

Masters Thesis

The design, implementation, and applications of a novel numerical solver: the Lagrangian solver

Vincent Paktoo Chan

Advisor:

Prof. Eliot Fried

February 2018



Okinawa Institute of Science and Technology
Graduate University

Acknowledgements

I spent a long time at OIST, much longer than I had originally intended. Ignoring the details, let me briefly thank a few people and sections that contributed to my current positive conclusion at OIST. I thank them in no particular order of importance.

Firstly, I thank my advisor, Eliot Fried, for letting me into his unit, which provided me with many opportunities to learn about interesting topics such as Rational Thermodynamics and configurational forces. Secondly, I thank Johannes Schönke, a postdoc in the Fried unit, for letting me play with his magnetic spheres project as part of this thesis. Hannes also had the dubious honor of being the main person to read my drafts and point out mistakes.

Lastly, I thank the Student Support Section and the Academic Support Section. Although they are separate sections, I'm afraid I never managed to separate them in my head. The two sections provided me with all sorts of administrative support, which is why I was able focus on the fun part of graduate student life (the research) and not worry about the boring stuff (the various paper work required by OIST, Okinawa, and/or Japan).

Thanks.

Contents

1	Introduction	1
2	Automatic differentiation	2
2.1	The algebraic structure of automatic differentiation	2
2.2	Summary	6
3	Collocation methods	7
3.1	Collocation method	8
3.1.1	Differentiation matrix	8
3.2	A few examples	10
3.3	Applying the collocation method to the Euler–Lagrange equations	13
3.4	A few extra notes	15
3.4.1	Numerical order	15
3.4.2	Timestep	16
3.5	Summary	16
4	Test case: Hénon–Heiles model	17
5	A chain of three magnetized, steel balls	20
5.1	Kinematics	21
5.2	Kinetic energy	22
5.3	Potential energy	22
5.4	Dimensionless energies	23
6	Results	24
6.1	Survey of the energy landscape	24
6.2	Results	29
6.3	Discussion and conclusion	33

7 Summary **37**

References **39**

Chapter 1

Introduction

For this thesis, we set out to achieve two main goals:

1. design and implement a new numerical solver with properties appropriate for studying the long-term dynamics of Lagrangian systems; and
2. applying it to a system of magnetized, steel balls stacked on top of each other with the bottom ball fixed in place for the purpose of understanding how the interplay between the stabilizing, magnetic force and the destabilizing, gravitational force influences its dynamics.

In the design of the numerical solver for Lagrangian systems, we notice that the collocation method can be coupled with automatic differentiation to produce a novel solver that takes as its only input the Lagrangian itself. For such a solver, there is no need for the user to explicitly derive the equations of motion. A formalism for the automatic differentiation technique is laid out in Chapter 2 and the collocation method is detailed in Chapter 3.

In Chapter 4, the novel and newly developed numerical solver is applied to the Hénon-Heiles model as a simple test case. Particularly, we wish to understand just how well our new numerical solver manages to conserve energy.

Finally, with the test case dealt with, we apply our method on a system of magnetized, steel balls in Chapter 5. As mentioned above, we are primarily interested in understanding how the interplay between stabilizing and destabilizing forces affects the dynamics of the system.

Chapter 2

Automatic differentiation

To avoid having to personally calculate the gradient and Hessian when supplied with a multivariate, scalar function, we introduce a technique known as automatic differentiation. It is not to be confused with symbolic differentiation or numerical differentiation (e.g. via finite differences). In particular, it does not output an analytical expression (symbolic differentiation) and it is not based on some sort of discretization or approximation (finite differences). As such, it is much less memory intensive (unlike symbolic differentiation) and it is essentially exact (unlike finite differences, which is an approximation).

While there are many types of automatic differentiation, they are broadly separated into two classes: forward mode and reverse mode. Forward mode is the version that will be detailed in this section, as it is the easier to implement. A version of the reverse mode exists under the name *backpropagation*, a technique familiar to those working with/on neural networks. The interested reader may refer to “Evaluating Derivatives” by Griewank and Walther [1] or “The Art of Differentiating Computer Programs” by Naumann [2].

Here, I provide a new, simple implementation of the forward mode automatic differentiation, even though it is not the most efficient. In particular, I show how, by explicitly keeping track of value the gradient and Hessian of a function, differentiation of twice-differentiable functions simplifies to multiplication and addition.

