eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Bachelor's Degree in Informatics Engineering

Computation

Bachelor's Thesis

Time-Renormalization for the Search of Periodic Solutions to the Three-Body Problem

Author

Markel Zubia Aldaburu

2021

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Bachelor's Degree in Informatics Engineering
Computation

Bachelor's Thesis

Time-Renormalization for the Search of Periodic Solutions to the Three-Body Problem

Author

Markel Zubia Aldaburu

Director
Ander Murua

Abstract

Researchers Antoñana et al. developed a technique for global time-renormalization of the gravitational N-body problem. In their paper, it is speculated that it may be useful for finding periodic orbits, but they do not perform any experiments to test this hypothesis. Influenced by their work, the aim of this project is to find planar three-body choreographies of different topologies.

This project takes a lot of inspiration from Simó's work done on N-body choreographies and the figure eight. In his paper, he proposes efficient methods for finding planar choreographies. The main driver of our work is that some of the problems Simó faced in his work could be lessened by making use of global time-regularization.

Two completely different approaches were taken to tackle the problem. The first approach consists of finding choreographies by solving an optimization problem in the space of solutions to the differential equations of Newton's law of gravitation. The second approach involves generating curves with the desired topology, and then using variational calculus to find solutions that satisfy Newton's laws.

With the first approach, we found thousands of choreographies of many different topologies. We also managed to show that the second approach is viable, although the results were not anywhere close those of the first approach. Experiments showed that global time renormalization reduces the number of Fourier coefficients for curve representation. It was also experimentally verified that the integration of differential equations was much more accurate with time-renormalization when using a constant step-size.

Two conclusions can be drawn from the results of the experiments. For one, both approaches greatly benefit from global time-renormalization. Secondly, our first approach is more effective than the second in finding the most choreographies. However, the ability to control the topology of the solutions is limited with the first approach, and not with the second.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The general N-body problem is a problem where the initial conditions of N point particles are given, and their subsequent motion needs to be determined, where the only forces acting are of mutual attraction. The many variants of the generalized problem have interested researchers in a broad range of disciplines, among others: mathematics, astrophysics, molecular dynamics, and quantum physics.

The most common variant of the problem is the gravitational N-body problem, where the bodies interact exclusively under Newton's law of universal gravitation. This is precisely the problem that has been tackled in this project. For vector positions $q_i = (x_i, y_i, z_i)$, the motion of the bodies can be described with a set of $3N$ second order differential equations:

$$\ddot{q}_i = G \sum_{j \neq i} m_j \frac{(q_j - q_i)}{\|q_j - q_i\|^3}$$

In classical mechanics, the Kepler problem was one of the first problems to be addressed concerning the gravitational N-body problem, which involves the interaction between two particles. The first particle is fixed at the origin, and the second one is moving under gravitational attraction. The method can be used to approximate the interaction between two bodies when one of them has much greater mass than the other. For instance, a real life example of this would be the motion of the Earth around the Sun. The solutions to the Kepler problem are conic sections, such as circles, ellipses, parabolas, and hyperbolas.

In the 18th century, Newton's formulations of his laws of motion and his law of universal gravitation had huge scientific impact. These simple principles allowed him to completely solve the two-body problem. He proved that when $N = 2$ the system can be reduced to the system of ordinary differential equations of the Kepler problem. Yet, Newton was not so successful in finding the solution to the three-body problem. When he was approximating the motions of the Earth, Moon, and Sun, astronomer John Machin made the remark that *"... his head never ached but with his studies on the moon."* [Schaffer, 1981].

In the late 19th century, a substantial prize was established for finding a global solution to the N-body problem (or comparably, to prove that there is no such solution) by Oscar II of Sweden. The goal was to find a solution for systems where $N \geq 3$ in series expansion which is valid for all time. The prize was awarded to Henri Poincaré because of his contributions to classical mechanics, even though he was not able to solve the problem. In 1909, a solution to the original problem was found by Karl Fritiof Sundman for a special case of the three-body problem with non-zero angular momentum [Sundman et al., 1913].

Nowadays, most scientists believe that there is no general closed-form solution to the N-body problem like the one there is for the two-body problem, as it is supported by overwhelming experimental evidence. Despite that, Sundman's work inspired scientists to continue looking for solutions to the problem, and in 1990, Qiu-Dong Wang found a generalization of Sundman's solution for $N \geq 3$. Wang's method defines a time-renormalization function in order to reparameterize the solution with a new independent variable. This reparameterization leads to the existence of a globally convergent power series expansion of the solution [Qiu-Dong, 1990].

Sundman's and Wang's contributions were of great theoretical interest, but their practicality is limited due to the slow convergence of the power series expansions. This is why the use of methods such as numerical integration of ordinary differential equations is needed to approximate solutions. It is challenging to find numerical solutions to the problem in the case that there are close encounters; that is, two of the bodies pass close to each other. To decrease numerical error, either a method for strategically adapting the step-size or some kind of time-renormalization is needed.

In 2020, Antoñana et al. published a paper proposing global time-renormalization functions of the N-body problem [Antoñana et al., 2020]. This paper has practicality in mind, and the renormalization methods proposed are useful for numerically approximating solutions with close encounters. In their paper, it is explained how the technique can be used in place of an adaptive method, and how periodic time-renormalized solutions with close

encounters tend to require far less Fourier coefficients. Because of that, the technique is speculated to be efficient for finding periodic solutions to the N-body problem.

The search of periodic N-body orbits has always been an area of interest. In the year 2000, Chenciner and Montgomery explained the idea of an N-body choreography. In such an orbit, the bodies follow a periodic curve where they all switch places uniformly [Chenciner and Montgomery, 2000]. The first choreography ever found dates back to 1772, when Lagrange described the periodic orbit of three bodies that form an equilateral triangle. In 1993, Cris Moore discovered the figure eight choreography, named after its shape [Moore, 1993].

In 2001, in response to Chenciner and Montgomery, Carles Simó proposed methods for the search of three-body planar choreographies [Simó, 2001]. In his paper, he explains a special symmetry shared among many choreographies and how it can be exploited in order to obtain new choreographies. He also proposes a variational method for approaching the search. Lastly, he explains the need for adaptive methods for the numerical approximation of choreographies with close encounters, as well as the use of many Fourier coefficients.

This project takes great inspiration from Simó's work, with the aim of improving upon it. The main idea of the project is to reimplement the methods proposed by Simó, but by introducing time-renormalization methods proposed by Antoñana et al. Nevertheless, several original ideas have also been proposed and tested in our work, with the ultimate intention of finding new families of choreographies of different topologies.

CHAPTER 2

Preliminaries

2.1 N-body choreography

An N-body choreography is a special kind of T periodic curve that satisfies Newton's law of gravity. This kind of periodic solution has the property that all N bodies share a common orbit and are uniformly spread out around this orbit [Chenciner and Montgomery, 2000]. In other words, all of the bodies switch places at every $1/N$th of the period of the orbit.

More precisely, solution u is said to be an absolute choreography with relative period T/N if

$$u(T/N) = Pu(0)$$

where matrix P is the permutation matrix denoting cycle $(1 \to 2 \to \cdots \to N \to 1)$ of the bodies:

$$P = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

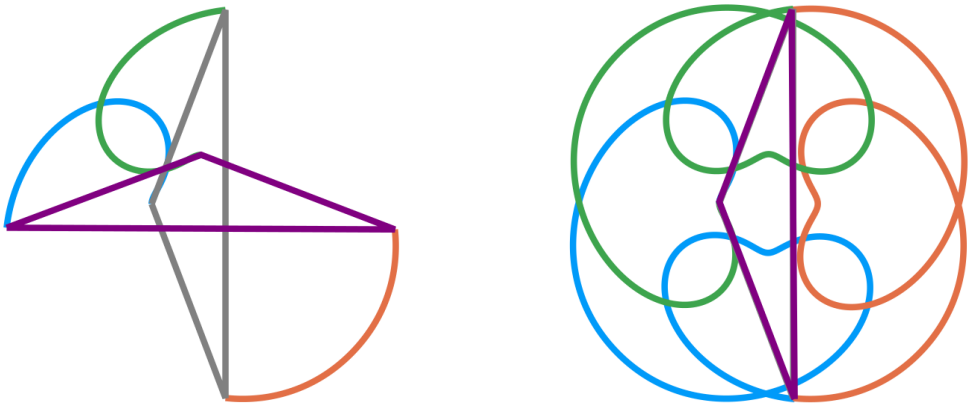


Figure 2.1: Diagram of a choreography with relative angle $\beta = \pi/2$ at $t \in [0, T/12)$ (left) and at $t \in [0, T/3)$ (right).

Another kind of choreography can be defined where permutation can occur at angle β, called relative choreography. A choreography is a relative choreography with relative period T/N if there exists $\beta \in (0, 2\pi]$ such that:

$$u(T/N) = P(\beta)\, u(0)$$

where matrix $P(\beta) = R_\beta P$ denotes the cycling of the bodies composed with the rotation of the plane by angle β.

Absolute choreographies can be considered to be a special case of relative choreographies where $\beta/(2\pi) = p/q$ is an irreducible fraction. The absolute period of the choreography would then be qT/N, where T/N is the relative period (see Figure 2.1). A special kind of absolute choreography occurs when $\beta = 2\pi$, called simple choreography. In case of $\beta \neq 2\pi$, the choreography is said to be a non-simple absolute choreography.

2.2 Topology of three-body choreographies

The phase space of the original planar three-body problem has 6 dimensions for the positions of bodies and 6 for their velocities, totaling 12 dimensions. In Richard Mongomery's paper, a method for reducing the dimensionality of the solutions to the problem is proposed [Montgomery, 2014].

Because of Galilean invariance, congruent triangles with congruent velocities must have congruent motions under Newton's equations of motion. It is impossible to define a sys-

tem of 2nd order ODEs for the three side lengths of the triangles formed by the three bodies. Despite that, the author manages to derive three alternative variables, w_1, w_2, w_3, that can be used similarly to the side lengths.

The shape space of solutions to the ODEs is homeomorphic to $\mathbb{R}^3$ and it is parameterized by the vector (w_1, w_2, w_3), called the shape sphere. The shape sphere has all collinear solutions lying on the disc delimited by the equator. Three radial lines on the collinear disc represent all three two-body collisions, while the three body-collision is in the center of the sphere (see Figure 2.2). Because of these properties, this representation brings new insight into the geometry of choreographies.

Periodic solutions to the three-body problem are closed loops in the shape sphere, and choreographies are periodic solutions with the permutation symmetry. As an example for this, the figure eight choreography can be seen represented in Figure 2.3.

The collinear triangles of a choreography can be used to encode it in a way that its topological properties are represented. More specifically, this encoding is a finite sequence $c = (c_1, \ldots, c_k) \in \{1,2,3\}^k$ representing the collinear triangles in chronological order.

In the case of the figure eight choreography, its code is $(3,1,2,3,1,2)$. Permutation symmetry can be observed in the code if it is split into three chunks: $(3,1) \to (2,3) \to (1,2)$. Therefore, a simpler code can be obtained, namely the relative code, in the case of the figure eight being $(3,1)$.

In order to define code equivalence, the shape space defined by Montgomery needs to be modified. The modified space is simply the shape sphere minus all four points of collision. This way, if two choreographies are homeomorphic, then there is a continuous transformation between them where no collision occurs.

Lastly, it is important to note that because of the periodicity of choreographies, any cyclic permutation of the code will result in an equivalent representation. Not only that but a choreography can be modified so that more collinear triangles occur, introducing repeating patterns into the code. In other words, two choreographies with codes of different lengths can be homeomorphic to each other. That is the case of the figure eight and the celtic knot:

- Code of the figure eight: $(3,1,2,3,1,2)$
- Code of the celtic knot: $(3,1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,1,2,3,1,2)$

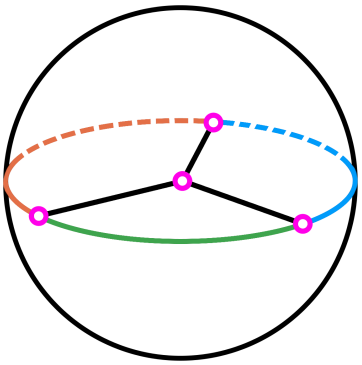


Figure 2.2: Diagram of the shape sphere.

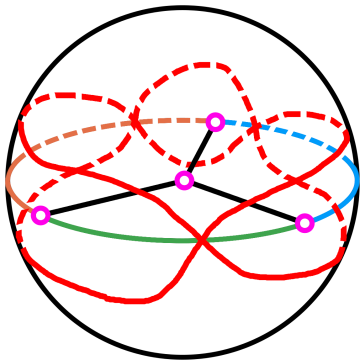


Figure 2.3: The figure eight choreography on the shape sphere.

The similarities between these two curves can very easily be seen by looking at their respective codes. Because of this, we are especially interested in choreographies with the simplest possible code. That is the case of simple choreographies, because of $\beta = 2\pi$ their code is as simple as it can be. In the case of the two choreographies mentioned above, we see that the celtic knot is homeomorphic to the figure eight, the latter being a simple choreography.

2.3 Global time-renormalization

The equations of motion are said to be renormalized if no critical points can be reached starting from a non-degenerate state. With global time-renormalization time is reparameterized by introducing a new independent variable τ in order to globally time-renormalize the equations of motion, by moving the critical points to infinity. The time-renormalization technique proposed by Antoñana et. al is also efficient for numerical integration with fixed steps, as it behaves similarly to adaptive methods [Antoñana et al., 2020]. The basic idea is that fictitious time is dilated in the presence of close encounters, and it is contracted whenever the bodies are far from each other. Intuitively, it makes sense that critical points are moved to infinity, as time would get infinitely dilated.

