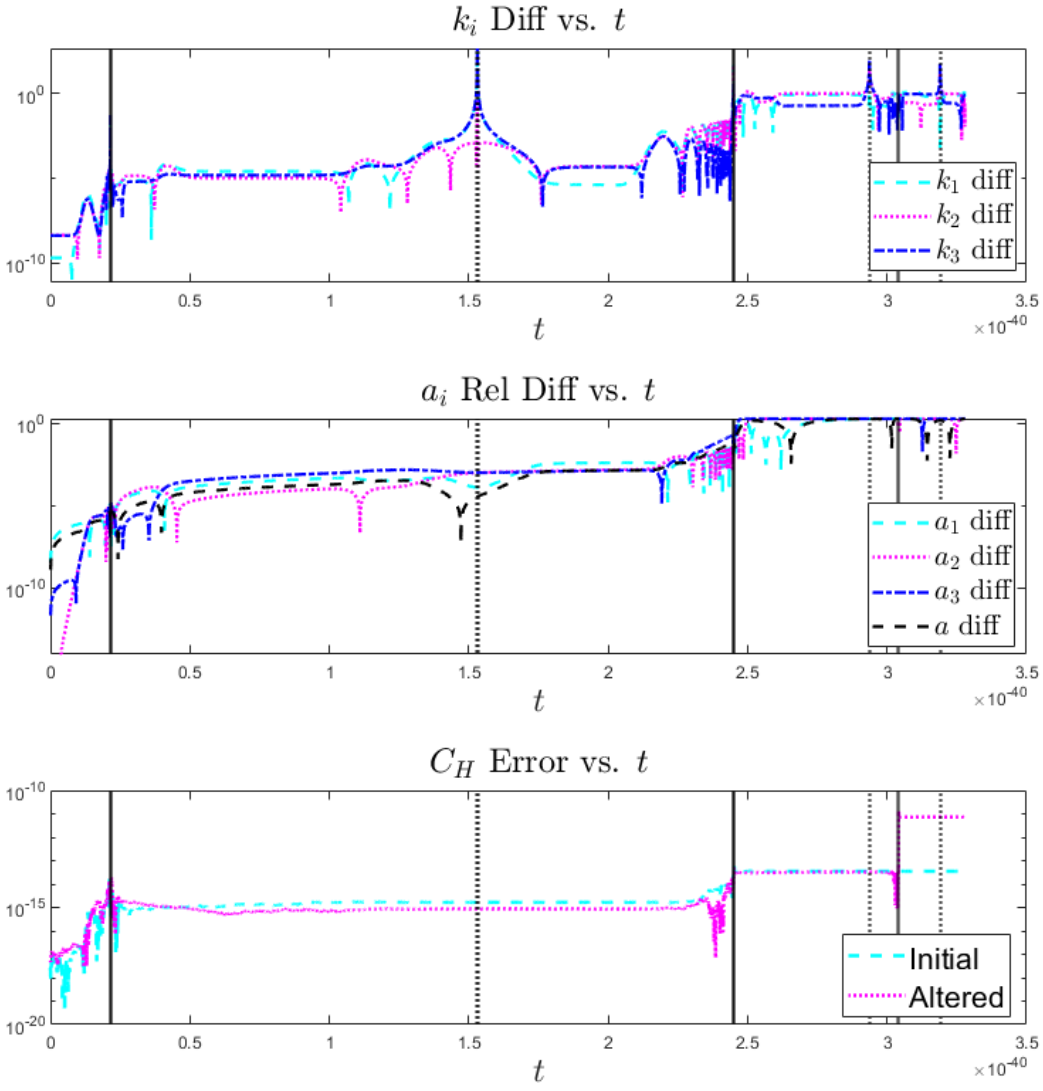

Figure B.11: This is the LQC Bianchi IX- 11 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

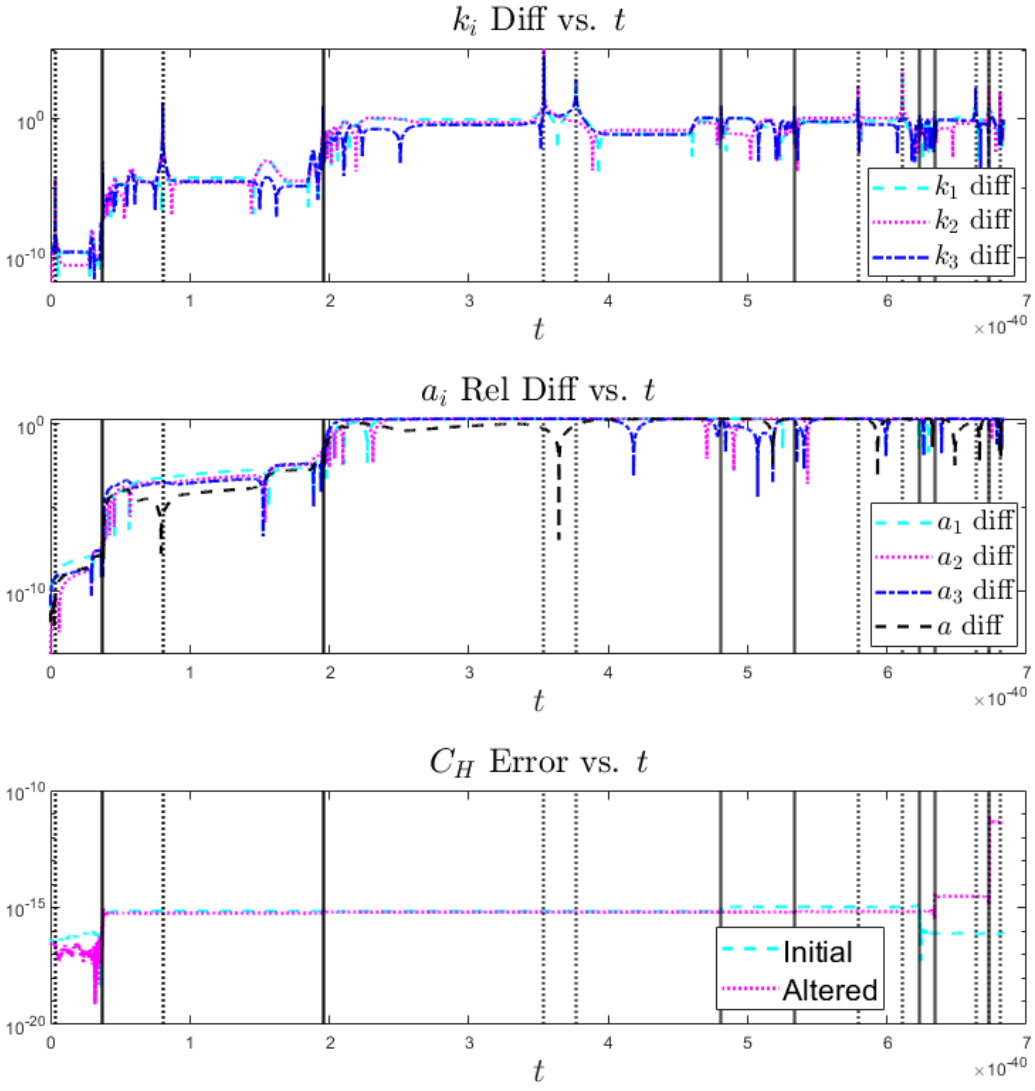

Figure B.12: This is the LQC Bianchi IX- 12 comparison plot. The top subplot displays the difference in k_i values for the original and altered solution vs. time. The middle subplot displays $|2(a_i - \tilde{a}_i)/(a_i + \tilde{a}_i)|$ vs time. Here $\tilde{a}$ refers to the scale factors from the slightly altered solutions. Vertical solid lines are shown when there is a recollapse in either the original and altered solution. Vertical dotted lines are shown when there is an LQG bounce in either the original or altered solution.

Appendix C

Kasner Transitions for High Precision Run

Here, all 57 Kasner transitions between the first and last recollapse in Figure 5.14 are tested. Note that both the forward and backward transition rule are tested for each Kasner transition.

C.1 First Recollapse to Recollapse

1.

Maximum a_2: k_i values taken at time: 1.268779e-40 and $\bar{k}_i$ values taken at time: 1.273328e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.816016	-0.469332	0.653316
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-1.999594	7.651797	-4.652203
Numerical	-0.198277	0.992546	0.205732
Percent Error	90.084113%	87.028594%	104.422247%

Table C.1: First Recollapse to Recollapse- Forward Transition Rules- 1

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.198277	0.992546	0.205732
	k_1	k_2	k_3
Predicted	0.598579	-0.332501	0.733922
Numerical	0.816016	-0.469332	0.653316
Percent Error	36.325533%	41.152059%	10.982932%

Table C.2: First Recollapse to Recollapse- Backward Transition Rules- 1

2.

Maximum a_1: k_i values taken at time: 1.273328e-40 and $\bar{k}_i$ values taken at time: 1.288318e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.198277	0.992546	0.205732
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.328576	0.987647	-0.316223
Numerical	0.584474	0.743083	-0.327557
Percent Error	77.881042%	24.762305%	3.584234%

Table C.3: First Recollapse to Recollapse- Forward Transition Rules- 2

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.584474	0.743083	-0.327557
	k_1	k_2	k_3
Predicted	-0.269473	0.881548	0.387926
Numerical	-0.198277	0.992546	0.205732
Percent Error	26.420429%	12.591274%	46.966197%

Table C.4: First Recollapse to Recollapse- Backward Transition Rules- 2

3.

