

Periodic Solutions to the n -Body Problem

by

Joel A. Dyck

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
September 2015

© Copyright 2015 by Joel A. Dyck

Thesis advisor

Dr. William Kocay

Author

Joel A. Dyck

Periodic Solutions to the n -Body Problem

Abstract

This thesis develops methods to identify periodic solutions to the n -body problem by representing gravitational orbits with Fourier series. To find periodic orbits, a minimization function was developed that compares the second derivative of the Fourier series with Newtonian gravitation acceleration and modifies the Fourier coefficients until the orbits match. Software was developed to minimize the function and identify the orbits using gradient descent and quadratic curves. A Newtonian gravitational simulator was developed to read the initial orbit data and numerically simulate the orbits with accurate motion integration, allowing for comparison to the Fourier series orbits and investigation of their stability. The orbits found with the programs correlate with orbits from literature, and a number remain stable when simulated.

Acknowledgments

I would like to thank my advisor Dr. Kocay for his support and guidance. Your knowledge and insight have been an inspiration to me. Thank you to my parents, for instilling a sense of wonder and a thirst for knowledge from an early age. Finally, my utmost thanks to my wife and children for their unending support, encouragement, and patience through this entire process.

To my family

Contents

Abstract	ii
Acknowledgments	iii
Dedication	iv
Table of Contents	vi
List of Tables	vii
List of Figures	ix
1 Introduction	1
2 Periodic Orbits	3
2.1 Imperfect Orbits	4
3 Related Work	7
4 Algorithm Concepts	10
4.1 Integration of Motion	10
4.1.1 Euler Method of Integration	11
4.1.2 Runge-Kutta Integration	12
4.2 Steepest Descent	13
5 The n-Body Problem and Periodic Orbits	15
5.1 Newtonian Gravitation	15
5.2 Fourier Series	17
5.3 Orbit Categories	19
5.3.1 Choreographed Orbits	20
5.3.2 Unchoreographed Orbits	21
6 Primary Algorithms	22
6.1 Minimization Function	23
6.2 Additional Constraints	24
6.3 Modified Gradient Descent	25

6.4	Quadratic Curve Fitting	29
7	Software	37
7.1	Generating Orbits	37
7.1.1	Main Program Loop	40
7.1.2	Additional User Control	43
7.1.3	Number of Variables	45
7.2	Newtonian Gravitation Simulator	46
7.2.1	Initial Values	47
7.2.2	Main Program Loop	48
8	Orbits Found	51
8.1	Choreographed Orbits	53
8.2	Unchoreographed Orbits	62
9	Additional Observations	67
9.1	The Lemniscate of Bernoulli	67
9.2	Stable Precession of Periodic Orbits	69
10	Future Work	74
11	Conclusion	76
	Appendix A Fourier Series Coefficients	79
	Bibliography	92

List of Tables

8.1	Choreographed three-body circle initial values	54
8.2	Choreographed four-body circle initial values	54
8.3	Choreographed three-body figure-eight initial values	55
8.4	Choreographed four-body trefoil initial values	56
8.5	Choreographed four-body chain initial values	57
8.6	Choreographed four-body inner loop initial values	58
8.7	Choreographed five-body figure-eight initial values	59
8.8	Choreographed five-body pretzel loop initial values	60
8.9	Choreographed five-body trefoil initial values	61
8.10	Three-body circle initial values	62
8.11	Three-body circled cross initial values	63
8.12	Three-body enveloped figure-eight initial values	64
8.13	Four-body nuclear symbol initial values	65
8.14	Three-body offset rings initial values	66
9.1	Initial values for figure-eight orbit	68
A.1	Choreographed three-body circle coefficients for Fig. 8.1	80
A.2	Choreographed four-body circle coefficients for Fig. 8.2	80
A.3	Choreographed three-body figure-eight coefficients for Fig. 8.3	81
A.4	Choreographed four-body trefoil coefficients for Fig. 8.4	82
A.5	Choreographed four-body chain coefficients for Fig. 8.5	83
A.6	Choreographed four-body inner loop coefficients for Fig. 8.6	84
A.7	Choreographed five-body figure-eight coefficients for Fig. 8.7	85
A.8	Choreographed five-body pretzel loop coefficients for Fig. 8.8	86
A.9	Choreographed five-body trefoil coefficients for Fig. 8.9	87
A.10	Three-body circle coefficients for Fig. 8.10	88
A.11	Three-body circled cross coefficients for Fig. 8.11	89
A.12	Three-body enveloped figure-eight coefficients for Fig. 8.12	90

A.13 Four-body nuclear symbol coefficients for Fig. 8.13	91
--	----

List of Figures

7.1	Choreographed orbits with error overlay	44
8.1	Choreographed three-body circle	53
8.2	Choreographed four-body circle	54
8.3	Choreographed three-body figure-eight	55
8.4	Choreographed four-body trefoil	56
8.5	Choreographed four-body chain	57
8.6	Choreographed four-body inner loop	58
8.7	Choreographed five-body figure-eight	59
8.8	Choreographed five-body pretzel loop	60
8.9	Choreographed five-body trefoil	61
8.10	Three-body circle	62
8.11	Three-body circled cross	63
8.12	Three-body enveloped figure-eight	64
8.13	Four-body nuclear symbol	65
8.14	Three-body offset rings	66
9.1	Figure-eight orbit with superimposed Lemniscate of Bernoulli	69
9.2	Precession of choreographed figure-eight orbits	71
9.3	Precession of choreographed circular orbits	72

Chapter 1

Introduction

Gravity affects the motion of every body in the universe, from interstellar dust to black holes, causing many fascinating patterns and motions to occur. Simulating the entire universe is currently impossible, but simulating a limited number of bodies, or n -body simulation, can lead to interesting results. A specific class of the n -body problem is the periodic orbit in which all the bodies in the system follow a repeating pattern of motion.

We will be searching for periodic solutions to the n -body problem. To facilitate in finding periodic orbits, software was developed to generate and simulate potential orbits.

The orbit generation software searches for orbits by defining the paths of potential orbits with Fourier series and then iteratively modifying the Fourier series coefficients to approach a potential solution. The modifications needed are determined by comparing the Fourier series orbits to orbits defined by Newton's law of

universal gravitation. The difference between the two orbits is minimized over successive iterations using a minimization function. We will look at how the function is created and at the algorithms used to efficiently minimize it.

Once potential periodic orbits are generated, they are run in the n -body gravitational simulator. We will observe the paths of the bodies under the effect of gravity, and compare the results to the orbits defined by the generated Fourier series.

Chapter 2

Periodic Orbits

Periodic orbits are found in the Kepler two-body problem. In its most pure form a periodic orbit is where some number of bodies, given a starting position and velocity, move through space affected only by gravitational forces from the other bodies, and all return to their original positions and velocities after some period of time. Nearly periodic orbits can be loosely described within these parameters. In many natural cases there is a large primary body with multiple secondary bodies orbiting it, such as the planets orbiting the Sun. In other cases, there are two bodies in a periodic orbit where one body is much more massive than the other, such as the Moon orbiting the Earth. While forces are exerted on both bodies, the difference in mass results in the larger body staying relatively stationary while the smaller body orbits around it on an elliptical path. The point at which the smaller body is closest to the larger body is the perihelion and is a convenient location to define as the start and end of each period.

Other forms of orbits exist such binary or multiple star systems. In these cases stars (or other large masses such as black holes) have a smaller order of magnitude difference between their masses and larger movement can be observed in all the bodies, in effect orbiting each other. The same gravitational laws controlling planets orbiting a sun define the motions of these systems, but more exotic orbits may be present when every body has a larger impact on each other.

In all systems the bodies can be seen as orbiting a center of mass, which is a point where the weighted average of the masses of all the bodies in the system cancel each other out. For example, in a two-body system where both masses are equal, the center of mass will be the point halfway between the two masses. In cases where one body is significantly larger than the other bodies in a system the center of mass may be inside the larger body, but this is not a requirement and in cases of bodies with masses of similar orders of magnitude, the center of mass will generally not be inside any of the bodies.

2.1 Imperfect Orbits

Other orbits exist that do not fit the description quite as cleanly. A prime example of such an orbit is that of Mercury and the Sun. Mercury is considered to be in a periodic orbit around the Sun, as it continues to travel on a stable elliptical orbit over a fixed period of time, but it does not come back to its original starting point. This is referred to as its perihelion precession. Mazarico et al. [7] used data from the MESSENGER probe to determine that Mercury has an orbit which takes

87.969216879 days ± 6 seconds. However, observational data has determined the perihelion of Mercury precesses at a rate of 5599.74 ± 0.41 arcseconds per century [4]. Much of this precession can be explained due to the gravitational effects of the other planets, but Einstein [5] showed that approximately 43 arcseconds per century can be accounted for by effects described by the general theory of relativity. While this precession of each orbit is small, it means that after a single orbit Mercury will not return to its original location.

Even taking precession into account, after one full rotation of the Mercury's perihelion around the Sun the perihelion will not be back at the same point it started. For this to happen, the precession rate would need to be a perfect multiple of Mercury's orbit length. This shows that due to effects such as precession, simply searching for orbits where all the bodies return to their exact initial locations excludes too many possible orbits, and so some perturbation of orbits must be allowed to occur.

The same ideas used to define the movements of a planet around a star or multiple star systems can be extended to apply to all celestial bodies such as planetary systems, asteroids, and comets. For the purposes of this thesis, some restrictions will be placed on the systems being used to focus on an idealized subset of the periodic orbits. All of the bodies focused on will be of equal mass. This will simplify a number of the formulas and concepts without any loss to the general ideas being proposed. All of the methods used could be easily extended to bodies of differing mass. Additionally, the period length of all orbits will be fixed as 2π . A period of any length could realistically be used, but this assumption simplifies the Fourier series formulas that will be discussed later, and a period of any length can be mapped to a period of

length 2π . Other restrictions on the orbits will be defined in future sections as they become relevant.

Chapter 3

Related Work

Periodic n -body orbits is currently a popular topic of research. A simple 3-body stable system in an equilateral triangle following a circular path has been known since it was found by Lagrange in 1772 [10], but other patterns were unknown until very recently.

Chenciner and Montgomery [2] demonstrated a new family of 3-body, stable, periodic orbits in the form of a figure-eight. They showed that by minimizing the difference between the potential and kinetic energy over the period of the orbit, they could find an orbit path which was stable and periodic.

Simó [10] provided starting positions and velocities for the figure eight described by Chenciner and Montgomery, and he discovered a number of new orbit systems where the objects all followed the same path, which he called choreographies. He expanded [11] on his findings and demonstrated how he found the minima of functions to identify choreographic orbits and investigated [12] how small perturbations could

result in dynamic orbits which maintained stability.

Chunhua and Zhang [3] proved the existence of $2n$ -body (for $n \geq 2$) stable periodic orbits in R^3 where the bodies follow two perpendicular circular paths. They did so by defining a minimizer which determined the starting positions and velocities of the points such that they followed periodic orbits, and then proving that the orbits were stable and without collisions.

Roberts [9] analyzed the figure eight solution to determine its stability. He proved it was linearly stable by reducing the problem with orbit symmetries and then analyzed the remaining problem using differential geometry to prove the reduced problem was stable.

Zhang and Zhou [15] proved that there exists at least one periodic solution without collisions for any number of bodies in an n -body system. They controlled their functions so the orbits would fall into a similar pattern for any number of objects.

Zhang et al. [16] proved the existence of new stable orbits containing bodies with non-uniform mass. Previous research had focused on bodies with uniform mass in order to simplify the equations, but by adjusting the mass they were able to develop new periodic orbits with unique shapes.

Vanderbei [13] discovered a large number of new periodic orbits, both choreographed and not, by starting with random Fourier series and minimizing the difference between the potential and kinetic energies in the orbits until local minima were found. He was able to find the known stable triangle and figure eight orbits, but all of the other orbits he found proved to be unstable.

Šuvakov and Dmitrašinović [14] discovered new classes of 3-body orbits by cal-

culating the topology of the orbits on the surface of a sphere and determining the planar periodic orbits from the results.

Aarseth [1] describes different ways of writing n -body simulations. He provides information on selecting simulation time steps and how to adjust them dynamically, how to accurately represent a continuous system with discrete time steps and correct for error, and how to determine the amount of error which accumulates in a simulation.

All of these works show that there are many possible methods available for finding new periodic n -body orbits.

Chapter 4

Algorithm Concepts

This chapter will introduce some of the algorithms that were required for developing applications to find and simulate periodic n -body orbits. The concepts considered include Runge-Kutta integration and an introduction to the Steepest Descent algorithm.

4.1 Integration of Motion

An inherent problem with simulating gravitational orbit mechanics is that gravitational motion is a continuous function, but simulations are based on discrete functions. To reduce the errors that can very quickly add up during a simulation, a small timestep is used. At each timestep the instantaneous position and velocity of each body is known. With the positions the acceleration due to gravity can be calculated using Newton's law of universal gravitation, and the requirement is to find the position and velocity of each body at the next timestep.

4.1.1 Euler Method of Integration

The first and simplest form of motion integration tried was to calculate using the Euler method of integration which is used to solve first order differential equations of the form $x'(t) = f(t, x(t))$ by breaking the time into some small number of steps and having a step size h where $h = 1/steps$. The integration then iterates with the form

$$\begin{aligned}x_{n+1} &= x_n + hf(t_n, x_n) \\t_{n+1} &= t_n + h\end{aligned}\tag{4.1}$$

To solve for motion on the x -axis, this requires the initial position x_0 , initial velocity v_0 . We can define the velocity as $v(t) = x'(t)$ and the acceleration in a gravitational simulator is determined by Newton's law of universal gravitation, indicated as $f(t, x(t))$. We can now define the system as

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ f(t, x(t)) \end{pmatrix}\tag{4.2}$$

Using Euler's method we then get

$$\begin{aligned}x_{n+1} &= x_n + hv_n \\v_{n+1} &= v_n + hf(t_n, x_n(t)) \\t_{n+1} &= t_n + h\end{aligned}\tag{4.3}$$

Euler's method is considered to not be very accurate, so to try to get more accuracy, the number of steps can be increased (decreasing the step size). The primary limitation of this method is that the acceleration due to gravity is only calculated once at the beginning of each step, and then it is used as a constant over the entire step.

This introduces error that can increase quickly. At the next timestep, the positions and velocities will be slightly wrong, which will produce an incorrect gravitational acceleration, causing the next timestep to be even more inaccurate.

4.1.2 Runge-Kutta Integration

A much more accurate method of integration is the Runge-Kutta method [8]. The software uses the fourth-order Runge-Kutta method, also known as RK4. This starts at time t_0 with the initial position x_0 , initial velocity v_0 , and initial acceleration a_0 , just as with the Euler method. It also uses a value h , which is the step size to take. The algorithm starts with the definition of

$$x' = f(t, x) \tag{4.4}$$

Using x_n at time t_n , x_{n+1} can be calculated at time $t_n + h$ by calculating several steps between t_n and t_{n+h} and using a weighted average of the values. It starts by first calculating the first derivative at the start time which is

$$k_1 = f(t_n, x_n) \tag{4.5}$$

This gives the slope of the function at the beginning of the time interval, just as the Euler method. Instead of using this slope over the full timestep, the result then gets fed back into the equation at the half-way point between the start and end time as

$$k_2 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1\right) \tag{4.6}$$

This gives an approximation of the slope halfway through the interval. Feeding it back into exactly the same formula at the midpoint results in another approximation

of the slope at the midpoint, starting with a different initial slope. This gives

$$k_3 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2\right) \quad (4.7)$$

This second approximation at the midpoint is now used to determine the slope at the end of the time interval with the formula

$$k_4 = f(t_n + h, x_n + hk_3) \quad (4.8)$$

Now that the slope for the beginning, the end, and twice for the middle, a weighted average of the slopes can be used to calculate the final answer of

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (4.9)$$

Using this function gives a much more accurate representation of the behaviour of the function without relying on the incredibly small time-steps that would be required for the Euler method.

4.2 Steepest Descent

Press et al. [8] describe the steepest descent algorithm as taking a function at a point, finding the gradient, or slope, of the function at that point which is the direction of steepest descent, and then following the gradient to the minimum along that direction. This is repeated many times, recalculating the gradient and approaching closer and closer to the minimum at each step. Given some function $f(\mathbf{x})$, γ which is a value to control how much influence the gradient has on each step, and the current value \mathbf{x}_1 , moving from a point \mathbf{x}_1 to point \mathbf{x}_2 on the function is done by calculating

$$\mathbf{x}_2 = \mathbf{x}_1 - \gamma \nabla f(\mathbf{x}_1) \quad (4.10)$$

There are some limitations to the steepest descent algorithm. It is able to quickly and easily approach a minimum when starting from far away, but as it gets closer to the minimum of the function, the minimization slows down and attempting to make it speed up can cause the function to overshoot the minimum. Additionally, in complex functions with a large number of parameters, modifying all the parameters to follow a single gradient change may not approach the actual minimum of the function but may simply cause the path to zigzag using many short line segments. Limitations of the steepest descent algorithm and the modifications made to create a modified gradient descent to find periodic orbits will be explored in greater detail in Chapter 6.

Chapter 5

The n -Body Problem and Periodic Orbits

5.1 Newtonian Gravitation

Newton's law of universal gravitation states that every body attracts every other body along a line intersecting the two bodies with a force equal to

$$F = G \frac{m_1 m_2}{r^2} \tag{5.1}$$

where F is the force between the two bodies, G is the gravitational constant defined by the International Astronomical Union [6] as $6.67428 \times 10^{-11} \text{ N(m/kg)}^2$, m_1 is the mass of the first body, m_2 is the mass of the second body, and r is the distance between the centers of masses of the bodies. This relationship holds for any bodies in the universe, although due to the force falling off at a rate inversely proportional to the square of the distance between the two bodies, distant objects exert much less

force on each other compared to closer objects of the equivalent masses.

When there are only two bodies in a closed system, their motion is defined by Kepler's Laws. Given the positions, masses, and starting velocities of the two bodies, their positions and velocities can be determined at any point in time.

Newtonian gravitation can be extended to multiple bodies via a simple summation of the forces. Say there are n bodies 1, 2, ..., n with the positions on a plane of $\mathbf{p}_1(t)$, $\mathbf{p}_2(t)$, ..., $\mathbf{p}_n(t)$ at time t where

$$\mathbf{p}_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} \quad (5.2)$$

The force on body i by body j is

$$\mathbf{F}_{i,j}(t) = -G \frac{(\mathbf{p}_i(t) - \mathbf{p}_j(t)) m_i m_j}{|\mathbf{p}_i(t) - \mathbf{p}_j(t)|^3} \quad (5.3)$$

where $|\mathbf{p}_i(t) - \mathbf{p}_j(t)|$ is the distance between the two bodies. The total force on a body is the sum of the forces from all the other bodies, defined as

$$\mathbf{F}_i(t) = -G \sum_{\substack{j=1 \\ j \neq i}}^n \frac{(\mathbf{p}_i(t) - \mathbf{p}_j(t)) m_i m_j}{|\mathbf{p}_i(t) - \mathbf{p}_j(t)|^3} \quad (5.4)$$

When there are 3 or more bodies, there is no general closed solution that can determine the positions and velocities at any given point in time. Instead, the system must be solved numerically given the starting values.

For each body, given that we know the position, we can define its velocity as

$$\mathbf{v}_i(t) = \frac{\partial}{\partial t} \mathbf{p}_i(t) \quad (5.5)$$

and its acceleration as

$$\mathbf{a}_i(t) = \frac{\partial}{\partial t} \mathbf{v}_i(t) \quad (5.6)$$

Since the primary purpose of this research was to investigate the concepts of periodic n -body orbits among equal-mass bodies so to simplify the equations and reduce computational complexity, a few assumptions will be defined:

1. All distances between bodies will be on the order of magnitude of 1
2. The gravitational constant will be defined as $G = 1$
3. All the bodies will have an equal mass of $m_i = 1$
4. All the bodies will exist on a single plane
5. The center of mass will be at the origin

By making these assumptions, the force on a body can be simplified to

$$\mathbf{F}_i(t) = - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_i(t) - \mathbf{p}_j(t)}{|\mathbf{p}_i(t) - \mathbf{p}_j(t)|^3} \quad (5.7)$$

and given $F = ma$ and $m = 1$, finding \mathbf{F}_i provides us with the acceleration of each body in the system. This can be split into separate x and y components as

$$\begin{aligned} F_{i_x}(t) &= - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{x_i(t) - x_j(t)}{((x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2)^{\frac{3}{2}}} \\ F_{i_y}(t) &= - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{y_i(t) - y_j(t)}{((x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2)^{\frac{3}{2}}} \end{aligned} \quad (5.8)$$

5.2 Fourier Series

A Fourier series is a method of defining a periodic function as a combination of sines and cosines with coefficients. In its simplest form, an elliptical periodic orbit of

length 2π could be defined using the Fourier series

$$\mathbf{z}(t) = \mathbf{a}_0 + \mathbf{a}_1 \cos t + \mathbf{b}_1 \sin t \quad (5.9)$$

where $2\mathbf{a}_0$ is an offset from the origin and the coefficients \mathbf{a}_1 and \mathbf{b}_1 control the size and shape of the orbit.

To more easily calculate the complex orbits, the x and y components can be separated out as

$$x(t) = a_0 + a_1 \cos t + b_1 \sin t \quad (5.10)$$

and

$$y(t) = c_0 + c_1 \cos t + d_1 \sin t \quad (5.11)$$

This was done in the software, with the x and y coefficients handled completely independently.

In order to get more complex orbits, more terms must be added to the Fourier series by adding more sine and cosine terms with coefficients. The position of a body can be determined using the formula

$$\mathbf{z}(t) = \mathbf{a}_0 + \mathbf{a}_1 \cos t + \mathbf{b}_1 \sin t + \mathbf{a}_2 \cos 2t + \mathbf{b}_2 \sin 2t + \mathbf{a}_3 \cos 3t + \mathbf{b}_3 \sin 3t + \dots \quad (5.12)$$

or in general

$$\mathbf{z}(t) = \mathbf{a}_0 + \sum_{k=1}^{\infty} [\mathbf{a}_k \cos(kt) + \mathbf{b}_k \sin(kt)] \quad (5.13)$$

It is possible to perfectly represent any periodic orbit with an infinite number of terms, however the software will have limits on the number of terms, which will be discussed in Chapter 7.