The motion of the three bodies on the plane in compact form

$$\frac{d}{dt}u = f(u),$$

where

$$u = (x_1, y_1, x_2, y_2, x_3, y_3, \dot{x}_1, \dot{y}_1, \dot{x}_2, \dot{y}_2, \dot{x}_3, \dot{y}_3)$$

is the system's state vector.

In order to apply time-renormalization, if an appropriate function $s(u)$ is chosen, the new independent variable τ (fictitious time) can be defined

$$\tau = \int_0^t \frac{dt'}{s(u(t'))}.$$

State vector u is defined as a function of τ

$$\frac{d}{d\tau}u = s(u)\, f(u)$$

Several time-renormalization functions are defined in their paper, each designed with a specific purpose in mind. In our case, we are working with choreographies, which have three bodies of equal mass. Because of that, the function defined in page 10 of their paper is suitable for the task. It only takes into account the positions of the bodies:

$$\begin{aligned}
\frac{d}{d\tau}q_i &= s(q)\,\dot{q}_i, \\
\frac{d}{d\tau}\dot{q}_i &= s(q)\sum_{j\neq i}\frac{1}{\|q_j - q_i\|^3}\,(q_j - q_i), \\
\frac{d}{d\tau}t &= s(q),
\end{aligned}$$

where

$$s(q_1,\ldots,q_N) = \left(\sum_{1\leq i<j\leq N}\frac{1}{\|q_i - q_j\|^2}\right)^{-1/2}\left(\sum_{1\leq i<j\leq N}\frac{1}{\|q_i - q_j\|}\right)^{-1/2}$$

2.4 Numerical integration of ordinary differential equations

Numerical Integration is the computation of an approximate solution to a definite integral. The requirement of numerical methods for finding solutions to ordinary differential equations becomes apparent in situations where the computation of the antiderivative is either intractable, or an analytical expression does not exist at all.

There are several algorithms that numerical integration covers, such as Monte Carlo meth-

ods and Bayesian methods. We are only interested in numerical methods for approximating solutions to ordinary differential equations from the initial value, to a given degree of accuracy.

Given the system of differential equations:

$$\frac{dx_i}{dt} = f_i(t, x_1, \ldots, x_M; p)$$

It can be rewritten into compact form:

$$\frac{du}{dt} = f(t, u, p), \quad u(t_0) = u_0$$

where,

- t_0 is the initial value of the free variable, usually time.
- u_0 is the initial state of the system.
- p is the parameter vector, with constant values.

The simplest method for numerical integration is known as the Euler method, named after Leonhard Euler. Time is discretized $t_0, \ldots, t_k$, where $t_k = t_{k-1} + h$ for step size $h > 0$. Generally, the lower h is, the more accurate is the solution obtained, but that comes at the cost of increased computational load.

The main difficulty of this process is that $u_k \approx u(t_k)$ has to be computed despite the lack of a precise expression for $u(t)$. To achieve this, for indices $k = 1, \ldots, n$ the following equation is used to sequentially calculate u_k:

$$u_k = u_{k-1} + hf(t_{k-1}, u_{k-1}, p)$$

The insight into the method is that for $h \approx 0$,

$$\frac{u(t+h) - u(t)}{h} \approx \frac{d}{dt} f(t, u(t), p)$$

and consequently, by taking $t = t_{k-1}$, and $h = t_k - t_{k-1}$,

$$u(t_k) \approx u(t_{k-1}) + hf(t_{k-1}, u(t_{k-1}), p)$$

2.5 Newton-Raphson method

The Newton-Raphson method, also known as Newton's method, is an algorithm for quickly finding a root of a real-valued continuous differentiable function starting from an initial guess. It iteratively produces better approximations until the desired precision is reached. It was named after Isaac Newton and Joseph Raphson.

The intuition behind it is that if the initial point is close enough to the root, then the function can be approximated by a straight line. A straight line can be drawn tangent to the initial guess, and the intersection between the x-axis and the tangent line is a better approximation than the initial guess. The process can then be repeated, successively increasing the precision of the result.

More precisely, let $f \in C^1$ be a real-valued function with variable x and f' its derivative. Provided that the initial guess x_0 is close enough to a root of f, then:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Value x_1 is a better approximation of the root than x_0. Then this step can be repeated starting from x_1, iteratively bettering the approximation, until convergence is obtained. Given x_i, the i-th approximation, next term x_{i+1} is obtained as

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{2.1}$$

Besides the geometric explanation of formula (2.1), it can also be explained as the truncation of the function's Taylor series expansion. The Taylor series expansion of f in $(x - x_0)$ is

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

Therefore, by truncating the series at the second term and assuming that x is a root of f, we obtain:

$$
\begin{aligned}
f(x) = f(x_0) + f'(x_0)(x - x_0) &= 0 \\
\frac{f(x_0)}{f'(x_0)} + x - x_0 &= 0 \\
x &= x_0 - \frac{f(x_0)}{f'(x_0)}
\end{aligned}
$$

Newton's method has some limitations. The biggest issue to encounter is that the algorithm can enter a cycle. If there is a point of inflexion in between x_0 and the root, the method may oscillate back and forth, never reaching convergence.

2.6 Trigonometric interpolation via DFT

Trigonometric interpolation is a technique of interpolation by using trigonometric polynomials. The task is to find a function that passes through some given sample points, and this is achieved with a sum of sines and cosines. Because of the use of trigonometric functions, this method is especially useful for periodic functions. In this section only uniformly spaced sample points are considered, as the trigonometric coefficients can be obtained with the discrete Fourier transform.

Given function f, uniformly distributed $2M+1$ sample points $f(t_0), f(t_0+h) \ldots, f(t_0+2Mh)$ are given, for some positive integer $M > 0$. The complex $\hat{f}_k$ trigonometric coefficients for $k = 0, \ldots, M$:

$$
\hat{f}_k = \frac{1}{2M+1} \sum_{j=0}^{2M} \exp\left(-ik\frac{2\pi j}{2M+1}\right) f_j
$$

The trigonometric polynomial that interpolates between the given sample points:

$$
\begin{aligned}
p(t) &= \hat{f}_0 + \sum_{k=1}^{M} \left(\hat{f}_k \exp(ik\omega t) + \overline{\hat{f}_k} \exp(-ik\omega t)\right) \\
&= \hat{f}_0 + 2\sum_{k=1}^{M} \left(\mathrm{Re}(\hat{f}_k)\cos(k\omega t) - \mathrm{Im}(\hat{f}_k)\sin(k\omega t)\right),
\end{aligned}
$$

where $\omega = 2\pi/(h(2M+1))$.

In other words, $p(t_0 + jh) = f(t_0 + jh)$ is satisfied for indices $j = 0, 1, \ldots, 2M$.

In case an even number of sample points is given, the process of obtaining the coefficients and evaluating the trigonometric polynomial is very similar. For $2M$ sample points $f(t_0), f(t_0 + h) \ldots, f(t_0 + (2M-1)h)$

$$\hat{f}_k = \frac{1}{2M} \sum_{j=0}^{2M-1} \exp\left(-ik\frac{\pi j}{M}\right) f_j$$

The trigonometric polynomial:

$$\begin{aligned} p(t) = \hat{f}_0 + 2 \sum_{k=1}^{M-1} \left(\mathrm{Re}(\hat{f}_k)\cos(k\omega t) - \mathrm{Im}(\hat{f}_k)\sin(k\omega t)\right) \\ + \mathrm{Re}(\hat{f}_M)\cos(M\omega t), \end{aligned}$$

It can be proven that the the process of obtaining the derivative of the trigonometric polynomial that interpolates uniformly distributed sample points $f_0, f_1 \ldots, f_{N-1}$ can be interpreted as a linear transformation:

$$\begin{pmatrix} p'(t_a) \\ p'(t_a + h) \\ p'(t_a + 2h) \\ \vdots \\ p'(t_b - h) \\ p'(t_b) \end{pmatrix} = \omega D_N \begin{pmatrix} f(t_a) \\ f(t_a + h) \\ p(t_a + 2h) \\ \vdots \\ f(t_b - h) \\ f(t_b) \end{pmatrix}$$

Matrix D_N is an $N \times N$ matrix where the entries on the diagonal equal zero. For $j \neq k$, entry of the jth row, kth column is:

- If N is odd,
$$\frac{(-1)^{j-k}}{2} \csc\left((j-k)\frac{\pi}{N}\right)$$
- If N is even,
$$\frac{(-1)^{j-k}}{2} \cot\left((j-k)\frac{\pi}{N}\right)$$

2.7 Floating-point arithmetic

Floating-point numbers are the most common representation of real numbers in computing. As opposed to integer arithmetic, floating-point numbers are approximations of real numbers.

For $x \in \mathbb{R} - \{0\}$, its normalized binary representation is the following:

$$x = \pm[a_1.a_2a_3\ldots] \times 2^e$$

where digits $a_i \in \{0,1\}$ and exponent $e \in \mathbb{Z}$. This representation is unique.

Computers have to work with a limited number of bits, most common being 32 and 64. For this reason, there is a trade-off between range and precision. Real numbers are represented very similarly to their normalized form, but with a fixed number of significant digits. The set of machine numbers is the set of numbers represented in the form of $x = \pm[a_1.a_2a_3\ldots a_d] \times 2^e$, for d digits. Consequently, floating-point arithmetic is just an approximation of real number arithmetic and computers have to deal with round-off error.

For the real number x, its floating point representation $fl(x)$ is the machine number closest to it. The absolute representation error of x is the following:

$$|x - fl(x)| \leq 2^{e-d}$$

The relative error for any non-zero real number is:

$$\left|\frac{x - fl(x)}{x}\right| \leq 2^{-d}$$

Machine epsilon is $\varepsilon = 2^{-d}$.

For any x, y real numbers, addition and multiplication are defined such that:

$$x \oplus y = fl(fl(x) + fl(y))$$
$$x \otimes y = fl(fl(x) \times fl(y))$$

Floating point arithmetic operations do not follow the typical properties of precise arith-

metic. They satisfy the commutative property, but not the associative property nor the distributive property. Their inverses, subtraction and division, are defined similarly:

$$x \ominus y = fl(fl(x) - fl(y))$$
$$x \oslash y = fl(fl(x)/fl(y))$$

For function $y = f(x)$, to measure how much output y changes for a small change in x, the condition number κ is defined:

$$\kappa := \lim_{\bar{x} \to x} \frac{|(f(\bar{x}) - f(x))/f(x)|}{|(\bar{x} - x)/x|} = \left| x \frac{f'(x)}{f(x)} \right|$$

Therefore, when $|\bar{x} - x/x| \approx 0$,

$$\left| \frac{f(\bar{x}) - f(x))}{f(x)} \right| \approx \kappa \left| \frac{\bar{x} - x}{x} \right|$$

For functions with two real inputs $z = f(x,y)$, the condition number is similarly defined:

$$\kappa_x := x \frac{\partial f(x,y)/\partial x}{f(x,y)} \kappa_y := y \frac{\partial f(x,y)/\partial y}{f(x,y)}$$

When relative errors are small, $|(\bar{x} - x)/x| \approx 0$ and $|(\bar{y} - y)/y| \approx 0$,

$$f(\bar{x},\bar{y}) - f(x,y) \approx \frac{\partial f(x,y)}{\partial x}(\bar{x} - x) + \frac{\partial f(x,y)}{\partial y}(\bar{y} - y)$$

Thus,

$$\begin{aligned} \left| \frac{f(\bar{x},\bar{y}) - f(x,y)}{f(x,y)} \right| &\approx \left| x \frac{\partial f(x,y)/\partial x}{f(x,y)} \left(\frac{\bar{x} - x}{x} \right) + |y \frac{\partial f(x,y)/\partial y}{f(x,y)} \left(\frac{\bar{y} - y}{y} \right) \right| \\ &= \left| \kappa_x \left(\frac{\bar{x} - x}{x} \right) + \kappa_y \left(\frac{\bar{y} - y}{y} \right) \right| \\ &\leq |\kappa_x| \left| \frac{\bar{x} - x}{x} \right| + |\kappa_y| \left| \frac{\bar{y} - y}{y} \right| \end{aligned}$$

When the condition number for a problem is low it is said that the problem is well-conditioned, whereas a problem with a high condition number is said to be ill-conditioned. The conditioning of elementary arithmetic operations is the following:

- Addition is ill-conditioned when $y \approx -x$, as $\kappa_x = \frac{x}{x+y}, \kappa_y = \frac{y}{x+y}$.
- Subtraction is ill-conditioned when $y \approx x$, as $\kappa_x = \frac{x}{x-y}, \kappa_y = \frac{y}{x-y}$.
- Multiplication is well-conditioned, as $\kappa_x = \kappa_y = 1$.
- Division is well-conditioned, as $\kappa_x = 1, \kappa_y = -1$.

2.8 Formal description of the problem

As explained in the introduction, the goal of this project is to find many three-body choreographies that have different encodings. In this section a more precise definition is given.

In section 2.7 the basics on floating point arithmetic are explained. Since subtraction is ill-conditioned, the original differential equations may not be ideal for numerical methods. In order to avoid an excessive cumulative error, relative coordinates can be used instead:

$$x_{ij} := x_j - x_i$$
$$y_{ij} := y_j - y_i$$

where $(i, j) = (1,2), (2,3), (3,1)$

The differential equations can be rewritten for $q_{ij} = (x_{ij}, y_{ij})$:

$$\begin{aligned} \ddot{q}_{12} &= -2\frac{q_{ij}}{\|q_{ij}\|} + \frac{q_{31}}{\|q_{31}\|} + \frac{q_{23}}{\|q_{23}\|} \\ \ddot{q}_{23} &= -2\frac{q_{23}}{\|q_{23}\|} + \frac{q_{12}}{\|q_{12}\|} + \frac{q_{31}}{\|q_{31}\|} \\ \ddot{q}_{31} &= -2\frac{q_{31}}{\|q_{31}\|} + \frac{q_{23}}{\|q_{23}\|} + \frac{q_{12}}{\|q_{12}\|} \end{aligned} \tag{2.2}$$

The condition for a curve to be a relative choreography remains unchanged:

$$u(T) = P(\beta)\, u(0) \tag{2.3}$$

Therefore, our aim is to find many $u(\tau)$ parametric curves that satisfy both (2.2) and (2.3). Also, it is important that the solutions obtained have different codes.