Maximum a_3: k_i values taken at time: 1.288318e-40 and $\bar{k}_i$ values taken at time: 1.342532e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.584474	0.743083	-0.327557
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.204820	0.255067	0.949753
Numerical	-0.224059	0.295298	0.928762
Percent Error	9.393258%	15.772590%	2.210156%

Table C.5: First Recollapse to Recollapse- Forward Transition Rules- 3

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.224059	0.295298	0.928762
	k_1	k_2	k_3
Predicted	0.571636	0.753387	-0.325023
Numerical	0.584474	0.743083	-0.327557
Percent Error	2.245772%	1.367732%	0.779480%

Table C.6: First Recollapse to Recollapse- Backward Transition Rules- 3

4.

Maximum a_1: k_i values taken at time: 1.342532e-40 and $\bar{k}_i$ values taken at time: 1.744989e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.224059	0.295298	0.928762
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.405991	-0.276907	0.870917
Numerical	0.407005	-0.277316	0.870314
Percent Error	0.249771%	0.147511%	0.069281%

Table C.7: First Recollapse to Recollapse- Forward Transition Rules- 4

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.407005	-0.277316	0.870314
	k_1	k_2	k_3
Predicted	-0.224368	0.295861	0.928509
Numerical	-0.224059	0.295298	0.928762
Percent Error	0.137501%	0.190189%	0.027245%

Table C.8: First Recollapse to Recollapse- Backward Transition Rules- 4

5.

Maximum a_2: k_i values taken at time: 1.744989e-40 and $\bar{k}_i$ values taken at time: 2.284489e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.407005	-0.277316	0.870314
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.331471	0.622666	0.708812
Numerical	-0.333699	0.619812	0.713888
Percent Error	0.672234%	0.458301%	0.716134%

Table C.9: First Recollapse to Recollapse- Forward Transition Rules- 5

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.333699	0.619812	0.713888
	k_1	k_2	k_3
Predicted	0.404499	-0.276748	0.872250
Numerical	0.407005	-0.277316	0.870314
Percent Error	0.619554%	0.205052%	0.221953%

Table C.10: First Recollapse to Recollapse- Backward Transition Rules- 5

Here is where the LQG bounce occurs

6.

Maximum a_1: k_i values taken at time: 3.011989e-40 and $\bar{k}_i$ values taken at time: 3.316474e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.998181	0.043317	-0.041499
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.333131	0.680718	0.652412
Numerical	-0.331879	0.704059	0.627819
Percent Error	0.375922%	3.428829%	3.769536%

Table C.11: First Recollapse to Recollapse- Forward Transition Rules- 6

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.331879	0.704059	0.627819
	k_1	k_2	k_3
Predicted	0.987021	0.119859	-0.106882
Numerical	0.998181	0.043317	-0.041499
Percent Error	1.130650%	63.860284%	61.172671%

Table C.12: First Recollapse to Recollapse- Backward Transition Rules- 6

7.

Maximum a_2: k_i values taken at time: 3.316474e-40 and $\bar{k}_i$ values taken at time: 3.374420e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.331879	0.704059	0.627819
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.446921	-0.292369	0.845447
Numerical	0.493575	-0.307212	0.813636
Percent Error	10.438937%	5.076747%	3.762633%

Table C.13: First Recollapse to Recollapse- Forward Transition Rules- 7

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.493575	-0.307212	0.813636
	k_1	k_2	k_3
Predicted	-0.313423	0.796760	0.516662
Numerical	-0.331879	0.704059	0.627819
Percent Error	5.888479%	11.634740%	21.514452%

Table C.14: First Recollapse to Recollapse- Backward Transition Rules- 7

8.

Maximum a_1: k_i values taken at time: 3.374420e-40 and $\bar{k}_i$ values taken at time: 3.405551e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.493575	-0.307212	0.813636
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.248383	0.342168	0.906216
Numerical	-0.269374	0.387926	0.881448
Percent Error	8.450982%	13.373085%	2.733073%

Table C.15: First Recollapse to Recollapse- Forward Transition Rules- 8

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.269374	0.387926	0.881448
	k_1	k_2	k_3
Predicted	0.584007	-0.326985	0.742978
Numerical	0.493575	-0.307212	0.813636
Percent Error	15.484759%	6.047225%	9.510194%

Table C.16: First Recollapse to Recollapse- Backward Transition Rules- 8

9.

Maximum a_2: k_i values taken at time: 3.405551e-40 and $\bar{k}_i$ values taken at time: 3.426396e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.269374	0.387926	0.881448
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.285203	-0.218445	0.933242
Numerical	0.325639	-0.240131	0.914492
Percent Error	14.178008%	9.927538%	2.009120%

Table C.17: First Recollapse to Recollapse- Forward Transition Rules- 9

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.325639	-0.240131	0.914492
	k_1	k_2	k_3
Predicted	-0.297503	0.462024	0.835479
Numerical	-0.269374	0.387926	0.881448
Percent Error	9.455070%	16.037692%	5.502118%

Table C.18: First Recollapse to Recollapse- Backward Transition Rules- 9

10.

Maximum a_1: k_i values taken at time: 3.426396e-40 and $\bar{k}_i$ values taken at time: 3.441928e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.325639	-0.240131	0.914492
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.197204	0.248987	0.948217
Numerical	-0.217964	0.284349	0.933615
Percent Error	10.527255%	14.202536%	1.539982%

Table C.19: First Recollapse to Recollapse- Forward Transition Rules- 10

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.217964	0.284349	0.933615
	k_1	k_2	k_3
Predicted	0.386413	-0.268724	0.882311
Numerical	0.325639	-0.240131	0.914492
Percent Error	15.727723%	10.640141%	3.647417%

Table C.20: First Recollapse to Recollapse- Backward Transition Rules- 10

11.

Maximum a_2: k_i values taken at time: 3.441928e-40 and $\bar{k}_i$ values taken at time: 3.454251e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.217964	0.284349	0.933615
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.223583	-0.181264	0.957681
Numerical	0.254746	-0.200698	0.945952
Percent Error	13.937897%	10.720910%	1.224781%

Table C.21: First Recollapse to Recollapse- Forward Transition Rules- 11

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.254746	-0.200698	0.945952
	k_1	k_2	k_3
Predicted	-0.244986	0.335276	0.909710
Numerical	-0.217964	0.284349	0.933615
Percent Error	11.029778%	15.189430%	2.627761%

Table C.22: First Recollapse to Recollapse- Backward Transition Rules- 11

12.

Maximum a_1: k_i values taken at time: 3.454251e-40 and $\bar{k}_i$ values taken at time: 3.464438e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.254746	-0.200698	0.945952
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.168763	0.204568	0.964194
Numerical	-0.186849	0.232328	0.954521
Percent Error	10.717153%	13.569852%	1.003229%

Table C.23: First Recollapse to Recollapse- Forward Transition Rules- 12

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.186849	0.232328	0.954521
	k_1	k_2	k_3
Predicted	0.298337	-0.225723	0.927385
Numerical	0.254746	-0.200698	0.945952
Percent Error	14.611511%	11.086716%	2.002015%

Table C.24: First Recollapse to Recollapse- Backward Transition Rules- 12

13.

Maximum a_2: k_i values taken at time: 3.464438e-40 and $\bar{k}_i$ values taken at time: 3.473108e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.186849	0.232328	0.954521
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.189673	-0.158623	0.968949
Numerical	0.214660	-0.175459	0.960799
Percent Error	13.173633%	10.614284%	0.841144%

Table C.25: First Recollapse to Recollapse- Forward Transition Rules- 13

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.214660	-0.175459	0.960799
	k_1	k_2	k_3
Predicted	-0.209925	0.270320	0.939605
Numerical	-0.186849	0.232328	0.954521
Percent Error	10.992554%	14.054484%	1.587471%

Table C.26: First Recollapse to Recollapse- Backward Transition Rules- 13

14.

Maximum a_1: k_i values taken at time: 3.473108e-40 and $\bar{k}_i$ values taken at time: 3.480647e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.214660	-0.175459	0.960799
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.150183	0.177610	0.972574
Numerical	-0.165895	0.200310	0.965585
Percent Error	10.461564%	12.780862%	0.718563%

Table C.27: First Recollapse to Recollapse- Forward Transition Rules- 14

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.165895	0.200310	0.965585
	k_1	k_2	k_3
Predicted	0.248268	-0.196765	0.948497
Numerical	0.214660	-0.175459	0.960799
Percent Error	13.536770%	10.827863%	1.297008%

Table C.28: First Recollapse to Recollapse- Backward Transition Rules- 14

15.