The Fourier series provides the position of a body in a periodic orbit at any point in time. The first derivative of the series can be used to determine the velocity of the body at any point in the orbit according to the Fourier series. This gives the formula

$$\mathbf{z}'(t) = -\mathbf{a}_1 \sin t + \mathbf{b}_1 \cos t - 2\mathbf{a}_2 \sin 2t + 2\mathbf{b}_2 \cos 2t - 3\mathbf{a}_3 \sin 3t + 3\mathbf{b}_3 \cos 3t + \dots \quad (5.14)$$

which can be generalized as

$$\mathbf{z}'(t) = \sum_{k=1}^{\infty} [-k\mathbf{a}_k \sin(kt) + k\mathbf{b}_k \cos(kt)] \quad (5.15)$$

Continuing on to the second derivative, the acceleration of the body according to the Fourier series can be determined with the formula

$$\mathbf{z}''(t) = -\mathbf{a}_1 \cos t - \mathbf{b}_1 \sin t - 4\mathbf{a}_2 \cos 2t - 4\mathbf{b}_2 \sin 2t - 9\mathbf{a}_3 \cos 3t - 9\mathbf{b}_3 \sin 3t + \dots \quad (5.16)$$

Finding the general form of this equation gives

$$\mathbf{z}''(t) = -\sum_{k=1}^{\infty} [k^2\mathbf{a}_k \cos(kt) + k^2\mathbf{b}_k \sin(kt)] \quad (5.17)$$

This acceleration formula will play a key role in identifying potentially stable orbits.

5.3 Orbit Categories

The orbits being considered will fall into one of two groups: choreographed orbits and unchoreographed orbits. It is entirely possible for orbits from the two groups to have the same behaviour, but they will be found in different manners.

5.3.1 Choreographed Orbits

A choreographed orbit is one in which a single Fourier series can be used to define the motion of all the bodies in the system. All of the bodies will follow the same path and appear to be chasing each other. The positions of the bodies can be determined using the same formula for all the bodies by simply offsetting the time used in the formula by a fixed ratio along the period of the function. Given a series $\mathbf{z}(t)$ with a period of 2π and n bodies, the position of each body i can be found by evaluating

$$\mathbf{p}_i(t) = \mathbf{z}\left(t + \frac{i \cdot 2\pi}{n}\right) \quad 0 \leq i < n \quad (5.18)$$

For example, consider the simple case of 3 bodies moving in a circle around the origin with a radius of 1 unit. In this case, the shape of the orbit can be defined using the formulas $x(t) = \cos(t)$ and $y(t) = \sin(t)$. Using equation 5.18 the positions of each body can be determined using the same formulas. At time t the positions would be

$$\begin{aligned} x_0 &= \cos(t) & y_0 &= \sin(t) \\ x_1 &= \cos\left(t + \frac{1 \cdot 2\pi}{3}\right) & y_1 &= \sin\left(t + \frac{1 \cdot 2\pi}{3}\right) \\ x_2 &= \cos\left(t + \frac{2 \cdot 2\pi}{3}\right) & y_2 &= \sin\left(t + \frac{2 \cdot 2\pi}{3}\right) \end{aligned} \quad (5.19)$$

The same time offsetting may be done with the first and second derivatives of the Fourier series to determine the velocities and accelerations of every body in the orbit.

5.3.2 Unchoreographed Orbits

Unchoreographed orbits exist when each body has a separate Fourier series used to describe its motion, $\mathbf{z}_1(t)$, $\mathbf{z}_2(t)$, etc. In this case the orbits as described by the Fourier series are completely uncoupled from the orbits of the other bodies. To calculate the values at a given time t , each function at t is calculated without any offsets. Given the appropriate sets of Fourier series this of course may result in the same values as a choreographed orbit, such as three sets of Fourier series which would result in the identical orbits as the choreographed 3-body circle. However, it is also possible to have much more diverse orbits, such as two bodies in a cross pattern with a third body circling around it which will be seen in Chapter 8.

Chapter 6

Primary Algorithms

The concepts discussed can now be brought together to identify periodic orbits, starting with the two distinct acceleration formulas as defined earlier. For the body i , the first acceleration formula comes from finding the force due to Newtonian gravitation giving the equation

$$\mathbf{F}_i(t) = - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_i(t) - \mathbf{p}_j(t)}{|\mathbf{p}_i(t) - \mathbf{p}_j(t)|^3} \quad (6.1)$$

which can be split into the components

$$\begin{aligned} F_{i_x}(t) &= - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{x_i(t) - x_j(t)}{\left((x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \right)^{\frac{3}{2}}} \\ F_{i_y}(t) &= - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{y_i(t) - y_j(t)}{\left((x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \right)^{\frac{3}{2}}} \end{aligned} \quad (6.2)$$

The position of each body comes from

$$\mathbf{z}_i(t) = \mathbf{a}_{i_0} + \sum_{k=1}^{\infty} [\mathbf{a}_{i_k} \cos(kt) + \mathbf{b}_{i_k} \sin(kt)] \quad (6.3)$$

which can be split into

$$\begin{aligned} x_i(t) &= a_{i_0} + \sum_{k=1}^{\infty} [a_{i_k} \cos(kt) + b_{i_k} \sin(kt)] \\ y_i(t) &= c_{i_0} + \sum_{k=1}^{\infty} [c_{i_k} \cos(kt) + d_{i_k} \sin(kt)] \end{aligned} \quad (6.4)$$

The second acceleration formula is the second derivative of the Fourier series given as

$$\mathbf{z}_i''(t) = - \sum_{k=1}^{\infty} [k^2 \mathbf{a}_{i_k} \cos(kt) + k^2 \mathbf{b}_{i_k} \sin(kt)] \quad (6.5)$$

or in its separate components

$$\begin{aligned} x_i''(t) &= - \sum_{k=1}^{\infty} [k^2 a_{i_k} \cos(kt) + k^2 b_{i_k} \sin(kt)] \\ y_i''(t) &= - \sum_{k=1}^{\infty} [k^2 c_{i_k} \cos(kt) + k^2 d_{i_k} \sin(kt)] \end{aligned} \quad (6.6)$$

6.1 Minimization Function

Looking at the two acceleration functions, the goal is to find the Fourier coefficients such that $\mathbf{z}_i''(t) = \mathbf{F}_i(t)$ for all points on the orbit. One way of achieving this is to define a new function to be minimized as

$$\mathbf{E}_i(t) = (\mathbf{z}_i''(t) - \mathbf{F}_i(t))^2 \quad (6.7)$$

Note that the squaring of the vector is taken to mean that each component, i.e., the x and the y components of the orbit, is individually squared. This means the vector formula can be split into the scalar formulas

$$\begin{aligned} E_{i_x}(t) &= (z_{i_x}''(t) - F_{i_x}(t))^2 \\ E_{i_y}(t) &= (z_{i_y}''(t) - F_{i_y}(t))^2 \end{aligned} \quad (6.8)$$

In order to describe a potentially stable periodic orbit with a Fourier series, these function needs to be minimized independently to 0, approaching from a positive value. If the acceleration as determined by Newtonian gravitation and the second derivative of the Fourier series is equivalent along the entire period of the orbit, then the Fourier series describes a periodic orbit.

To minimize this function, start by solving the Fourier series at a given time to find the x and y coordinates of each body. In the case of choreographed orbits, the Fourier series is solved using the time offsets as described in Section 5.3.1. In the case of unchoreographed orbits, the series for each body is solved individually.

Substituting in the appropriate equations gives the function

$$\mathbf{E}_i(t) = \left(- \sum_{k=1}^{\infty} [k^2 \mathbf{a}_{i_k} \cos(kt) + k^2 \mathbf{b}_{i_k} \sin(kt)] - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_j(t) - \mathbf{p}_i(t)}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right)^2 \quad (6.9)$$

Note that the negative sign from the Newtonian summation has been multiplied through, causing the swapping of \mathbf{p}_i and \mathbf{p}_j . This was done to help some of the simplifications when working with this formula.

6.2 Additional Constraints

Leaving the function as a comparison of the accelerations would work, but an additional constraint can be added to the function to converge on a minimization more rapidly. In order for a periodic orbit system to be stable, the center of mass of all the bodies must not be accelerating, allowing for a frame of reference to be chosen so the center of mass can be defined to be at the origin without loss of generality.

This allows for the definition of a new function

$$\mathbf{g}(t) = \left(\sum_{i=1}^n \mathbf{p}_i(t) \right)^2 \quad (6.10)$$

Note that just as how the main minimization function vector $\mathbf{E}_i(t)$ was handled separately as the scalar E_{i_x} and E_{i_y} , the constraint $\mathbf{g}(t)$ is also split into separate components to be independently minimized to 0 as

$$\begin{aligned} g_x(t) &= \left(\sum_{i=1}^n p_{i_x}(t) \right)^2 \\ g_y(t) &= \left(\sum_{i=1}^n p_{i_y}(t) \right)^2 \end{aligned} \quad (6.11)$$

A new term λ is also defined which will allow for tuning the influence the constraint has on the full function as we're approaching a solution. Putting this all together, the new function to minimize is now

$$\boldsymbol{\phi}_i(t) = \mathbf{E}_i(t) + \lambda_i \mathbf{g}(t) \quad (6.12)$$

When a solution is found, we will have

$$\frac{\partial}{\partial \lambda_i} \boldsymbol{\phi}_i(t) = \mathbf{g}(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (6.13)$$

6.3 Modified Gradient Descent

Given the function to minimize, $\boldsymbol{\phi}_i(t)$, the next step is to define how to minimize the function. This requires finding the gradient of the function, or $\nabla \boldsymbol{\phi}_i(t)$. One of the complexities of the function being minimized is that modifying the coefficients of the Fourier series not only changes the position and acceleration of the bodies, but the

new position of each body will cause the Newtonian acceleration to also be changed as well as modifying the center of mass. Due to this, following the gradient of the function can be difficult, and the standard steepest descent algorithm from Section 4.2 is not sufficient, so we will create a modified gradient descent algorithm. In order to minimize the function, each Fourier series coefficient will be handled separately.

The first step is taking the partial derivatives of $\phi_i(t)$ with respect to the Fourier series coefficients. As an example, consider a 3-body choreographed system and look at $\phi_1(t)$ with respect to \mathbf{a}_{1_2} which is the \mathbf{a}_2 coefficient for the Fourier series defining $\mathbf{p}_1(t)$. The other coefficients follow a similar form.

Start with the full function

$$\phi_1(t) = \mathbf{E}_1(t) + \lambda_1 \mathbf{g}(t) \quad (6.14)$$

Now take the partial derivative as

$$\frac{\partial}{\partial a_{1_2}} \phi_1(t) = \frac{\partial}{\partial a_{1_2}} \mathbf{E}_1(t) + \frac{\partial}{\partial a_{1_2}} \lambda_1 \mathbf{g}(t) \quad (6.15)$$

The two parts of the function $\phi_1(t)$, that is the acceleration difference $\mathbf{E}_1(t)$ and the center of mass constraint $\lambda_1 \mathbf{g}(t)$, will be handled separately to simplify the writing and then combined again afterwards. To keep the formulas manageable, define \mathbf{D}_i for the i th body in an n -body system as

$$\mathbf{D}_i = - \sum_{k=1}^{\infty} [k^2 \mathbf{a}_{1_k} \cos(kt) + k^2 \mathbf{b}_{1_k} \sin(kt)] - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_j(t) - \mathbf{p}_i(t)}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \quad (6.16)$$

Expanding the partial derivative of the acceleration differences gives

$$\begin{aligned}
\frac{\partial}{\partial a_{1_2}} \mathbf{E}_1(t) &= \frac{\partial}{\partial a_{1_2}} (\mathbf{D}_1)^2 \\
&= 2\mathbf{D}_1 \left(\frac{\partial}{\partial a_{1_2}} \mathbf{D}_1 \right) \\
&= 2\mathbf{D}_1 \left(-\frac{\partial}{\partial a_{1_2}} \sum_{k=1}^{\infty} [k^2 \mathbf{a}_{1_k} \cos(kt) + k^2 \mathbf{b}_{1_k} \sin(kt)] \right. \\
&\quad \left. - \frac{\partial}{\partial a_{1_2}} \sum_{\substack{j=1 \\ j \neq i}}^3 \frac{\mathbf{p}_j(t) - \mathbf{p}_1(t)}{|\mathbf{p}_j(t) - \mathbf{p}_1(t)|^3} \right) \\
&= 2\mathbf{D}_1 \left(-4 \cos(2t) - \frac{\partial}{\partial a_{1_2}} \left(\frac{\mathbf{p}_2(t) - \mathbf{p}_1(t)}{|\mathbf{p}_2(t) - \mathbf{p}_1(t)|^3} + \frac{\mathbf{p}_3(t) - \mathbf{p}_1(t)}{|\mathbf{p}_3(t) - \mathbf{p}_1(t)|^3} \right) \right) \\
&= 2\mathbf{D}_1 \left(-4 \cos(2t) - \cos(2t) \left(\frac{3(\mathbf{p}_2(t) - \mathbf{p}_1(t))^2}{|\mathbf{p}_2(t) - \mathbf{p}_1(t)|^5} - \frac{1}{|\mathbf{p}_2(t) - \mathbf{p}_1(t)|^3} \right) \right. \\
&\quad \left. - \cos(2t) \left(\frac{3(\mathbf{p}_3(t) - \mathbf{p}_1(t))^2}{|\mathbf{p}_3(t) - \mathbf{p}_1(t)|^5} - \frac{1}{|\mathbf{p}_3(t) - \mathbf{p}_1(t)|^3} \right) \right)
\end{aligned} \tag{6.17}$$

This can be generalized for \mathbf{a}_{i_k} and \mathbf{b}_{i_k} , the k th Fourier coefficients \mathbf{a} and \mathbf{b} of body i in a system with n bodies, to the equations

$$\begin{aligned}
\frac{\partial}{\partial a_{i_k}} \mathbf{E}_i(t) &= 2\mathbf{D}_i (-k^2 \cos(kt) \\
&\quad - \cos(kt) \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{3(\mathbf{p}_j(t) - \mathbf{p}_i(t))^2}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^5} - \frac{1}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \Bigg) \\
\frac{\partial}{\partial b_{i_k}} \mathbf{E}_i(t) &= 2\mathbf{D}_i (-k^2 \sin(kt) \\
&\quad - \sin(kt) \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{3(\mathbf{p}_j(t) - \mathbf{p}_i(t))^2}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^5} - \frac{1}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \Bigg)
\end{aligned} \tag{6.18}$$

Looking at the partial derivative of the second part of the equation, the center of

mass constraint, again using the coefficient \mathbf{a}_{1_2} , gives

$$\begin{aligned}\frac{\partial}{\partial a_{1_2}} \lambda_1 \mathbf{g}(t) &= \frac{\partial}{\partial a_{1_2}} \lambda_1 (\mathbf{p}_1(t) + \mathbf{p}_2(t) + \mathbf{p}_3(t))^2 \\ &= 2\lambda_1 \cos(2t)(\mathbf{p}_1(t) + \mathbf{p}_2(t) + \mathbf{p}_3(t))\end{aligned}\quad (6.19)$$

In general, the partial derivatives for \mathbf{a}_{i_k} and \mathbf{b}_{i_k} in the center of mass constraint for the k th Fourier coefficient of body i in a system with n bodies is

$$\begin{aligned}\frac{\partial}{\partial a_{i_k}} \lambda_i \mathbf{g}(t) &= 2\lambda_i \cos(kt) \left(\sum_{j=1}^n \mathbf{p}_j(t) \right) \\ \frac{\partial}{\partial b_{i_k}} \lambda_i \mathbf{g}(t) &= 2\lambda_i \sin(kt) \left(\sum_{j=1}^n \mathbf{p}_j(t) \right)\end{aligned}\quad (6.20)$$

Putting Equations 6.16, 6.18, and 6.20 together, finding the partial derivative for a Fourier coefficient can be done with the formulas

$$\begin{aligned}\frac{\partial}{\partial a_{i_k}} \phi_i(t) &= 2 \left(- \sum_{l=1}^{\infty} [l^2 \mathbf{a}_{i_l} \cos(lt) + l^2 \mathbf{b}_{i_l} \sin(lt)] - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_j(t) - \mathbf{p}_i(t)}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \\ &\quad \cdot \left(-k^2 \cos(kt) - \cos(kt) \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{3(\mathbf{p}_j(t) - \mathbf{p}_i(t))^2}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^5} \right. \right. \\ &\quad \left. \left. - \frac{1}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \right) + 2\lambda_i \cos(kt) \left(\sum_{j=1}^n \mathbf{p}_j(t) \right) \\ \frac{\partial}{\partial b_{i_k}} \phi_i(t) &= 2 \left(- \sum_{l=1}^{\infty} [l^2 \mathbf{a}_{i_l} \cos(lt) + l^2 \mathbf{b}_{i_l} \sin(lt)] - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\mathbf{p}_j(t) - \mathbf{p}_i(t)}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \\ &\quad \cdot \left(-k^2 \sin(kt) - \sin(kt) \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{3(\mathbf{p}_j(t) - \mathbf{p}_i(t))^2}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^5} \right. \right. \\ &\quad \left. \left. - \frac{1}{|\mathbf{p}_j(t) - \mathbf{p}_i(t)|^3} \right) \right) + 2\lambda_i \sin(kt) \left(\sum_{j=1}^n \mathbf{p}_j(t) \right)\end{aligned}\quad (6.21)$$

To minimize the function, the partial derivative with respect to λ_i is also required. Finding the partial derivative gives

$$\frac{\partial}{\partial \lambda_i} \phi_i(t) = \left(\sum_{j=1}^n \mathbf{p}_j(t) \right)^2 \quad (6.22)$$

Note that when the center of mass is 0, $\frac{\partial}{\partial \lambda_i} \phi_i(t)$ will also be 0. This purpose of adding the variable λ_i is to move the center of mass to the origin, so when the center of mass is far from the origin, the constraint will have a large impact, but it no longer affects the minimization function once the constraint is met.

Once the partial derivatives with respect to variables are found, they can be used to modify the values of each variable. Changing a Fourier coefficient not only changes the calculated acceleration due to the Fourier series, but it also moves the bodies in the orbit, causing the Newtonian gravitation to change. Due to the chaotic nature of the changes, only small steps may be made towards a minimum, which is done by introducing a scaling factor h , the value of which will be discussed in Chapter 7. Each variable is modified by subtracting the partial derivative of the minimization function with respect to the coefficient, multiplied by h . For example, the coefficient a_{12} would be modified as

$$\mathbf{a}_{12} = \mathbf{a}_{12} - h \frac{\partial \phi_1(t)}{\partial a_{12}} \quad (6.23)$$

6.4 Quadratic Curve Fitting

When starting with the modified gradient descent, the initial coefficient modifications causes the Fourier series and the resulting minimization function to be modified

chaotically. After a number of iterations, the gradient descent will start stabilizing and converging smoothly to a minimum. Once this happens, a point is quickly reached where the convergence slows down. At this point another algorithm is introduced to allow the program to more quickly jump towards a minimum. This function is a quadratic curve fitting.

When the function is approaching a minimum, the curve of the function will behave similar to a quadratic curve. We will fit a quadratic to the three most recent values of the function, and then go directly to the minimum of the quadratic. This will quickly bring us close to a minimum of the minimization function.

To do the curve fitting, a list of the previous coefficient values in the Fourier series is maintained. At a point when the gradient descent function slows down, the previous gradient functions are used to generate a quadratic curve that can be followed to the minimum as follows.

Let \mathbf{C} be the vector of all the coefficients in a Fourier series for a single dimension. As a reminder, the x and y components of each orbit are handled independently by separate Fourier series. For the quadratic curve fitting of a single body, the following algorithm must be run twice: once for \mathbf{C} being the vector of coefficients of the Fourier series defining the x position of the body, and once for \mathbf{C} being the coefficients of the Fourier series defining the y position of the body.

We can now define a measure of fitness function as Φ . We start by defining three

fitness functions as

$$\begin{aligned}\text{Fitness}(\mathbf{C}_0) &= \Phi_0 \\ \text{Fitness}(\mathbf{C}_1) &= \Phi_1 \\ \text{Fitness}(\mathbf{C}_2) &= \Phi_2\end{aligned}\tag{6.24}$$

where \mathbf{C}_0 , \mathbf{C}_1 , and \mathbf{C}_2 are the coefficients from the three previous steps during the gradient descent function.

We define a new restricted quadratic function as

$$q(\mathbf{C}) = a(\mathbf{C} - \mathbf{C}_0) \cdot (\mathbf{C} - \mathbf{C}_0) + b(\mathbf{C} - \mathbf{C}_1) \cdot (\mathbf{C} - \mathbf{C}_1) + c(\mathbf{C} - \mathbf{C}_2) \cdot (\mathbf{C} - \mathbf{C}_2)\tag{6.25}$$

where the dot product of two vectors $\mathbf{r} = (r_0, r_1, \dots, r_m)$ and $\mathbf{s} = (s_0, s_1, \dots, s_m)$ is

$$\mathbf{r} \cdot \mathbf{s} = r_0 s_0 + \dots + r_m s_m\tag{6.26}$$

Now we want to find \mathbf{C} such that $q(\mathbf{C}_0) = \Phi_0$, $q(\mathbf{C}_1) = \Phi_1$, and $q(\mathbf{C}_2) = \Phi_2$. We start by setting \mathbf{C} to \mathbf{C}_0 , \mathbf{C}_1 , and \mathbf{C}_2 into $q(\mathbf{C})$. Using the definitions in Equation 6.24 and cancelling out values we get

$$\begin{aligned}\Phi_0 &= b(\mathbf{C}_0 - \mathbf{C}_1) \cdot (\mathbf{C}_0 - \mathbf{C}_1) + c(\mathbf{C}_0 - \mathbf{C}_2) \cdot (\mathbf{C}_0 - \mathbf{C}_2) \\ \Phi_1 &= a(\mathbf{C}_1 - \mathbf{C}_0) \cdot (\mathbf{C}_1 - \mathbf{C}_0) + c(\mathbf{C}_1 - \mathbf{C}_2) \cdot (\mathbf{C}_1 - \mathbf{C}_2) \\ \Phi_2 &= a(\mathbf{C}_2 - \mathbf{C}_0) \cdot (\mathbf{C}_2 - \mathbf{C}_0) + b(\mathbf{C}_2 - \mathbf{C}_1) \cdot (\mathbf{C}_2 - \mathbf{C}_1)\end{aligned}\tag{6.27}$$

Before continuing we will introduce some shorthand for the dot products. We will define $C_{ab} = (\mathbf{C}_a - \mathbf{C}_b) \cdot (\mathbf{C}_a - \mathbf{C}_b)$ so, for example, $C_{01} = (\mathbf{C}_0 - \mathbf{C}_1) \cdot (\mathbf{C}_0 - \mathbf{C}_1)$. It should also be noted that taking a dot product of a vector with itself gives us the square of its magnitude, meaning that we can further simplify the number of shorthand definitions since $C_{01} = C_{10}$, $C_{02} = C_{20}$, and $C_{12} = C_{21}$.