CHAPTER 3

Approach 1: Optimization problem

3.1 Description of the approach

Our task is to find three-body choreographies with different topological properties. The precise description of this idea can be found in section 2.8.

The problem can be posed as an optimization problem. That is, a space of candidate solutions composed of candidates for choreographies is defined, together with an objective function that measures the proximity to being a choreography of said candidates.

To solve the optimization problem, algorithms for continuous optimization are used. Local minima are expected to be crude approximations of relative choreographies, and they can later be made more accurate.

One special aspect of our method is the continuation algorithm. The search space of the optimization problem is reduced in such a way that it is very easy to find solutions. In exchange, some choreographies are lost, but these missing choreographies may be found later with this technique.

In order to find solutions of different topologies, we have developed multiple objective functions. Each function takes into account a kind of symmetry, and also some consider what we have named the symmetry index of a choreography.

3.2 Encoding of candidate solutions

3.2.1 Encoding of asymmetrical solutions

The initial state of a candidate u_0 encodes the entire curve $u(t)$, as it can be approximated through numerical methods. That is why only the initial state is used to represent a candidate solution. This way, only 12 variables have to be optimized, the positions and velocities of the particles on the plane. Of course, this implies that there is always some amount of numerical error originating from the integration of the differential equations that needs to be taken into consideration.

The naive search space consisting of all possible initial states has many redundancies. For example, the solution to the differential equations encoded by u_0 and the same curve with some translation applied to it will have the same geometry, and therefore it should have the same evaluation of the objective function. Some restrictions are set to the encoding of the candidates in order to get rid of some redundant solutions. These restrictions are based on the ones Simó defines in his paper [Simó, 2001].

Firstly, we will fix the center of mass to the origin, since the geometry of a curve does not depend on the location of said curve. This is achieved by defining the following equation:

$$\|q_1 + q_2 + q_3\|^2 = 0$$

Similarly, we will also set the linear momentum to zero:

$$\|\dot{q}_1 + \dot{q}_2 + \dot{q}_3\|^2 = 0$$

This lets us solve for the initial position and speed of one of the bodies from the values of the other two bodies. Consequently, only the initial conditions of two of the bodies are needed to represent a candidate solution. This way we are reducing the search space, making optimization easier. In this case it is body number one that is inferred from the values of the other two.

For any three-body choreography it is true that at some point in time the three bodies must form an isosceles triangle. This is explained by the symmetrical properties of choreographies. An intuition to understand this is by drawing a line between two of the bodies. In case of a choreography, the third body must cross the line, forming an isosceles triangle.

With this in mind, because of the periodicity of choreographies, we can assume that the initial state of a candidate solution is an isosceles triangle. This triangle can be placed with its symmetry axis laying on top of the x axis because of rotational symmetry. We can also assume that body number 1 is located on the asymmetrical vertex.

Lastly, we can set the total energy of the system to a fixed value. This is because of the scale invariance property of the N-body problem. The only detail to take into account when setting the total energy of the system, is that it must be negative as the three bodies need to be orbiting each other. In this case the total energy of the system will be set to $-1/2$, as it is frequently done in related work.

With all of these restrictions set, it is possible to completely determine an initial state with only the values of x_2, y_2, $\dot{x}_1$, $\dot{x}_2$, and $\dot{y}_2$.

$$\dot{y}_1 + \dot{y}_2 + \dot{y}_3 = 0,$$
$$\frac{1}{2}\sum_{i=1}^{3} \|\dot{q}_i\|^2 - \sum_{(i,j)\in I} \frac{1}{\|q_{ij}\|} = -\frac{1}{2}$$

Combining both equations,

$$\frac{1}{2}(\dot{y}_1^2 + \dot{y}_3^2) = \sum \frac{1}{\|q_{ij}\|} - \frac{1}{2} - \frac{1}{2}(\dot{x}_1^2 + \dot{x}_2^2 + \dot{y}_2^2 + \dot{x}_3^2)$$

$$\dot{y}_1^2 + \dot{y}_1\dot{y}_2 + \frac{1}{2}\dot{y}_2^2 = \sum \frac{1}{\|q_{ij}\|} - \frac{1}{2} - \frac{1}{2}(\dot{x}_1^2 + \dot{x}_2^2 + \dot{y}_2^2 + \dot{x}_3^2)$$

From here, we can solve for $\dot{y}_1$. So done, the initial states of the bodies can be defined in terms of x_2, y_2, $\dot{x}_1$, $\dot{x}_2$, and $\dot{y}_2$. To define the space in a nicer way, the search space is formed by the 5-tuple $(x_2, y_2, \alpha, \gamma_2, \gamma_3)$, where $\alpha \in [0, \pi/2]$, $\gamma_2, \gamma_3 \in [0, 2\pi)$. The initial state determined by these values is the following:

$$\begin{aligned}
&x_{12} = 3x_2, \quad y_{12} = y_2, \quad x_{23} = 0, \quad y_{23} = -2y_2, \quad x_{31} = -3x_2, \quad y_{31} = y_2,\\
&\dot{x}_1 = -\dot{x}_2 - \dot{x}_3, \quad \dot{y}_1 = -\dot{y}_2 - \dot{y}_3,\\
&\dot{x}_2 = \lambda\cos(\alpha)\cos(\gamma_2), \quad \dot{y}_2 = \lambda\cos(\alpha)\sin(\gamma_2),\\
&\dot{x}_3 = \lambda\sin(\alpha)\cos(\gamma_3), \quad \dot{y}_3 = \lambda\sin(\alpha)\sin(\gamma_3),
\end{aligned}$$

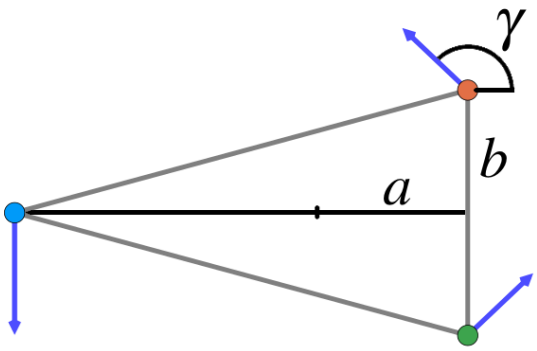


Figure 3.1: Simó's initial isosceles configuration.

where $\lambda > 0$ is determined by the restriction that the energy must be equal to $-1/2$.

The initial state of the system is obtained with function ψ_5:

$$\begin{aligned}\psi_5(a,b,\alpha,\gamma_2,\gamma_3) = (&3a,b,0,-b,-3a,b,\\ &-\lambda\cos(\alpha)\cos(\gamma_2)-\lambda\sin(\alpha)\cos(\gamma_2),-\lambda\cos(\alpha)\sin(\gamma_2)-\lambda\sin(\alpha)\sin(\gamma_2),\\ &\lambda\cos(\alpha)\cos(\gamma_2),\lambda\cos(\alpha)\sin(\gamma_2),\lambda\sin(\alpha)\cos(\gamma_2),\lambda\sin(\alpha)\sin(\gamma_2))\end{aligned}$$

3.2.2 Encoding of symmetrical solutions

A special kind of symmetry shared among several choreographies is described in Simó's paper [Simó, 2001]. The bodies form an isosceles triangle with the symmetry axis on the x axis and the first body on the asymmetrical vertex. The velocity of the first body does not have a horizontal component, and the speeds of the second and third bodies have the same y component but different x. To get a better understanding of the shape, see Figure 3.1.

On top of the restrictions defined in section 3.2.1, we will also assume that the choreographies satisfy Simó's symmetry. This will let us represent a candidate solution with only $x_2, y_2, \dot{x}_2$, and $\dot{y}_2$:

$$x_3 = x_2, \quad y_3 = -y_2, \quad \dot{x}_3 = -\dot{x}_2, \quad \dot{y}_3 = \dot{y}_2.$$

and,

$$x_1 = -2x_2, \quad y_1 = 0, \quad \dot{x}_1 = 0, \quad \dot{y}_1 = -2\dot{y}_2.$$

Scale invariance allows us to fix the total energy of the system to a value of our choice, in

this case, $-\frac{1}{2}$. This way, the total energy will be the following:

$$\rho^2 - \frac{1}{2|y_2|} - \frac{2}{\sqrt{9x_2^2+y_2^2}} = -\frac{1}{2},$$

where $\rho = \sqrt{3\dot{y}_2^2+\dot{x}_2^2}$, thus

$$\dot{x}_2 = \rho\cos(\gamma), \quad \dot{y}_2 = \frac{\rho}{\sqrt{3}}\sin(\gamma)$$

where $\gamma \in [0, 2\pi]$

Therefore, the triplet (x_2, y_2, γ) will determine the initial state of the system, a candidate solution. Variable γ will be in the range $[0, 2\pi]$, and x_2 and y_2 will both be positive real numbers.

We have the additional constraint that:

$$\frac{1}{2|y_2|} + \frac{2}{\sqrt{9x_2^2+y_2^2}} - \frac{1}{2} \geq 0 \tag{2}$$

Finally, function ψ_3 is defined to turn the triplet (x_2, y_2, γ) into an initial configuration u_0:

$$\psi_3(a,b,\gamma) = \Big(-2a, \quad 0, \quad a, \quad b, \quad a, \quad -b, \\ 0, \frac{\rho}{-2\sqrt{3}}\sin(\gamma), \rho\cos(\gamma), \frac{\rho}{\sqrt{3}}\sin(\gamma), -\rho\cos(\gamma), \frac{\rho}{\sqrt{3}}\sin(\gamma)\Big)$$

3.3 Objective functions

3.3.1 General definition of the objective function

Several objective functions were used for the search of three-body choreographies. Each of these functions has been designed to target a specific kind of symmetry. Even if they each serve a different purpose, all of them have the same underlying structure.

In general, the objective functions will consist of what we call a trigger function $g(t)$ and a callback function $h(u_t)$. The idea behind the trigger function is that it is used to detect the

values of τ for which a specific configuration of the bodies happens, for example, forming an isosceles triangle. Then, the callback function is called at those specific timestamps to determine the quality of the solution.

The objective functions are all of the following form:

$$\mathcal{F}(q) = \min_{\tau \in G_q} h(q(\tau))$$

where

$$G_q = \{\tau : g(q(\tau)) = 0\}$$

To compute the objective function, a discretization of the process is necessary. The step-by-step algorithm is the following:

1. Firstly, using either ψ_3 or ψ_5, initial state u_0 is obtained from the encoded candidate solution from the optimization problem.
2. Once the initial state of the system is known, states $u_1, \ldots, u_n$ are calculated via numerical integration. A Runge-Kutta method of order 8 was used for the integration.
3. Then, g is evaluated in each discrete step u_i and an approximation of the roots is obtained with linear interpolation.
4. Lastly, h is evaluated in each of these steps and the minimum is returned.

3.3.2 Symmetry-based objective functions

Asymmetrical objective function

The asymmetrical objective function is designed to be used with solutions with asymmetrical encodings, defined in section 3.2.1. The choreographies targeted are not assumed to have any kind of special symmetry at all, besides the permutation symmetry explained in section 2.1.

This function needs to return a heuristic on how close to being a choreography the candidate solution is. To achieve that, it has to find τ such that $\|Pq(0) - q(\tau)\|^2$ is minimized. This minimization can be done analytically. To maximize the chance of finding a choreography, we will also minimize for $\|P^{-1}q(0) - q(\tau)\|^2$, that is, we will allow the permutation to happen in either direction.

The basic idea behind the function is the following: because we know that the initial position of the bodies is an isosceles triangle, g is defined such that its roots are on isosceles triangles. Then, h simply compares the angles of these triangles with those of q_0.

First off, we need to detect an isosceles configuration with g. But, it is only necessary to check whether there is an isosceles triangle where body number one is on either symmetrical vertex. This is because initially this body is on the asymmetrical vertex, so either body two or body three has taken its place after the permutation. This is done by comparing dot products:

$$g_{\text{iso}}(u(\tau)) = (q_{23} \cdot q_{31} - q_{31} \cdot q_{12})(q_{32} \cdot q_{12} - q_{12} \cdot q_{23})$$

Then, with function h the cosines of the angles of triangles q_0 and q_τ are compared:

$$\begin{aligned} h(q_\tau) =& (c(q_{31}(0), q_{12}(0)) - c(q_{23}(\tau), q_{31}(\tau)))^2 + \\ & (c(q_{12}(0), q_{23}(0)) - c(q_{31}(\tau), q_{12}(\tau)))^2 + \\ & (c(q_{23}(0), q_{31}(0)) - c(q_{12}(\tau), q_{23}(\tau)))^2 + \\ & \|\dot{q}_1 - \dot{q}_3\|^2 + \|\dot{q}_2 - \dot{q}_1\|^2 + \|\dot{q}_3 - \dot{q}_2\|^2 \end{aligned}$$

where c is proportional to the cosine of the angle between two vectors $c(u, v) = u \times v$.

It is trivial that if $\mathcal{F}(q) = 0$ then $q(t)$ is a relative choreography.

Isosceles objective function

The isosceles objective function is used with solutions with the isosceles symmetry defined by Simó, explained in section 3.2.2. These choreographies form a very special configuration that can be exploited with a specialized objective function.

Simó explains in his paper how if a choreography satisfies the initial conditions defined in section 3.2.2, then this choreography has an extra symmetry. At $\tau = T/6$ these choreographies form an isosceles triangle, not necessarily the same as the one at $u(0)$, where the moduli of the velocities on the asymmetrical vertices are equal to each other.