Maximum a_2: k_i values taken at time: 3.480647e-40 and $\bar{k}_i$ values taken at time: 3.487310e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.165895	0.200310	0.965585
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.167586	-0.143015	0.975429
Numerical	0.188373	-0.157727	0.969354
Percent Error	12.403641%	10.286829%	0.622815%

Table C.29: First Recollapse to Recollapse- Forward Transition Rules- 15

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.188373	-0.157727	0.969354
	k_1	k_2	k_3
Predicted	-0.185642	0.230411	0.955231
Numerical	-0.165895	0.200310	0.965585
Percent Error	10.637221%	13.064055%	1.083915%

Table C.30: First Recollapse to Recollapse- Backward Transition Rules- 15

16.

Maximum a_1: k_i values taken at time: 3.487310e-40 and $\bar{k}_i$ values taken at time: 3.493590e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.188373	-0.157727	0.969354
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.136825	0.159085	0.977740
Numerical	-0.153831	0.172787	0.981044
Percent Error	12.429512%	8.613215%	0.337952%

Table C.31: First Recollapse to Recollapse- Forward Transition Rules- 16

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.153831	0.172787	0.981044
	k_1	k_2	k_3
Predicted	0.222191	-0.194813	0.972621
Numerical	0.188373	-0.157727	0.969354
Percent Error	15.220451%	19.036615%	0.335903%

Table C.32: First Recollapse to Recollapse- Backward Transition Rules- 16

C.2 Second Recollapse to Recollapse

1.

Maximum a_3: k_i values taken at time: 3.501152e-40 and $\bar{k}_i$ values taken at time: 3.504082e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.896488	0.691085	-0.587573
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	1.591003	2.763751	-3.354755
Numerical	-0.110778	0.062843	1.047935
Percent Error	106.962767%	97.726184%	131.237316%

Table C.33: Second Recollapse to Recollapse- Forward Transition Rules- 1

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.110778	0.062843	1.047935
	k_1	k_2	k_3
Predicted	0.641207	0.697288	-0.338495
Numerical	0.896488	0.691085	-0.587573
Percent Error	39.812638%	0.889530%	73.584241%

Table C.34: Second Recollapse to Recollapse- Backward Transition Rules- 1

2.

Maximum a_1: k_i values taken at time: 3.504082e-40 and $\bar{k}_i$ values taken at time: 3.512088e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.110778	0.062843	1.047935
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.142307	-0.203885	1.061578
Numerical	0.281600	-0.216529	0.934930
Percent Error	97.882247%	6.201703%	11.930205%

Table C.35: Second Recollapse to Recollapse- Forward Transition Rules- 2

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.281600	-0.216529	0.934930
	k_1	k_2	k_3
Predicted	-0.180143	0.221770	0.958374
Numerical	-0.110778	0.062843	1.047935
Percent Error	38.505667%	71.663109%	9.345161%

Table C.36: Second Recollapse to Recollapse- Backward Transition Rules- 2

3.

Maximum a_2: k_i values taken at time: 3.512088e-40 and $\bar{k}_i$ values taken at time: 3.527577e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.281600	-0.216529	0.934930
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.267151	0.381925	0.885226
Numerical	-0.248312	0.342022	0.906291
Percent Error	7.051743%	10.447955%	2.379586%

Table C.37: Second Recollapse to Recollapse- Forward Transition Rules- 3

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.248312	0.342022	0.906291
	k_1	k_2	k_3
Predicted	0.258741	-0.203096	0.944355
Numerical	0.281600	-0.216529	0.934930
Percent Error	8.834429%	6.614454%	0.998009%

Table C.38: Second Recollapse to Recollapse- Backward Transition Rules- 3

4.

Maximum a_1:

k_i values taken at time: 3.527577e-40 and $\bar{k}_i$ values taken at time: 3.551206e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.248312	0.342022	0.906291
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.493293	-0.307131	0.813839
Numerical	0.447632	-0.292617	0.844986
Percent Error	9.256425%	4.725537%	3.827268%

Table C.39: Second Recollapse to Recollapse- Forward Transition Rules- 4

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.447632	-0.292617	0.844986
	k_1	k_2	k_3
Predicted	-0.236185	0.317975	0.918210
Numerical	-0.248312	0.342022	0.906291
Percent Error	5.134756%	7.562425%	1.298090%

Table C.40: Second Recollapse to Recollapse- Backward Transition Rules- 4

5.

Maximum a_2: k_i values taken at time: 3.551206e-40 and $\bar{k}_i$ values taken at time: 3.596381e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.447632	-0.292617	0.844986
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.331758	0.705499	0.626263
Numerical	-0.333326	0.663996	0.669330
Percent Error	0.472647%	5.882748%	6.876961%

Table C.41: Second Recollapse to Recollapse- Forward Transition Rules- 5

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.333326	0.663996	0.669330
	k_1	k_2	k_3
Predicted	0.427264	-0.285223	0.857959
Numerical	0.447632	-0.292617	0.844986
Percent Error	4.767185%	2.592472%	1.512067%

Table C.42: Second Recollapse to Recollapse- Backward Transition Rules- 5

6.

Maximum a_1: k_i values taken at time: 3.596381e-40 and $\bar{k}_i$ values taken at time: 4.373888e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.333326	0.663996	0.669330
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.999934	-0.007967	0.008035
Numerical	0.999303	-0.026303	0.027025
Percent Error	0.063089%	230.170672%	236.339401%

Table C.43: Second Recollapse to Recollapse- Forward Transition Rules- 6

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.999303	-0.026303	0.027025
	k_1	k_2	k_3
Predicted	-0.333256	0.657740	0.675524
Numerical	-0.333326	0.663996	0.669330
Percent Error	0.021044%	0.951207%	0.916871%

Table C.44: Second Recollapse to Recollapse- Backward Transition Rules- 6

Here is where the LQG bounce occurs

7.

Maximum a_2: k_i values taken at time: 4.820134e-40 and $\bar{k}_i$ values taken at time: 5.108540e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.341678	0.682435	0.659241
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.432663	-0.288572	0.855908
Numerical	0.439835	-0.289853	0.850017
Percent Error	1.657626%	0.444116%	0.688282%

Table C.45: Second Recollapse to Recollapse- Forward Transition Rules- 7

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.439835	-0.289853	0.850017
	k_1	k_2	k_3
Predicted	-0.332796	0.689646	0.643146
Numerical	-0.341678	0.682435	0.659241
Percent Error	2.668848%	1.045632%	2.502494%

Table C.46: Second Recollapse to Recollapse- Backward Transition Rules- 7

8.

Maximum a_3: k_i values taken at time: 5.108540e-40 and $\bar{k}_i$ values taken at time: 5.240205e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.439835	-0.289853	0.850017
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.792534	0.522282	-0.314817
Numerical	0.792505	0.522320	-0.314827
Percent Error	0.003627%	0.007188%	0.003083%

Table C.47: Second Recollapse to Recollapse- Forward Transition Rules- 8

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.792505	0.522320	-0.314827
	k_1	k_2	k_3
Predicted	0.439728	-0.289820	0.850087
Numerical	0.439835	-0.289853	0.850017
Percent Error	0.024327%	0.011650%	0.008323%

Table C.48: Second Recollapse to Recollapse- Backward Transition Rules- 8

9.

Maximum a_2: k_i values taken at time: 5.240205e-40 and $\bar{k}_i$ values taken at time: 5.288251e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.792505	0.522320	-0.314827
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.898518	-0.255458	0.356940
Numerical	0.898864	-0.255987	0.357122
Percent Error	0.038570%	0.207136%	0.051099%

Table C.49: Second Recollapse to Recollapse- Forward Transition Rules- 9

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.898864	-0.255987	0.357122
	k_1	k_2	k_3
Predicted	0.792765	0.524537	-0.317304
Numerical	0.792505	0.522320	-0.314827
Percent Error	0.032819%	0.422643%	0.780793%

Table C.50: Second Recollapse to Recollapse- Backward Transition Rules- 9

10.

Maximum a_1: k_i values taken at time: 5.288251e-40 and $\bar{k}_i$ values taken at time: 5.296075e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.898864	-0.255987	0.357122
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.321284	0.551069	0.770214
Numerical	-0.288069	0.414446	0.873622
Percent Error	10.338164%	24.792321%	13.425885%

Table C.51: Second Recollapse to Recollapse- Forward Transition Rules- 10

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.288069	0.414446	0.873622
	k_1	k_2	k_3
Predicted	0.679628	-0.381471	0.701843
Numerical	0.898864	-0.255987	0.357122
Percent Error	32.258292%	32.894750%	49.116527%

Table C.52: Second Recollapse to Recollapse- Backward Transition Rules- 10

11.

Maximum a_3: Kasner Transition: k_i values taken at time: 5.296075e-40 and $\bar{k}_i$ values taken at time: 5.298077e-40

	k_1	k_2	k_3
Numerical	-0.288069	0.414446	0.873622
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.531142	0.786858	-0.317999
Numerical	0.671070	0.734799	-0.405869
Percent Error	26.344851%	6.616014%	27.632088%

Table C.53: Second Recollapse to Recollapse- Forward Transition Rules- 11

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.671070	0.734799	-0.405869
	k_1	k_2	k_3
Predicted	-0.747199	-0.408685	2.155884
Numerical	-0.288069	0.414446	0.873622
Percent Error	61.446856%	201.409643%	59.477293%

Table C.54: Second Recollapse to Recollapse- Backward Transition Rules- 11

C.3 Third Recollapse to Recollapse

1.