By combining these shorthand definitions, we now have the three functions

$$\Phi_0 = bC_{01} + cC_{02} \quad (6.28)$$

$$\Phi_1 = aC_{01} + cC_{12} \quad (6.29)$$

$$\Phi_2 = aC_{02} + bC_{12} \quad (6.30)$$

To find the coefficients a , b , and c we start from Equation 6.30 to get

$$a = \frac{\Phi_2}{C_{02}} - \frac{bC_{12}}{C_{02}} \quad (6.31)$$

Rearranging Equation 6.29 and substituting in Equation 6.31 we get

$$\begin{aligned} c &= \frac{\Phi_1}{C_{12}} - \frac{aC_{01}}{C_{12}} \\ &= \frac{\Phi_1}{C_{12}} - \frac{C_{01}}{C_{12}} \left(\frac{\Phi_2}{C_{02}} - \frac{bC_{12}}{C_{02}} \right) \\ &= \frac{\Phi_1}{C_{12}} - \frac{\Phi_2 C_{01}}{C_{02} C_{12}} + \frac{bC_{01}}{C_{02}} \end{aligned} \quad (6.32)$$

Now rearranging Equation 6.28, substituting in Equation 6.31 and Equation 6.32, and cancelling terms we find

$$\begin{aligned} b &= \frac{\Phi_0}{C_{01}} - \frac{cC_{02}}{C_{01}} \\ &= \frac{\Phi_0}{C_{01}} - \frac{C_{02}}{C_{01}} \left(\frac{\Phi_1}{C_{12}} - \frac{\Phi_2 C_{01}}{C_{02} C_{12}} + \frac{bC_{01} C_{12}}{C_{02} C_{12}} \right) \\ &= \frac{\Phi_0}{C_{01}} - \frac{\Phi_1 C_{02}}{C_{01} C_{12}} + \frac{\Phi_2 C_{01} C_{02}}{C_{01} C_{02} C_{12}} - \frac{bC_{01} C_{02}}{C_{01} C_{02}} \\ &= \frac{\Phi_0}{C_{01}} - \frac{\Phi_1 C_{02}}{C_{01} C_{12}} + \frac{\Phi_2}{C_{12}} - b \\ &= \frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01} C_{12}} + \frac{\Phi_2}{2C_{12}} \end{aligned} \quad (6.33)$$

Substituting Equation 6.33 into Equation 6.31 results in

$$\begin{aligned}
a &= \frac{\Phi_2}{C_{02}} - \frac{C_{12}}{C_{02}} \left(\frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01} C_{12}} + \frac{\Phi_2}{2C_{12}} \right) \\
&= \frac{-\Phi_0 C_{12}}{2C_{01} C_{02}} + \frac{\Phi_1}{2C_{01}} + \frac{\Phi_2}{C_{02}} - \frac{\Phi_2}{2C_{02}} \\
&= \frac{-\Phi_0 C_{12}}{2C_{01} C_{02}} + \frac{\Phi_1}{2C_{01}} + \frac{\Phi_2}{2C_{02}}
\end{aligned} \tag{6.34}$$

Finally, substituting Equation 6.33 into Equation 6.32 we find

$$\begin{aligned}
c &= \frac{\Phi_1}{C_{12}} - \frac{\Phi_2 C_{01}}{C_{02} C_{12}} + \frac{C_{01}}{C_{02}} \left(\frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01} C_{12}} + \frac{\Phi_2}{2C_{12}} \right) \\
&= \frac{\Phi_0}{2C_{02}} + \frac{\Phi_1}{C_{12}} - \frac{\Phi_1}{2C_{12}} - \frac{\Phi_2 C_{01}}{C_{02} C_{12}} + \frac{\Phi_2 C_{01}}{2C_{02} C_{12}} \\
&= \frac{\Phi_0}{2C_{02}} + \frac{\Phi_1}{2C_{12}} - \frac{\Phi_2 C_{01}}{2C_{02} C_{12}}
\end{aligned} \tag{6.35}$$

In summary, we can now define the coefficients a , b , and c such that

$$\begin{aligned}
a &= \frac{-\Phi_0 C_{12}}{2C_{01} C_{02}} + \frac{\Phi_1}{2C_{01}} + \frac{\Phi_2}{2C_{02}} \\
b &= \frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01} C_{12}} + \frac{\Phi_2}{2C_{12}} \\
c &= \frac{\Phi_0}{2C_{02}} + \frac{\Phi_1}{2C_{12}} - \frac{\Phi_2 C_{01}}{2C_{02} C_{12}}
\end{aligned} \tag{6.36}$$

The next step is to find the minimum of the quadratic function from Equation 6.25 which is a function of a vector \mathbf{C} . To minimize the function we want to find the differential of $q(\mathbf{C})$ where $dq(\mathbf{C}) = 0$, using the gradient of the function $\nabla q(\mathbf{C})$, giving

$$dq(\mathbf{C}) = \nabla q(\mathbf{C}) \cdot d\mathbf{C} = \mathbf{0} \tag{6.37}$$

To find the minimization function's Fourier coefficients such that the function is minimized along the entire orbit, we want to all the components of $\nabla q(\mathbf{C})$ to be 0, independent of $d\mathbf{C}$.

Finding the gradient $\nabla q(\mathbf{C})$ and setting it to 0 gives

$$\nabla q(\mathbf{C}) = 2a(\mathbf{C} - \mathbf{C}_0) + 2b(\mathbf{C} - \mathbf{C}_1) + 2c(\mathbf{C} - \mathbf{C}_2) = \mathbf{0} \quad (6.38)$$

Expanding and solving for the vector x , we can determine the new minimized Fourier coefficients as

$$\mathbf{C} = \frac{a\mathbf{C}_0 + b\mathbf{C}_1 + c\mathbf{C}_2}{a + b + c} \quad (6.39)$$

Once this equation was implemented in software, it was discovered that it was possible to run into situations where the denominator of $a + b + c$ would be very close to 0, causing numerical precision issues with 64-bit floating-point arithmetic. To help resolve this issue, each time the quadratic is used, we pick the largest coefficient (a , b , or c) and multiply the numerator and denominator by $1/\text{coefficient}$. This results in the equations

$$\begin{aligned} \mathbf{C} &= \frac{\mathbf{C}_0 + \frac{b}{a}\mathbf{C}_1 + \frac{c}{a}\mathbf{C}_2}{1 + \frac{b}{a} + \frac{c}{a}} \text{ when } a > b, c \\ \mathbf{C} &= \frac{\frac{a}{b}\mathbf{C}_0 + \mathbf{C}_1 + \frac{c}{b}\mathbf{C}_2}{\frac{a}{b} + 1 + \frac{c}{b}} \text{ when } b > a, c \\ \mathbf{C} &= \frac{\frac{a}{c}\mathbf{C}_0 + \frac{b}{c}\mathbf{C}_1 + \mathbf{C}_2}{\frac{a}{c} + \frac{b}{c} + 1} \text{ when } c > a, b \end{aligned} \quad (6.40)$$

For completeness, working out the fractions we get

$$\begin{aligned} \frac{b}{a} &= \frac{\frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01}C_{12}} + \frac{\Phi_2}{2C_{12}}}{\frac{-\Phi_0 C_{12}}{2C_{01}C_{02}} + \frac{\Phi_1}{2C_{01}} + \frac{\Phi_2}{2C_{02}}} \\ &= \frac{C_{02}}{C_{12}} \left(\frac{\Phi_0 C_{12} - \Phi_1 C_{02} + \Phi_2 C_{01}}{-\Phi_0 C_{12} + \Phi_1 C_{02} + \Phi_2 C_{01}} \right) \end{aligned} \quad (6.41)$$

$$\begin{aligned} \frac{c}{a} &= \frac{\frac{\Phi_0}{2C_{02}} + \frac{\Phi_1}{2C_{12}} - \frac{\Phi_2 C_{01}}{2C_{02}C_{12}}}{\frac{-\Phi_0 C_{12}}{2C_{01}C_{02}} + \frac{\Phi_1}{2C_{01}} + \frac{\Phi_2}{2C_{02}}} \\ &= \frac{C_{01}}{C_{12}} \left(\frac{\Phi_0 C_{12} + \Phi_1 C_{02} - \Phi_2 C_{01}}{-\Phi_0 C_{12} + \Phi_1 C_{02} + \Phi_2 C_{01}} \right) \end{aligned} \quad (6.42)$$

$$\begin{aligned} \frac{b}{c} &= \frac{\frac{\Phi_0}{2C_{01}} - \frac{\Phi_1 C_{02}}{2C_{01}C_{12}} + \frac{\Phi_2}{2C_{12}}}{\frac{\Phi_0}{2C_{02}} + \frac{\Phi_1}{2C_{12}} - \frac{\Phi_2 C_{01}}{2C_{02}C_{12}}} \\ &= \frac{C_{02}}{C_{01}} \left(\frac{\Phi_0 C_{12} - \Phi_1 C_{02} + \Phi_2 C_{01}}{\Phi_0 C_{12} + \Phi_1 C_{02} - \Phi_2 C_{01}} \right) \end{aligned} \quad (6.43)$$

The inverse fractions (such as a/b) can be found by simply inverting the equations as needed.

Additionally, if the quadratic fit degenerates to a linear function, it can cause numerical issues. In this case, the quadratic fit is ignored until the function is no longer linear.

By using the equations on each coefficient in the Fourier series, we are able to follow a smooth quadratic curve to its minimum, quickly jump closer to a solution. Since the curve fitting does not do any comparisons between Fourier and Newtonian acceleration we cannot switch to this purely once we are moving towards a minimum. Instead it is done as a single step once every number of iterations of the gradient descent minimization. After each quadratic curve fitting, the gradient descent minimization function takes over again and corrects for any over- or under-shooting of coefficient modifications.

By alternating between both the modified gradient descent and the quadratic

curve fitting functions, we are able to quickly and efficiently approach a minimum.

Chapter 7

Software

To identify potentially stable n -body orbits, three pieces of software are needed. The first two pieces of software are for finding Fourier series which defined potentially stable orbits; one is for generating choreographed orbits where a single set of Fourier series defines the orbits of all the bodies, and the other is for generating unchoreographed orbits where each orbit is defined by its own set of Fourier series. The third piece of software is a Newtonian gravitational simulator where the initial positions and velocities of each body are defined, and thereafter all motion is determined by standard Newtonian gravitation.

7.1 Generating Orbits

The two pieces of software for generating orbits are very similar, so they will be discussed together. The main goal of the software is to find Fourier series that define stable orbits for a given number of bodies. As seen in Section 5.2, the Fourier series

is defined as

$$\mathbf{z}(t) = \mathbf{a}_0 + \sum_{k=1}^K [\mathbf{a}_k \cos(kx) + \mathbf{b}_k \sin(kx)] \quad (7.1)$$

An infinite Fourier series has $K = \infty$, but in software an approximation must be made of the infinite series so the programs start with a smaller number of coefficients. Through experimenting with different initial numbers of coefficients, a good starting point which allowed for a variety of different orbits to emerge while still being able to minimize the function was $K = 7$. This initial value can be easily modified in the programs and additionally may be increased dynamically as the programs run to refine Fourier series. The coefficients are all initialized randomly in the range $(-3.0, 3.0)$, which was again found to be a good starting range through experimenting with ranges of values between $(-1, 1)$ and $(-20, 20)$. The value of λ for the additional constraint is set randomly in the range $(0, 3.0)$, since it is required to be positive.

The programs start by initializing all of the coefficients and choosing the timesteps to be considered during each main iteration of process. The timesteps cover the entire period of the Fourier series. During each iteration of the program, the minimization function from Equation 6.7, comparing the difference between the Fourier and Newtonian acceleration, is evaluated at each timestep to determine if they match. The results of the minimization function at each timestep are summed over all the timesteps. The purpose of summing the results is so the Fourier series can be modified so the function is minimized along the entire length of the orbit curve. The partial derivatives with respect to each coefficient, used to find the gradient with Equation 6.21, are also summed over all the timesteps and the results are used to update the coefficients for the next iteration of the program.

In order to quickly converge towards a solution, the programs initially use 31 timesteps in $[0, 2\pi]$, which was a number that was found through experimenting to be a good starting point. A prime number was chosen as the starting value because the function is minimized at a finite number of points along the orbit, and choosing a prime number ensures the time points are not broken up in an even number of positions to reduce the chance of oscillations or fitting only specific intervals on the series. Additionally, the timesteps are not evenly spaced. The period of 2π is broken up into 31 equal intervals, and a random location in each interval is used as the timestep value. In other words for timestep t_l , where $0 \leq l < 31$, the time is randomly assigned in the range

$$\frac{2\pi}{31}l \leq t_l < \frac{2\pi}{31}(l+1) \quad (7.2)$$

The timesteps are used to find the partial derivatives at a number of points along the orbit. To determine the modification needed for each Fourier coefficient, the program finds the sum of the partial derivatives of the minimization function (using Equation 6.21). The sum is then used with Equation 6.23 to find the new value of the coefficient. For example, to modify the Fourier series for the second coefficient of the first body using 31 timesteps and a scaling factor h , the new value would be

$$\mathbf{a}_{1_2} = a_{1_2} - h \sum_{l=1}^{31} \frac{\partial \phi_1(t_l)}{\partial \mathbf{a}_{1_2}} \quad (7.3)$$

The programs can be run in either fixed or variable timestep modes, each with advantages and disadvantages. In fixed timestep mode, the timestep points are defined when the program starts and remain constant throughout the run of the program. This allows for the program to initially fit the timesteps quicker while minimizing large

initial oscillations when the Fourier coefficients are still far from a solution. However, fixed timestep mode has the problem that it can overfit the set timesteps and oscillations or large adjustments may be introduced when more timesteps are added. In variable timestep mode, the timestep points are redefined after every minimization loop. While this can initially cause more oscillations and moving towards an initial possible solution is less likely, but once a potential solution is being approached it reduces the oscillations that are introduced when more timesteps are added. In both cases potential orbits were able to be identified and once additional Fourier series coefficients and timesteps were added, the differences between the two approaches were negligible.

To generate accurate results, the number of timesteps needs to be increased as a solution is approached, to ensure the Fourier coefficients are not overfitting the timesteps. Adding timesteps is done manually by the user as the function is minimizing, the process of which is described in Section 7.1.2.

7.1.1 Main Program Loop

Once the initial Fourier coefficients and timesteps are defined, the main algorithm loop begins using the formulas from Chapter 6. For each time through the loop, the positions of each body is first evaluated at each time interval using the Fourier series (again, the Fourier series uses independent vector coefficients for the x -position and the y -position). As described in Section 5.3, there is a difference between finding the orbits with the choreographed and unchoreographed orbits. In the choreographed

orbits program, a single Fourier series with time offsets equally spaced along the period of the orbit is used to find the positions. In the unchoreographed orbits, each orbit has its own Fourier series.

Once the positions of each body at a given timestep is determined, the Newtonian acceleration of each body can be determined using Equation 6.1 and the Fourier acceleration, using the second derivative of the Fourier series, can be determined using Equation 6.5.

With the positions and accelerations calculated, the minimization step may occur. Initially, all of the minimization is done using the modified gradient descent algorithm from Section 6.3. For each timestep, the partial derivatives with respect to each Fourier coefficient and with respect to the constraint factor λ are determined. For each coefficient the sum of the partial derivatives over all the timesteps is used to modify the values for the next iteration using Equation 6.23. As the minimization progresses, the values of all the coefficients from the previous three iterations are stored in preparation for the quadratic curve fitting.

The quadratic curve fitting from Section 6.4 is not initially enabled, as the Fourier coefficients start out at random values and may take a number of iterations before the measurement of fitness function starts smoothly approaching a minimum. When the function starts approaching a minimum, the user may enable the quadratic curve fitting by enabling a checkbox in the user interface. Once enabled, every fifth time through the main program loop the quadratic curve fitting algorithm is used instead of the modified gradient descent minimization function.

After the quadratic curve fitting step is complete, it is common for the gradient

descent scaling factor h to be too large which can cause oscillations. To prevent this problem, after each quadratic curve fitting the gradient minimization is allowed to run a single time. The squares of the differences between the Newtonian and Fourier accelerations at each timestep are summed and compared to the sum of the squares of the differences before the gradient descent step (but after the quadratic curve fitting). If both the x -coordinate and y -coordinate differences got worse after a single gradient step, or one coordinate got worse by more than 10 times the amount the other coordinate improved, then the value of h , the gradient scaling factor, is reduced to 80% of its value. This can happen up to three times in a row in an attempt to reduce the amount of oscillation. If, after reducing h three times, the coordinate differences still are not improving, an additional set of Fourier coefficients (\mathbf{a}_i and \mathbf{b}_i) are added to the end of each Fourier series being used. This allows for the Fourier series to be further refined by the gradient minimization in an attempt to reduce error that may be resulting from a too-coarse Fourier series.

The user may allow the programs to run as long as desired to allow the minimization functions to refine the orbits. At any point, the user may press the “Dump” button on the user interface which will write all of the Fourier series coefficients to a file. This file can then be re-imported at a later time into the minimization program to continue investigation into a particular periodic orbit, or can be loaded into the Newtonian gravitation simulator to explore the actual orbit that has been generated.

7.1.2 Additional User Control

In order to converge on a solution, the programs start with a limited number of coefficients in the Fourier series, as well as a limited number of time increments along the series to compare the bodies at. As a potential solution is approached, the coefficients starts fitting these cases nicely but any other points along the series will still be incorrect. When this happens, more coefficients and time increments need to be added to refine the solution. Additionally, as a solution approaches a minimum, the scaling factor h for the gradient minimization can cause the modifications to the Fourier coefficients to overshoot the correct if it h stays too large. The programs include additional ways for the user to mitigate these issues.

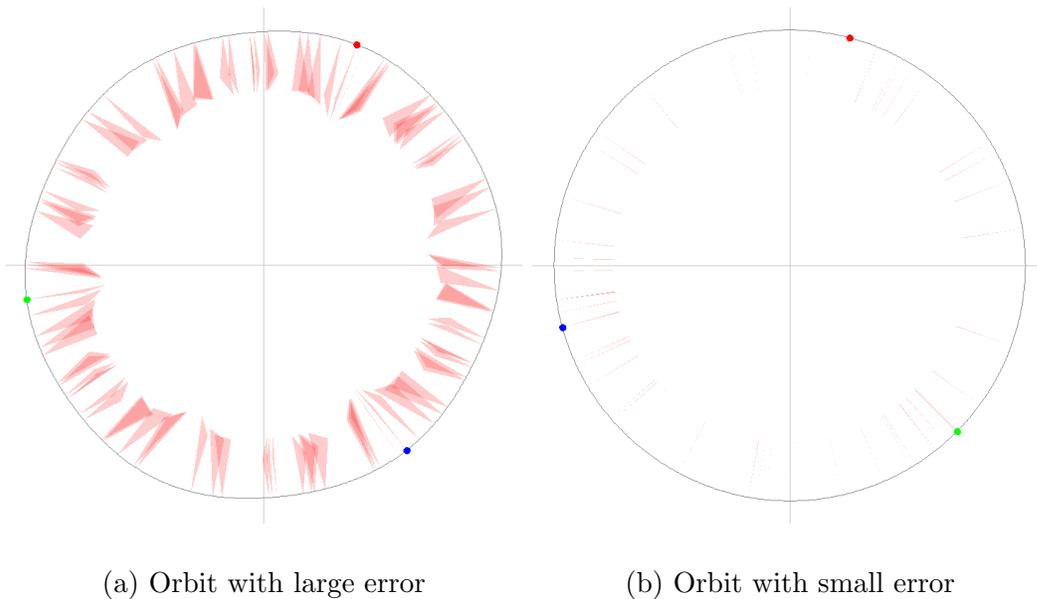
By clicking the “Add Var” button, two additional sets of coefficients (two a coefficients and two b coefficients) are added to the end of the Fourier series. These additional values are initially set to 0, but they are allowed to move freely according to the algorithm minimization algorithms. Adding these coefficients allows for more fine-tuning of the curves, especially when the solution has small loops in it. Starting the values at 0 ensures that adding more coefficients does not initially introduce wild oscillations to the Fourier series.

The “Add Timesteps” button doubles the current number of timesteps being used by the algorithm. If using variable timestep mode, the period 2π is divided into double the intervals for randomly selecting the timesteps each iteration. If using fixed timestep mode, all of the existing timesteps are maintained and a new set of timesteps are randomly interspersed within the existing values, using the same interval ranges

as the initial values.

The value of h can also be modified via user interface buttons. The user is able to increase h by multiplying it by 2 for each click, or decrease it by dividing by 2 for each click. Generally, h starts off larger but as a solution is approached, the user needs to reduce the value of h in order to keep approaching a minimum without overshooting it repeatedly and causing oscillations around the minimum.