Because of a property of the N-body problem called reversibility, it is ensured that the bodies will trace a symmetrical path from then on, and that they will form an isosceles

triangle at $T/3$ with the exact dimensions of the initial one. In other words, if the bodies form this special triangle at $T/6$, it can be proven that it is a choreography.

Trigger function g is the same as in section 3.3.2, because we also need to evaluate h on isosceles configurations only.

Callback function h is used for calculating whether the bodies form this special intermediate isosceles triangle. First, two helper functions are defined. Function h_2 is used in case body number two is on the asymmetrical axis. Similarly, h_3 is used for body three:

$$h_{\text{iso2}}(u(\tau)) = ((x_{23} - x_{12})\dot{x}_{23} + (y_{23} - y_{12})\dot{y}_{23})^2 + (\|\dot{q}_{12}\|^2 - \|\dot{q}_{31}\|^2)^2$$

$$h_{\text{iso3}}(u(\tau)) = ((x_{31} - x_{23})\dot{x}_{31} + (y_{31} - y_{23})\dot{y}_{31})^2 + (\|\dot{q}_{12}\|^2 - \|\dot{q}_{23}\|^2)^2$$

Again, we do not need to worry about body number one being on the asymmetrical axis.

The callback function calculates which body is on the asymmetrical axis and the corresponding function is used based on that:

$$h_{\text{iso}}(u(\tau)) = \begin{cases} h_{\text{iso2}}(u(\tau)) & \text{if} \quad \nu(u(\tau)) \geq 0 \\ h_{\text{iso3}}(u(\tau)) & \text{else} \end{cases}$$

where

$$\nu(u(\tau)) = 2(q_{31} \cdot q_{12}) - q_{12} \cdot q_{23} - q_{23} \cdot q_{31}$$

Collinear objective function

The collinear objective function detects a special collinear configuration of the bodies. This configuration was first described by Simó in 2002 when analyzing the properties of the figure eight [Simó, 2002]. All three bodies align such that one of the bodies is at the origin. The moduli of the velocities of the two bodies at each end of the line segment are equal.

Collinear configurations can be detected as follows

$$g_{\text{col}}(u(\tau)) = x_{12}y_{23} - y_{12}x_{23}$$

The callback function needs to detect that one body is perfectly centered and that the

moduli of the velocities of the two edgemost bodies are equal. Similarly to the isosceles function, we will define one helper function for each of the possible bodies that can be at the origin:

$$h_{\text{col2}}(u(\tau)) = \|q_{23} - q_{12}\|^2 + \|\dot{q}_3 - \dot{q}_1\|^2$$

$$h_{\text{col3}}(u(\tau)) = \|q_{31} - q_{23}\|^2 + \|\dot{q}_1 - \dot{q}_2\|^2$$

The callback function calculates the centermost body in the collinear formation and the corresponding function is used based on that:

$$h_{\text{iso}}(u(\tau)) = \begin{cases} h_{\text{col2}}(u(\tau)) & \text{if} \quad |x_{31}(\tau)| - |x_{12}(\tau)| \geq 0 \\ h_{\text{col3}}(u(\tau)) & \text{else} \end{cases}$$

3.3.3 Indexed objective functions

Several objective functions are defined in section 3.3.2 with the aim of increasing the diversity of the choreographies to be found. In this section we introduce the concept of symmetry index, with the idea of further increasing the diversity of the solutions.

The symmetry index of a choreography is the number of times an isosceles triangle is formed within the timespan of a relative period. More precisely, the index of a choreography is

$$\iota(q(t)) = |\{\tau \in [0, T/3) : g(q(\tau)) = 0\}|$$

Different choreographies have different indices, so the usefulness for increasing diversity is evident. Two choreographies with the same absolute code may have different numbers, as they may have different relative codes. Two simple choreographies with the same absolute and relative code must have the same index.

All choreographies with the symmetry defined in section 3.3.2 must have an even symmetry number. With the same reasoning, all choreographies with colinear symmetry defined in 3.3.2 must have a symmetry number that is a multiple of 4.

A new kind of generic objective function can be defined so that the index of the chore-

ographies is taken into account. All of these objective functions will have the following structure:

$$\mathcal{F}_i(q) = h(q(\tau_i))$$

where τ_i is the element number i of G_q.

With this method, we can take the functions defined in section 3.3.2 and define new ones by fixing the index. It was experimentally verified that indices 1 through 4 are the most common, so those are the ones chosen. Together with the original objective functions, 15 functions were defined:

- General asymmetrical function. $\mathcal{F}^{\text{asy}}$
- Asymmetrical functions with indices 1 through 4. $\mathcal{F}_i^{\text{asy}}$
- General isosceles function. $\mathcal{F}^{\text{iso}}$
- Isosceles functions with indices 1 through 4. $\mathcal{F}_i^{\text{iso}}$
- General collinear function. $\mathcal{F}^{\text{col}}$
- Colinear functions with indices 1 through 4. $\mathcal{F}_i^{\text{col}}$

These functions will be used independently to each other in the local search, as described in section 3.4.

3.3.4 Properties of the objective functions

The definition of the general structure of the objective functions as well as that of several objective functions is given in section 3.3. Despite the different definitions of these functions, they all share many properties.

All objective functions are computed in linear time, $O(n)$:

1. The computation of ψ_3 or ψ_5 is done in $O(1)$ time.
2. The numerical integration step to obtain $u_1, \ldots, u_n$ is done in linear time.
3. The obtaining of the roots of g is done in linear time.

4. The evaluation of h is technically done in linear time at worst, although for continuous curves (such as solutions to Newton's law of gravitation) this is negligible.

Another property of all objective functions is that they are well-behaved in several bounded regions of their domain, because they are piecewise compositions of elementary functions with infinitely differentiable curves as inputs.

Also, since all objective functions are bounded in $[0,\infty)$, a minimization algorithm would be expected to converge for relatively many different candidate solutions, because of these well-behaved regions.

It was experimentally verified that attractors are a problem with this kind of objective function. Attractors are solutions to which many initial points converge, which would prevent our algorithm from finding solutions other than these. This is detrimental to our method because our aim is to find choreographies of different kinds. Of course, the goal of using many different objective functions is to lessen this kind of problem.

3.4 Local search

For finding local minima of the functions defined in section 3.3, local search was used. Techniques for unconstrained optimization were used for the search, and therefore some adjustments needed to be made in order for the constraints to be satisfied. Most of the constraints were about the scope of the variables, but these are loose as they are used in order to get rid of redundancy.

The only constraint that is problematic is related to the energy of the system. We know that the potential energy of the system in terms of x_2 and y_2 is

$$T(x_2,y_2) = \frac{1}{2|y_2|} + \frac{2}{\sqrt{9x_2^2+y_2^2}}$$

Since the total energy of the system equals $1/2$, in order for the kinetic energy to be positive, we have following constraint:

$$\frac{1}{2|y_2|} + \frac{2}{\sqrt{9x_2^2+y_2^2}} - \frac{1}{2} \geq 0$$

The objective function needs to be adjusted so that the constraint is met:

$$\tilde{\mathcal{F}}(q) = \begin{cases} \mathcal{F}(q) & \text{if } \rho^2 > 0 \\ \infty & \text{else} \end{cases}$$

Using this newly defined function, local search can be applied. First, the initial point (α, β, γ) or $(x_2, y_2, \alpha, \gamma_2, \gamma_3)$ is randomly chosen from a uniform distribution, such that all constraints are satisfied. Then, the Nelder-Mead algorithm is used in order to find a local minimum of $\tilde{\mathcal{F}}$. As mentioned earlier, this method is unconstrained. Lastly, if the algorithm converges and $\tilde{\mathcal{F}}(q) = 0$, then q is a choreography.

3.5 Curve continuation

3.5.1 Insight

Let C^1 be the space of all real valued continuous functions that are continuously differentiable on $\mathbb{R}$. Then, the set of all relative choreographies where no collision between any two bodies happens forms multiple continuous curves in C^1, as was shown by Simó in his paper. Every choreography is part of a curve, and throughout the entirety of any curve the code remains unchanged.

This property can be exploited to generate even more choreographies starting with the ones found with the local search. The idea behind it is that because every choreography is part of a curve, slight perturbations may be applied in a precise direction in order to get a new one. This can be then repeated iteratively to explore a curve in its entirety. A curve ends at points of collision between the bodies, known as critical points. It is possible to skip over these critical points, but that would change the code of the choreographies.

3.5.2 Formal definition of the curves

Solution $q(\tau)$ is a choreography if there exists $T > 0$ and $\beta \in [-\pi, \pi]$ where for indices $i = 1, \ldots, 3$

$$\begin{aligned}
x_i(T) &= \cos(\beta)\, x_{i+1}(0) - \sin(\beta)\, y_{i+1}(0), \\
y_i(T) &= \sin(\beta)\, x_{i+1}(0) + \cos(\beta)\, y_{i+1}(0), \\
\dot{x}_i(T) &= \cos(\beta)\, \dot{x}_{i+1}(0) - \sin(\beta)\, \dot{y}_{i+1}(0), \\
\dot{y}_i(T) &= \sin(\beta)\, \dot{x}_{i+1}(0) + \cos(\beta)\, \dot{y}_{i+1}(0).
\end{aligned}$$

Here, indices are cyclic. For example where it says $i = 3 + 1$ it should be interpreted as $i = 1$.

When performing the local search for relative choreographies, we assumed that the initial values $q(0)$ satisfied some symmetry conditions when using the symmetrical encoding defined at section 3.2.2. This way we obtained the approximate initial values $\tilde{q}$ of some choreographies.

We will increase the precision of these approximations by making use of Newton's method. No symmetry condition will be assumed, as this will allow for a processing of both symmetric and asymmetric relative choreographies.

Considering that if $q(\tau)$ is a choreography then $q(T_0 + \tau)$ is also a choreography, for every T_0, and also that by rotating q we also get a choreography, in general we can assume that:

$$x_3(0) = x_2(0), \quad y_3(0) = -y_2(0),$$

thus,

$$x_1(0) = -2x_2(0), \quad y_1(0) = 0, \quad \dot{x}_1(0) = -\dot{x}_2(0) - \dot{x}_3(0), \quad \dot{y}_1(0) = -\dot{y}_2(0) - \dot{y}_3(0).$$

Nine free variables are used to determine a relative choreography, which will be arranged in vector $z = (z_1, z_2, \ldots, z_9)$ where

$$\begin{aligned}
&x_2(0) = z_1, \quad y_2(0) = z_2, \quad \dot{x}_2(0) = z_3, \quad \dot{y}_2(0) = z_4, \quad \dot{x}_3(0) = z_5, \\
&\dot{y}_3(0) = z_6, \quad T = z_7, \quad \beta = z_8, \quad P = z_9,
\end{aligned}$$

where $P = t(T)$ is the physical relative period.

The initial conditions are defined:

$$u(0) = \chi(z)$$

where

$$\chi(z) = (3z_1, z_2, 0, -2z_2, -3z_1, z_2, -z_3 - z_5, -z_4 - z_6, z_3, z_4, z_5, z_6, 0)$$

In order for solution $u(\tau)$ to be a relative choreography with rotation angle β it is enough that it satisfies these equations:

$$\begin{aligned}
x_{12}(T) &= \cos(\beta)\, x_{23} - \sin(\beta)\, y_{23}, \\
y_{12}(T) &= \sin(\beta)\, x_{23} + \cos(\beta)\, y_{23}, \\
x_{23}(T) &= \cos(\beta)\, x_{31} - \sin(\beta)\, y_{31}, \\
y_{23}(T) &= \sin(\beta)\, x_{31} + \cos(\beta)\, y_{31}, \\
\dot{x}_1(T) &= \cos(\beta)\, \dot{x}_2 - \sin(\beta)\, \dot{y}_2, \\
\dot{y}_1(T) &= \sin(\beta)\, \dot{x}_2 + \cos(\beta)\, \dot{y}_2.
\end{aligned}$$

The remaining conditions to be a choreography are obtained straight away with the four conditions involving the center of mass and because of the conservation of energy and angular momentum.

To the six conditions listed we will add that the total energy is equal to $-1/2$, and lastly that $t(T) = z_9$.

These eight conditions can be written in compact form as $g(u(T), u(0), \beta, P) = 0$ where

$$g(u^*, u, \beta, P) = \begin{pmatrix}
x_{12}^* - \cos(\beta)\, x_{23} + \sin(\beta)\, y_{23}, \\
y_{12}^* - \sin(\beta)\, x_{23} - \cos(\beta)\, y_{23}, \\
x_{23}^* - \cos(\beta)\, x_{31} + \sin(\beta)\, y_{31}, \\
y_{23}^* - \sin(\beta)\, x_{31} - \cos(\beta)\, y_{31}, \\
\dot{x}_1^* - \cos(\beta)\, \dot{x}_2 + \sin(\beta)\, \dot{y}_2 \\
\dot{y}_1^* - \sin(\beta)\, \dot{x}_2 - \cos(\beta)\, \dot{y}_2 \\
H(u) + \frac{1}{2} \\
u_{13}^* - P
\end{pmatrix}$$

Let's assume that $u(T)$ is obtained with some integration method with n steps and constant step-size. Then, function ϕ_n is defined such that this approximation is a function of z where

$u(z) \approx \phi_n(z)$.

Lastly, in order for z to represent a relative choreography, the following system of equations needs to be satisfied:

$$g(\phi_n(z), \chi(z), z_8, z_9) = 0 \tag{3.1}$$

There are nine free variables and eight equations. In fact, system (3.1) is an implicit definition of the curves of relative choreographies.