Maximum a_1: k_i values taken at time: 5.300682e-40 and $\bar{k}_i$ values taken at time: 5.302995e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.506744	0.047335	1.459409
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-37.569320	71.629285	-33.059969
Numerical	0.189567	-0.159622	0.970055
Percent Error	100.504578%	100.222844%	102.934229%

Table C.55: Third Recollapse to Recollapse- Forward Transition Rules- 1

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.189567	-0.159622	0.970055
	k_1	k_2	k_3
Predicted	-0.137453	0.159166	0.978287
Numerical	-0.506744	0.047335	1.459409
Percent Error	268.666049%	70.260649%	49.180033%

Table C.56: Third Recollapse to Recollapse- Backward Transition Rules- 1

2.

Maximum a_2: k_i values taken at time: 5.302995e-40 and $\bar{k}_i$ values taken at time: 5.304999e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.189567	-0.159622	0.970055
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.190489	0.234477	0.956012
Numerical	-0.164746	0.198603	0.966143
Percent Error	13.514213%	15.299306%	1.059649%

Table C.57: Third Recollapse to Recollapse- Forward Transition Rules- 2

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.164746	0.198603	0.966143
	k_1	k_2	k_3
Predicted	0.166375	-0.142143	0.975768
Numerical	0.189567	-0.159622	0.970055
Percent Error	13.939021%	12.296365%	0.585465%

Table C.58: Third Recollapse to Recollapse- Backward Transition Rules- 2

3.

Maximum a_1: k_i values taken at time: 5.304999e-40 and $\bar{k}_i$ values taken at time: 5.307163e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.164746	0.198603	0.966143
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.245703	-0.195208	0.949505
Numerical	0.208247	-0.171220	0.962974
Percent Error	15.244410%	12.288329%	1.418497%

Table C.59: Third Recollapse to Recollapse- Forward Transition Rules- 3

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.208247	-0.171220	0.962974
	k_1	k_2	k_3
Predicted	-0.147016	0.173156	0.973861
Numerical	-0.164746	0.198603	0.966143
Percent Error	12.059975%	14.696378%	0.792508%

Table C.60: Third Recollapse to Recollapse- Backward Transition Rules- 3

4.

Maximum a_2: k_i values taken at time: 5.307163e-40 and $\bar{k}_i$ values taken at time: 5.309543e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.208247	-0.171220	0.962974
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.204077	0.260387	0.943692
Numerical	-0.178530	0.219360	0.959171
Percent Error	12.518069%	15.756065%	1.640326%

Table C.61: Third Recollapse to Recollapse- Forward Transition Rules- 4

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.178530	0.219360	0.959171
	k_1	k_2	k_3
Predicted	0.180848	-0.152469	0.971621
Numerical	0.208247	-0.171220	0.962974
Percent Error	15.150313%	12.298289%	0.890019%

Table C.62: Third Recollapse to Recollapse- Backward Transition Rules- 4

5.

Maximum a_1: k_i values taken at time: 5.309543e-40 and $\bar{k}_i$ values taken at time: 5.312185e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.178530	0.219360	0.959171
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.277678	-0.214173	0.936497
Numerical	0.232351	-0.186864	0.954513
Percent Error	16.323639%	12.751274%	1.923845%

Table C.63: Third Recollapse to Recollapse- Forward Transition Rules- 5

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.232351	-0.186864	0.954513
	k_1	k_2	k_3
Predicted	-0.158634	0.189689	0.968945
Numerical	-0.178530	0.219360	0.959171
Percent Error	12.542506%	15.641680%	1.008687%

Table C.64: Third Recollapse to Recollapse- Backward Transition Rules- 5

6.

Maximum a_2: k_i values taken at time: 5.312185e-40 and $\bar{k}_i$ values taken at time: 5.315153e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.232351	-0.186864	0.954513
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.225742	0.298374	0.927369
Numerical	-0.196477	0.247798	0.948680
Percent Error	12.963955%	16.950642%	2.297967%

Table C.65: Third Recollapse to Recollapse- Forward Transition Rules- 6

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.196477	0.247798	0.948680
	k_1	k_2	k_3
Predicted	0.200000	-0.165685	0.965686
Numerical	0.232351	-0.186864	0.954513
Percent Error	16.175738%	12.782428%	1.156950%

Table C.66: Third Recollapse to Recollapse- Backward Transition Rules- 6

7.

Maximum a_1: k_i values taken at time: 5.315153e-40 and $\bar{k}_i$ values taken at time: 5.318538e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.196477	0.247798	0.948680
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.323662	-0.239120	0.915460
Numerical	0.266553	-0.207723	0.941170
Percent Error	17.644515%	13.130309%	2.808512%

Table C.67: Third Recollapse to Recollapse- Forward Transition Rules- 7

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.266553	-0.207723	0.941170
	k_1	k_2	k_3
Predicted	-0.173865	0.212238	0.961627
Numerical	-0.196477	0.247798	0.948680
Percent Error	13.005839%	16.754731%	1.346368%

Table C.68: Third Recollapse to Recollapse- Backward Transition Rules- 7

8.

Maximum a_2: k_i values taken at time: 5.318538e-40 and $\bar{k}_i$ values taken at time: 5.322476e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.266553	-0.207723	0.941170
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.254711	0.355352	0.899360
Numerical	-0.221091	0.289934	0.931158
Percent Error	13.199413%	18.409425%	3.535679%

Table C.69: Third Recollapse to Recollapse- Forward Transition Rules- 8

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.221091	0.289934	0.931158
	k_1	k_2	k_3
Predicted	0.227093	-0.183518	0.956426
Numerical	0.266553	-0.207723	0.941170
Percent Error	17.376060%	13.189402%	1.595035%

Table C.70: Third Recollapse to Recollapse- Backward Transition Rules- 8

9.

Maximum a_1: k_i values taken at time: 5.322476e-40 and $\bar{k}_i$ values taken at time: 5.327164e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.221091	0.289934	0.931158
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.396349	-0.272934	0.876588
Numerical	0.320119	-0.237296	0.917178
Percent Error	19.233137%	13.057371%	4.630558%

Table C.71: Third Recollapse to Recollapse- Forward Transition Rules- 9

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.320119	-0.237296	0.917178
	k_1	k_2	k_3
Predicted	-0.195166	0.245661	0.949506
Numerical	-0.221091	0.289934	0.931158
Percent Error	13.283413%	18.022258%	1.932382%

Table C.72: Third Recollapse to Recollapse- Backward Transition Rules- 9

10.

Maximum a_2: k_i values taken at time: 5.327164e-40 and $\bar{k}_i$ values taken at time: 5.332969e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.320119	-0.237296	0.917178
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.294006	0.451641	0.842367
Numerical	-0.257320	0.360930	0.896391
Percent Error	12.478024%	20.084849%	6.413345%

Table C.73: Third Recollapse to Recollapse- Forward Transition Rules- 10

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.257320	0.360930	0.896391
	k_1	k_2	k_3
Predicted	0.269790	-0.209616	0.939827
Numerical	0.320119	-0.237296	0.917178
Percent Error	18.655001%	13.204932%	2.409897%

Table C.74: Third Recollapse to Recollapse- Backward Transition Rules- 10

11.

Maximum a_1: k_i values taken at time: 5.332969e-40 and $\bar{k}_i$ values taken at time: 5.340536e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.257320	0.360930	0.896391
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.530163	-0.316692	0.786532
Numerical	0.419858	-0.282396	0.862538
Percent Error	20.805764%	10.829654%	9.663479%

Table C.75: Third Recollapse to Recollapse- Forward Transition Rules- 11

Backwards Kasner Transition: k_i values taken at time:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.419858	-0.282396	0.862538
	k_1	k_2	k_3
Predicted	-0.228219	0.302938	0.925281
Numerical	-0.257320	0.360930	0.896391
Percent Error	12.751296%	19.142976%	3.122288%

Table C.76: Third Recollapse to Recollapse- Backward Transition Rules- 11

12.

Maximum a_2: k_i values taken at time: 5.340536e-40 and $\bar{k}_i$ values taken at time: 5.351321e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.419858	-0.282396	0.862538
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.333020	0.648874	0.684147
Numerical	-0.312637	0.513579	0.799059
Percent Error	6.120560%	20.850741%	16.796251%

Table C.77: Third Recollapse to Recollapse- Forward Transition Rules- 12

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.312637	0.513579	0.799059
	k_1	k_2	k_3
Predicted	0.352474	-0.253349	0.900875
Numerical	0.419858	-0.282396	0.862538
Percent Error	19.117416%	11.464951%	4.255537%

Table C.78: Third Recollapse to Recollapse- Backward Transition Rules- 12

13.

Maximum a_1: k_i values taken at time: 5.351321e-40 and $\bar{k}_i$ values taken at time: 5.369606e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.312637	0.513579	0.799059
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.834309	-0.298071	0.463764
Numerical	0.684948	-0.332993	0.648045
Percent Error	17.902353%	11.715765%	39.735940%

Table C.79: Third Recollapse to Recollapse- Forward Transition Rules- 13

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.684948	-0.332993	0.648045
	k_1	k_2	k_3
Predicted	-0.289020	0.437531	0.851489
Numerical	-0.312637	0.513579	0.799059
Percent Error	8.171337%	17.381172%	6.157540%

Table C.80: Third Recollapse to Recollapse- Backward Transition Rules- 13

14.