Figure 7.1: Choreographed orbits with error overlay



The user is also able to enable viewing of an overlay showing the difference between the Newtonian and Fourier acceleration. In this mode, a triangle is overlaid on top of the orbit for each timestep along the orbit. The points representing the position of the body on the Fourier series orbit for a given timestep, the direction and magnitude of the Fourier acceleration on the body, and the direction and magnitude of the Newtonian acceleration on the body form the triangle. This means that in the case

where the Fourier and Newtonian acceleration are equal, a straight line is drawn from the point in the direction of the acceleration, but in the case where a large difference exists there are distinct triangles drawn. An example of two choreographed Fourier series orbits, one with a large amount of error and one with a small amount of error, can be seen in Figure 7.1. Showing the error allows the user to visibly inspect the existing error in the orbit function being minimized and adjust the number of coefficients, number of timesteps, or the value of h in response. It also allows the user to determine when the error is no longer being reduced enough through the gradient minimization and the quadratic curve fitting is required.

7.1.3 Number of Variables

There are a large number of variables required in the minimization function. The first set of variables is the coefficients of the Fourier series. The series is truncated at some length K , which is initially set to 7. For both the x and y Fourier series, there are K values of a , K values of b , plus the offset value a_0 . This gives a total of $4K + 2$ coefficients for the Fourier series. In the case of choreographed orbits, each body uses the same series, but for unchoreographed orbits with n bodies, each body has its own Fourier series. In this case, there will be $(4K + 2)n$ coefficients in total.

Additionally, for each Fourier minimization there will be a λ for both the x and y minimization. With choreographed orbits this results in 2 values of λ , and in the unchoreographed orbits we have $2n$ values of λ .

Two other variables are the scaling factor h and the number of timesteps. Neither

of these are dependent on the number of coefficients or the type of orbit. The number of timesteps will affect the number of comparisons required in each iteration of the minimization algorithm, and so increasing the number of timesteps will affect the running time, but the minimization function maintains running totals for partial derivatives of the Fourier coefficients and so does not require additional variables when increasing timesteps.

To evaluate the quadratic curve fitting, three iterations of Fourier coefficients must be maintained to define the quadratic. Given that the choreographed orbits require $4K + 2$ coefficients and the unchoreographed orbits require $(4K + 2)n$ coefficients, the quadratic curve fitting requires an additional set of $12K + 6$ and $(12K + 6)n$ variables respectively.

Given a starting point of $K = 7$, a choreographed system with $n = 3$ would require a total of 30 coefficients plus 2 values of λ , as well as an additional 90 coefficients being maintained for the quadratic curve fitting. In an unchoreographed system, these totals would increase to a total of 90 coefficients plus 6 values of λ for the gradient minimization function, plus an additional 270 coefficients being maintained for the quadratic curve fitting.

7.2 Newtonian Gravitation Simulator

In order to determine if an orbit is potentially stable, software was created to accurately model the motion of bodies with an n -body Newtonian gravitation simulator. Although the simulator can theoretically handle any number of bodies, it has

been designed for only a handful in mind. Most of the orbits investigated have three to six bodies in them. Every body adds complexity to the simulation and as such the simulator is much more suited for a small number of bodies.

7.2.1 Initial Values

Every body in the simulation requires an initial position and velocity. Initially these values were hard-coded into the program and recompiled, but as a greater number of potential orbits were found this proved unwieldy, so an import function was created. When the simulator starts, the user may choose a file that has been outputted from the orbit generation software described in Section 7.1. These files contain all of the Fourier coefficients used to define the orbits. In the case of the choreographed orbits, a single Fourier series is used to determine the orbits of every body using the time offsets described in Section 5.3.1. For the unchoreographed orbits, the files contain a unique Fourier series for each body in the system and each series is evaluated at time $t = 0$. The initial location of each body is determined by evaluating the Fourier series, and the initial velocity is found using the first derivative of the series. Once the initial positions of all the bodies have been calculated, the center of mass of the system is calculated by summing the x and y positions of each body. If the center of mass is offset from the origin, all the bodies are translated to move the center of mass to the origin to allow for easier observation of the system.

The orbits defined by the Fourier series have an orbit period time of 2π . In order to provide accurate simulation results, it was found that an initial timestep

of $t_{step} = 10^{-5}$ provided good results. The size of the timestep can change as the simulation progress, which will be described further in the next section.

7.2.2 Main Program Loop

Once the simulation starts the motion of the bodies is solely defined by Newton's law of universal gravitation described in Section 5.1. As a reminder, for the purposes of simplifying the calculations the gravitational constant is set to $G = 1$ and the mass of each body is $m = 1$. At the beginning of each loop, the position and velocity of each body is known.

To find the position and velocity of each body after a single timestep, the motion of each body needs to be integrated over the timestep. As described in Section 4.1.1, the initial tests using the Euler method of integration proved to be very inaccurate within even a single period of simulation. To provide more accurate results, the simulation uses the fourth order Runge-Kutta integration for the motion, known as RK4, described in Section 4.1.2. This provides a much more accurate numerical integration.

Each time through the main program loop, the position and velocity of each body is used in the RK4 integration to find the positions and velocities after a single timestep. The functions used in the RK4 integration are the standard position and velocity formulas discussed in Section 4.1.

Once the new positions and velocities of each body have been calculated, the program analyses the results to determine if the current timestep is small enough to

provide accurate results. When the bodies in the simulation are further apart, a larger timestep will provide accurate results due to the small variation in forces over the length of the step while allowing the simulation to progress more rapidly. However, as bodies near each other, the forces and acceleration can change very rapidly within a single timestep. In this case, a large timestep will provide inaccurate results and so a smaller and more accurate (albeit longer to simulate) timestep is needed.

To support shorter timesteps when bodies approach each other, the simulator is able to adapt the timestep dynamically as the simulation progresses. To determine if the timestep is too large, the RK4 integration of the system is run in reverse.

At each step of the simulation, each body starts with initial positions and velocities \mathbf{x}_0 and \mathbf{v}_0 . After the system has been calculated in the forward direction from $t_i \rightarrow t_{i+1}$, giving a timestep of $t_{step} = |t_{i+1} - t_i|$, each body has a new position and velocity \mathbf{x}_1 and \mathbf{v}_1 . Using the negative of the velocity to run the RK4 integration from $t_{i+1} \rightarrow t_i$, starting with \mathbf{x}_1 and $-\mathbf{v}_1$ for the time t_{step} , the bodies end up at a position \mathbf{x}_2 . If the full system was calculated with a perfect continuous function, the result would be $\mathbf{x}_0 = \mathbf{x}_2$. Since there is some approximation in the RK4 integration, there may be some difference in the results.

If the difference in either the x or y position between the forward and backward computations values is past an error threshold of 10^{-5} for any body in the system, the timestep is too large. At that point, the timestep is divided by 2 and the motion integration is recalculated. The process of comparing the forward and backward motion integration repeats until the error is below the threshold, at which point the simulation may continue iterating using the new timestep.

To allow the simulation timesteps to slowly increase, to speed up the simulation as the bodies move away from each other and the error decreases, the timestep is allowed to increase by a factor of 10% for every 20 timestep iterations that do not pass the error threshold. If the increase becomes too large or the bodies move closer and the error threshold is surpassed, the timestep will again be shortened by the reverse RK4 integration logic.

Using the initial positions and velocities of the bodies based on the Fourier series orbits generated via the minimization functions, the simulator can be used to run a Newtonian simulation to investigate if the generated orbits are in fact stable. The stability of the orbits is determined by observing the orbits over multiple periods of the orbit. While it is possible for a periodic orbit to exist in seemingly chaotic motion, the expected observation of a periodic orbit should closely match the orbit defined by the Fourier series.

Chapter 8

Orbits Found

A number of potential choreographed and unchoreographed orbits were found using the minimization programs. The orbits from the Fourier series, as well as the resulting Newtonian simulation, are presented here using the software described in Chapter 7.

For each orbit two images are presented. The images on the left are from the minimization programs and show the orbits as defined by the Fourier series using the minimized coefficients. Each minimization image is labeled with ϵ , the average remaining difference of the goodness of fit minimization function from Equation 6.9 over all the timesteps. ϵ is defined as $\epsilon = \sqrt{\epsilon_x^2 + \epsilon_y^2}$ where ϵ_x and ϵ_y are the minimization function values in the x and y directions (since the Fourier series are handled separately in the two directions). Larger values of ϵ may indicate an unstable orbit, such as when the minimization function results in a local minimum, but is not definitive.

The images on the right are from the Newtonian gravitation simulator. For the simulation, the starting position of each body is found by evaluating the Fourier series and the starting velocity of each body is found by evaluating the first derivative of the Fourier series, both at time $t = 0$. As each body moves through the simulation, it leaves a trail on the image so the complete orbits can be seen. The images from the gravitation simulator are labeled with the value of t when they were taken (where a single period is $t = 2\pi$).

The tables for each set of images contain the starting positions and velocities determined for each body from the Fourier series. The Fourier series coefficients used for the bodies in each system can be found in Appendix A.

There are three common possible results in the orbits presented. The first possibility is when the Newtonian simulation results in an orbit that is able to maintain stability for a number of periods, which indicates it has good potential to be a truly stable orbit.

The second possible result is that the Newtonian orbit is not stable and quickly becomes chaotic. In these cases, the minimization does not result in a stable Newtonian orbit. This can be due to an insufficient resolution in the timesteps or number of coefficients of the Fourier series with an orbit that is very sensitive to perturbations, or the minimization function getting stuck in a local minimum. Additionally, even the buildup of the small error inherent in the floating-point arithmetic of the simulator's numerical integration can be enough to cause a sensitive orbit to decay.

The third possible result is when the orbit remains almost periodic, but the orbit has some degree of precession. When this occurs, the simulation image does not

match the Fourier series image, but the simulation orbit appears to follow the Fourier series orbit with rotation added to the orbit. These orbits may significantly decay after only a few periods, but in other cases they appear to remain stable. This orbit precession will be discussed in Chapter 9.

It can be noted that a number of the orbits found are similar to the orbits found by both Simó [11] and Vanderbei [13].

8.1 Choreographed Orbits

The bodies in each system in this section are determined by a single set of Fourier series with each body starting at specific intervals along the period. The starting offsets for each body around the orbit can be found using the formulas in Section 5.3.1.

Figure 8.1: Choreographed three-body circle

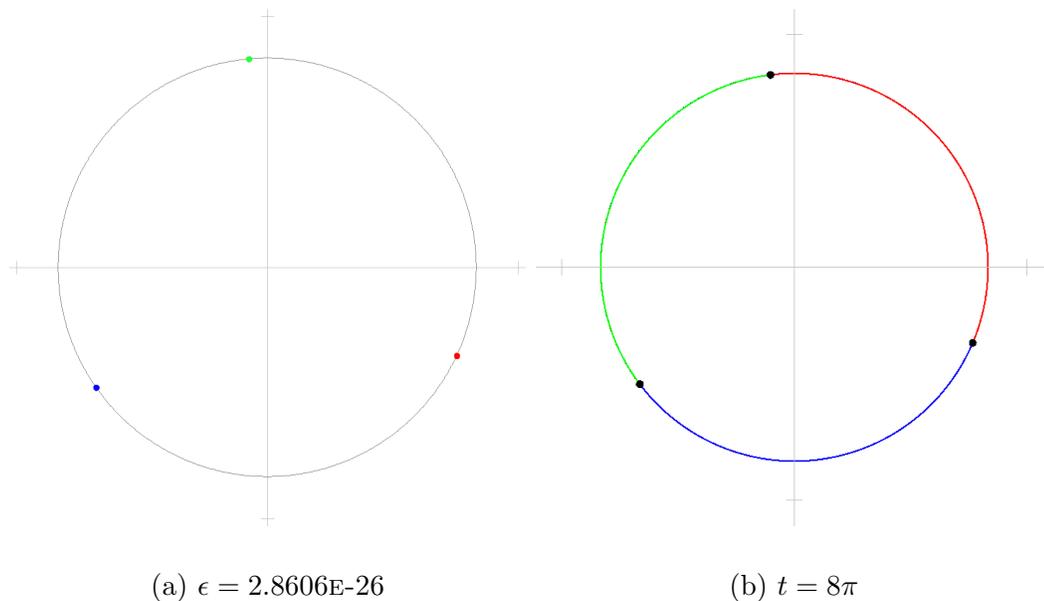


Table 8.1: Choreographed three-body circle initial values

Body	x	y	x'	y'
1	0.7548983603205632	-0.3514110697402242	-0.35141106974012876	-0.7548983603206323
2	-0.6817800937262974	-0.4780556224427804	-0.4780556224427675	0.6817800937264157
3	-0.07311826659426574	0.8294666921830046	0.8294666921828963	0.07311826659421666

Figure 8.2: Choreographed four-body circle

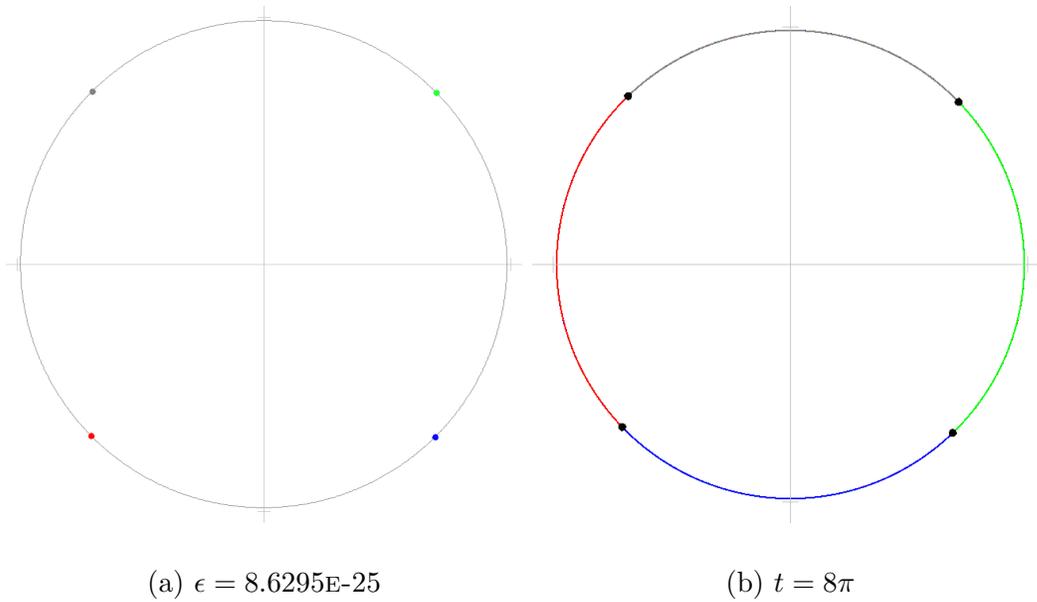


Table 8.2: Choreographed four-body circle initial values

Body	x	y	x'	y'
1	-0.699242298027974	-0.6944467629710684	0.6944467629708051	-0.6992422980282715
2	0.6944467629708471	-0.6992422980282327	0.6992422980279377	0.694446762971098
3	0.699242298027974	0.6944467629710683	-0.6944467629708043	0.6992422980282715
4	-0.6944467629708471	0.6992422980282328	-0.6992422980279385	-0.694446762971098

Figure 8.3: Choreographed three-body figure-eight

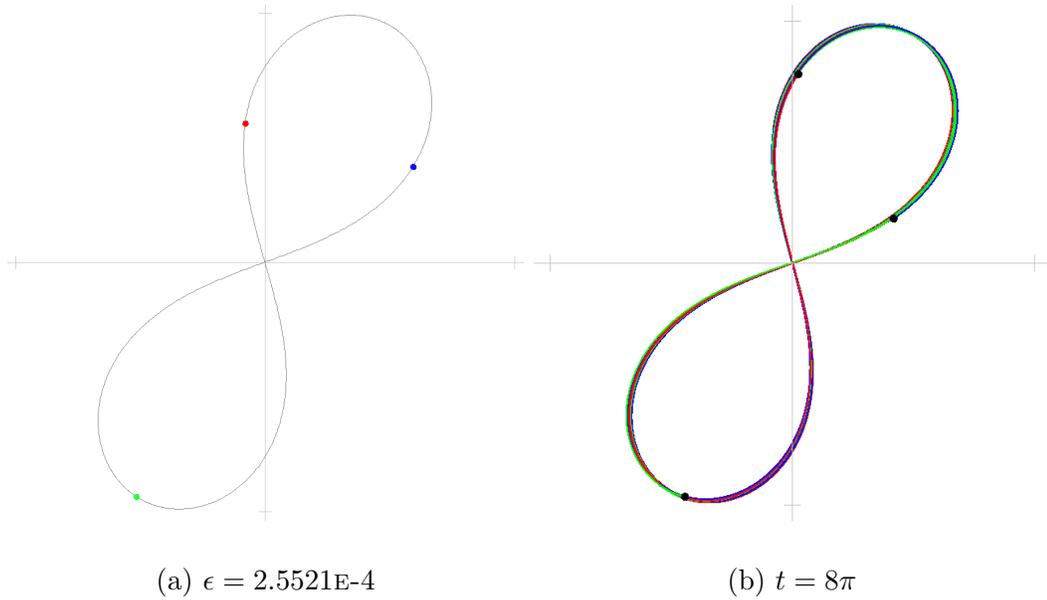


Table 8.3: Choreographed three-body figure-eight initial values

Body	x	y	x'	y'
1	-0.07880227334416882	0.5570371897354746	0.15998292728488323	1.1593418791674066
2	0.5940359608209828	0.383319210563721	-0.5557289806160467	-0.9029539156799118
3	-0.5152336874768139	-0.9403564002991956	0.39574605333116347	-0.2563879634874948

Figure 8.4: Choreographed four-body trefoil

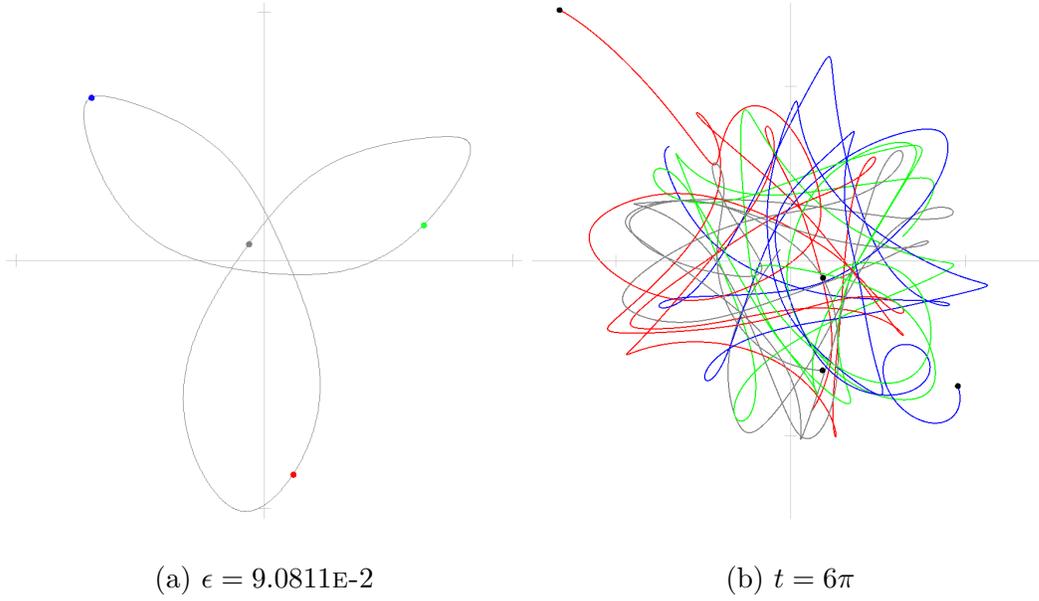


Table 8.4: Choreographed four-body trefoil initial values

Body	x	y	x'	y'
1	0.11661716221150759	-0.8628229948869036	0.48336456850271736	0.7305598083360698
2	-0.6963959862721382	0.6556306737971199	-0.2682255717908091	-0.1249827016194934
3	0.641932494625764	0.14183543304631654	0.8123289473484319	0.8218875611377845
4	-0.06215367056513341	0.06535688804346723	-1.0274679440603403	-1.4274646678543608

Figure 8.5: Choreographed four-body chain

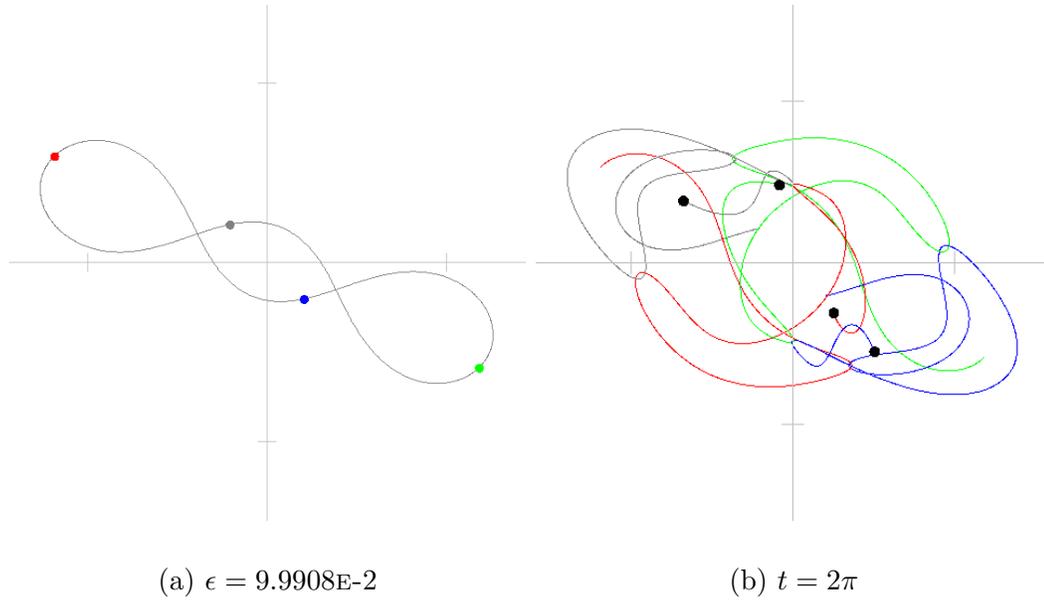


Table 8.5: Choreographed four-body chain initial values

Body	x	y	x'	y'
1	-1.1844475691074707	0.5898810975762604	0.4087319255099535	0.39755580006528823
2	0.20705693570351705	-0.20720580654667664	1.3649438991518443	0.2819370010443289
3	1.1835106293686215	-0.5912376198019829	-0.4103624168353448	-0.39612196466777105
4	-0.20611999596466776	0.20856232877239902	-1.363313407826453	-0.2833708364418462