3.5.3 Curve continuation method

A naive approach to the continuation of the curves would be to gradually increase and decrease $z_8 = \beta$ and to solve for the rest of the variables. This way, the algorithm would be processing choreographies of changing relative angle, which would guarantee finding new choreographies. The problem with this technique is that if the curve bends towards the β axis, the algorithm will not be able to continue the curve.

To avoid that issue, the intersection between the curve and a carefully chosen hyperplane are considered for the continuation of the curve.

Specifically, to determine a solution to system (3.1) that is close to $z^0 \in \mathbb{R}^9$, the intersection between that system and hyperplane $\alpha \cdot (z - z^0) = 0$ is calculated, where an appropriate value for $\alpha \in \mathrm{R}^9$ is chosen. In other words, we will calculate a solution $f(z) = 0$ that is close to z^0 (for example, with Newton's method), where:

$$f(z) := \begin{pmatrix} g(\phi_n(z), \chi(z), z_8) \\ \alpha \cdot (z - z^0) \end{pmatrix} \tag{3.2}$$

Let $z^1 \in \mathbb{R}^9$ represent a relative choreography, where $\beta^1 = z_8^1$. To begin the continuation of the curve, let $\beta^2 = \beta^1 + \Delta\beta$, for a small $\Delta\beta$. Using a non-linear solver z^2 can be obtained where $z_8^2 = \beta^2$ and $f(z^2) = 0$, $\alpha = (0, \ldots, 0, 1)$.

The criteria for choosing $\Delta\beta$ is that Newton's method takes the desired number of iterations to get from z^1 to z^2. The number of iterations is chosen based on experiments performed with different choreographies.

Afterwards, from points z^1 and z^2 of the curve of relative choreographies, a third point z^3 is obtained. To explain the general procedure, let's assume that points z^{i-2} and z^{i-1} are given, and z^i needs to be obtained. First off, let unit vector α equal

$$\alpha = \frac{1}{\|z^{i-1} - z^{i-2}\|}(z^{i-1} - z^{i-2})$$

An initial approximation $\tilde{z}^i$ of point z^i is obtained by linear interpolation:

$$\tilde{z}^i = z^{i-1} + \alpha\,\lambda^i,$$

where lambda is a small number.

Then, point $\tilde{z}^i$ will be the initial point z^0 of the equation (3.2). Then Newton's method is applied to get the desired solution z^i. Again, λ^i needs to be determined. Our criteria will be the same as before, to choose a value so that the number of iterations of Newton's method is in the range $[it_0, it_1]$.

This same method can also be used to increase the precision of the initial values obtained. In case of a simple choreography, simply fixing $\beta^1 = 0$ and applying Newton's method to solve the system (3.2) will give us the z vector of this same choreography but to a higher precision. Then, the initial state of the system is just $\chi(z)$. In case of a relative choreography, it is desirable that β^1 is set to a simple fraction of 2π close to β^0.

3.5.4 Continuation algorithm

Given an approximation of the initial state of a relative choreography found with the local search, this state is not very accurate. Therefore, the first step is to apply the Newton Raphson method so that system of equations (3.1) is satisfied, in order to increase the numerical precision of the initial state. In this step, $\beta = z_8$ is fixed since the curve does not need to be continued yet. In case of non-convergence, the curve is discarded.

For the continuation of the curve, it is necessary to represent it with an appropriate data structure. Farey secuence F_l is used to generate a list of fractions of 2π. When performing the continuation of the curve, whenever β passes over one of said fractions, the precise choreography concerning the angle is interpolated linearly and the initial state z stored in a dynamic list containing all relative choreographies with said angle. Thus, when the continuation algorithm finishes, the curve is represented by a set of lists, each containing relative choreographies with a set angle.

Because of the high computational cost of doing the continuation of a curve, it is in our interest to perform checks in order to avoid repeating the continuation of the same curve

multiple times. Every time a fraction of 2π in F_l is reached, the relative choreography is compared with other relative choreographies with the same angle in other curves. In case the choreography was already found in some other curve, the continuation of the curve is halted as there is no need to keep going.

Finally, the representation of the curves enables the easy obtention of simple choreographies. For each curve, simple choreographies are in the list containing choreographies with $\beta = 0$. Relative choreographies with simple fractions of 2π can also be obtained, which usually have interesting shapes.

3.6 Experiments

With all the techniques defined in this chapter, three tasks were carried out:

1. The first task was to perform a local search for three-body choreographies using the optimization method described in section 3.4. Since fifteen different objective functions were defined, fifteen independent local searches needed to be performed.

   The device chosen for the task has 16 cores. This is very convenient for the search as it enables the execution of all fifteen searches in parallel, each on its own thread, plus one additional thread for the operating system and miscellaneous processes.

   The search was carried out throughout 22 days. In order to avoid loss of data, all solutions were stored in the hard drive of the device at the time of their obtention, and backups were made periodically.

2. The second task was to employ the continuation algorithm to generate new choreographies starting from the ones found with the local search. These initial solutions were first filtered in order to get rid of repeated curves and any non-choreographies.

   The initial idea was to run the continuation algorithm defined in section 3.5.4 with all the choreographies obtained. However, because of lack of time, the algorithm was only applied to choreographies with the most interesting properties.

3. Lastly, an experiment to evaluate the effectiveness of global time-renormalization with uniform step-size was performed. Some of the choreographies obtained via the local search and the continuation algorithm were used for this. They were chosen based on their relative angle and how extreme their close encouters are.

The initial states of said choreographies were taken, and then numerical approximations of the solutions were made both with and without time-renormalization. A highly accurate approximation was also obtained by making use of high-precision floating point numbers and a very small step-size. To make an estimation of the error we simply took the solutions obtained with and without renormalization and compared them with the highly accurate solutions.

Of course, similar tests were done at the beginning of the project, as a sanity check. However, these tests were done with simpler orbits, as their purpose was to just verify that our implementation was correct.

3.7 Results

- In total 5392 relative choreographies were found with the local search, 166 of which being simple. The amount of relative choreographies obtained with each objective function is shown in Table 3.1 and the amounts of simple choreographies are shown in Table 3.2.

  Examples of the kind of choreographies found with the collinear function can be found in appendix A, together with their initial state vectors.

- The continuation algorithm was applied to the figure eight, heart, and ladybug. The continuation of the figure eight can be seen in Figure 3.3 and the heart in Figure 3.4. The continuation of the ladybug was also performed and another choreography with the same shape was found (see Figure 3.5).

- Nine choreographies were chosen for the experiment based on their relative angle and the closeness of their closest encounter, shown in Figure 3.2. The initial configurations of these choreographies are shown in Table 3.5, and their codes in Table 3.3.

  The average and maximum errors with and without time-renormalization of the choreographies selected are shown in Table 3.4, together with the closeness of the closest encounter.

3.8 Conclusions

The search with the isosceles objective function found the greates number of choreographies, as shown in Table 3.1. These numbers may mislead one into believing that it is the best method among those three. However, by taking a look at table 3.2, we can see that the collinear function is able to find many simple choreographies.

The solutions found with the asymmetrical function were very innaccurate and with many non-choreographies. The asymmetrical function was designed to find choreographies that do not follow Simó's symmetry, but all choreographies found with the function either have said symmetry or they belong to the family of a choreography with said curve. The isosceles function was successful at finding a great number of relative choreographies, but there was not a lot of diversity among them. This is because two very similar relative choreographies with slightly different relative angles are considered different. So the number of distinct simple choreographies is not too high compared to the sheer amount of relative choreographies found. The collinear function was the opposite, as it found very few solutions, but the vast majority of them had interesting properties (see A.1). They also were very precise in comparison with the solutions found with the other functions.

Lastly, on Table 3.1 no relative choreographies were found with the collinear function with index 2. This is most likely due to human error and has nothing to do with the objective function itself.

The relative choreographies obtained with the functions with different indices were similar to the ones obtained with the general function. Also, the same choreography could appear with two different indices. This happened because if a choreography is mirrorred it can result in the same shape but different index.

The most interesting choreography found was the ladybug. We believe that this choreography was not previously found. It is a simple choreography and it is homeomorphic to the figure eight, but it does not belong to the same family of relative choreographies; it lays on a completely different curve.

The continuation of the figure eight can be seen in Figure 3.3. The images are very instructive for understanding the connection between the figure eight and the celtic knot. However, the choreographies found were not of our interest because Simó already did the continuation of the figure eight. The same thing can be said about the continuation of heart, shown in Figure 3.4.

Function	General	Index = 1	Index = 2	Index = 3	Index = 4
$\mathcal{F}^{\text{asy}}$	727	13	484	159	181
$\mathcal{F}^{\text{iso}}$	1582	677	361	558	495
$\mathcal{F}^{\text{col}}$	71	30	0	32	22

Table 3.1: Number of distinct relative choreographies obtained with each objective function.

Function	General	Index = 1	Index = 2	Index = 3	Index = 4
$\mathcal{F}^{\text{asy}}$	14	1	5	5	4
$\mathcal{F}^{\text{iso}}$	38	7	15	3	11
$\mathcal{F}^{\text{col}}$	21	1	0	25	16

Table 3.2: Number of distinct simple choreographies obtained with each objective function.

The continuation of the ladybug on the other hand was very interesting, as this choreography was not found by others before. What was even more interesting is that the curve joined two choreographies with very similar geometry (see Figure 3.5). It turns out that for any choreography, there can be multiple versions of it with the same geometry. Another example of this phenomenon can be seen in Figure A.1 with the two versions of the spider.

Finally, nine choreographies were selected and used for the experiment testing the numerical errors (see Figure 3.2). The figure eight was selected for the experiment, together with the celtic knot and the star, which belong to the same family. The family sharing heart and four-leaf clover were also chosen. In the family of the ladybug we have the ladybug, ferret, and the cross. Finally, the spider does not belong to any of the other's families, but it is homeomorphic to the heart.

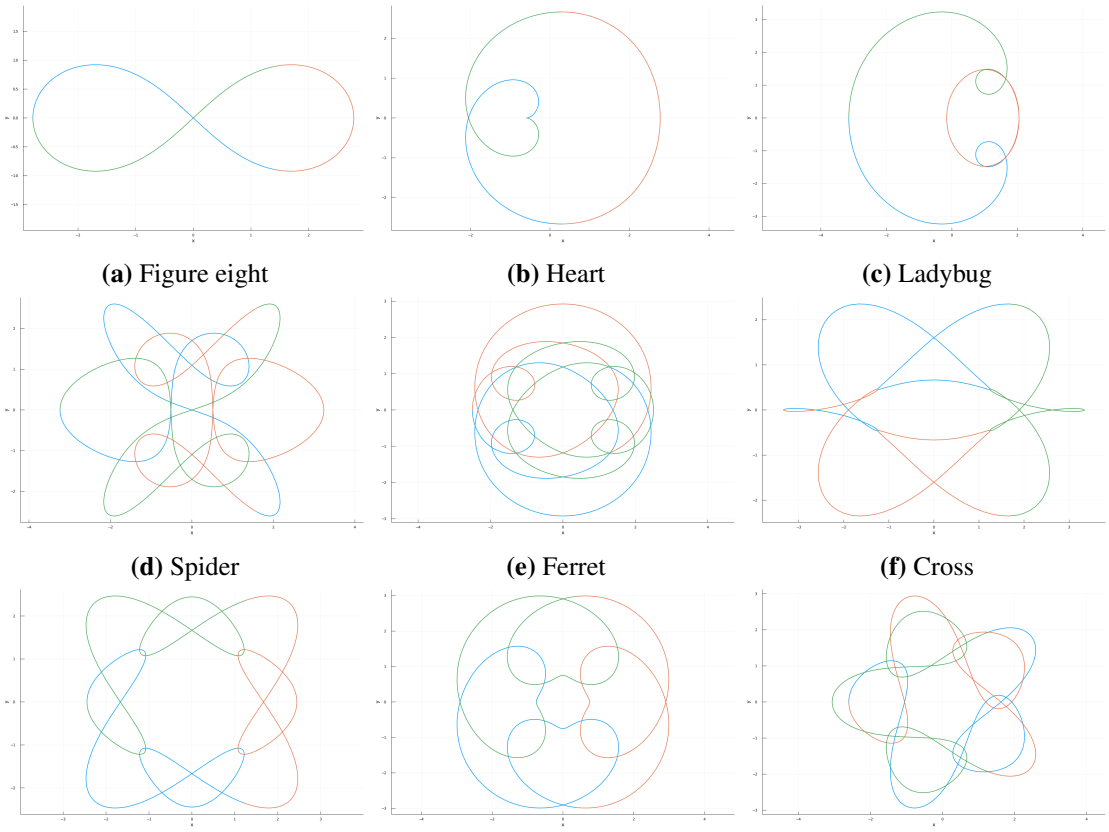
Table 3.4 shows that global time-renormalization is clearly necessary if a constant step-size is used when approximating choreographies with close encounters. The error increases the closer the bodies meet. An adaptive method could also be used with a similar effect.

Name	Code	β	Simple
Figure eight	312312	π	No
Heart	133221	0	Yes
Ladybug	312312	0	Yes
Spider	213213213213213213213213213	π	No
Ferret	133221133221	π	No
Cross	12233112233112233112233	π	No
Celtic knot	312312312312312312312	$\pi/2$	No
Four-leaf clover	133221133221133221133221	$\pi/2$	No
Star	312312312312312312312312312	$2\pi/5$	No

Table 3.3: Geometric properties of the choreographies selected.