Maximum a_2: k_i values taken at time: 5.369606e-40 and $\bar{k}_i$ values taken at time: 5.420869e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.684948	-0.332993	0.648045
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.056771	0.996940	-0.053710
Numerical	-0.121055	0.983002	0.138057
Percent Error	313.233843%	1.398095%	357.039763%

Table C.81: Third Recollapse to Recollapse- Forward Transition Rules- 14

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.121055	0.983002	0.138057
	k_1	k_2	k_3
Predicted	0.622032	-0.331423	0.709393
Numerical	0.684948	-0.332993	0.648045
Percent Error	10.114557%	0.473606%	8.647854%

Table C.82: Third Recollapse to Recollapse- Backward Transition Rules- 14

15. Maximum a_1: k_i values taken at time: 5.420869e-40 and $\bar{k}_i$ values taken at time: 5.657869e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.121055	0.983002	0.138057
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.159726	0.977572	-0.137293
Numerical	0.159429	0.973313	-0.132739
Percent Error	0.186021%	0.435662%	3.316750%

Table C.83: Third Recollapse to Recollapse- Forward Transition Rules- 15

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.159429	0.973313	-0.132739
	k_1	k_2	k_3
Predicted	-0.120884	0.979765	0.141121
Numerical	-0.121055	0.983002	0.138057
Percent Error	0.141246%	0.330375%	2.171434%

Table C.84: Third Recollapse to Recollapse- Backward Transition Rules- 15

Here is where the LQG bounce occurs

16.

Maximum a_1: k_i values taken at time: 5.877619e-40 and $\bar{k}_i$ values taken at time: 6.252619e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.448740	-0.280109	0.831366
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.236493	0.325364	0.911127
Numerical	-0.246166	0.337653	0.908507
Percent Error	4.090280%	3.777040%	0.287607%

Table C.85: Third Recollapse to Recollapse- Forward Transition Rules- 16

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.246166	0.337653	0.908507
	k_1	k_2	k_3
Predicted	0.484895	-0.304685	0.819778
Numerical	0.448740	-0.280109	0.831366
Percent Error	7.456244%	8.066010%	1.413570%

Table C.86: Third Recollapse to Recollapse- Backward Transition Rules- 16

17.

Maximum a_3: k_i values taken at time: 6.252619e-40 and $\bar{k}_i$ values taken at time: 6.391412e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.246166	0.337653	0.908507
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.557629	0.764876	-0.322507
Numerical	0.556762	0.765579	-0.322343
Percent Error	0.155420%	0.091890%	0.050936%

Table C.87: Third Recollapse to Recollapse- Forward Transition Rules- 17

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.556762	0.765579	-0.322343
	k_1	k_2	k_3
Predicted	-0.247453	0.340243	0.907205
Numerical	-0.246166	0.337653	0.908507
Percent Error	0.520053%	0.761378%	0.143559%

Table C.88: Third Recollapse to Recollapse- Backward Transition Rules- 17

18.

Maximum a_2: k_i values taken at time: 6.391412e-40 and $\bar{k}_i$ values taken at time: 6.414112e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.556762	0.765579	-0.322343
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.824887	-0.302462	0.477574
Numerical	0.830986	-0.319134	0.488148
Percent Error	0.739282%	5.511976%	2.214095%

Table C.89: Third Recollapse to Recollapse- Forward Transition Rules- 18

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.830986	-0.319134	0.488148
	k_1	k_2	k_3
Predicted	0.532764	0.882236	-0.415000
Numerical	0.556762	0.765579	-0.322343
Percent Error	4.504374%	13.222843%	22.327104%

Table C.90: Third Recollapse to Recollapse- Backward Transition Rules- 18

19.

Maximum a_1: k_i values taken at time: 6.414112e-40 and $\bar{k}_i$ values taken at time: 6.417392e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.830986	-0.319134	0.488148
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.312169	0.504452	0.807717
Numerical	0.006400	-0.059481	1.053081
Percent Error	102.050262%	111.791220%	30.377463%

Table C.91: Third Recollapse to Recollapse- Forward Transition Rules- 19

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.006400	-0.059481	1.053081
	k_1	k_2	k_3
Predicted	-0.006319	-0.046091	1.052410
Numerical	0.830986	-0.319134	0.488148
Percent Error	13249.760952%	592.406085%	53.616181%

Table C.92: Third Recollapse to Recollapse- Backward Transition Rules- 19

C.4 Fourth Recollapse to Recollapse

1.

Maximum a_1: k_i values taken at time: 6.423785e-40 and $\bar{k}_i$ values taken at time: 6.428928e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.152014	1.024991	0.127024
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.218420	1.035908	-0.254328
Numerical	0.240052	0.951655	-0.191705
Percent Error	9.903704%	8.133202%	24.622893%

Table C.93: Fourth Recollapse to Recollapse- Forward Transition Rules- 1

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.240052	0.951655	-0.191705
	k_1	k_2	k_3
Predicted	-0.162186	0.967337	0.194851
Numerical	-0.152014	1.024991	0.127024
Percent Error	6.271577%	5.960056%	34.809661%

Table C.94: Fourth Recollapse to Recollapse- Backward Transition Rules- 1

2.

Maximum a_3: k_i values taken at time: 6.428928e-40 and $\bar{k}_i$ values taken at time: 6.437223e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.240052	0.951655	-0.191705
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.232501	0.921593	0.310911
Numerical	-0.216491	0.934748	0.281744
Percent Error	6.886006%	1.427432%	9.381156%

Table C.95: Fourth Recollapse to Recollapse- Forward Transition Rules- 2

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.216491	0.934748	0.281744
	k_1	k_2	k_3
Predicted	0.221938	0.958265	-0.180202
Numerical	0.240052	0.951655	-0.191705
Percent Error	8.161699%	0.689809%	6.383143%

Table C.96: Fourth Recollapse to Recollapse- Backward Transition Rules- 2

3.

Maximum a_1: k_i values taken at time: 6.437223e-40 and $\bar{k}_i$ values taken at time: 6.448777e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.216491	0.934748	0.281744
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.381805	0.884921	-0.266724
Numerical	0.347916	0.903257	-0.251169
Percent Error	8.875932%	2.071967%	5.831606%

Table C.97: Fourth Recollapse to Recollapse- Forward Transition Rules- 3

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.347916	0.903257	-0.251169
	k_1	k_2	k_3
Predicted	-0.205160	0.942952	0.262209
Numerical	-0.216491	0.934748	0.281744
Percent Error	5.523041%	0.870037%	7.449982%

Table C.98: Fourth Recollapse to Recollapse- Backward Transition Rules- 3

4.

Maximum a_3: k_i values taken at time: 6.448777e-40 and $\bar{k}_i$ values taken at time: 6.467102e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.347916	0.903257	-0.251169
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.310296	0.805604	0.504699
Numerical	-0.299292	0.831779	0.467514
Percent Error	3.546436%	3.249147%	7.367739%

Table C.99: Fourth Recollapse to Recollapse- Forward Transition Rules- 4

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.299292	0.831779	0.467514
	k_1	k_2	k_3
Predicted	0.328541	0.913066	-0.241606
Numerical	0.347916	0.903257	-0.251169
Percent Error	5.897235%	1.074280%	3.958274%

Table C.100: Fourth Recollapse to Recollapse- Backward Transition Rules- 4

5.

Maximum a_1: k_i values taken at time: 6.467102e-40 and $\bar{k}_i$ values taken at time: 6.504807e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.299292	0.831779	0.467514
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.745588	0.580933	-0.326515
Numerical	0.717841	0.612726	-0.330564
Percent Error	3.721384%	5.472749%	1.240155%

Table C.101: Fourth Recollapse to Recollapse- Forward Transition Rules- 5

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.717841	0.612726	-0.330564
	k_1	k_2	k_3
Predicted	-0.294719	0.841000	0.453720
Numerical	-0.299292	0.831779	0.467514
Percent Error	1.551566%	1.096364%	3.040232%

Table C.102: Fourth Recollapse to Recollapse- Backward Transition Rules- 5

6.

Maximum a_3: k_i values taken at time: 6.504807e-40 and $\bar{k}_i$ values taken at time: 6.714771e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.717841	0.612726	-0.330564
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.167358	-0.142836	0.975486
Numerical	0.159303	-0.136983	0.977683
Percent Error	4.812884%	4.097906%	0.225169%

Table C.103: Fourth Recollapse to Recollapse- Forward Transition Rules- 6

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.159303	-0.136983	0.977683
	k_1	k_2	k_3
Predicted	0.715535	0.615282	-0.330816
Numerical	0.717841	0.612726	-0.330564
Percent Error	0.322283%	0.415456%	0.076132%

Table C.104: Fourth Recollapse to Recollapse- Backward Transition Rules- 6

Here is where the LQG bounce occurs

7.