Figure 8.6: Choreographed four-body inner loop

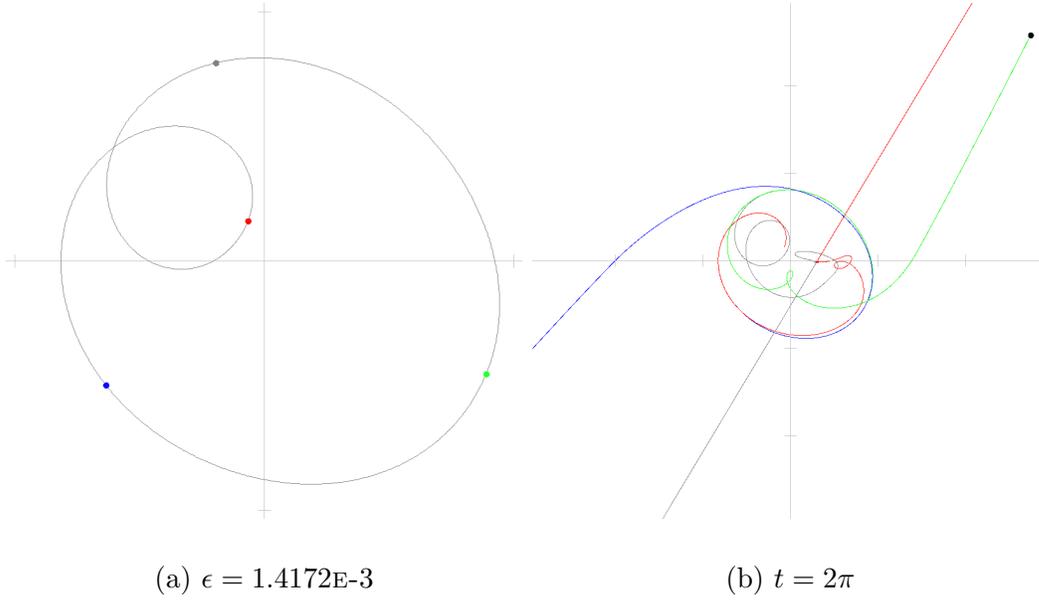


Table 8.6: Choreographed four-body inner loop initial values

Body	x	y	x'	y'
1	-0.06415594629608656	0.15963611651760048	0.19231903002674716	0.498135742337882
2	-0.6332455383219576	-0.49885112427588185	0.8611257278494242	-1.0944264406332394
3	0.8905726094445537	-0.45444379724006034	0.386736125112952	0.9468444486242256
4	-0.19317112482650967	0.7936588049983417	-1.4401808829891232	-0.3505537503288683

Figure 8.7: Choreographed five-body figure-eight

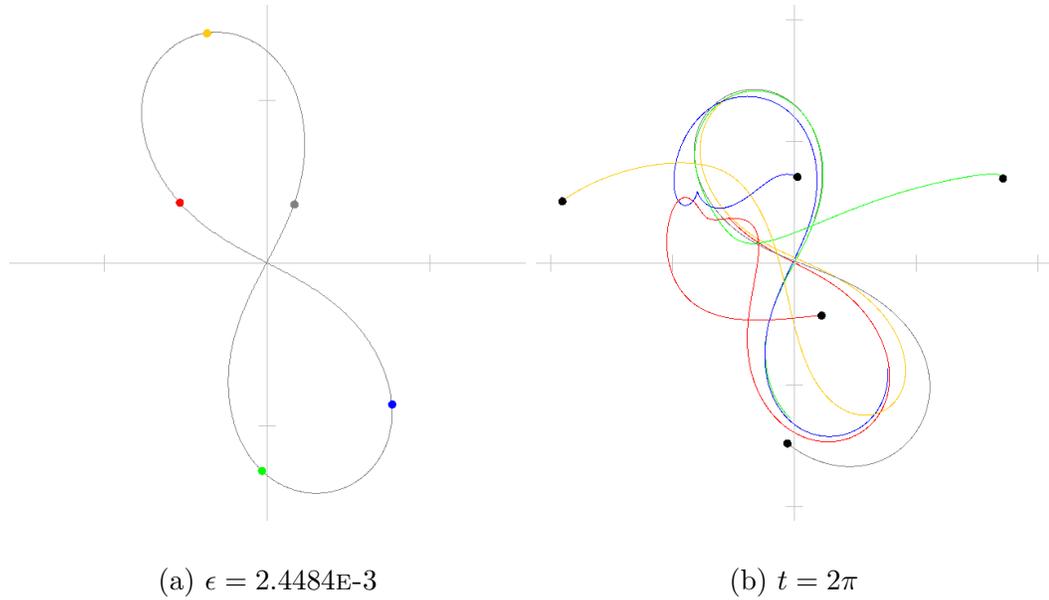


Table 8.7: Choreographed five-body figure-eight initial values

Body	x	y	x'	y'
1	-0.5358076316429841	0.37180556472793874	0.9743317053313775	-1.096362116657104
2	0.7676771558912595	-0.867823428450871	0.08096334827522714	-1.0175492452692891
3	-0.031055488577342624	-1.2763443049957865	-0.6990773093771361	0.672388539946965
4	0.1675715831569598	0.3604411202596589	0.5493639322512589	1.5662040016531198
5	-0.36838561882789267	1.41192104845906	-0.9055816764807272	-0.12468117967369167

Figure 8.8: Choreographed five-body pretzel loop

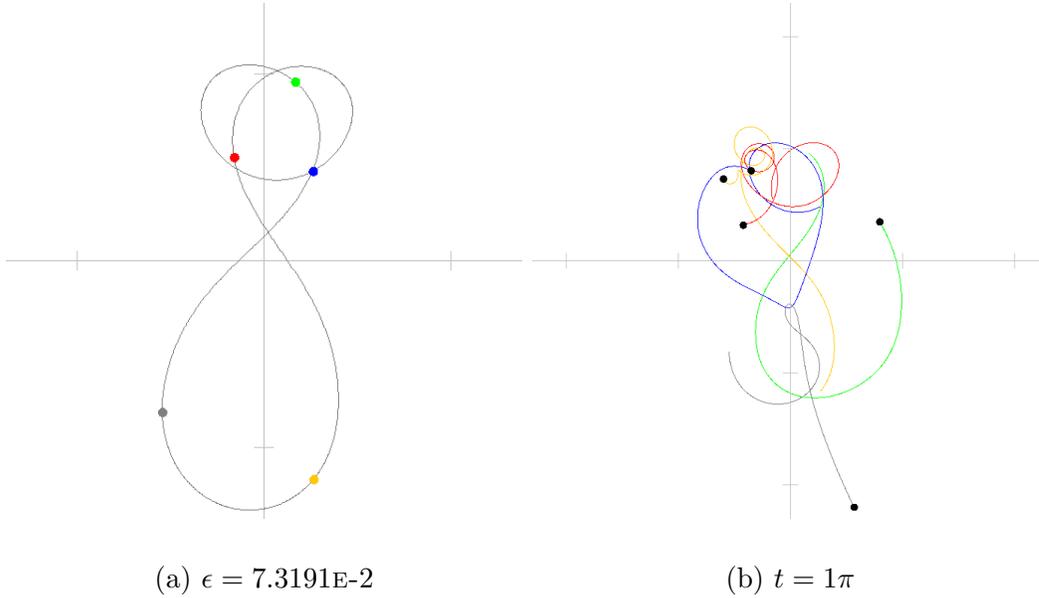


Table 8.8: Choreographed five-body pretzel loop initial values

Body	x	y	x'	y'
1	-0.1580890992735324	0.5522601063500079	-0.3549681458323808	2.1208189974173686
2	0.2643361663758888	0.4777863235898944	-1.4815977423589528	-0.7930211016542689
3	0.1688808045879178	0.9563693451915746	1.2128692819240836	-1.0274109751221157
4	-0.5428002432275009	-0.8134290379137666	0.019545784227261165	-1.0338061864372619
5	0.2676723715372266	-1.17298673721771	0.604150822039989	0.7334192657962778

Figure 8.9: Choreographed five-body trefoil

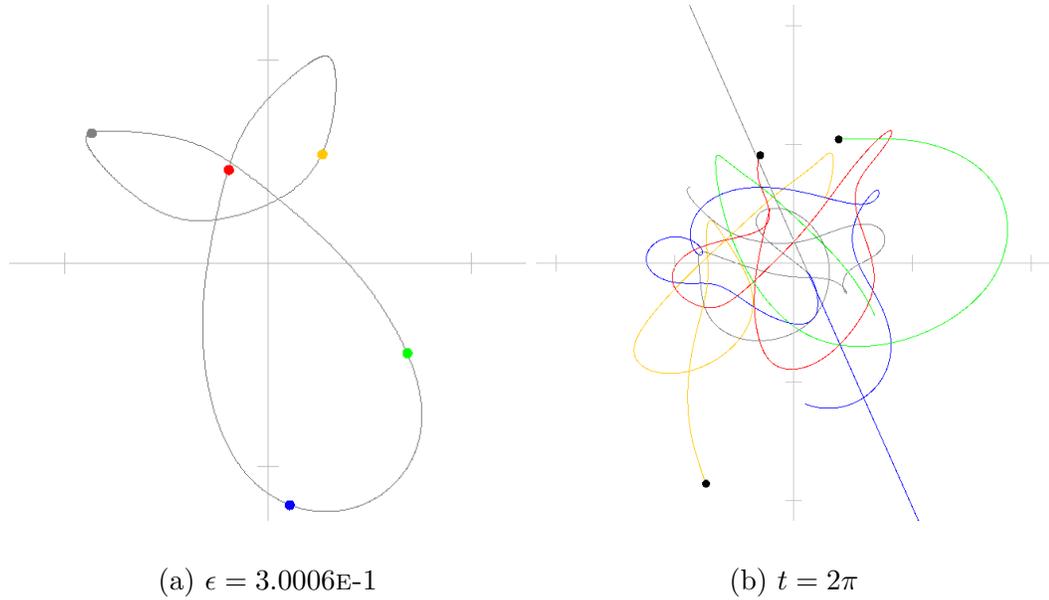


Table 8.9: Choreographed five-body trefoil initial values

Body	x	y	x'	y'
1	-0.19303202724324267	0.4585444026490357	-0.622244545113255	-2.0430154410644796
2	0.10545527999486655	-1.190486988139539	0.9134832494681733	-0.3588341636318668
3	0.6860208939238829	-0.443093326526616	-0.5105853805336759	1.1737050359224457
4	-0.865708183987005	0.6396338125865522	-0.4008266738461247	-0.12489063559067837
5	0.2672640373114983	0.5354020994305673	0.620173350024882	1.3530352043645792

8.2 Unchoreographed Orbits

Each body in this section is determined by a unique set of Fourier series. The starting positions and velocities for each body are determined by solving the Fourier series and its first derivative, respectively, at time $t = 0$ with no offsets for any body as described in Section 5.3.2.

Figure 8.10: Three-body circle

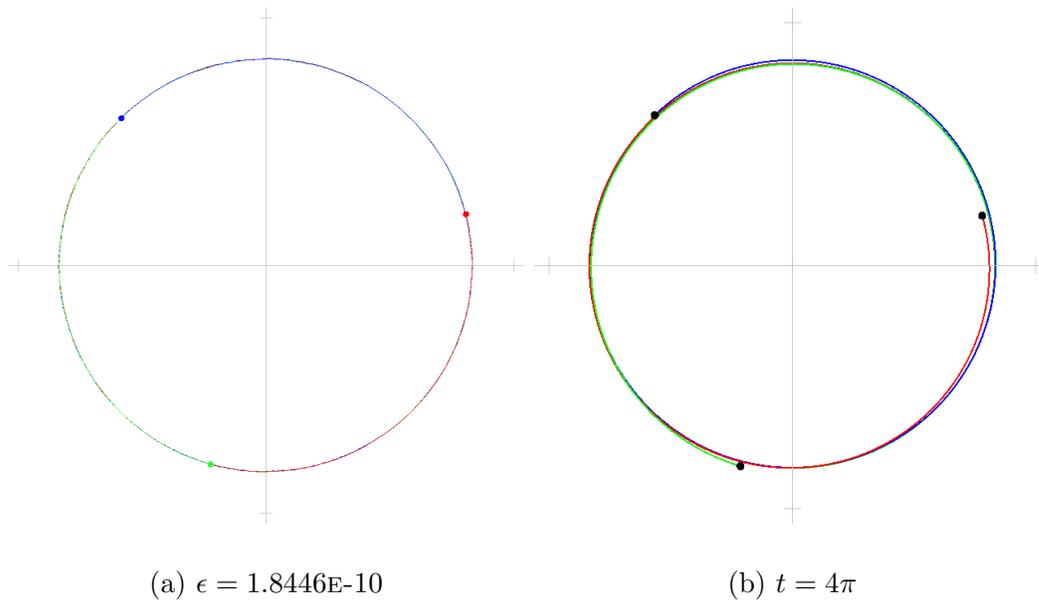


Table 8.10: Three-body circle initial values

Body	x	y	x'	y'
1	0.8064717222934421	0.20727055247344797	-0.20727366251751161	0.806466168737808
2	-0.5827339118953009	0.5947956312501173	-0.5947853254134745	-0.5827324482138609
3	-0.2237378103981412	-0.8020661837235651	0.8020589879309861	-0.22373372052394694

Figure 8.11: Three-body circled cross

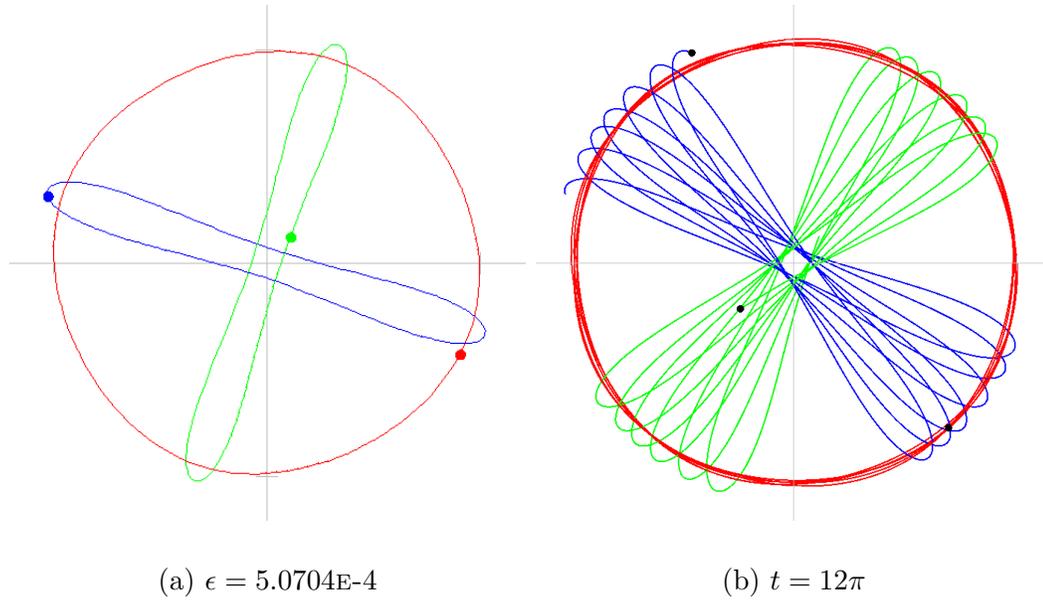


Table 8.11: Three-body circled cross initial values

Body	x	y	x'	y'
1	0.9092123039000347	-0.43040535790199963	0.47797186363889416	0.9089826224520586
2	-1.0229414945397166	0.3111968046982191	-0.06412936032621364	0.21850122248844084
3	0.11372919063968188	0.11920855320378049	-0.41384250331268047	-1.1274838449404994

Figure 8.12: Three-body enveloped figure-eight

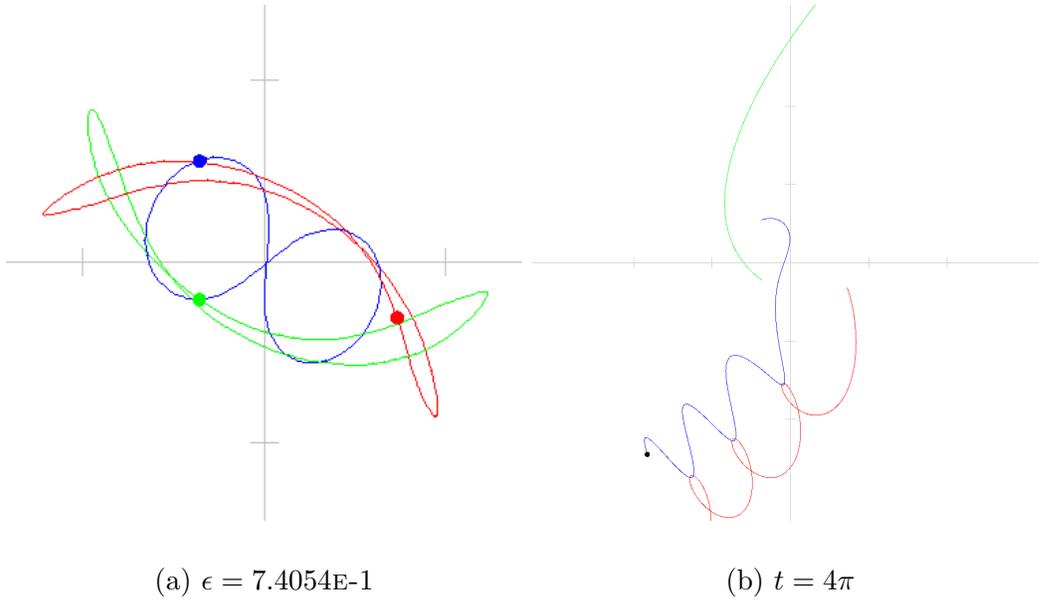


Table 8.12: Three-body enveloped figure-eight initial values

Body	x	y	x'	y'
1	0.7275242381011939	-0.3209974644044415	0.42161865424089007	-1.182271917940274
2	-0.36268417274378945	0.5426769104001704	0.7884831954777513	0.29893144651911696
3	-0.36484006535740454	-0.22167944599572623	-1.2101018497186413	0.8833404714211572

Figure 8.13: Four-body nuclear symbol

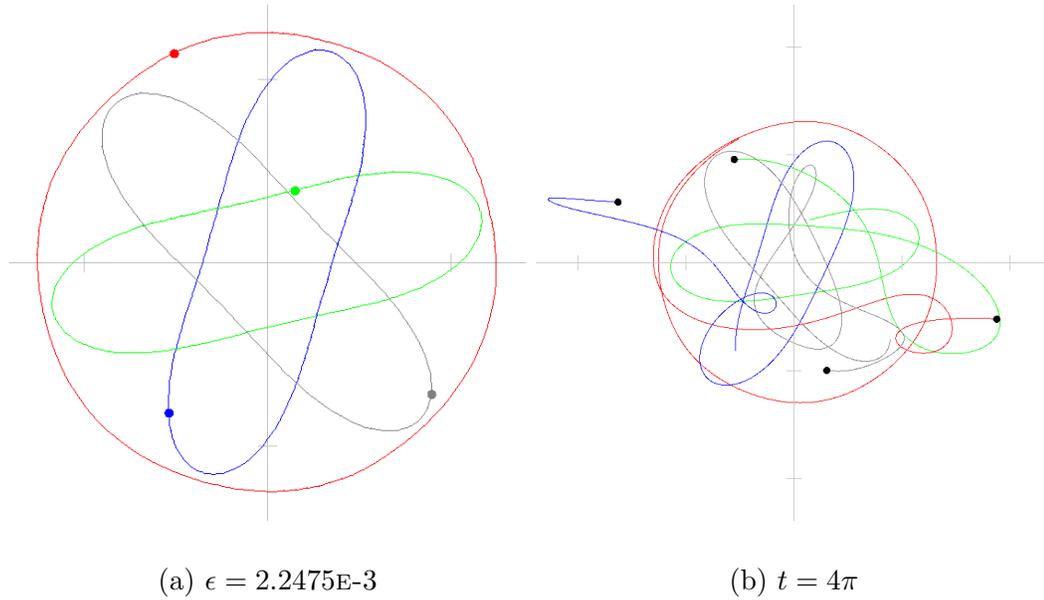


Table 8.13: Four-body nuclear symbol initial values

Body	x	y	x'	y'
1	-0.5091966069861722	1.1420126037068834	-1.1449094769247894	-0.5602702750794802
2	-0.536844041872087	-0.8187464228819548	-0.04350261985812843	0.7708499999957881
3	0.15059481202038588	0.39389689651781934	1.247618869002686	0.32317677688965085
4	0.8954458368378734	-0.7171630773427479	-0.05920677221976818	-0.5337565018059588

The final orbit, seen in Figure 8.14, was not found directly through the orbit generator programs but was instead found while experimenting with different initial ratios for the starting values. It is of particular interest because it remains stable for five periods ($t = 10\pi$) before starting to unravel, which makes it more stable than the choreographed circular orbit with small perturbations that will be presented in Chapter 9.