Name	Closest encounter	Mean No TR	Maximum No TR	Mean GTR	Maximum GTR
Cross	4.3226e-3	3975.8	8543.6	3.8951e-11	6.2544e-10
Celtic knot	1.0069e-1	288.64	658.51	2.9598e-14	1.4322e-13
Star	3.7212e-1	1.1884e-8	1.1794e-7	2.1502e-14	4.3520e-14
Spider	5.5382e-1	5.1530e-11	5.1350e-10	5.7244e-13	5.4267e-12
Ladybug	7.2609e-1	2.2214e-14	4.7832e-1	6.4651e-15	1.8324e-14
Ferret	7.3118e-1	7.6542e-9	1.2075e-7	6.7977e-9	1.2629e-7
Four-leaf clover	8.6789e-1	1.8527e-8	3.4578e-7	3.2259e-9	5.3489e-8
Heart	1.6785	5.4987e-14	3.0065e-13	1.0636e-14	4.6109e-14
Figure eight	1.7776	2.0173e-15	5.0202e-15	2.8045e-15	5.9702e-15

Table 3.4: Estimation of the approximation error made in the integration of each curve with and without global time-renormalization.



(a) Figure eight **(b)** Heart **(c)** Ladybug

(d) Spider **(e)** Ferret **(f)** Cross

(g) Celtic knot **(h)** Four-leaf clover **(i)** Star

Figure 3.2: The nine selected figures for the experiments.

Name	x_2	y_2	γ
Figure eight	1.3914	-0.8888	-0.3533
Heart	0.2973	-2.6624	0.000
Ladybug	1.5708	-1.2761	1.0237
Spider	1.6198	-2.5128	2.4112
Ferret	1.2503	-1.1980	0.5630
Cross	1.6702	-2l3497	-3.0858
Celtic knot	1.2227	2.3562	-0.5588
Four-leaf clover	0.3755	-2.9764	-0.1836
Star	1.3009	-1.9462	-0.5477

Table 3.5: Initial states of the choreographies selected.

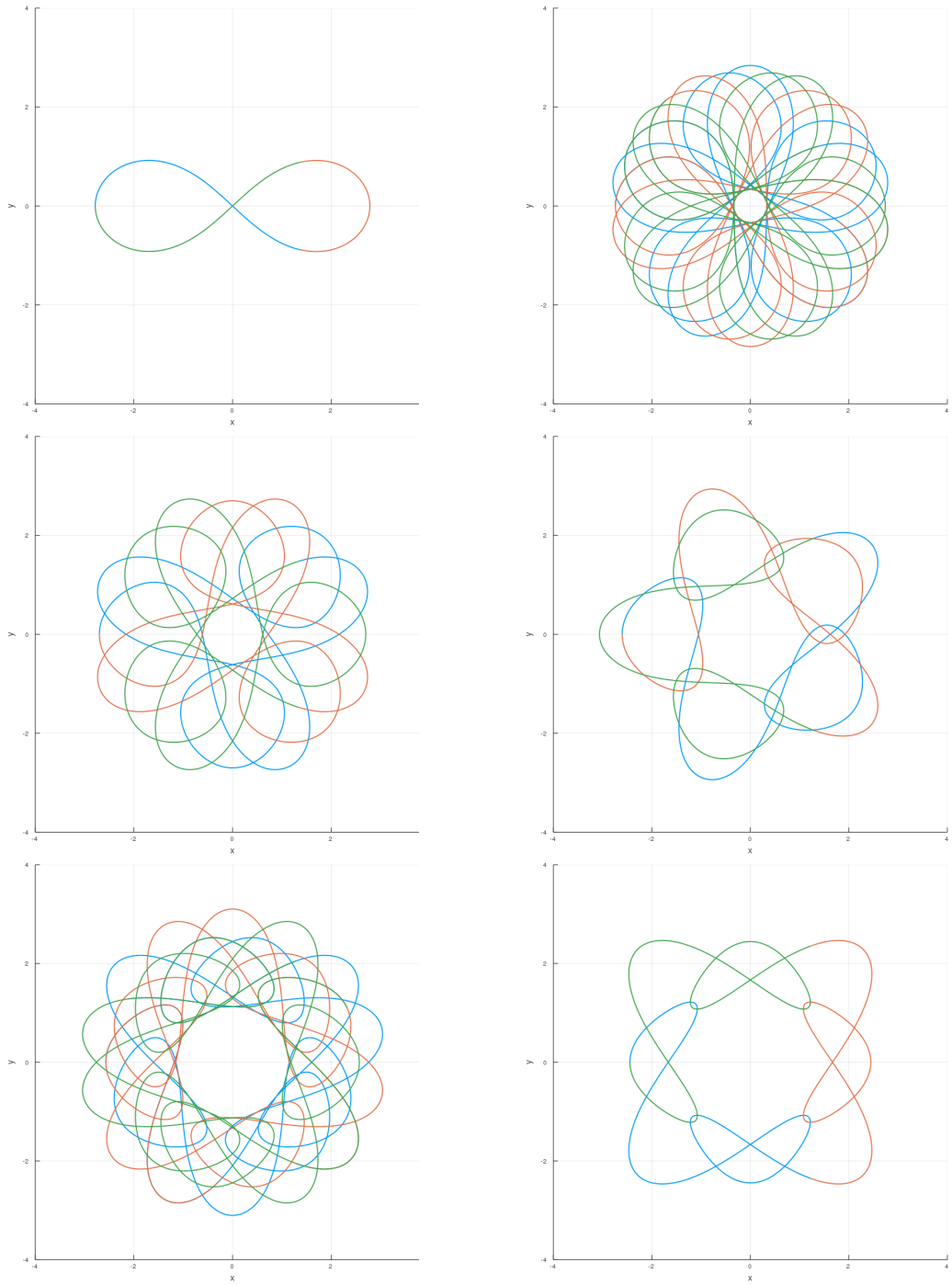


Figure 3.3: The continuation of the figure eight choreography. Top-left to bottom-right.

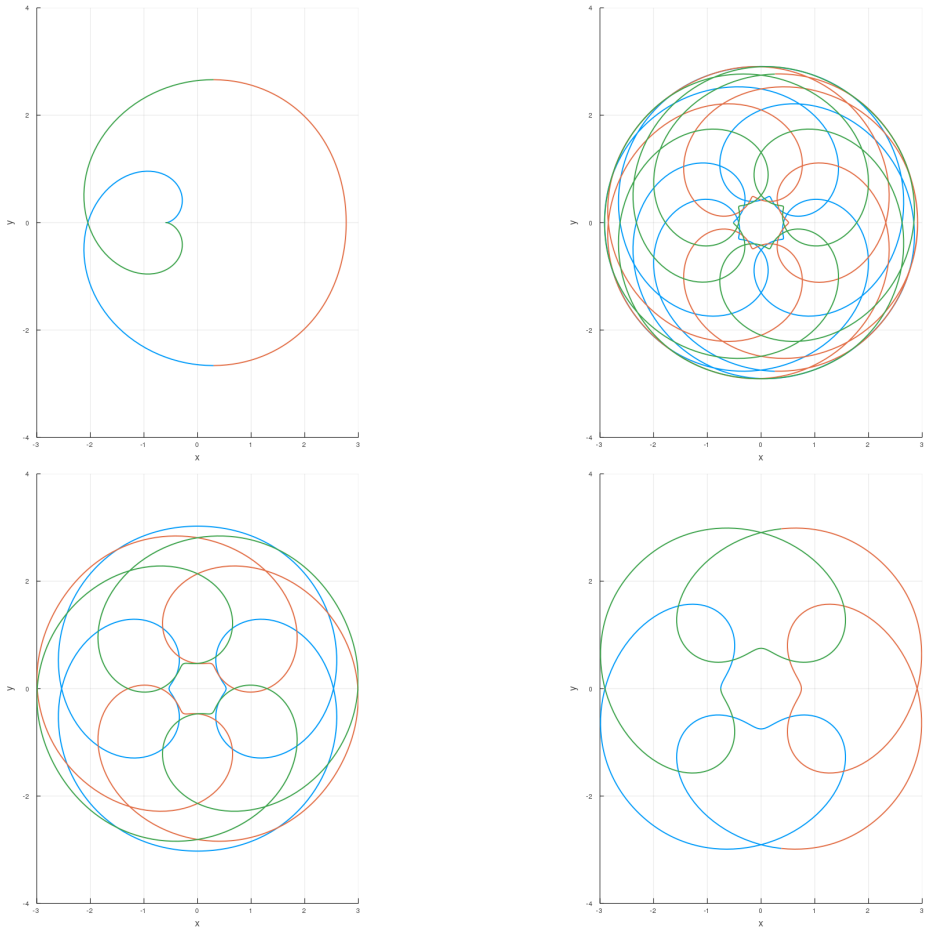


Figure 3.4: The continuation of the heart choreography. Top-left to bottom-right.

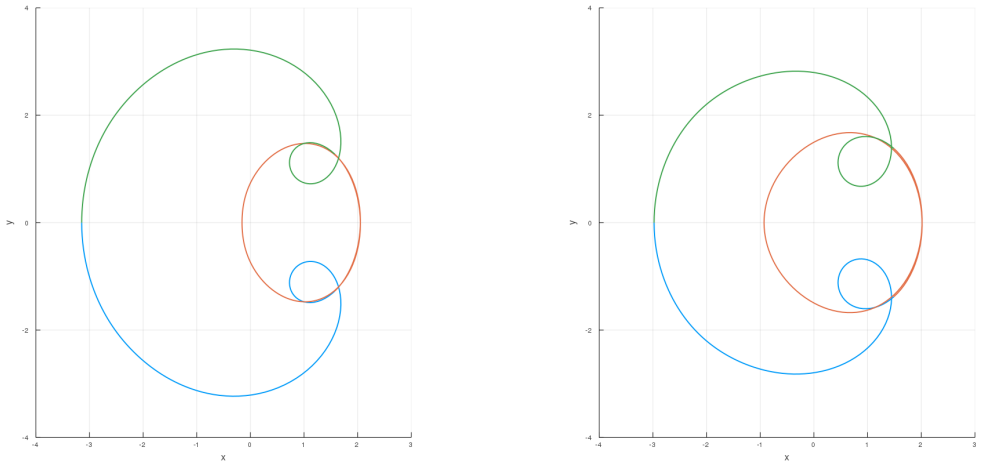


Figure 3.5: Both ladybug choreographies side by side.

CHAPTER 4

Approach 2: Variational method

4.1 Description of the approach

Our task is to find three-body choreographies of different topological properties. A detailed explanation of this idea can be found in section 2.8.

In section 3 the problem is posed as an optimization problem. In this section, a completely different approach is used. The problem is posed as a variational calculus problem, inspired by Vanderbei's paper [VANDERBEI, 2004]. Instead of trying to find solutions that satisfy the permutation symmetry of choreographies in the space of solutions to Newton's law of universal gravitation, the opposite is done. Curves with the desired topology are generated that satisfy permutation symmetry, and then they are modified in order to satisfy gravitational force.

To generate an initial solution, two methods were used, the naive method and the corrected method. The idea behind the naive method is to generate the desired collinear configurations and then to interpolate between them to generate the entire curve. This method is flawed and therefore another method is defined, called the corrected method. The corrected method takes the desired code and works backwards from it, forcing the generated curve to have said code.

To solve the variational calculus problem, the curve is passed as the initial point to an algorithm for continuous optimization. The algorithm then tries to minimize the action functional starting from that initial point. If the algorithm converges into a local minimum,

it is guaranteed to be a solution to Newton's equations as it is a stationary point of the action.

With the approach of the optimization problem, the ability to control the codes of the choregraphies obtained was completely out of our reach. On the other hand, this method allows for a more specialized search of choreographies. Given a code, the method would find a choreography that matches said code, in case of convergence.

4.2 Code inference

Given any P periodic N-body curve (not necessarily a choreography), a code can be assigned to said curve in such a way that it encodes its topological features, as explained in section 2.2. To achieve this, the algorithm needs to list the index of the middlemost body of each collinear formation, chronologically sorted.

Because of the discrete structure of computers, discretized curves are used to estimate their code. This means that since rounding error is inevitable, the inferred codes may be inaccurate if the correct hyperparameters are not chosen.

To detect that all three bodies form a straight line, the same function is used as in section 3.3.2. This function is equal to zero if and only if all three bodies are aligned:

$$g_{\text{col}}(u(\tau)) = x_{12}y_{23} - y_{12}x_{23}$$

To detect for which values of tau $g(q(\tau))$ equals zero, naively checking for which values of τ the sign of the output changes does not suffice. This is because we can have that g just barely reaches zero at a certain value of τ but never changes signs, similar to the function x^2 at the origin for example.

The roots of g are found by detecting when the derivative of its square changes from negative to positive, and then checking whether $g \approx 0$ at that specific point. The derivative of its square will only go from negative to positive at local minima of the square of g. We will call this new function h:

$$h(t) = \frac{1}{2}\frac{d}{dt}(g(q(t))^2) = g(q(t))g'(q(t))$$

where

$$g'(q) = (\dot{x}_2 - \dot{x}_1)(y_3 - y_2) + (x_2 - x_1)(\dot{y}_3 - \dot{y}_2) - (\dot{x}_3 - \dot{x}_2)(y_2 - y_1) - (x_3 - x_2)(\dot{y}_2 - \dot{y}_1)$$

For every time h changes signs, the exact instant at which h equals zero is estimated by linear interpolation. This will result in the sequence $\tau_1, \ldots, \tau_{k'}$.

Because all local minima of g^2 will be selected, the values for τ detected with the above method also include some that are not roots of g. In other words, even though the method detects all roots of g, it also detects some non-roots of g (false positives). To deal with this, we will simply evaluate g on each of these points and discard any non-roots. Of course, because of the rounding error we will be checking whether $g(\tau_i)^2 - \varepsilon < 0$, for some small ε. Since for any continuous curve the number of false positives is very low, efficiency-wise it is not a matter of concern.

Lastly, to determine which body is closest to the middle, the moduli of vectors q_{12}, q_{23}, and q_{31} can be compared. For example, if body number one is in the middle then $\|q_{23}\|^2$ will be greater than both $\|q_{12}\|^2$ and $\|q_{31}\|^2$.