Maximum a_1: k_i values taken at time: 7.274271e-40 and $\bar{k}_i$ values taken at time: 7.621456e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.507362	0.803649	-0.311014
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.251827	0.902542	0.349284
Numerical	-0.251826	0.902543	0.349281
Percent Error	0.000559%	0.000173%	0.000945%

Table C.105: Fourth Recollapse to Recollapse- Forward Transition Rules- 7

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.251826	0.902543	0.349281
	k_1	k_2	k_3
Predicted	0.507357	0.803653	-0.311013
Numerical	0.507362	0.803649	-0.311014
Percent Error	0.001127%	0.000529%	0.000254%

Table C.106: Fourth Recollapse to Recollapse- Backward Transition Rules- 7

8.

Maximum a_2: k_i values taken at time: 7.621456e-40 and $\bar{k}_i$ values taken at time: 7.700943e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.251826	0.902543	0.349281
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.553730	-0.321752	0.768022
Numerical	0.554889	-0.321979	0.767090
Percent Error	0.209250%	0.070566%	0.121284%

Table C.107: Fourth Recollapse to Recollapse- Forward Transition Rules- 8

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.554889	-0.321979	0.767090
	k_1	k_2	k_3
Predicted	-0.250168	0.904332	0.345834
Numerical	-0.251826	0.902543	0.349281
Percent Error	0.662654%	0.197805%	0.996477%

Table C.108: Fourth Recollapse to Recollapse- Backward Transition Rules- 8

9.

Maximum a_1: k_i values taken at time: 7.700943e-40 and $\bar{k}_i$ values taken at time: 7.729208e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.554889	-0.321979	0.767090
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	-0.263008	0.373403	0.889605
Numerical	-0.265337	0.378638	0.886699
Percent Error	0.885483%	1.401757%	0.326597%

Table C.109: Fourth Recollapse to Recollapse- Forward Transition Rules- 9

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	-0.265337	0.378638	0.886699
	k_1	k_2	k_3
Predicted	0.565358	-0.323947	0.758588
Numerical	0.554889	-0.321979	0.767090
Percent Error	1.851775%	0.607281%	1.120784%

Table C.110: Fourth Recollapse to Recollapse- Backward Transition Rules- 9

10.

Maximum a_2: k_i values taken at time: 7.729208e-40 and $\bar{k}_i$ values taken at time: 7.740229e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	-0.265337	0.378638	0.886699
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.279944	-0.215469	0.935525
Numerical	0.149965	-0.137520	0.987554
Percent Error	46.430171%	36.176319%	5.561562%

Table C.111: Fourth Recollapse to Recollapse- Forward Transition Rules- 10

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.149965	-0.137520	0.987554
	k_1	k_2	k_3
Predicted	-0.172526	0.189693	0.982833
Numerical	-0.265337	0.378638	0.886699
Percent Error	53.795235%	99.605269%	9.781286%

Table C.112: Fourth Recollapse to Recollapse- Backward Transition Rules- 10

11.

Maximum a_3: k_i values taken at time: 7.740229e-40 and $\bar{k}_i$ values taken at time: 7.742723e-40

Kasner Transition:

	k_1	k_2	k_3
Numerical	0.149965	-0.137520	0.987554
	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Predicted	0.714285	0.617654	-0.331939
Numerical	0.569901	0.791770	-0.361671
Percent Error	20.213768%	28.189804%	8.957002%

Table C.113: Fourth Recollapse to Recollapse- Forward Transition Rules- 11

Backwards Kasner Transition:

	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$
Numerical	0.569901	0.791770	-0.361671
	k_1	k_2	k_3
Predicted	-0.554621	0.247339	1.307282
Numerical	0.149965	-0.137520	0.987554
Percent Error	127.039247%	155.599781%	24.457407%