Figure 8.14: Three-body offset rings

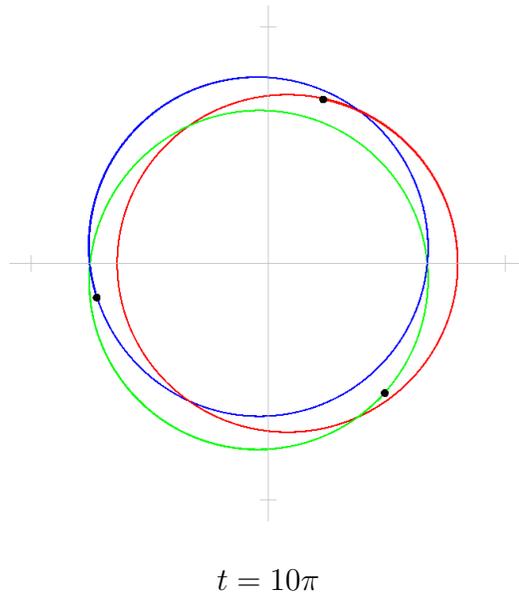


Table 8.14: Three-body offset rings initial values

Body	x	y	x'	y'
1	0.0	0.8	-0.8	0.0
2	$0.8 \cos \frac{2\pi}{3}$	$0.8 \sin \frac{2\pi}{3}$	$-0.8 \sin \frac{2\pi}{3}$	$0.8 \cos \frac{2\pi}{3}$
3	$0.8 \cos \frac{4\pi}{3}$	$0.8 \sin \frac{4\pi}{3}$	$-0.8 \sin \frac{4\pi}{3}$	$0.8 \cos \frac{4\pi}{3}$

Chapter 9

Additional Observations

In developing the software and identifying potential periodic orbits, two interesting observations were made which resulted in additional investigation. These were the similarity of the figure-eight orbits to the Lemniscate of Bernoulli, and how perturbing certain orbits results in adding precession to the orbit.

9.1 The Lemniscate of Bernoulli

The three-body figure-eight choreographed orbit has an interesting property in that it closely resembles the Lemniscate of Bernoulli, and it is worth investigating if they match up.

Jacob Bernoulli (1694) defined what is now called the Lemniscate of Bernoulli as

$$(x^2 + y^2)^2 = 2a^2(x^2 - y^2) \tag{9.1}$$

where a is a constant defined by the focal points of the lemniscate and x and y are

the coordinates on the lemniscate.

To determine the value of a , the starting values provided by Chenciner and Montgomery [2] based on values determined by Carles Simó for a stable 3-body figure-eight pattern were used. These values can be seen in Table 9.1. The figure-eight was run through the Newtonian gravitation simulator and to find the value of x when one of the bodies crosses the positive x axis (i.e. the x crossing furthest to the right when $y = 0$). This gave a value of $x = 1.0810170862565978$. Putting this into the equation, along with $y = 0$ results in a value of $a = 0.7643945122705633$.

Table 9.1: Initial values for figure-eight orbit

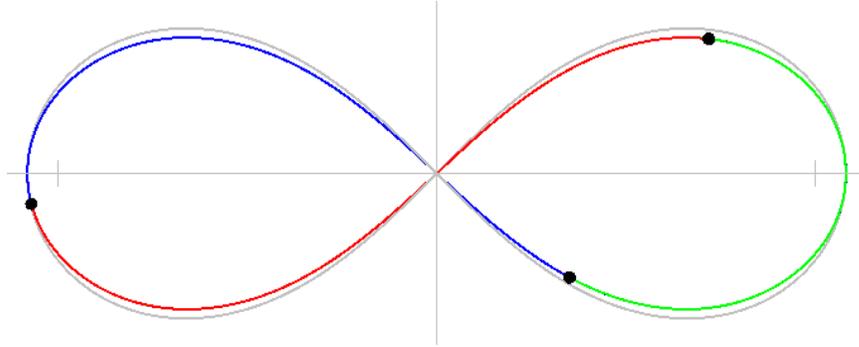
Body	x	y	x'	y'
1	-0.97000436	0.24308753	0.93240737/-2	0.86473146/-2
2	0.97000436	-0.24308753	0.93240737/-2	0.86473146/-2
3	0	0	0.93240737	0.86473146

Using the value of a , an extra curve can be drawn on the simulation. This curve is drawn by using the x -position of one of the bodies along with the value of a to determine the y -position predicted by the lemniscate. The resulting output can be seen in Figure 9.1.

In the figure, the light grey exterior curve is the one defined by the lemniscate and the inner curve containing the three bodies is defined by the motion of the bodies due to Newtonian gravitation.

While the figure-eight orbit is very similar that of the Lemniscate of Bernoulli,

Figure 9.1: Figure-eight orbit with superimposed Lemniscate of Bernoulli



overlaying them shows a distinct difference between them. Modification of the constant value a in an attempt to make the curves fit better at the minimum and maximum y positions would cause the curve to stop fitting at the minimum and maximum x values, so it can be concluded that the Lemniscate of Bernoulli does not match the lemniscate shape of the three-body figure-eight orbit.

9.2 Stable Precession of Periodic Orbits

As was seen in the orbits found in Chapter 8, as well as looking at the orbit of Mercury as discussed in Section 2.1, it is possible for orbits to be stable with precession. It is worth looking at how much an orbit can be perturbed before it no longer remains stable, as this is a test for stability.

To see how a single body affects the system, a body in the system is modified moved along a line passing through the origin and the initial position of the body. Given the initial starting position $\mathbf{v}_0 = (x_0, y_0)$, the body can be moved in or out to

a new starting position by multiplying the position by a factor δ as

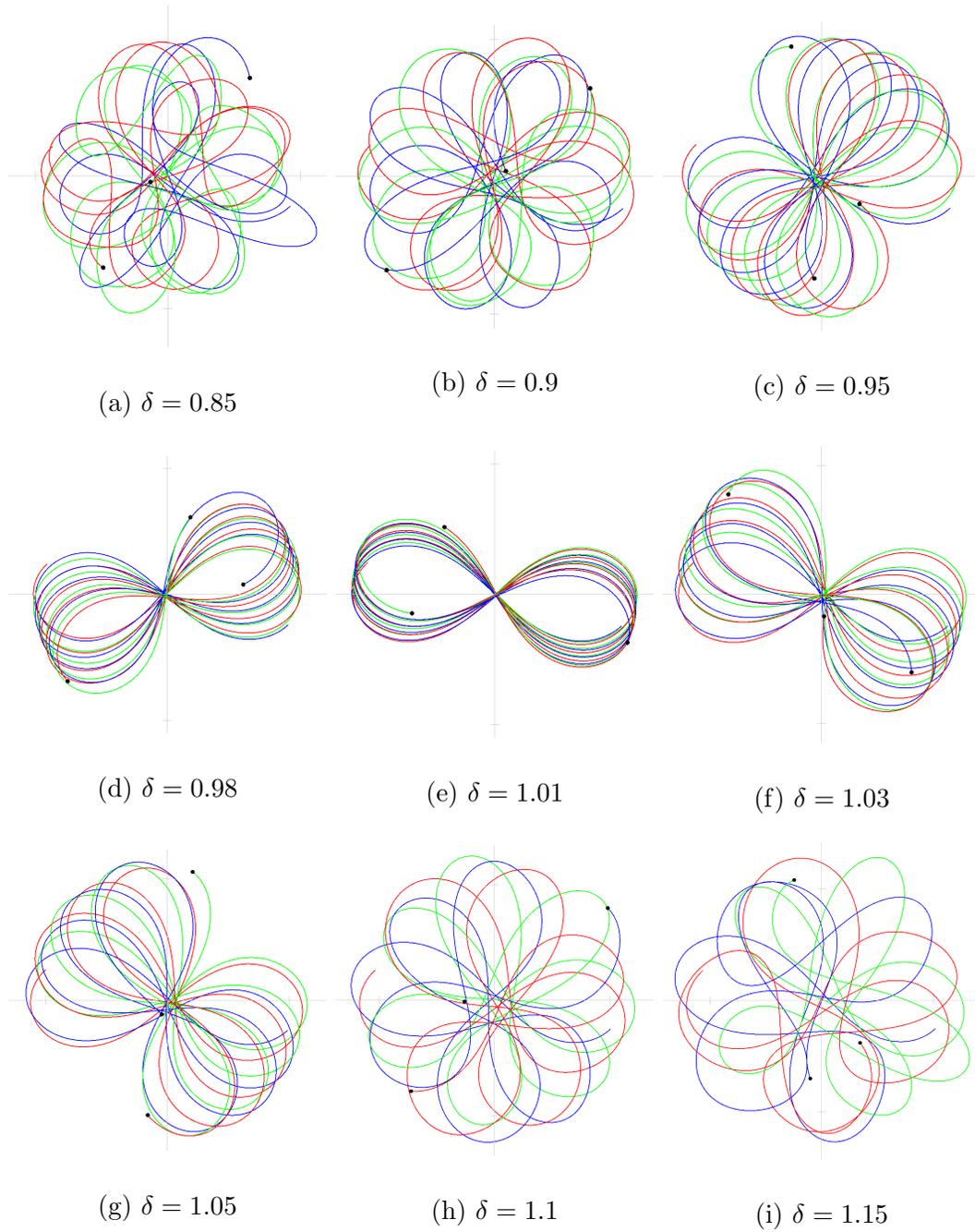
$$\mathbf{v}_1 = \delta \mathbf{v}_0 \tag{9.2}$$

The first case to investigate is the three-body figure-eight choreographed orbit. Again, starting with the values provided by Chenciner and Montgomery [2] as seen in Table 9.1, the initial position of the first body is modified and then the Newtonian simulation is allowed to run. Moving the first body in and out, and allowing the simulation to run for a time, results in the orbit paths seen in Figure 9.2. In each figure, the caption indicates the δ used to modify the initial position. The simulation was allowed to run until $t = 4 \cdot 2\pi$, or four complete periods and the paths each body took is displayed in the figures.

As can be seen in the figures, shifting by up to $\pm 3\%$ does not introduce enough error to cause the path to become chaotic and instead just introduces precession to the orbit. Moving up to $\pm 5\%$ shift, the paths are slowly becoming chaotic but they stay relatively stable for a large number of periods. At $\pm 10\%$ the paths are maintained for a short period of time before becoming chaotic, but modifying the initial value by $\pm 15\%$ introduces too large an error. In the case of $\delta = 1.01$, the figure-eight loops precess much slower than in the case $\delta = 0.98$ which causes the orbit overlaps to appear much tighter.

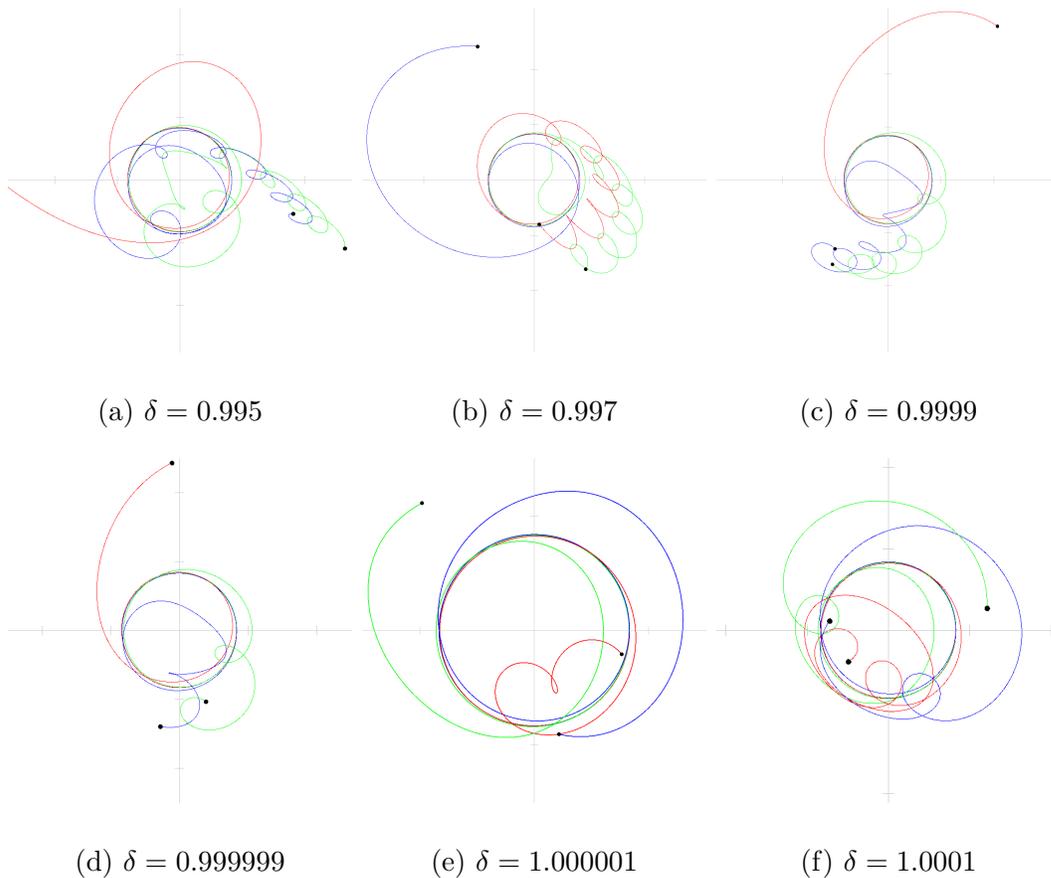
The second case to investigate is the three-body choreographed circular orbits. The initial positions of this orbit can be seen in Table 8.1. Again the initial position of the first body was moved in and out by a factor of δ and the simulation was allowed to run until $t = 4 \cdot 2\pi$. From the resulting orbits in Figure 9.3 it can be seen that

Figure 9.2: Precession of choreographed figure-eight orbits



even very small changes to the starting positions causes the paths to become chaotic almost immediately. Unlike the figure-eight which remained stable at $\pm 3\%$, a shift of only $\pm 0.0001\%$ causes the paths to become chaotic after only two orbit periods.

Figure 9.3: Precession of choreographed circular orbits



As can be seen by the difference in stability between the figure-eight and the circular orbits, the effects of modifying a single body's position can vary greatly. There is no obvious general rule that can be applied to determine the stability of a system after modifying the initial conditions, and so any precession and stability

analysis needs to be handled on a case-by-case basis.

Chapter 10

Future Work

There are a number of ways this work may be extended further. We found the modified gradient descent was quick and provided acceptable results, but there are a number of other n -variable minimizers that could be used. These could be used in place of the modified gradient descent, or used in addition to it, such as was done with the quadratic curve fitting. Additionally, the minimization could be more automated. Determining when the function has been sufficiently minimized, and controlling the scaling factor during iterations could be done algorithmically, to multiple minimizations to be run unattended.

Another addition to the minimization function would be to add additional constraints to allow for finer control over the minimization. One such constraint that could be added would be to ensure that angular momentum is conserved.

Finally, a number of more advanced mathematical theorems and conditions could also be applied to the systems to further investigate their stability. In the Newto-

nian gravitational simulator, the Courant-Friedrichs-Lewy condition could be used to formally determine the timestep size required for correct results. The virial theorem could be used to analyze the kinetic and potential energy of the orbits to help determine their stability. Finally, a more rigorous analysis of stability could be attempted for the orbits using the Hamilton-Jacobi equation and analyzing the eigenvalues of the Hamiltonian system using Jacobi fields.

Chapter 11

Conclusion

Finding new n -body periodic solutions in gravitational orbits is an open problem. Since the motion of each body must be solved iteratively, finding potentially stable orbits cannot be done with a simple formula. A few orbits have been proven to be stable, such as the choreographed circle and figure-eight, but many more potentially stable orbits exist.

To identify potentially stable periodic orbits, we developed software that generates Fourier series representing orbits. The software used a minimization function that compares the Fourier series orbit with the Newtonian orbit at a number of points on the period.

The minimization function began by calculating the second derivative of the Fourier series at a number of points and comparing it to the Newtonian gravitational acceleration on a body at that point. To find an orbit, the difference between the Fourier and Newtonian acceleration needed to be minimized. An additional con-

straint was added to the minimization function to keep the center of mass of all the bodies at the origin.

The initial minimization was done with a modified gradient descent algorithm. The partial derivative of the minimization function with respect to each Fourier coefficient was determined and for each coefficient, the sum of its partial derivatives over all the timesteps was used to update the coefficient for the next iteration.

Once a minimization function was smoothly approaching a minimum, the programs intersperse the descent algorithm with quadratic curve fitting. A quadratic curve for each Fourier coefficient was generated using its values over three gradient descent iterations and followed to its minimum, allowing the minimization function to quickly jump closer to a minimum value of 0.

Once a potentially stable period orbit was identified, it was run through the developed Newtonian gravitational simulator using fourth order Runge-Kutta numerical integration and variable timesteps for accuracy. The software found the starting positions and velocities of each body in a system by solving the Fourier series at time $t = 0$. The path was then allowed to progress using only Newtonian gravitation and the results were observed. In some cases the Fourier series orbits were interesting but were actually local minimum, and the resulting Newtonian orbits proved to not be stable. A number of stable periodic orbits were identified however, including the choreographed circle and figure-eight which match findings other researchers have made via alternate means. Some of the orbits found showed some amount of precession while still remaining stable. We found that while some possible periodic orbits became chaotic when a perturbation of the initial positions was introduced,

other orbits were much more resilient and small perturbations only caused the orbit to precess.

By using Fourier series to represent orbits and minimizing the difference between the acceleration on each body due to the Fourier and Newtonian acceleration, we were able to find a number of potential periodic solutions to the n -body problem.

Appendix A

Fourier Series Coefficients

The Fourier series for each orbit found are presented here. In each table the “Index” column is the coefficient index in the Fourier series. The columns a_x and b_x are the a and b coefficients for the x -position Fourier series and the columns a_y and b_y are the a and b coefficients for the y -position Fourier series. If the first coefficient a_0 is zero it is not included in the table. As a reminder, the Fourier series are in the form

$$x = a_{x_0} + \sum_{i=1}^n (a_{x_i} \cos(it) + b_{x_i} \sin(it))$$
$$y = a_{y_0} + \sum_{i=1}^n (a_{y_i} \cos(it) + b_{y_i} \sin(it))$$

In the case of the choreographed orbits only a single Fourier series is used, with time offsets used to determine the initial values for each body. For the unchoreographed orbits, the Fourier coefficients for each body are included.

Table A.1: Choreographed three-body circle coefficients for Fig. 8.1

Index	a_x	b_x	a_y	b_y
1	0.7548983603205649	-0.35141106974012754	-0.3514110697402252	-0.7548983603206388
2	-1.9150307611289275E-15	2.2360036541497445E-17	1.0746316306317935E-16	1.932412397911802E-15
3	-1.155278399273465E-16	1.586672222985794E-17	-1.2627005056661258E-16	1.4807338111954498E-16
4	2.1493226961371906E-16	-3.9623108916705373E-16	1.0344272047428232E-16	1.0469296740822034E-15
5	-2.192584409811363E-16	2.5454810690503273E-16	8.254995454316692E-16	-7.976576730858503E-17
6	5.892651453666586E-18	1.0799037031550896E-18	8.369160965856156E-18	-4.322064445878037E-18
7	8.911782171407457E-17	-1.162659174579948E-16	7.807524256218779E-17	3.5114136917452266E-16
8	7.351451010855196E-17	1.2505057795014027E-17	-1.2324568023800478E-16	-1.3037121716135135E-16
9	-9.067474666386747E-19	-6.187576736173928E-19	2.205967900199214E-18	-3.1383313572816968E-18
10	-6.310111074082526E-17	-4.4038817701780406E-18	1.0280602437442736E-16	-1.141116303043025E-16
11	4.990501404651108E-17	-6.338450189900628E-17	-2.3425608109295964E-16	2.6350610506766024E-17
12	-1.2351474069778525E-18	-1.8303978176322985E-18	-1.0695479637493678E-18	-2.1959375628251515E-19
13	-3.213383696043687E-17	4.0973510655277233E-17	-2.217178064646633E-17	-1.4646531101056533E-16
14	-1.7863484792721525E-17	1.4227484782168845E-18	3.5362666975103254E-17	1.8021048103201003E-17

Table A.2: Choreographed four-body circle coefficients for Fig. 8.2

Index	a_x	b_x	a_y	b_y
1	-0.6992422980279646	0.6944467629708361	-0.6944467629710738	-0.6992422980282387
2	1.5840764845037316E-17	-8.547797029174964E-18	-6.495442789092932E-18	-1.403794758841306E-17
3	-9.406667479226474E-15	-1.0959228427744459E-14	5.393378284949609E-15	-6.647486558551395E-15
4	7.32438875740547E-20	1.4508423396636423E-18	8.942314337190109E-18	4.477261138816237E-17
5	-1.4887221202381589E-16	1.3104157746358994E-16	-7.626187904709564E-16	-1.4786812358819632E-15
6	3.9290057688780394E-17	4.9410497910460896E-17	-7.213582935186658E-18	9.67355169225502E-19
7	1.2996661417398896E-16	1.286000356713211E-16	6.176405986731774E-16	-7.588825995378119E-16
8	3.0225849181050846E-19	2.5963504256457734E-18	6.57937586869542E-16	-9.341535914296715E-16

Table A.3: Choreographed three-body figure-eight coefficients for Fig. 8.3

Index	a_x	b_x	a_y	b_y
1	0.2176530096583163	0.4456800765154419	0.42803565283137085	0.8751469569744517
2	-0.23766250591571017	-0.18519048182149087	0.12089459041179575	0.0941848697746536
3	3.0387249528375605E-5	3.7906119159809098E-6	-6.218346808909589E-6	-2.5609960843165065E-6
4	-0.048127890614614935	0.012128894572191067	0.024455443217361286	-0.006254982637285189
5	-0.008651189112234942	0.00733478889270486	-0.017173910560992767	0.014542744891938058
6	9.632850445794124E-7	-1.2498384463979924E-6	-8.463865878180795E-7	2.3093088649608327E-6
7	1.1574921606010391E-4	0.002597276853219944	2.2165415328532142E-4	0.005152471623695889
8	-0.0012459951314085815	-0.0023196020823530424	6.298973590986714E-4	0.0011477807001238456
9	5.058442834108797E-7	6.447192094018169E-7	2.0936288528594602E-8	3.1158760059785906E-8
10	-6.846983486635172E-4	-1.255756510266923E-4	3.5726881027828993E-4	6.269960769177977E-5
11	-1.7768877308305413E-4	5.7117130718706356E-5	-3.4401770372315257E-4	9.789521018362353E-5
12	2.1938148696231915E-8	-7.789613920687731E-8	4.55864106106472E-9	1.4790314679749991E-7
13	-2.1064322829912827E-5	3.8279878367731235E-5	-3.938878299959574E-5	9.061525979218574E-5