4.3 Curve generation

4.3.1 Naive method for curve generation

The code of a P periodic curve represents in chronological order the middlemost body each time all three bodies are aligned through $[0, T)$. This is explained in-depth in section 2.2.

The first step of the variational approach is to generate a P periodic curve for a given code, not necessarily a solution to Newton's differential equations. This naive approach involves first generating the points of the curve concerning the collinear configurations, and then interpolating between them to obtain a continuous curve.

More precisely, we have the code $c = (c_1, c_2, \ldots, c_l)$, where c_i is the index of the body closest to the center at collinear configuration number i. We fix $P = 2\pi$, and then the positions of the bodies at l discrete points are defined for $i = 1, \ldots, l$:

$$
\begin{aligned}
t_i &= P\frac{(i-1)}{l} \\
\theta_i &= \frac{2\pi}{a}(i-1) \\
x_{c_i}(t_i) &= 0 \\
y_{c_i}(t_i) &= 0 \\
x_{(c_i \mod 3)+1}(t_i) &= cos(\theta_i) \\
y_{(c_i \mod 3)+1}(t_i) &= sin(\theta_i) \\
x_{(c_i+1 \mod 3)+1}(t_i) &= -cos(\theta_i) \\
y_{(c_i+1 \mod 3)+1}(t_i) &= -sin(\theta_i)
\end{aligned}
$$

for some constant value of $a > 0$. In our case $a = 3$ was chosen, as the curve can get quite cluttered for high values of a. Also, note that if $a \in 1,2$ then the curves generated are degenerate and not ideal for the task.

In order for the curve defined to be a choreography, another condition has to be satisfied: all three bodies must permute. To achieve this, points will be generated for $t \in [0,T)$, for $T = P/3$. Then, the rest of the points regarding the collinear configurations will be generated via permutation. That is, the rest of the points are generated in order to satisfy

$$q_i(t) = q_j(t + T((i-j) \mod 3)) \tag{4.1}$$

Once all l points are obtained, trigonometric interpolation is used to interpolate between them and to obtain a continuous P periodic curve.

The generated curve has three nice properties:

- Because of the symmetry of the initial points generated, the interpolated points also satisfy equation (4.1).
- For all l points, the center of mass is at the origin. The equivalent for that statement is that the interpolating trigonometric polynomial of order $l-1$ has its coefficients 0, 3, 6, 9, ... equal to zero. Because of this, all interpolated points also have the center of mass at the origin.
- Collision is impossible when θ is not a multiple of π. We have no formal proof for this property, but it is suggested by experimental evidence.

Finally, P was fixed to 2π at first, but the scale of the system is not appropriate for the minimization of the system for that value of P. By using the scale invariance principle, curve q is rescaled to $Q = \nu q$, for a value of ν that minimizes the action functional.

Trivially,

$$A(\nu q) = K(q)\nu^2 + T(q)/\nu$$

thus,

$$\arg\min_{\nu>0} \; A(\nu q) = \left(\frac{T(q)}{2K(q)}\right)^{1/3}$$

Even if this method is guaranteed to generate a curve that has the collinear configurations that appear in the code, undesired collinear configurations also appear in some cases. For example, let's say a curve with relative code 123 is needed to be generated. Then, the method successfully generates a curve with collinear configurations with body 1 in the middle first, body 2 and then body 3. But, other collinear configurations may also appear in between them, as a result of the interpolation. Consequently, the actual code of the curve generated may be something like 2132223.

4.3.2 Corrected method for curve generation

The incorrect behavior of the naive method defined in section 4.3.1 taken into account, a new method is defined that fixes it. The goal is the same, from a given code to generate the according choreography. The insight behind this method is to force the generated curve to have the given code, with what can be understood as reversing the logic for the algorithm for code inference defined in section 4.2.

Because this method involves rotations of the curves, instead of using tuples of parametric curves $q_{jk}(t) = (x_jk(t), y_{jk}(t))$, parametric curves of complex numbers are used such that $z_{jk}(t) = x_jk(t) + y_{jk}(t)i$. This way, they can easily be represented in polar coordinates, $z_{jk} = r_{jk}e^{i\theta_{jk}}$

By comparing angles we can detect collinear configurations, and also which body is in the middle. For example, when $\theta_{12} = \theta_{31}$, then body number 2 is closest to the center, and also $|\theta_{12} - \theta_{23}| = \pi$. Therefore, by defining the parametric curve $\theta(\sigma) := \theta_{12}(t) - \theta_{31}(t)$,

the l discrete values for σ that make $\theta = 0$ can be set. A uniform distribution can be used, $\sigma = 0, 2\pi\frac{1}{l}, 2\pi\frac{2}{l}, \ldots, 2\pi\frac{l-1}{l}$, and reparameterization $\theta(\sigma)$ to $\theta(t)$ can be done afterwards.

Once l discrete points for $\theta(\sigma)$ are calculated, we need to obtain θ_{12} and r_{12}, as z_{12} can be obtained from them. Of course, since we only have information about $\theta(\sigma)$, this is not enough to determine the values for $\theta_{12}(\sigma)$ and $r_{12}(\sigma)$. Because we have some freedom, parameters $\alpha_1, \ldots, \alpha_m$ are defined such that $\theta_{12}(\sigma)$ and $r_{12}(\sigma)$ can be determined. These α_i parameters concern r and the angular momentum. This way, $\theta_{12}(\sigma)$ and $r_{12}(\sigma)$ can be defined as functions of α_i.

Then, using trigonometric interpolation N discrete points can be obtained with uniform distribution $\sigma = 0, 2\pi\frac{1}{N}, 2\pi\frac{2}{N}, \ldots, 2\pi\frac{N-1}{N}$. From there $z_{12}(\sigma)$ is obtained for those N values of $\theta_{12}(\sigma)$ and $r_{12}(\sigma)$.

Once obtained $z_{12}(\sigma)$, a change of parameters can be done. The connection between σ and t is defined as

$$\left.\frac{dt}{d\sigma}\right|_{\sigma=\sigma_j} = s_j, \tag{4.2}$$

where $\sigma_j = \frac{2\pi j}{N}$ for $j = 0, 1, \ldots, N$. The values for s are obtained by fixing the total energy to $-\frac{1}{2}$:

With notation $\frac{d}{dt}z = \dot{z}$ and $\frac{d}{d\sigma}z = z'$, then $z' = \dot{z}s$.

We have the following formula for the energy:

$$H = \sum_{j=1}^{3}\left(\frac{|\dot{z}_j|^2}{6} - \frac{1}{|z_j|}\right)$$

By fixing the energy to $-\frac{1}{2}$:

$$\begin{aligned} -\frac{1}{2} &= \sum_{j=1}^{3}\left(\frac{|\dot{z}_j|^2}{6} - \frac{1}{|z_j|}\right) \\ &= \sum_{j=1}^{3}\left(\frac{|z'_j|^2}{6s^2} - \frac{1}{|z_j|}\right) \\ &= \frac{1}{s^2}\sum_{j=1}^{3}\frac{|z'_j|^2}{6} - \sum_{j=1}^{3}\frac{1}{|z_j|} \end{aligned}$$

Solving for s

$$s = \sqrt{\frac{\sum_{j=1}^{3} \frac{|z_j'|^2}{6}}{\sum_{j=1}^{3} \frac{1}{|z_j|} - \frac{1}{2}}}$$

The N discrete values for t can be obtained with these $s_1, \ldots, s_N$ by solving the system of differential equations (111).

With this procedure, given a code and parameters α_i, we are able to obtain N values of z_{12}. It can be proven that the curve generated has the code specified by reductio ad absurdum.

Finally, note that the curve z_{12} depends on values α_i, which have not been fixed yet. This means that its action can be defined in terms of α_i: $A(\alpha_1, \ldots, \alpha_m)$. It can now be posed as an optimization problem to find the values of α_i that minimize the action. Once the optimal values for α_i are found, z_{12} will be completely known and the entire curve z can be determined by permutation. It is ensured that z has the desired code.

It is important to note that by minimizing the action in relation to α_i we are not looking for a solution to Newton's laws of gravitation, we do not expect to find a solution to those equations. Instead, the goal of that optimization is to find an initial curve z_0 that is reasonably scaled for later minimization of the action.

4.4 Action functional

4.4.1 Definition of the action functional for choreographies

The action functional is a functional that takes a curve $z(t) = (z_1(t), z_2(t), z_3(t))$ and returns a real number and it is comprised of kinetic and potential energy functions T and V. Curve $z(t)$ is a stationary point of the action, $\nabla A(z) = \vec{0}$, if and only if z is a solution to Newton's differential equations for universal gravitation.

Functions T and V for three-body curves:

$$T(\dot{z}) = \frac{1}{2} \sum_{i=1}^{3} \|\dot{z}_i(t)\|^2$$

$$V(z) = \sum_{1 \leq i < j \leq 3} \frac{1}{\|z_i(t) - z_j(t)\|}$$

Action for P periodic curves:

$$A(z;P) = \int_0^P (T(\dot{z}(t)) + V(z(t)))dt$$

If curve $z(t)$ is a choreography, it is enough to compute the action for a third of the period $T = P/3$. Variable z is redefined so that it only represents the motion of the first body:

$$x(t) := x_1(t)$$
$$y(t) := y_1(t)$$

Then, because of permutation symmetry:

$$x_2(t) = x(t + \frac{P}{3})$$
$$y_2(t) = y(t + \frac{P}{3})$$
$$x_3(t) = x(t + 2\frac{P}{3})$$
$$y_3(t) = y(t + 2\frac{P}{3})$$

The action can then be redefined to only integrate on the bounded interval $[0, P/3)$:

$$A(z;P) = 3\int_0^{\frac{P}{3}} (T(\dot{z}(t)) + V(z(t)))dt$$

4.4.2 Discretization of the action

No closed form solution for calculating the action functional is known. Therefore, a discretization of the functional is necessary in order to numerically approximate it. Uniform discretization is enough for this purpose because we are working with periodic curves. As the number of discrete samples N increases, the method converges exponentially. The intuition behind this feature is that because of the periodicity of the curve, the approximation errors cancel each other out.

Uniform discretization of the action for discrete points $t_1, \ldots, t_N$:

$$\tilde{A}_N(z;P) = 3\sum_{j=1}^{N/3}(T(\dot{z}(t_j)) + V(z(t_j)))$$

Let $N = 3M$, we will use the following notation:

$$Z = (z_1, \ldots, z_N)$$
$$\dot{Z} = (\dot{z}_1, \ldots, \dot{z}_N)$$

The action can be rewritten with the newly defined notation:

$$\hat{A}_M(Z;P) = 3\sum_{j=1}^{M}(T_j(\dot{Z}) + V_j(Z)),$$

where

$$\begin{aligned} T_j(\dot{Z}) &= \frac{1}{2}(\|\dot{z}_j\|^2 + \|\dot{z}_{j+M}\|^2 + \|\dot{z}_{j+2M}\|^2) \\ V_j(Z) &= \|z_j(t) - z_{j+M}(t)\|^{-1} \\ &\quad + \|z_{j+M}(t) - z_{j+2M}(t)\|^{-1} \\ &\quad + \|z_{j+2M}(t) - z_j(t)\|^{-1} \end{aligned}$$

Because it is in our interest to define the action as a function of positions z_i only, derivatives $\dot{z}_i$ need to be estimated. To do so, differentiation matrix D_N is used (defined in section 2.6):

$$\begin{pmatrix} \dot{z}_1 \\ \ldots \\ \dot{z}_N \end{pmatrix} = \omega D_N \begin{pmatrix} z_1 \\ \ldots \\ z_N \end{pmatrix} \tag{4.3}$$

where $\omega = \frac{2\pi}{P}$.

We will assume that all three bodies are lined up on the x axis in the initial state, without

loss of generalization. In that case, we have that:

$$y_M = y_{2M} = y_{3M} = 0.$$

Note that $x = \Re(z)$ and $y = \Im(z)$ for a simpler notation.

This is important in order for the solutions that satisfy $\nabla A(z) = \vec{0}$ to be isolated. In other words, this initial collinear configuration has to be assumed so that the Hessian matrix is not singular.

Therefore, when minimizing the discrete action there will be $6M-2$ variables:

- $3M$ variables for the x coordinate.
- $3M-3$ variables for the y coordinate.
- One variable for $\omega > 0$ (or equivalently $T = 2\pi/\omega$).

One more assumption can be done in order to reduce the dimensionality of the problem, to make the minimization of the gradient easier. For indices $j = 1, \ldots, M$:

$$z_{j+2M} = -z_j - z_{j+M} \tag{4.4}$$

and as a result of equations (4.3) and (4.4):

$$\dot{z}_{j+2M} = -\dot{z}_j - \dot{z}_{j+M}$$

In this case, there will be $4M-1$ variables to optimize:

- $2M$ variables for the x coordinate.
- $2M-2$ variables for the y coordinate.
- One variable for $\omega > 0$ (or equivalently $T = 2\pi/\omega$).

4.5 Experiments

Two tasks were carried out applying the techniques defined in this chapter:

Name	Closest encounter	No TR $e = 10^{-6}$	No TR $e = 10^{-12}$	GTR $e = 10^{-6}$	GTR $e = 10^{-12}$
Cross	4.3226e-3	6713	19843	478	6461
Celtic knot	1.0069e-1	6623	7455	326	650
Star	3.7212-1	2956	6235	346	676
Spider	5.5382e-1	1725	3596	338	1693
Ferret	7.3118e-1	1166	9479	274	8965
Ladybug	7.2609e-1	460	920	115	208
Four-leaf clover	8.6789e-1	649	10841	202	9389
Heart	1.6785	64	118	44	80
Figure eight	1.7776	58	110	48	106

Table 4.1: Number of Fourier coefficients needed for the interpolation of the curves.