Table C.114: Fourth Recollapse to Recollapse- Backward Transition Rules- 11

Appendix D

Code for Project

D.1 LQC Bianchi IX Code

This is the code to generate LQC Bianchi IX solutions in C++. Note that this code can be modified to generate solutions for both Classical and LQC Bianchi I, II, and IX.

```
//C++ program to implement Runge Kutta method in LQC Bianchi IX
#define _USE_MATH_DEFINES
#include<stdio.h>
#include<iostream>
#include<fstream>
#include<string>
#include <cmath>
#include <iomanip>

using namespace std;

//Defining constants
long double l_p1 = 1.0;
long double gamma = 1.0;
long double Delta = 4*sqrt(3)*M_PI*gamma*l_p1*l_p1;
long double G = 1.0;//This is the gravitational constant
long double placeholder = 1.0;

long double dP1dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return(P1*sqrt(P1*P2*P3)/(sqrt(::Delta)*::gamma*::l_p1))*
    cos(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    (sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2) +
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3));
}

long double dP2dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return(P2*sqrt(P1*P2*P3)/(sqrt(::Delta)*::gamma*::l_p1))*
    cos(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    (sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3) +
```

```
    sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1));
}

long double dP3dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return(P3*sqrt(P1*P2*P3)/(sqrt(::Delta)*::gamma*::l_p1))*
    cos(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1) +
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2));
}

long double dc1dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return-((P2*P3)/(::Delta*::gamma*::l_p1*::l_p1))*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
    sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)

    +((sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)/2.0)*
    cos(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    (sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    -((sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)/2.0)*
    cos(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    -((sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)/2.0)*
    cos(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)))

    -(P2*P2*P3*P3)/(2.0*(P1*P1*P1)) + (P1*P3*P3)/(2.0*(P2*P2))
    +(P2*P2*P1)/(2.0*(P3*P3)) - P1;
}

long double dc2dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return-((P1*P3)/(::Delta*::gamma*::l_p1*::l_p1))*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
    sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)

    -((sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)/2.0)*
    cos(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    (sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
```

```
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    +((sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)/2.0)*
    cos(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    -((sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)/2.0)*
    cos(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)))

    -(P3*P3*P1*P1)/(2.0*(P2*P2*P2)) + (P2*P1*P1)/(2.0*(P3*P3))
    + (P3*P3*P2)/(2.0*(P1*P1)) - P2;
}

long double dc3dt(long double P1, long double P2, long double P3,
                  long double c1, long double c2, long double c3)
{
return-((P1*P2)/(::Delta*::gamma*::l_p1*::l_p1))*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
    sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)

    -((sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)/2.0)*
    cos(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
    (sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    -((sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)/2.0)*
    cos(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))

    +((sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)/2.0)*
    cos(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)*
    (sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)+
    sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)))

    -(P1*P1*P2*P2)/(2.0*(P3*P3*P3)) + (P3*P2*P2)/(2.0*(P1*P1))
    + (P1*P1*P3)/(2.0*(P2*P2)) - P3;
}

long double rungeKutta(long double t0, long double t,
                       long double P1_0, long double P2_0,
                       long double P3_0, long double c1_0,
                       long double c2_0, long double c3_0,
                       long double h)
{

//Calculating total numer of iterations, n
long long int n = ((t - t0)/h);

//Initializing variables for Runge-Kutta method
```

```
long double k1_P1, k2_P1, k3_P1, k4_P1, k5_P1, k6_P1, k7_P1;
long double k1_P2, k2_P2, k3_P2, k4_P2, k5_P2, k6_P2, k7_P2;
long double k1_P3, k2_P3, k3_P3, k4_P3, k5_P3, k6_P3, k7_P3;
long double k1_c1, k2_c1, k3_c1, k4_c1, k5_c1, k6_c1, k7_c1;
long double k1_c2, k2_c2, k3_c2, k4_c2, k5_c2, k6_c2, k7_c2;
long double k1_c3, k2_c3, k3_c3, k4_c3, k5_c3, k6_c3, k7_c3;
long double k1_d1, k2_d1, k3_d1, k4_d1, k5_d1, k6_d1, k7_d1;
long double k1_d2, k2_d2, k3_d2, k4_d2, k5_d2, k6_d2, k7_d2;
long double k1_d3, k2_d3, k3_d3, k4_d3, k5_d3, k6_d3, k7_d3;

long double P1 = P1_0;
long double P2 = P2_0;
long double P3 = P3_0;
long double c1 = c1_0;
long double c2 = c2_0;
long double c3 = c3_0;


//Here I calculate the relative error of the Hamiltonian
long double C = -((P1*P2*P3)/(8*M_PI*::G*::gamma*::gamma*::Delta))*
(sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)+
sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))
+ (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1) + (P1*P1*P3*P3)/(P2*P2) +
(P1*P1*P2*P2)/(P3*P3)) -(1/(16*M_PI*::G))*(P1*P1 + P2*P2 + P3*P3);

long double C10 = -((P1*P2*P3)/(8*M_PI*::G*::gamma*::gamma*::Delta))*
(sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3));

long double C20 = -((P1*P2*P3)/(8*M_PI*::G*::gamma*::gamma*::Delta))*
(sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3));

long double C30 = -((P1*P2*P3)/(8*M_PI*::G*::gamma*::gamma*::Delta))*
(sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2));

long double U0 = (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1)
+ (P1*P1*P3*P3)/(P2*P2) +(P1*P1*P2*P2)/(P3*P3))
-(1/(16*M_PI*::G))*(P1*P1 + P2*P2 + P3*P3);

long double U1 = (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1));

long double U2 = (1/(32*M_PI*::G))*((P1*P1*P3*P3)/(P2*P2));

long double U3 = (1/(32*M_PI*::G))*((P1*P1*P2*P2)/(P3*P3));

long double U23 = -(1/(16*M_PI*::G))*P1*P1;

long double U13 = -(1/(16*M_PI*::G))*P2*P2;

long double U12 = -(1/(16*M_PI*::G))*P3*P3;

long double C_den = sqrt(C10*C10 + C20*C20 + C30*C30 + U0*U0);
```

```
long double C_rel = C/C_den;

//Here I create the file I am going to output data to
ofstream outputFile;
outputFile.open("Test_For_Paper.csv");

outputFile << std::setprecision(20);

//This point counter will output to our csv file when it equals 1
int Point_Counter = 2;

//This point counter will output to our command window when it
//equals 1
int Point_Counter_2 = 1;

int Points_Per_Output = 1000;
int Points_Per_Output_2 = 100000;

//Initializing directional scale factors and kasner exponents
long double a_new, a1_new, a2_new, a3_new, a_older;
long double a1_older, a2_older, a3_older, k1, k2, k3;

//Calculating initial scale factors
long double a1_old = sqrt((P2*P3)/P1);
long double a2_old = sqrt((P1*P3)/P2);
long double a3_old = sqrt((P1*P2)/P3);
long double a_old = pow(P1*P2*P3, 1.0/6.0);


//Here initial data is outputted to csv file
outputFile << t0 << "," << P1 << "," << P2 << "," << P3 << ","
<< c1 << "," << c2 << "," << c3 << "," << C_rel
<< "," << a1_old << "," << a2_old << "," << a3_old << ","
<< a_old <<"," << h << "," << Points_Per_Output << "," <<
::placeholder << "," << U0 << "," << U1 << "," << U2 << ","
<< U3 << "," << U23 << "," << U13 << "," << U12 << std::endl;

for (long long int i=0; i<=n; i++){


    // Apply 6th order Runge Kutta Formulas to find next values
    //of P_1, P_2, P_3, c_1, c_2, c_3
    k1_P1 = h*dP1dt(P1, P2, P3, c1, c2, c3);
    k1_P2 = h*dP2dt(P1, P2, P3, c1, c2, c3);
    k1_P3 = h*dP3dt(P1, P2, P3, c1, c2, c3);
    k1_c1 = h*dc1dt(P1, P2, P3, c1, c2, c3);
    k1_c2 = h*dc2dt(P1, P2, P3, c1, c2, c3);
    k1_c3 = h*dc3dt(P1, P2, P3, c1, c2, c3);


    k2_P1 = h*dP1dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                    P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
                    c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);
    k2_P2 = h*dP2dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                    P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
                    c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);
    k2_P3 = h*dP3dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                    P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
```

```
                c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);
k2_c1 = h*dc1dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
                c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);
k2_c2 = h*dc2dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
                c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);
k2_c3 = h*dc3dt(P1 + (1.0/3.0)*k1_P1, P2 + (1.0/3.0)*k1_P2,
                P3 + (1.0/3.0)*k1_P3, c1 + (1.0/3.0)*k1_c1,
                c2 + (1.0/3.0)*k1_c2, c3 + (1.0/3.0)*k1_c3);


k3_P1 = h*dP1dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);
k3_P2 = h*dP2dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);
k3_P3 = h*dP3dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);
k3_c1 = h*dc1dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);
k3_c2 = h*dc2dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);
k3_c3 = h*dc3dt(P1 + (2.0/3.0)*k2_P1, P2 + (2.0/3.0)*k2_P2,
                P3 + (2.0/3.0)*k2_P3, c1 + (2.0/3.0)*k2_c1,
                c2 + (2.0/3.0)*k2_c2, c3 + (2.0/3.0)*k2_c3);


k4_P1 = h*dP1dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);
k4_P2 = h*dP2dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);
k4_P3 = h*dP3dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
```

```
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);
k4_c1 = h*dc1dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);
k4_c2 = h*dc2dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);
k4_c3 = h*dc3dt(P1 + (1.0/12.0)*k1_P1 + (1.0/3.0)*k2_P1
                -(1.0/12.0)*k3_P1,
                P2 + (1.0/12.0)*k1_P2 + (1.0/3.0)*k2_P2
                -(1.0/12.0)*k3_P2,
                P3 + (1.0/12.0)*k1_P3 + (1.0/3.0)*k2_P3
                -(1.0/12.0)*k3_P3,
                c1 + (1.0/12.0)*k1_c1 + (1.0/3.0)*k2_c1
                -(1.0/12.0)*k3_c1,
                c2 + (1.0/12.0)*k1_c2 + (1.0/3.0)*k2_c2
                -(1.0/12.0)*k3_c2,
                c3 + (1.0/12.0)*k1_c3 + (1.0/3.0)*k2_c3
                -(1.0/12.0)*k3_c3);


k5_P1 = h*dP1dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);
```

```
k5_P2 = h*dP2dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);
k5_P3 = h*dP3dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);
k5_c1 = h*dc1dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);
k5_c2 = h*dc2dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);
k5_c3 = h*dc3dt(P1 - (1.0/16.0)*k1_P1 +(9.0/8.0)*k2_P1
                -(3.0/16.0)*k3_P1 - (3.0/8.0)*k4_P1,
                P2 - (1.0/16.0)*k1_P2 +(9.0/8.0)*k2_P2
                -(3.0/16.0)*k3_P2 - (3.0/8.0)*k4_P2,
                P3 - (1.0/16.0)*k1_P3 +(9.0/8.0)*k2_P3
                -(3.0/16.0)*k3_P3 - (3.0/8.0)*k4_P3,
                c1 - (1.0/16.0)*k1_c1 +(9.0/8.0)*k2_c1
                -(3.0/16.0)*k3_c1 - (3.0/8.0)*k4_c1,
                c2 - (1.0/16.0)*k1_c2 +(9.0/8.0)*k2_c2