Table A.4: Choreographed four-body trefoil coefficients for Fig. 8.4

Index	a_x	b_x	a_y	b_y
1	-0.21818179023741846	-0.35180865345607926	-0.48561306784621855	0.2870460991118993
2	0.37374585031919794	0.37176324429184593	-0.37192954077433876	0.37229356291694715
3	0.00229609142978605	-0.01165194328584045	0.009024411281896221	-0.0386567325504359
4	7.859387119088035E-4	-1.982822519229455E-4	-7.413359053393541E-5	-2.480329675339246E-4
5	-0.04542724072143844	0.027982356901784067	-0.02363369068966606	-0.04031454068882473
6	0.007408609536063562	-0.008904830214951856	0.00792376228622713	0.007227263309350796
7	2.0254338691290524E-4	0.006048225169384902	0.0025957158013560876	-5.035959219868777E-4
8	1.5288298213605577E-5	6.17331987803343E-5	-1.3396052708293783E-5	9.412871403905804E-6
9	0.001637950643496527	0.002828962693452802	-0.0032170212459698604	0.0026298272304605946
10	-0.0014850192374428613	-6.435873268237417E-4	4.839192240845605E-4	-2.4300375717753174E-4
11	-0.004910382270007302	3.0630617541027E-4	-4.8416931469498134E-4	-0.005898932703098607
12	4.491838461506581E-6	5.677027432000256E-7	3.1003525992139407E-7	-6.320682296045191E-6
13	3.15611190925027E-4	-3.2512667392058624E-4	1.825950700073688E-4	2.993127056136419E-4
14	5.124471528160966E-4	-0.0030287279229078193	0.0032528260964720963	5.289283149106808E-4
15	1.7172895817895294E-4	0.0011367210333987328	-0.0011074805147649129	1.530335102923528E-4
16	2.7146304178585593E-6	-1.3920373423701747E-6	-2.038489117714004E-7	-3.709143354145493E-8
17	8.363733760260929E-4	3.325469272797656E-4	-4.0190041164145796E-4	9.611451724150175E-4
18	-8.244810235537325E-4	-5.2463123656415344E-5	7.430707865178096E-5	-8.052588812492951E-4
19	1.814734018591389E-4	-6.81827524013189E-5	1.5791550327611686E-5	1.3979552521784134E-4
20	-1.2871423966060112E-6	5.838569285997445E-6	3.839526994450627E-7	1.4069626587090317E-6
21	1.1589931637595746E-4	-2.971980428129216E-4	3.0325810682029907E-4	9.412280840780452E-5
22	-4.875051188044247E-6	1.569738711998738E-4	-1.4560735347721594E-4	-2.026681228712151E-5
23	1.6210408113961298E-4	1.3702601936543135E-4	-1.6743383690820477E-4	1.611501032036193E-4
24	-2.3610877332772047E-7	-1.739363279164875E-6	-1.14994442794577E-7	-5.209844632984745E-7
25	-6.733486219457434E-5	1.2428222400238723E-5	3.767601282346654E-5	-6.313784492935416E-5
26	-1.237332254104557E-4	7.686729830485E-5	-7.101034201239648E-5	-1.2825954795042834E-4
27	6.949500502864025E-5	-7.525049845425157E-5	7.82302714901067E-5	7.726540841769711E-5
28	6.495534577755363E-7	3.568520445209692E-6	-2.176252015042144E-7	-3.2194865932180116E-7
29	-2.0975339834326778E-5	-5.0989600551447665E-5	4.39177647163428E-5	-1.8695460170696717E-5
30	4.6029948153356503E-5	6.932253746775258E-5	-8.243713590064795E-5	5.2443272982306294E-5
31	4.5884605540394235E-6	-1.4445915568122353E-5	1.915102450381389E-5	-6.552312469119472E-6
32	-2.176543909467466E-6	-1.8699241326040297E-7	2.909514417026145E-7	-3.56073115058382E-7
33	-4.380202651794108E-5	2.297112024017074E-5	-5.196990687410373E-6	-3.1809098904028264E-5

Table A.5: Choreographed four-body chain coefficients for Fig. 8.5

Index	a_x	b_x	a_y	b_y
1	-1.1920168613261075	0.28140929021797434	0.5087071050154558	0.006696236539724962
2	-5.392227219497806E-4	-4.34675665239019E-4	-6.859069541201709E-4	5.661030167270081E-4
3	0.02253250259526173	0.09468487425989221	0.13254001983013455	0.1736262722085104
4	-2.951440502466351E-6	-3.760482912279364E-7	-3.3910192810265936E-6	2.5760975166475356E-6
5	-0.020956866137226422	-0.014793273517415224	-0.05238675763064296	-0.033586016150804664
6	7.830907518401013E-5	3.911491564682996E-7	1.3475650066669615E-6	-8.101306060272862E-5
7	0.006307707198408547	-0.02043561583234652	0.0031748190601258424	0.009705320499903513
8	2.1039054875041925E-7	-3.2224171807862937E-7	3.677357433875016E-9	8.897106583048406E-8
9	6.790239758091741E-5	0.00907732355760707	-0.0044535677438441	-0.002863462236650776
10	-7.80266699952667E-6	7.211265269670601E-6	7.538514259448851E-6	7.401641338516054E-6
11	1.4714795344197363E-4	-0.002981764032375554	0.003998366605273357	-0.0016808361419626992
12	4.1612608450513985E-8	2.881229115116096E-8	-7.061266283336499E-8	2.8644271511799302E-8
13	6.315973664745375E-4	0.0016612059846027552	-0.0015908586313868659	0.0014553620466306212
14	6.631187611873235E-8	-1.7659620952787435E-6	-1.8227892171240462E-6	-2.1592504206714135E-7
15	-9.724667558314082E-4	-6.683925073513836E-4	7.280221328777237E-4	-6.862541285493843E-4
16	3.8447740123237237E-8	3.1031133214241217E-9	2.767237079298632E-8	6.541386768415347E-9
17	5.960771691626762E-4	9.91580938900138E-5	-2.6025913720021147E-4	5.944783309444898E-4
18	2.1141640098545532E-7	2.708338120840619E-7	4.5076283424241014E-7	1.8504533275064162E-8
19	-3.6326523651359407E-4	-5.797615617368972E-5	3.2376542755910666E-5	-4.429946208132072E-4
20	1.383846040054533E-9	2.46938596547957E-8	1.8087991213102835E-8	2.6292380842987386E-9
21	1.1184033398329467E-4	-7.490829798550757E-5	1.1634533558011079E-4	2.842799979274052E-4
22	-3.277548815511649E-8	-1.4384011999788404E-8	1.2321234850965914E-7	6.57498341870955E-8
23	-7.868623683522561E-5	9.00969047111248E-5	-1.0049351902966427E-4	-1.0209907967735063E-4
24	-5.391912283127185E-9	-1.708128200377265E-9	2.5346974509248556E-8	-2.0621360891411E-8
25	2.4955597318615343E-5	-7.09391359008292E-5	9.376316521139023E-5	1.993034622274571E-5
26	3.2322307823984357E-9	-1.7674781866450824E-8	-1.661849576189965E-8	-8.110539266817182E-8
27	8.269119779097485E-6	7.97338783859528E-5	-9.217654600754323E-5	2.3846515581268222E-5
28	-2.422147104950989E-9	-2.4858034706023494E-9	1.0784724689615103E-8	5.108190014642779E-10
29	2.5817174649721484E-5	-2.3293834214391517E-5	3.653211881098584E-5	-5.527821474491722E-5
30	-6.214341199293822E-9	4.87714675492204E-9	-1.9790883787810198E-8	-2.8310765971632514E-9
31	-1.8884752088026874E-5	5.276242962534998E-6	-8.80833818698521E-6	8.169133519668523E-6
32	2.3215381796762767E-9	-4.291277537895737E-9	-1.088933729569972E-8	-1.0058142865842846E-8
33	-3.2838417678791586E-6	1.0948496934270798E-5	6.953551503336392E-6	-2.6221327001475526E-6
34	4.473662329509446E-9	3.654678031660191E-9	4.4985406870529455E-8	6.195902387807065E-9
35	-9.02555710358268E-6	-1.4156326021058792E-5	9.622179948899615E-6	-1.4309000130823149E-5
36	4.298846696218809E-9	-3.451453041469761E-10	-1.8285684581648212E-8	4.346300920221835E-8
37	-1.3576300633508058E-5	-4.969299716295609E-6	8.354697741993223E-6	1.0144200293873792E-5

Table A.6: Choreographed four-body inner loop coefficients for Fig. 8.6

Index	a_x	b_x	a_y	b_y
1	-0.5863637254644544	-0.20360823032495845	0.33464309023604566	-0.4694719281970928
2	0.4341835714607591	0.12067815222478132	-0.1389947873413346	0.42325194078268613
3	0.14818362461934828	0.018760715466636057	-0.015605158135815162	0.14951953529464898
4	-4.723324146224298E-5	8.77541932839509E-5	6.200167930525036E-5	-8.184085239244896E-5
5	-0.034038768838151816	0.007198819836347653	-0.009253846657856635	-0.03407943071923219
6	-0.021419337896183947	0.009389625113206717	-0.009237882323114847	-0.021378333713062303
7	-0.006171819473500474	0.003927862767383613	-0.004186099115290647	-0.00599066151122887
8	1.0411116866289987E-6	-6.0809110312426415E-6	-5.1167474296045686E-6	7.80474190606329E-6
9	9.279556866028861E-4	-0.0012276899060907542	0.0012046874537808363	8.937377340183751E-4
10	4.480922095973371E-4	-8.240347099208988E-4	8.408323759388756E-4	4.2002713526071036E-4
11	1.1449790446259833E-4	-2.561319987812255E-4	2.784924207022036E-4	7.245472896241279E-5
12	-1.126379580811363E-6	-5.023108934126972E-7	1.2194079643552831E-6	4.3992133445212936E-7
13	-1.2019573355569363E-5	3.8081112879623844E-5	-3.394475591916479E-5	5.840798876709223E-6
14	-5.358807398840856E-6	5.029786697856876E-6	-1.3828708612055842E-5	-3.251322352236467E-6
15	-1.374139490447088E-5	-4.357805225789137E-6	-5.3877825905035425E-6	-9.605405254860716E-6
16	-4.558427930562271E-7	6.049536337971546E-7	4.999597204263851E-7	-6.283517077358721E-7
17	1.145378817839415E-5	5.714361304455382E-7	4.611733929045024E-6	1.36992541024505E-6
18	3.7077856426902888E-6	-1.2363851485042517E-6	7.675818232199572E-7	8.309261607025398E-6
19	1.2490582017684216E-7	3.7461431879084772E-6	-4.126852806782228E-6	5.678203685902371E-6
20	6.19809673448354E-7	9.866791751998575E-8	-6.652041066647746E-7	-1.5856831322978314E-7
21	-8.162809671998237E-7	-2.6758055641570888E-6	-3.449053040149737E-7	-4.200909813764541E-6
22	-2.2369491150566556E-6	9.143789938271856E-7	2.3304044290766413E-6	-2.051546515774726E-6
23	2.3832543867921347E-6	1.0333002758473463E-6	3.1657122849923416E-6	1.6998741393078394E-6
24	1.531017494777801E-7	-2.245697728711411E-7	-1.6049325091245718E-7	2.7451160166070176E-7
25	-1.9276431599386903E-6	2.7364527885152825E-7	-2.534328550059829E-6	8.620772142870813E-7
26	3.176795589826954E-7	1.0046258188421226E-6	-9.910976547718265E-7	-7.500630932786538E-7
27	2.3425283303927393E-7	-1.995359009275045E-7	7.52653528169294E-8	-6.024836128508093E-7
28	-2.1744364882980707E-7	-8.344706571530263E-8	2.685646591340352E-7	4.44984469203522E-8
29	6.368998003124848E-7	-1.9062507069869206E-7	9.443793170968771E-7	-7.003197410489741E-8
30	4.982145447994164E-8	-1.08329853214548E-6	-6.138148432872726E-7	2.3436599189714984E-7
31	-1.8044160977274974E-6	-9.57512268919394E-7	-1.5750264408700995E-6	-1.526112555246664E-6
32	4.085323651914081E-8	8.968417159616096E-8	-9.455852439394777E-8	-8.166660951584981E-8
33	1.1507558481911302E-6	1.0999915686426583E-7	1.182416363019268E-6	1.355272695890821E-8
34	-1.1945355831502473E-7	5.233122156644593E-7	1.532318600275573E-6	7.844541705617454E-7
35	1.8041130555172197E-6	2.411043034880743E-6	1.5143140242490936E-6	2.957098365481605E-6
36	-7.145686242578804E-8	3.152787377880425E-6	-1.4427503005564479E-7	-4.749517630328291E-7
37	-2.169242078227881E-6	-1.322933798154378E-6	-3.2465808245045024E-6	-2.556412804808969E-7
38	-6.545626771976379E-7	5.43142555715863E-7	-8.422016319467004E-7	-2.0167245200721667E-7
39	-8.194121003264663E-7	1.9259054204258075E-6	-8.867378497724163E-7	1.9164723646451874E-6
40	1.0274209102531651E-7	2.2292407131528963E-7	-1.1162439446760669E-7	-2.995777268191457E-7
41	2.466671671046875E-7	-1.9579754357575778E-7	1.1599274019952291E-7	-5.76403321713268E-7
42	3.002861543746831E-7	-6.300530114522537E-7	-3.575548297244651E-7	-2.2634664652442708E-7
43	-7.789790533134595E-7	-1.290389270553542E-6	-7.721664615909525E-7	-1.5193368719432616E-6
44	1.6743269716360202E-6	1.0519079543884208E-6	-1.8279609455888973E-6	-1.1690940625332435E-6

Table A.7: Choreographed five-body figure-eight coefficients for Fig. 8.7

Index	a_x	b_x	a_y	b_y
1	-0.16365157403042543	0.3962041385194827	0.5242689043637169	-1.251850854937724
2	-0.3370742874318663	0.33372627739393157	-0.10587389156118429	0.10351043364209389
3	0.010301714843625112	-0.004212961348000595	-0.03315626666875662	0.013167227328670712
4	-0.03455684024836918	-2.8103839535126865E-4	-0.01103638632058648	-5.562646361063848E-4
5	-1.0751390222479793E-5	4.769467178640183E-6	-2.542708661679079E-5	-6.055707727052003E-6
6	-0.009979533614770202	-0.010373101851709227	-0.003118292396659717	-0.003210899947670573
7	5.185086058990445E-4	0.0014471833820078547	-0.001772422597406034	-0.004479995490086375
8	7.814352784050354E-5	-0.0032311786028543427	1.1179817454685592E-4	-9.777889010666539E-4
9	-4.647885122376092E-4	7.947422028956417E-4	0.001272702980716266	-0.0027602603242219315
10	3.959564465567611E-7	-5.152693597268808E-7	1.319267208888729E-6	-7.038440173930091E-7
11	-3.0180017449211186E-4	8.119201081629171E-5	9.880352105695067E-4	-4.1118238014750837E-4
12	-4.2836565592801416E-4	-1.8659098212908335E-5	-1.3479522352301433E-4	-2.4379055995300683E-5
13	-6.712235546221255E-5	-4.043337939627013E-5	2.3461800078285083E-4	1.1013164512411674E-4
14	-1.7827827724017085E-4	-1.9746095230382558E-4	-6.121581335928178E-5	-6.202549262863486E-5
15	5.988033448263989E-8	2.3480620274289608E-7	-1.497685040658491E-7	4.909091554346601E-7
16	-4.479516501733158E-7	-1.101411090657505E-4	1.645642656046875E-5	-2.4040665663433124E-5
17	-7.734124885007479E-6	1.1171773528815274E-5	1.9918487333957393E-5	-3.699479769503142E-5
18	2.046687978006187E-5	-1.8938434096760044E-5	7.566999416422887E-6	-4.6388797217784125E-6
19	-1.001646732002023E-5	1.7594014264574727E-6	2.88224626420575E-5	-1.28247294630325E-5
20	5.550643891536263E-9	-2.3312133167287148E-8	1.7376684934232406E-7	-8.827434373217093E-8
21	-1.5707325294289703E-6	-3.220803219384208E-6	8.022332376886483E-6	5.281686851442154E-6
22	-3.3284839235260187E-6	-4.023115850025911E-6	-1.1720447228946292E-6	-1.4607447656640968E-6
23	-1.8684769678405506E-7	-1.1441178136295095E-6	1.3241362134223397E-6	2.5675832455558576E-6
24	-5.90591332782866E-7	-3.0720285785143786E-6	1.8377792613645944E-6	-2.6513077377160676E-6
25	1.552437539787133E-7	3.350106735499737E-8	3.032629145539551E-7	1.2059791689741756E-7

Table A.8: Choreographed five-body pretzel loop coefficients for Fig. 8.8

Index	a_x	b_x	a_y	b_y
1	0.2600512129518242	0.05754854590023973	0.17883836955028115	1.109440779001036
2	-0.005641765457476541	-0.1266346913200789	0.31969166688641154	-0.09364753551580997
3	-0.23682966656936402	0.16478636708142944	0.13660366308249516	0.16685749768093436
4	-0.10693554022701625	-0.10846461089573393	-0.06604269315362374	0.07763600817969096
5	2.3371376929076318E-4	-1.4840943042795685E-4	3.600592992319922E-4	-4.9886667712044574E-5
6	-0.027074066080924105	-0.014297722106623374	-0.001632511864470335	0.011890864039471623
7	-0.0028030081475470844	0.013909986192491489	0.008001825640031077	0.0012600222060709326
8	-0.029392297143526976	-2.752874520883621E-4	0.0013645989640289863	0.023791687568252503
9	-0.0033921608458276436	-0.012796773258599715	-0.01618818868615238	0.0028457910006658184
10	2.9349958935760645E-5	-2.835497252175668E-5	2.1003038438442993E-5	-9.299468096470575E-6
11	-0.0012280197729411513	-0.0010938141240566155	-0.0028880951007931844	0.001710937041952022
12	-0.0014882473005725625	0.0015572660675133897	0.0020094412560131185	0.002073568194694982
13	-0.004202364988365959	-0.002593415159370602	-0.0028727692844175974	0.004382420798195334
14	0.0012139871933786524	-0.002856220889104768	-0.0023751210514776477	-7.799250168776344E-4
15	6.086027275904716E-7	5.324719181023139E-7	3.07338814283453E-6	4.0735283248848767E-7
16	1.8599673163299438E-5	-5.349969392634466E-4	-4.2542082609010054E-4	1.1283315891582947E-4
17	-6.230383654160546E-4	1.3386329887041183E-4	1.4573352047148413E-4	5.013703868821608E-4
18	-5.461157532402132E-4	-0.0010756987417290611	-9.721727279068121E-4	5.303488764039622E-4
19	4.664954333596062E-4	-4.606324628962489E-4	-4.460337398585181E-4	-5.154310547349268E-4
20	-4.906068259990713E-6	-2.2699652907939908E-7	-4.483128594825174E-6	5.53640969970624E-7
21	2.243164399341345E-5	-1.2292635831029643E-4	-1.229270906559646E-4	-8.553917933182316E-5
22	-1.2457708197123528E-4	-2.5951814487372657E-5	-5.294803182921705E-5	1.3048143147873057E-4
23	7.769657526757024E-6	-2.979838331249011E-4	-2.7751979519965525E-4	-1.4917384193464355E-5
24	2.1186497484312276E-4	-5.0287413637972705E-5	-4.6429510848813066E-5	-1.787434879131297E-4
25	-2.774487592087521E-7	-2.0995267986100378E-7	-2.530820401830415E-7	2.2923383344225515E-7
26	6.83843191740339E-5	-4.591397460176739E-5	-2.8311525638088687E-6	-1.872444876144334E-5
27	-2.1258225319618278E-5	-7.4081622829000435E-6	-3.6737066182843166E-5	1.765498014495811E-5
28	4.671760469327048E-5	-6.384065379901193E-5	-5.353559235901486E-5	-3.957576778806548E-5
29	6.0631692970336796E-5	2.4828029124699414E-5	2.408113024710073E-5	-5.325143736801293E-5
30	-9.17221305113645E-8	7.518041980927958E-7	-1.1876063057768395E-6	6.494531870999119E-7
31	2.8773768938686142E-5	8.712918531787415E-6	4.393442792461907E-6	-5.81455556980233E-6
32	-1.4730058038402322E-5	3.1536487779184154E-6	-9.41722116214297E-6	-2.950979069820752E-6
33	2.1491597865740416E-5	-1.7189651140782895E-5	-5.946260980082242E-6	-2.060491131931754E-5
34	1.0430716575805613E-5	2.3323782748779693E-5	2.097897540850491E-5	-1.0798738929758907E-5
35	-9.958720259411818E-9	1.2531944409787623E-6	-5.188571394614994E-8	1.384522759393017E-6
36	4.078750788650047E-6	1.4823409412636199E-5	-1.480412756006276E-6	-5.7394028114649124E-6
37	-7.854920410076386E-6	1.820178895601459E-6	-1.2627397160758256E-6	-7.06081666194709E-6
38	2.7416853298687293E-6	-9.654040477676302E-7	9.395210870953566E-6	-1.0017070136785061E-5