1. It is convenient to verify the adecuacy of global time-renormalization for trigonometric interpolation. To do so, the same choreographies as in section 3.6 were selected because of their different close encounters. Then, the number of Fourier coefficients needed to represent each curve with an error lower than 10^-6 and 10^{-12} were calculated both with the regularized and the non-regularized equations.

   The error was calculated as the mean squared error between the discretized choreography and the interpolated points.

2. A search for choreographies with simple codes was performed. First off, curves with the desired code were generated using the generation algorithms defined in section 4.3. Then, the discretized action functional was minimized using Nelder-Mead. Finally, in case of convergence a relative choreography with the desired code is obtained.

4.6 Results

1. The number of Fourier coefficients needed for the representation of each curve are shown in Table 4.1. The distance between the bodies in the closest encounter is also shown.

2. We have applied our initial implementation of the variational approach to the figure eight choreography. The initially generated curve and the result obtained can be seen in Figure 4.1.

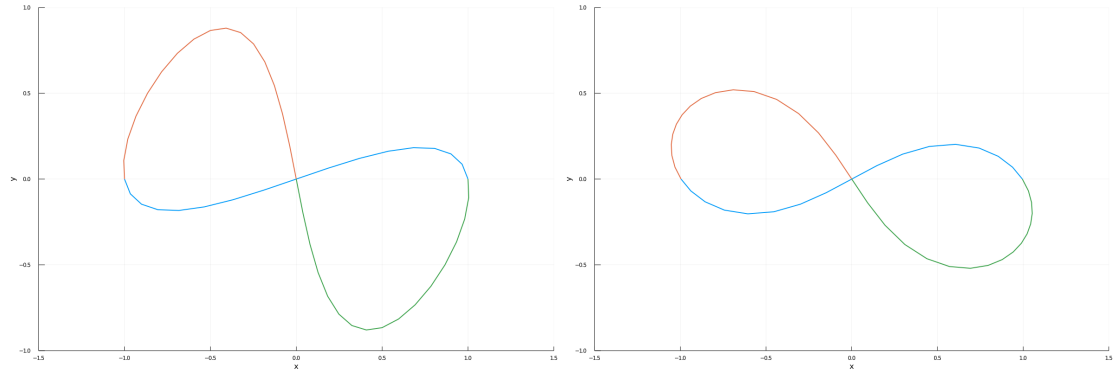


Figure 4.1: The curve generated with figure eight's code (left) and the result of minimizing its action (right).

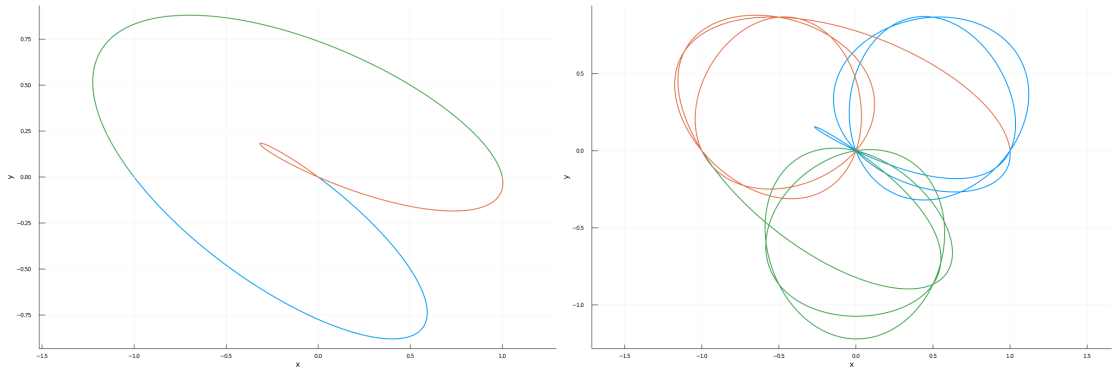


Figure 4.2: The curves generated with the hearts's code (left) and the celtic knot's code (right).

4.7 Conclusions

As seen in Table 4.1, the number of Fourier coefficients needed to interpolate a periodic curve with a close encounter is much greater when no time-regularization is used. This fact becomes especially apparent with the interpolation of the cross and the celtic knot. The use of global time-regularization for the minimization of the action functional is clearly beneficial, as the curves are discretized as a set of uniformly distributed points.

The obtention of the figure eight shows that the approach is viable for finding choreographies. No other choreographies were found, and the limiting factor here was the lack of time. Therefore, the idea that the approach is flawed based on the sole fact that only one choreography was found is inaccurate. As shown in section 4.4.2, the discretized action used in this method does not include any form of time-renormalization, thus making it much more challenging to find a stationary point of its gradient for curves with a close encounter, as proven by the first experiment.

By taking a look at Figure 4.1, it is clear that the initially generated curve was already close to the resulting solution, and we also know that the figure eight choreography does not have any close enconters. These factors combined create almost ideal conditions for the convergence of the minimization of the non-renormalized action. In contrast, the curves with the codes of the heart and the celtic knot can be seen in Figure 4.2, and it is clear that they are not as close to their corresponding choreographies compared to the figure eight.

CHAPTER 5

Conclusions and future work

5.1 Conclusions

The detailed conclusions of each approach can be found in sections 3.8 and 4.7. In this section general conclusions are drawn based on the results obtained with both approaches.

The first approach was successful at finding many choreographies. With the local search alone, 5392 relative choreographies and 166 simple choreographies were found in total. The most interesting family of choreographies discovered was the ladybug's, as we believe that it was not previously known. The collinear function seemed to be the most promising of them all. The implementation of indices did not result in a much greater variety of results, so that approach may not be ideal.

The continuation of the ladybug brought a new insight into our understanding of choreographies, which is that there can be multiple choreographies with the same shape but slightly altered. The algorithm also showed the connection between different choreographies, such as the figure eight and the celtic knot, and it enabled the discovery of new choreographies, such as the cross.

The variational method is definitely interesting, because of the ability of choosing the desired code. However, because of lack of time, only the figure eight choreography was obtained with the method.

The results of the experiments performed to determine the effectiveness of global time-

renormalization were similar to those obtained by Antoñana et al. Both experiments show that the method is definitely beneficial for both approaches.

5.2 Future work

It would be interesting to define new objective functions for the local search to exploit different kinds of symmetries, similarly to how the collinear function was defined. In fact, the choreographies obtained with the collinear function show many different kinds of symmetries.

The figure eight, heart, and ladybug were the only choreographies whose curve got continued. The continuation of other curves could be done in order to obtain new simple choreographies. Especially, the choreographies obtained with the collinear function would be interesting choices, because we believe that many of them were not found before. An example would be the spider, which could be continued to find a simple choreography isomorphic to the heart.

The variational approach needs two changes done. For one, time-renormalization should be done to the action in order to decrease the number of points needed to represent a curve, leading to a faster optimization process. Secondly, the curve generation process is not ideal, and it could be improved to generate curves that are closer to actual choreographies.

Appendices

APPENDIX A

Initial values of some choreographies

Some of the choreographies obtained with the collinear function can be seen in Figure A.1, and their initial values in Table A.1.

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9
1.6198	-2.5139	-0.1891	0.0978	0.1891	0.0978	29.8926	0.0003	43.3783
-1.3915	-0.8891	-0.6836	-0.1456	0.6836	-0.1456	13.4303	2.7182	21.7727
2.4369	-1.7981	-0.1924	0.0473	0.1924	0.0473	21.6628	2.6592	28.7017
1.6177	-2.8046	-0.1324	0.0765	0.1324	0.0765	16.1109	2.0950	21.6930
2.8652	-0.2537	-1.3033	0.0385	1.3033	0.0385	55.6791	0.0107	69.7850
1.7607	-0.8332	-0.6642	0.1048	0.6642	0.1048	28.2365	0.0077	43.4511

Table A.1: Initial z vectors of the selction of choreographies obtained with the collinear function.

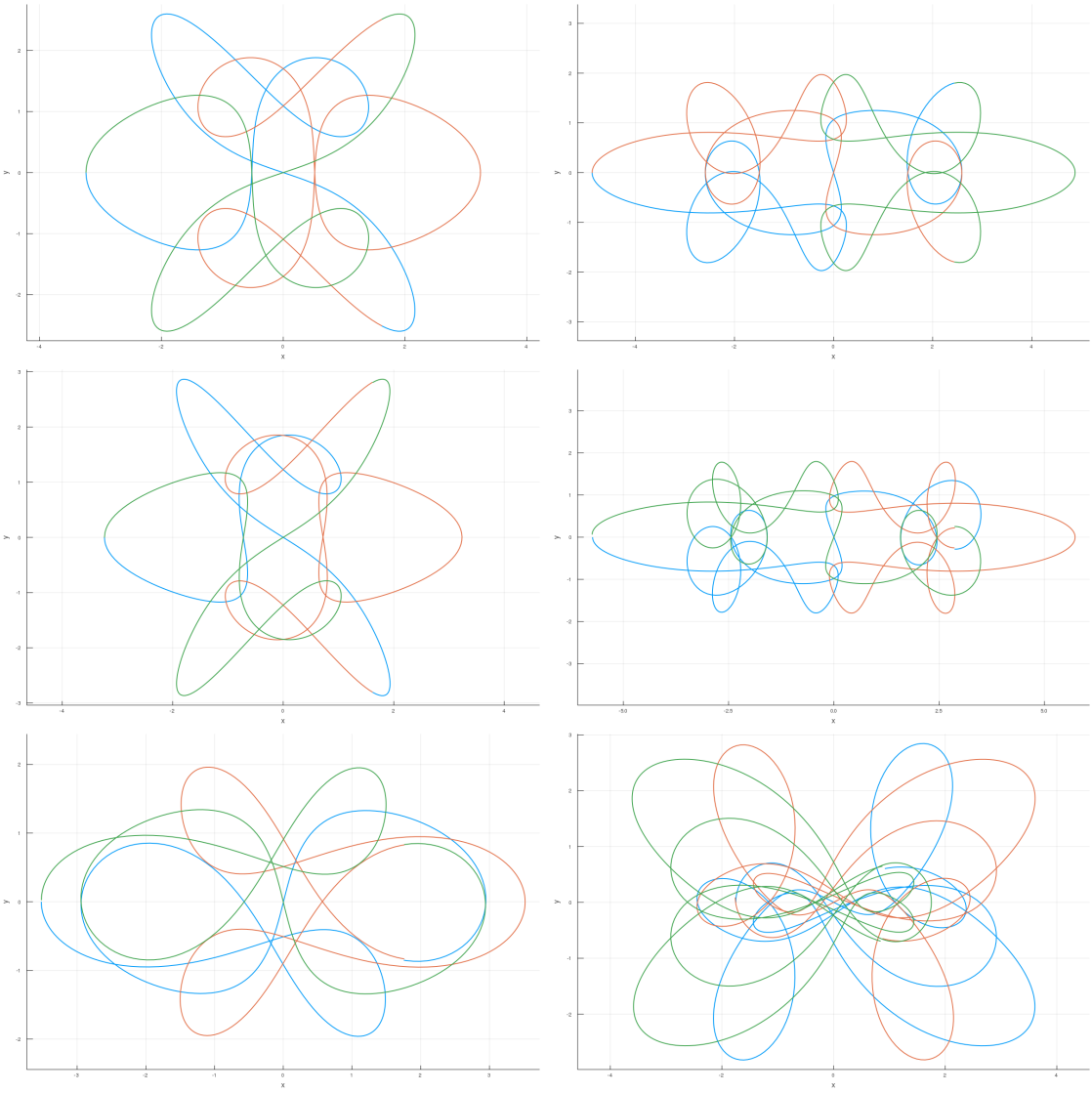


Figure A.1: A selection of choreographies obtained with the collinear objective function.

APPENDIX B

Source code of the project

The source code was written in the Julia programming language. This code was then put together into a set of Jupyter notebooks, which allow for an interactive execution of the code. They are licensed under the MIT license; a permissive free software license. The notebooks can be found in the following repository:

https://github.com/MarkelZ/3body-problem

Bibliography

[Antoñana et al., 2020] Antoñana, M., Chartier, P., Makazaga, J., and Murua, A. (2020). Global time-renormalization of the gravitational *n*-body problem.

[Chenciner and Montgomery, 2000] Chenciner, A. and Montgomery, R. (2000). A remarkable periodic solution of the three-body problem in the case of equal masses. *Annals of Mathematics*, pages 881–901.

[Montgomery, 2014] Montgomery, R. (2014). The three-body problem and the shape sphere.

[Moore, 1993] Moore, C. (1993). Braids in classical dynamics. *Phys. Rev. Lett.*, 70:3675–3679.

[Qiu-Dong, 1990] Qiu-Dong, W. (1990). The global solution of the n-body problem. *Celestial Mechanics and Dynamical Astronomy*, 50.

[Schaffer, 1981] Schaffer, S. (1981). Richard s. westfall, never at rest. a biography of isaac newton, cambridge university press, 1980, 8vo, pp. xviii, 908, illus., £25.00. *Medical History*, 25(4):432–434.

[Simó, 2001] Simó, C. (2001). New families of solutions in n-body problems. In *European Congress of Mathematics*, pages 101–115. Springer.

[Simó, 2002] Simó, C. (2002). Dynamical properties of the figure eight solution of the. In *Celestial Mechanics: Dedicated to Donald Saari for His 60th Birthday: Proceedings of an International Conference on Celestial Mechanics, December 15-19, 1999, Northwestern University, Evanston, Illinois*, volume 292, page 209. American Mathematical Soc.

[Sundman et al., 1913] Sundman, K. F. et al. (1913). Mémoire sur le problème des trois corps. *Acta mathematica*, 36:105–179.

[VANDERBEI, 2004] VANDERBEI, R. J. (2004). New orbits for then-body problem. *Annals of the New York Academy of Sciences*, 1017(1):422–433.