                -(3.0/16.0)*k3_c2 - (3.0/8.0)*k4_c2,
                c3 - (1.0/16.0)*k1_c3 +(9.0/8.0)*k2_c3
```

```
                -(3.0/16.0)*k3_c3 - (3.0/8.0)*k4_c3);

k6_P1 = h*dP1dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);
k6_P2 = h*dP2dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);
k6_P3 = h*dP3dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);
k6_c1 = h*dc1dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);
k6_c2 = h*dc2dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
```

```
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);
k6_c3 = h*dc3dt(P1 + (9.0/8.0)*k2_P1 -(3.0/8.0)*k3_P1
                -(3.0/4.0)*k4_P1 + (1.0/2.0)*k5_P1,
                P2 + (9.0/8.0)*k2_P2 -(3.0/8.0)*k3_P2
                -(3.0/4.0)*k4_P2 + (1.0/2.0)*k5_P2,
                P3 + (9.0/8.0)*k2_P3 -(3.0/8.0)*k3_P3
                -(3.0/4.0)*k4_P3 + (1.0/2.0)*k5_P3,
                c1 + (9.0/8.0)*k2_c1 -(3.0/8.0)*k3_c1
                -(3.0/4.0)*k4_c1 + (1.0/2.0)*k5_c1,
                c2 + (9.0/8.0)*k2_c2 -(3.0/8.0)*k3_c2
                -(3.0/4.0)*k4_c2 + (1.0/2.0)*k5_c2,
                c3 + (9.0/8.0)*k2_c3 -(3.0/8.0)*k3_c3
                -(3.0/4.0)*k4_c3 + (1.0/2.0)*k5_c3);



k7_P1 = h*dP1dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                -(16.0/11.0)*k6_P1,
                P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
                -(16.0/11.0)*k6_P2,
                P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                -(16.0/11.0)*k6_P3,
                c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
                +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                -(16.0/11.0)*k6_c1,
                c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                -(16.0/11.0)*k6_c2,
                c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                -(16.0/11.0)*k6_c3);
k7_P2 = h*dP2dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                -(16.0/11.0)*k6_P1,
                P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
                -(16.0/11.0)*k6_P2,
                P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                -(16.0/11.0)*k6_P3,
                c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
                +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                -(16.0/11.0)*k6_c1,
                c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                -(16.0/11.0)*k6_c2,
                c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                -(16.0/11.0)*k6_c3);
k7_P3 = h*dP3dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                -(16.0/11.0)*k6_P1,
                P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
```

```
                  -(16.0/11.0)*k6_P2,
                  P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                  +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                  -(16.0/11.0)*k6_P3,
                  c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
                  +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                  -(16.0/11.0)*k6_c1,
                  c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                  +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                  -(16.0/11.0)*k6_c2,
                  c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                  +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                  -(16.0/11.0)*k6_c3);
k7_c1 = h*dc1dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                  +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                  -(16.0/11.0)*k6_P1,
                  P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                  +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
                  -(16.0/11.0)*k6_P2,
                  P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                  +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                  -(16.0/11.0)*k6_P3,
                  c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
                  +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                  -(16.0/11.0)*k6_c1,
                  c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                  +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                  -(16.0/11.0)*k6_c2,
                  c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                  +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                  -(16.0/11.0)*k6_c3);
k7_c2 = h*dc2dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                  +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                  -(16.0/11.0)*k6_P1,
                  P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                  +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
                  -(16.0/11.0)*k6_P2,
                  P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                  +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                  -(16.0/11.0)*k6_P3,
                  c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
                  +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                  -(16.0/11.0)*k6_c1,
                  c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                  +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                  -(16.0/11.0)*k6_c2,
                  c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                  +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                  -(16.0/11.0)*k6_c3);
k7_c3 = h*dc3dt(P1 + (9.0/44.0)*k1_P1 -(9.0/11.0)*k2_P1
                  +(63.0/44.0)*k3_P1 + (18.0/11.0)*k4_P1
                  -(16.0/11.0)*k6_P1,
                  P2 + (9.0/44.0)*k1_P2 -(9.0/11.0)*k2_P2
                  +(63.0/44.0)*k3_P2 + (18.0/11.0)*k4_P2
                  -(16.0/11.0)*k6_P2,
                  P3 + (9.0/44.0)*k1_P3 -(9.0/11.0)*k2_P3
                  +(63.0/44.0)*k3_P3 + (18.0/11.0)*k4_P3
                  -(16.0/11.0)*k6_P3,
                  c1 + (9.0/44.0)*k1_c1 -(9.0/11.0)*k2_c1
```

```
                +(63.0/44.0)*k3_c1 + (18.0/11.0)*k4_c1
                -(16.0/11.0)*k6_c1,
                c2 + (9.0/44.0)*k1_c2 -(9.0/11.0)*k2_c2
                +(63.0/44.0)*k3_c2 + (18.0/11.0)*k4_c2
                -(16.0/11.0)*k6_c2,
                c3 + (9.0/44.0)*k1_c3 -(9.0/11.0)*k2_c3
                +(63.0/44.0)*k3_c3 + (18.0/11.0)*k4_c3
                -(16.0/11.0)*k6_c3);


P1 = P1 + ((11.0/120.0)*k1_P1 + (27.0/40.0)*k3_P1
           +(27.0/40.0)*k4_P1 -(4.0/15.0)*k5_P1
           -(4.0/15.0)*k6_P1 + (11.0/120.0)*k7_P1);
P2 = P2 + ((11.0/120.0)*k1_P2 + (27.0/40.0)*k3_P2
           +(27.0/40.0)*k4_P2 -(4.0/15.0)*k5_P2
           -(4.0/15.0)*k6_P2 + (11.0/120.0)*k7_P2);
P3 = P3 + ((11.0/120.0)*k1_P3 + (27.0/40.0)*k3_P3
           +(27.0/40.0)*k4_P3 -(4.0/15.0)*k5_P3
           -(4.0/15.0)*k6_P3 + (11.0/120.0)*k7_P3);
c1 = c1 + ((11.0/120.0)*k1_c1 + (27.0/40.0)*k3_c1
           + (27.0/40.0)*k4_c1 -(4.0/15.0)*k5_c1
           -(4.0/15.0)*k6_c1 + (11.0/120.0)*k7_c1);
c2 = c2 + ((11.0/120.0)*k1_c2 + (27.0/40.0)*k3_c2 +
           (27.0/40.0)*k4_c2 -(4.0/15.0)*k5_c2
           -(4.0/15.0)*k6_c2 + (11.0/120.0)*k7_c2);
c3 = c3 + ((11.0/120.0)*k1_c3 + (27.0/40.0)*k3_c3
           + (27.0/40.0)*k4_c3 -(4.0/15.0)*k5_c3
           -(4.0/15.0)*k6_c3 + (11.0/120.0)*k7_c3);

//Updating the time with the stepsize
t0 = t0 + h;

//Calculating new directional scale factors
a_new = pow(P1*P2*P3, 1.0/6.0);
a1_new = sqrt((P2*P3)/P1);
a2_new = sqrt((P1*P3)/P2);
a3_new = sqrt((P1*P2)/P3);


//Calculating and outputting data to csv file if
//Point_Counter == 1
if(Point_Counter == 1){
//Calculating Relative Error
C = -((P1*P2*P3)/(8*M_PI*::G*::gamma*::gamma*::Delta))*
(sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)+
sin(sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/(abs(P2*P3)))*c1)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3)+
sin(sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/(abs(P1*P3)))*c2)*
sin(sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/(abs(P1*P2)))*c3))
+ (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1) +
(P1*P1*P3*P3)/(P2*P2) + (P1*P1*P2*P2)/(P3*P3))
-(1/(16*M_PI*::G))*(P1*P1 + P2*P2 + P3*P3);

C_rel = C/C_den;

//Calculating Kasner Exponents
k1 = (a_new/(3*a1_new))*((a1_new-a1_old)/(a_new-a_old));
```

```
    k2 = (a_new/(3*a2_new))*((a2_new-a2_old)/(a_new-a_old));
    k3 = (a_new/(3*a3_new))*((a3_new-a3_old)/(a_new-a_old));

    //Calculating Potentials
    U0 = (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1) +
    (P1*P1*P3*P3)/(P2*P2) + (P1*P1*P2*P2)/(P3*P3))
    -(1/(16*M_PI*::G))*(P1*P1 + P2*P2 + P3*P3);

    U1 = (1/(32*M_PI*::G))*((P2*P2*P3*P3)/(P1*P1));
    U2 = (1/(32*M_PI*::G))*((P1*P1*P3*P3)/(P2*P2));
    U3 = (1/(32*M_PI*::G))*((P1*P1*P2*P2)/(P3*P3));
    U23 = -(1/(16*M_PI*::G))*P1*P1;
    U13 = -(1/(16*M_PI*::G))*P2*P2;
    U12 = -(1/(16*M_PI*::G))*P3*P3;

    //Outputting data
    outputFile << t0 << "," << P1 << "," << P2 << "," << P3 << ","
    << c1 << "," << c2 << "," << c3 << "," << C_rel<< ","
    << a1_new << "," << a2_new << "," << a3_new << ","
    << a_new <<"," << k1 << "," << k2 << "," << k3<< ","
    << U0 << "," << U1 << "," << U2 << "," << U3 <<  ","
    << U23 << "," << U13 << "," << U12 << std::endl;
    }

    Point_Counter++;

    //Resetting Point_Counter if it reaches Points_Per_Output
    if(Point_Counter > Points_Per_Output){
        Point_Counter = 1;
    }

    //Canonical variable transformation if LQG bounce occurs
    if (i!=0 && a_older > a_old && a_new > a_old){
    c1 = (M_PI + sqrt((abs(P1)*::Delta*::l_p1*::l_p1)/
                      (abs(P2*P3)))*c1)/sqrt((abs(P1)*
                      ::Delta*::l_p1*::l_p1)/(abs(P2*P3)));
    c2 = (M_PI + sqrt((abs(P2)*::Delta*::l_p1*::l_p1)/
                      (abs(P1*P3)))*c2)/sqrt((abs(P2)*
                      ::Delta*::l_p1*::l_p1)/(abs(P1*P3)));
    c3 = (M_PI + sqrt((abs(P3)*::Delta*::l_p1*::l_p1)/
                      (abs(P1*P2)))*c3)/sqrt((abs(P3)*
                      ::Delta*::l_p1*::l_p1)/(abs(P1*P2)));
    }


//This is a block of code that will end the calculation if
//all variables are Nan ie. the calculation fails
if(isnan(P1) && isnan(P2) && isnan(P3) && isnan(c1)
   && isnan(c2) && isnan(c3)){
break;
}

// This block of code outputs progress to the command window
if(Point_Counter_2 == 1){
cout << t0/t;
printf("\n");
}
```

```
Point_Counter_2++;

//Resetting_2 Point_Counter if it reaches Points_Per_Output_2
if(Point_Counter_2 > Points_Per_Output_2){
Point_Counter_2 = 1;
}

//Updating directional scale factor values
a1_older = a1_old;
a2_older = a2_old;
a3_older = a3_old;
a_older = a_old;

a1_old = a1_new;
a2_old = a2_new;
a3_old = a3_new;
a_old = a_new;

}

return c1;
}

int main(int argc, char** argv)
{

long double t0, t, p1_0, p2_0, p3_0, c1_0, c2_0, c3_0, h;


//Test For Paper
p1_0 = 8.039999999999999e+25;
p2_0 = 9e23;
p3_0 = 1e30;
c1_0 = 30000000.000000008134767904279735241544;
c2_0 = 3.72232352997251587e+10;
c3_0 = -2.24994857268156738e+03;
h = 1e-39;
t = 2.5455e-33;
t0 = 0.0;

    printf("\nThe value of y at d is : %f",
            rungeKutta(t0, t, p1_0, p2_0, p3_0, c1_0,
                        c2_0, c3_0, h));
    return 0;
}
```

Vita

Candidate's full name: Timothy Ernest Blackmore
University attended (with dates and degrees obtained): BSc. Mathematics-Physics (Hon), University of New Brunswick, 2021
Publications: NA
Conference Presentations: Atlantic General Relativity Meeting 2021, UNB Graduate Research Conference 2021, CAP 2021 Virtual Congress, AUPAC 2020 & 2019