Table A.9: Choreographed five-body trefoil coefficients for Fig. 8.9

Index	a_x	b_x	a_y	b_y
1	0.020512312921712076	0.3315402529668549	0.050881773776277464	-0.912658807176195
2	-0.10849493250204766	-0.5714715102087544	0.4333504108776496	-0.14783638056784168
3	-0.08602538484987002	0.0520674030101522	-0.004212457206805521	-0.17473604172005247
4	0.011914032151299327	2.2992738957267875E-4	-0.03251815044797603	-0.006245160085559099
5	1.461837667248285E-4	-1.5974200684101331E-4	3.367538723590266E-4	-3.1170641665606965E-4
6	-0.0014406873317469335	-0.025075102417597285	0.02210507385673589	-0.002998832778496828
7	-0.02153513713928313	9.95062019377028E-4	-0.0016146134518039666	-0.020557922350274635
8	-2.6238847091858127E-4	0.006428553021973196	-0.004808888293012105	0.0016503905206129918
9	-5.449422624238472E-4	8.815927526749619E-4	0.0010640150149012339	-0.0026132547378914473
10	-7.320570166776272E-6	1.1576760375541459E-5	-8.578886565821875E-6	7.380330669921115E-6
11	-0.00318702904338432	1.1454294030459832E-4	5.797739204853349E-4	-0.003950447540710948
12	-2.7487582623374033E-4	0.0019807035033689293	-0.0018882118693909471	8.017859370399472E-5
13	-0.0019163289637130286	-6.595424093176794E-5	9.871981662135712E-4	-0.002543814148917482
14	-4.098794448144552E-4	0.002287523980393218	-0.0026297245915735696	-8.276178107786607E-4
15	3.897746011656107E-6	-6.944614040718093E-7	1.5832644218906636E-6	-1.80286550130598E-6
16	-4.581797617019268E-5	3.2169352675278024E-4	-3.614207034826104E-4	-6.68838182357426E-5
17	-0.0011730142731161523	-4.4352473687061145E-4	5.699659842140809E-4	-0.0012346593567045855
18	-5.623570331162263E-4	0.0015362993356759966	-0.0016189851253708668	-6.80321361117968E-4
19	5.66843638142207E-4	1.9956946819300357E-4	-2.057591750244331E-4	5.751860771597646E-4
20	-4.4550901301833354E-7	-1.1132589628563052E-6	-1.0250993701302292E-6	-1.2835601889737744E-6
21	-3.5555264621124074E-4	-2.137957126001533E-4	2.4453666739853537E-4	-3.6237009806418025E-4
22	-3.822817373871913E-4	6.71289029640319E-4	-6.625945811679489E-4	-3.901183916829599E-4
23	3.8346628880465657E-4	2.087817803956814E-4	-1.633336538832861E-4	3.6939896694250307E-4
24	-1.7070224269343697E-5	2.997925429112581E-6	-1.4783263805028812E-5	-6.26381958535967E-5
25	-8.032613751210515E-7	9.042932350562856E-8	-1.9306401198066466E-6	-1.1631293172652837E-7
26	-1.3538697419031794E-4	1.8066949957183285E-4	-1.8739306449626626E-4	-1.3993220526142198E-4
27	1.50503882087091E-4	1.1081980954948394E-4	-8.655992342811338E-5	1.5235094992997798E-4
28	-3.999418231635606E-5	1.532106292758917E-5	-4.3655974987490416E-5	-5.808081565813274E-5
29	8.164832983790772E-5	8.320267023968079E-5	-9.476037597444115E-5	9.10183359264538E-5
30	2.9301548791564023E-7	-4.211631589426431E-7	3.734120236928149E-7	-5.553204208308868E-7
31	4.152135310667031E-5	2.7445502058328748E-5	-3.0169821201317507E-5	2.702055093463697E-5
32	-4.159622524079175E-5	1.0575118407680011E-5	-3.890556661745416E-5	-5.1851334046087356E-5
33	5.764831941628786E-5	9.799998375083604E-5	-1.0276907954477884E-4	5.2170843393281195E-5
34	4.579748442786917E-5	-3.279795670183894E-5	2.7602354575120832E-5	4.680563135004926E-5
35	1.5038322525903967E-7	5.631568296621946E-7	1.5210580252729217E-7	1.246823177790435E-6
36	-2.264840564832622E-5	7.53874433636617E-6	5.758624233227442E-7	-2.3702218634270516E-5
37	2.784578947126513E-5	6.102090372126453E-5	-5.064503027607154E-5	2.8929156591768456E-5
38	3.731269483323855E-5	-2.464004622102332E-5	6.30874424170572E-5	4.7972007604451425E-5
39	-8.601953937633656E-6	-2.413918821151523E-6	7.569221427139716E-8	-2.258382714523404E-5
40	-1.2015247951539485E-7	-6.072677802277099E-8	4.382684419835515E-7	-1.039797346675782E-6
41	4.1843162692165935E-6	1.8909672868756703E-5	-2.1096667430378742E-5	8.330780107223321E-6
42	1.5331479015438242E-5	-8.478243036221689E-6	1.9718680183336463E-5	2.720626761510263E-5
43	-1.7034018274659808E-6	-1.4657865960152732E-6	2.6628648299113466E-6	3.039371697128631E-6
44	9.727526486200372E-6	-1.9192119663470965E-6	2.0842437149440007E-6	8.962628398293324E-6
45	1.4242664538777184E-8	-1.671618191268762E-7	3.289922400925879E-7	-4.556899519709534E-8
46	-2.592550285284001E-6	3.2061803888245005E-6	7.251120544363338E-7	-1.5300585352039073E-5

Table A.10: Three-body circle coefficients for Fig. 8.10

(a) Coefficients for body 1

Index	a_x	b_x	a_y	b_y
0	1.4605605023968157E-9	0.0	-1.4604536330254136E-9	0.0
1	0.80647271105693492	-0.20727782667716863	0.20727242775853796	0.8064688722555448
2	-8.472847859534296E-7	1.224933578021879E-6	-1.104850911400487E-6	-7.931864702838543E-7
3	-2.1157743277582987E-7	6.150853944624093E-7	-7.121988646216031E-7	-1.727908927179269E-7
4	-4.6755573177964155E-8	-9.458052550523652E-8	-4.614923286861201E-8	-1.3433251338603074E-7
5	2.09878953430838E-8	6.622607491352736E-10	5.364689753328171E-8	-1.4348112962101537E-8
6	2.8477043939111803E-8	2.5661054707540648E-8	1.2562608622713474E-8	2.737964533512327E-8

(b) Coefficients for body 2

Index	a_x	b_x	a_y	b_y
0	-1.1879432059057798E-9	0.0	1.1880097959876623E-9	0.0
1	-0.5827363158047619	-0.594786384346318	0.594795274506634	-0.5827386900459348
2	1.7440760269745606E-6	2.6620532586167326E-7	3.025515799317771E-7	1.5930596686635705E-6
3	5.745816946874389E-7	-1.1108891998442633E-8	1.3536630154355804E-7	7.973737072982364E-7
4	3.312223656255306E-8	7.683642271047498E-8	1.3268944708880325E-8	1.1224514250099403E-7
5	2.1372295854366846E-8	2.2561733600274758E-8	7.573035673097334E-9	4.577280164888958E-8
6	-3.44712348301349E-8	8.268953573700722E-9	-2.6369254821170643E-8	2.3287734683263075E-8

(c) Coefficients for body 3

Index	a_x	b_x	a_y	b_y
0	-2.7261729649103677E-10	0.0	2.724438370377538E-10	0.0
1	-0.22373660137859494	0.8020638323644752	-0.8020674648432556	-0.22372973941643787
2	-8.955580257976309E-7	-1.402275147920433E-6	7.848494475089005E-7	-8.279901051968732E-7
3	-3.566553231206447E-7	-6.091708361880141E-7	5.897233256862484E-7	-6.180727323600622E-7
4	9.116214030303089E-9	1.3990956143568473E-9	3.6905909239951186E-8	3.68896286642893E-8
5	-4.071301865038872E-8	-2.349780742374822E-8	-6.499011367284745E-8	-2.978348864830188E-8
6	8.646839389620824E-9	-3.175980293219664E-8	1.1193811379619509E-8	-5.259513354448923E-8

Table A.11: Three-body circled cross coefficients for Fig. 8.11

(a) Coefficients for body 1

Index	a_x	b_x	a_y	b_y
0	-2.1217845337777366E-7	0.0	-1.4669322080367649E-7	0.0
1	0.8960721489109486	0.4265128480864686	-0.42205074952576593	0.8888382827413434
2	2.2104931184610615E-5	3.666774942253576E-4	2.8753547241472486E-4	-9.075550963448619E-5
3	0.0046324542473141295	-9.685306736364869E-4	0.0015373407449030118	-0.004373519779609295
4	-4.48079494674178E-5	5.287768908781594E-5	1.3110651114526698E-4	8.279440215639287E-5
5	0.006265459228163849	0.008870750998035621	-0.008896698533359705	0.00624148934633266
6	-5.1956790336414215E-5	-2.0260515663783344E-5	-2.1721571625022914E-5	6.250047859495722E-5
7	1.3759126502136066E-4	-2.7173673294041318E-5	1.5234089352717233E-4	2.0761028520245264E-5
8	1.037031407243873E-5	-4.915807165648985E-6	1.5394939068988876E-5	-1.9026083848575475E-5
9	5.959123162488419E-5	6.432443788202887E-4	-6.96508600133468E-4	9.867424495206803E-5

(b) Coefficients for body 2

Index	a_x	b_x	a_y	b_y
0	-1.1954023309410316E-7	0.0	5.1432316661146E-7	0.0
1	-1.0405448426949055	-0.1076789986856698	0.32817038504878787	0.14682047031772133
2	-7.202018245006315E-4	1.0818320895937176E-5	3.313878790497535E-4	-3.802380517952626E-4
3	0.020676305996031995	0.020130166915380422	-0.019646471362576783	0.025190990350168106
4	2.1135020626674982E-7	-1.0051477435522349E-4	-2.698400468480114E-5	-1.5060494903146494E-4
5	-0.004744798704371927	-0.005105624077232802	0.0040854299404422985	-0.0014245090226489014
6	6.156792172503539E-6	6.527171822549903E-5	3.390774580369679E-5	-3.4418062864506754E-5
7	3.768210673778716E-4	0.0012119961967867415	-0.001227493208695589	4.219336779966284E-4
8	-3.8126033441265335E-5	-2.6701401851250535E-5	-1.78883984371225E-5	5.291566035607207E-5
9	-6.24616380146515E-5	-3.580227455867407E-4	3.572682743170702E-4	8.57018397784925E-5

(c) Coefficients for body 3

Index	a_x	b_x	a_y	b_y
0	3.317186864718768E-7	0.0	-3.6762994580778375E-7	0.0
1	0.13888351698141674	-0.32965952301639556	0.09626922685087222	-1.0379420306088813
2	-3.586816508525718E-4	3.2884311224027404E-4	-4.6373276390963456E-4	3.0104542006410167E-4
3	-0.02490460392178365	-0.01956914101609359	0.01814861405234655	-0.02079812539515158
4	-2.4339575379128998E-5	-1.8323688989749974E-4	-9.425725092864877E-5	6.46191177769445E-5
5	-0.0015748169324063049	-0.0037508737967153673	0.004802057739107326	-0.004806066169451975
6	9.625432631006567E-5	-2.0478336863141727E-5	-9.300025489386003E-6	-1.813252707547664E-5
7	-5.060740926712936E-4	-0.0012194562196326101	0.0010712834216137373	-4.388152814683785E-4
8	1.0331730812510423E-6	3.055996889025294E-5	3.7862719737577806E-6	-2.3579316627902326E-5
9	7.009923318326339E-6	-2.0658178213792174E-4	3.444940770942545E-4	-1.4001009643516574E-4

Table A.12: Three-body enveloped figure-eight coefficients for Fig. 8.12

(a) Coefficients for body 1

Index	a_x	b_x	a_y	b_y
0	-3.1159325902201866E-4	0.0	-5.443673528215832E-4	0.0
1	0.6477686898149714	0.9375892666714728	-0.3191470606590699	-0.4994971080767887
2	0.036439822687458115	-0.14521611698324563	0.0779937025095896	-0.3255841270774089
3	0.05565415247309742	-0.04571711416003409	-0.03171840855375271	0.015851887003139272
4	-0.010715827294020958	-0.005767089443367684	-0.031007576678544643	-0.012086540176509489
5	0.001525230279319403	-0.004690661620688317	0.002236382244331799	-0.0019261313000684484
6	8.281169692053454E-5	-0.004332892140648124	-0.0011253426209546391	-0.007289056909350979
7	0.001657080976623897	-0.0018571325102033909	-0.0015389608086192217	0.001339305335274105
8	-9.410061496087491E-4	-7.397660389929571E-4	-0.0022393693325797955	-0.0016003841533587557

(b) Coefficients for body 2

Index	a_x	b_x	a_y	b_y
0	0.0011794582131423063	0.0	-1.2108239835465657E-4	0.0
1	-0.4460156494065332	0.18912944533227496	0.26291200302428863	-0.08436462431542031
2	0.11546036023881161	0.12674270917193414	0.2366388969200578	0.2423008373255711
3	-0.04680060057789391	0.10029288197477984	0.019320562641299617	-0.05020183817087203
4	0.019680084302816533	6.515663832347175E-4	0.039649399818747426	2.2145737416079083E-4
5	0.0018144254651665183	0.006664568110750634	-0.0011380450799579494	-0.0019045700799537458
6	-4.59015848893161E-4	0.0014134494400341834	-0.0024337274111319415	0.0013825271411129507
7	-0.0032740987645638337	-0.0015421738763394548	0.0015715693045278843	4.5658702206963735E-4
8	-6.340132412972646E-4	0.0010406327080710325	1.8379673271669077E-4	0.0025585977217024514

(c) Coefficients for body 3

Index	a_x	b_x	a_y	b_y
0	-8.678649541202876E-4	0.0	6.654497511762396E-4	0.0
1	-0.19463578864129	-1.1272582606146764	0.1322939564573432	0.5357418645539433
2	-0.14931391191574706	0.021345871874835602	-0.34313367380574894	0.05479796947284386
3	-0.008686408358809718	-0.059530673686566976	0.009230535140215991	0.045455270552813835
4	-0.007630668015606829	0.005465493442713141	-0.012475557714830679	0.012092327228980007
5	-0.0036970722181058936	-0.0021840885362476876	-2.2443143331555307E-4	0.003859661790706773
6	4.710719839683093E-4	0.0028676745994498072	0.0036722080561865777	0.005307808972838867
7	0.00156038855231301	0.0035220765648408116	1.975165947707879E-4	-0.0017996033651452687
8	0.001595311334538997	-3.494331317891348E-4	0.0020010141104992176	-0.0014071587139672043

Table A.13: Four-body nuclear symbol coefficients for Fig. 8.13

(a) Coefficients for body 1

Index	a_x	b_x	a_y	b_y
1	-0.5072560392632973	-1.1419176012269356	1.1367260791526612	-0.5230744127665182
2	0.0017026374446621646	-7.629290046163467E-4	0.0018503644581783943	3.567830988249491E-4
3	0.0010403168778698384	-8.740332374787062E-4	0.001085036887712877	0.0013456015328645687
4	-1.544976012068956E-5	-6.349329731980418E-5	-4.363070851729055E-4	1.715091020201528E-4
5	8.004292667737137E-4	5.918542000629317E-4	0.0011964685480012453	-0.0011085568536699033
6	1.545431127776888E-4	-1.1066704539734375E-4	-4.686139294899034E-5	-3.0359207156227812E-5
7	-0.005503258536169576	-2.9950683617379497E-5	-3.682054799405173E-5	-0.005511293241008536
8	-1.1793939123678094E-4	6.729515127041693E-5	1.0728563083771065E-5	-1.5898639767501724E-5
9	9.070310639800915E-6	-1.3689303826919773E-4	7.928989197103801E-5	-8.75259368698663E-5

(b) Coefficients for body 2

Index	a_x	b_x	a_y	b_y
1	-0.5061841235108988	-0.04956602422852003	-0.8201946652388723	0.8423345182411991
2	-6.493096966616771E-4	0.002041364838710155	3.467320944143197E-4	-6.561717334275325E-4
3	-0.030186999855725274	0.010651073500584508	-0.006287621520648438	-0.02723244455068437
4	9.473602311653335E-5	5.588160055280728E-5	-1.6791604331728614E-5	-2.419244489013321E-4
5	-0.0013570468828405504	-0.0068628176724384905	0.006244305055920337	-8.16542891512311E-4
6	-1.520635320271948E-4	-7.476722688736255E-5	8.980180851346133E-5	9.988805889217246E-5
7	0.0017128822336416483	1.0917279544139478E-5	1.0101783631467704E-4	0.002041357079041868
8	-3.812533640325953E-5	-1.1066829057757682E-4	-6.967135341710783E-5	-8.413368024554176E-6
9	-7.307426621737814E-5	5.952697195099683E-4	-5.441551912386904E-4	-9.251056297942654E-5

(c) Coefficients for body 3

Index	a_x	b_x	a_y	b_y
1	0.12338889436404332	1.1828679090352618	0.4203212152728146	0.20498214157677902
2	7.685034604680557E-5	-8.437686184554988E-4	-5.88325718275988E-4	-0.0014033017803456885
3	0.01904950874099774	0.017789593163947192	-0.02504872582839151	0.02237498129248939
4	-1.9966819631677595E-4	-1.0766329636716875E-5	2.2612148362459754E-4	1.4956016572790986E-5
5	0.00577051627553504	0.0023964816609092663	-0.0029718799141467273	0.006436637721162242
6	3.7112860254812385E-5	7.152706391909867E-5	5.5255582602836395E-5	1.5221141274419379E-5
7	0.0019630365320714212	2.083459062145638E-4	2.7631066954303064E-5	0.0018017646099293349
8	3.7829380467170654E-5	3.670612199885622E-5	8.24933651919153E-5	4.285318927726107E-5
9	4.8164876535732287E-4	-1.1877504236208507E-4	2.0848597605452372E-4	6.666813846214944E-4

(d) Coefficients for body 4

Index	a_x	b_x	a_y	b_y
1	0.8899693322043268	0.008699625491285024	-0.7435807884956038	-0.5335370855016518
2	-0.0010156743842465266	-4.76616742036706E-4	-0.0014479802492393626	0.0012505133979280076
3	0.01011392285974696	-0.027596246202825574	0.03032121804248069	0.0034232369305808564
4	1.1834933380689016E-4	1.8145562443725688E-5	2.9766269831154327E-4	1.1507530605076966E-5
5	-0.005219985744873353	0.0038787378453081617	-0.00442410504088344	-0.004554344342263729
6	-3.638894315376717E-5	1.0979522268304609E-4	-8.909845439386874E-5	-5.626833793499864E-5
7	0.0018282585730035408	-1.8903649803794965E-4	-7.230138884501577E-5	0.0016990558217785164
8	1.1732716776039562E-4	6.2707299350935444E-6	-1.489883453417643E-5	-1.7576490707521562E-5
9	-4.1838718042661103E-4	-3.3716624320858334E-4	2.6258914856880654E-4	-4.740307036906672E-4

Bibliography

- [1] Sverre J. Aarseth. *Gravitational N-Body Simulations*. Cambridge University Press, Cambridge, UK, 2003.
- [2] Alain Chenciner and Richard Montgomery. A remarkable periodic solution of the three-body problem in the case of equal masses. *The Annals of Mathematics, Second Series*, 152(3):881–901, November 2000.
- [3] Deng Chunhua and Shiqing Zhang. New periodic solutions for $2n$ -body problems in R^3 . *Journal of Mathematical Analysis and Applications*, 330(1):312–321, June 2007.
- [4] G. M. Clemence. Relativity effects in planetary motion. *Proceedings of the American Philosophical Society*, 93(7):pp. 532–534, 1949.
- [5] Albert Einstein. Perihelion motion of Mercury. In *The Collected Paper of Albert Einstein, English Translation*, volume 6, pages 113–116. Princeton University Press, 1997.
- [6] Brian Luzum, Nicole Capitaine, Agnès Fienga, William Folkner, Toshio Fukushima, James Hilton, Catherine Hohenkerk, George Krasinsky, Gérard Pe-

- tit, Elena Pitjeva, Michael Soffel, and Patrick Wallace. The IAU 2009 system of astronomical constants: the report of the IAU working group on numerical standards for fundamental astronomy. *Celestial Mechanics and Dynamical Astronomy*, 110(4):293–304, August 2011.
- [7] Erwan Mazarico, Antonio Genova, Sander Goossens, Frank G. Lemoine, Gregory A. Neumann, Maria T. Zuber, David E. Smith, and Sean C. Solomon. The gravity field, orientation, and ephemeris of Mercury from MESSENGER observations after three years in orbit. *Journal of Geophysical Research: Planets*, 119(12):2417–2436, December 2014.
- [8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York, USA, 3rd edition, 2007.
- [9] Gareth E. Roberts. Linear stability analysis of the figure-eight orbit in the three-body problem. *Ergodic Theory and Dynamical Systems*, 27(6):1947–1963, December 2007.
- [10] Carles Simó. New families of solutions in n -body problems. In Carles Casacuberta, Rosa Maria Miró-Roig, Joan Verdera, and Sebastià Xambó-Descamps, editors, *European Congress of Mathematics (Barcelona, July 10–14, 2000, Volume I)*, volume 201 of *Progress in Mathematics*, pages 101–115. Birkhäuser Basel, 2001.
- [11] Carles Simó. Periodic orbits of the planar n -body problem with equal masses

- and all bodies on the same path. In Bonnie A. Steves and A. J. Maciejweski, editors, *The Restless Universe: Applications of Gravitational N-Body Dynamics to Planetary Stellar and Galactic Systems*. Institute of Physics Publishing, 2001.
- [12] Carles Simó. Dynamical properties of the figure eight solution of the three-body problem. In Alain Chenciner, Richard Cushman, Clark Robinson, and Zhihong Jeff Xia, editors, *Celestial Mechanics: Dedicated to Donald Saari for his 60th Birthday*, volume 292 of *Contemporary Mathematics*, pages 209–228. American Mathematical Society, 2002.
- [13] Robert J. Vanderbei. New orbits for the n -body problem. *Annals of the New York Academy of Sciences*, 1017:422–433, May 2004.
- [14] Milovan Šuvakov and V. Dmitrašinović. Three classes of newtonian three-body planar periodic orbits. *Physical Review Letters*, 110(11):114301, March 2013.
- [15] Shiqing Zhang and Qing Zhou. A noncollision periodic solution for n -body problems. *Journal of Mathematical Analysis and Applications*, 257(2):332–342, May 2001.
- [16] Shiqing Zhang, Qing Zhou, and Yurong Liu. New periodic solutions for 3-body problems. *Celestial Mechanics and Dynamical Astronomy*, 88(4):365–378, April 2004.