2.1 The algebraic structure of automatic differentiation

To begin with, let us see automatic differentiation in action with a simple example: Let $f, g, h : \mathbb{R}^N \rightarrow \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^N$ and consider, at point \mathbf{x} , the value the product $h = fg$, as

well as its gradient and Hessian. They are

$$h(\mathbf{x}) = f(\mathbf{x})g(\mathbf{x}), \quad (2.1)$$

$$\nabla h(\mathbf{x}) = f(\mathbf{x})\nabla g(\mathbf{x}) + g(\mathbf{x})\nabla f(\mathbf{x}), \text{ and} \quad (2.2)$$

$$\begin{aligned} \nabla\nabla h(\mathbf{x}) &= \nabla(f\nabla g + hg\nabla f)|_{\mathbf{x}} \\ &= f(\mathbf{x})\nabla\nabla g(\mathbf{x}) + \nabla f(\mathbf{x}) \otimes \nabla g(\mathbf{x}) + \nabla g(\mathbf{x}) \otimes \nabla f(\mathbf{x}) + g(\mathbf{x})\nabla\nabla f(\mathbf{x}), \end{aligned} \quad (2.3)$$

where \otimes is the outer product for vectors. From (2.1 – 2.3), we see that it is possible to calculate $h(\mathbf{x})$, $\nabla h(\mathbf{x})$, and $\nabla\nabla h(\mathbf{x})$ without knowing the function h itself; we merely need to know the values, at point \mathbf{x} , of the functions f and g , their gradients, and their Hessians. That is, suppose we store the values of f , its gradient, and its Hessian at \mathbf{x} as s_f , \mathbf{g}_f , and \mathbf{H}_f , respectively, and similarly store the values of g , its gradient, and its Hessian at point \mathbf{x} as s_g , \mathbf{g}_g , and \mathbf{H}_g , respectively. Then, using (2.1 – 2.3),

$$h(\mathbf{x}) = s_f s_g, \quad (2.4)$$

$$\nabla h(\mathbf{x}) = s_f \mathbf{g}_g + s_g \mathbf{g}_f, \text{ and} \quad (2.5)$$

$$\nabla\nabla h(\mathbf{x}) = s_f \mathbf{H}_g + \mathbf{g}_f \otimes \mathbf{g}_g + \mathbf{g}_g \otimes \mathbf{g}_f + s_g \mathbf{H}_f, \quad (2.6)$$

and so, $h(\mathbf{x})$, $\nabla h(\mathbf{x})$, and $\nabla\nabla h(\mathbf{x})$ were calculated without actually differentiating h itself.

We can do something similar to the example above for when $h = f + g$ or $h = \phi \circ f$, where $\phi : \mathbb{R} \rightarrow \mathbb{R}$. And so, we can imagine defining a new unitary, associative algebra \mathbf{A} (the “A” is for automatic differentiation) to replace real numbers \mathbb{R} and doing our calculations on the new algebra \mathbf{A} instead. That way, information regarding the gradient and Hessian comes free.

Definition 1. (*The unitary, associative algebra of automatic differentiation numbers*) *The field $\mathbf{A}(N)$ (or simply \mathbf{A}) is the set of ordered triplets*

$$(s, \mathbf{g}, \mathbf{H}), \quad (2.7)$$

where $s \in \mathbb{R}$, $\mathbf{g} \in \mathbb{R}^N$, and $\mathbf{H} \in \mathbb{R}^{N \times N}$, equipped with two binary operations: addition, $+ : \mathbf{A} \times \mathbf{A} \rightarrow \mathbf{A}$, and multiplication, $\cdot : \mathbf{A} \times \mathbf{A} \rightarrow \mathbf{A}$, and a scalar multiplication $\cdot : \mathbb{R} \times \mathbf{A} \rightarrow \mathbf{A}$

as defined by

$$(s_a, \mathbf{g}_a, \mathbf{H}_a) + (s_b, \mathbf{g}_b, \mathbf{H}_b) = (s_a + s_b, \mathbf{g}_a + \mathbf{g}_b, \mathbf{H}_a + \mathbf{H}_b) \quad (2.8)$$

$$\begin{aligned} (s_a, \mathbf{g}_a, \mathbf{H}_a) \cdot (s_b, \mathbf{g}_b, \mathbf{H}_b) &= (s_a s_b, \\ &\quad s_a \mathbf{g}_b + s_b \mathbf{g}_a, \end{aligned}$$

$$s_a \mathbf{H}_b + \mathbf{g}_a \otimes \mathbf{g}_b + \mathbf{g}_b \otimes \mathbf{g}_a + s_b \mathbf{H}_a), \text{ and} \quad (2.9)$$

$$a \cdot (s, \mathbf{g}, \mathbf{H}) = (as, a\mathbf{g}, a\mathbf{H}), \forall a \in \mathbb{R}. \quad (2.10)$$

The operator $+$ has also been overloaded to also act as the usual scalar, vector, and matrix addition operators, \cdot has been overloaded to also act as the usual scalar multiplication operators, and $\otimes : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^{N \times N}$ is the familiar outer product.

The additive and multiplicative identity elements are given by

$$\mathbf{0}_A = (0, \mathbf{0}_N, \mathbf{0}_{N \times N}) \text{ and} \quad (2.11)$$

$$\mathbf{1}_A = (1, \mathbf{0}_N, \mathbf{0}_{N \times N}), \quad (2.12)$$

respectively, and unique. The additive inverse element $-\mathbf{a}$ and the multiplicative inverse element \mathbf{a}^{-1} of element $\mathbf{a} = (s, \mathbf{g}, \mathbf{H})$ are

$$-\mathbf{a} = (-s, -\mathbf{g}, -\mathbf{H}) \text{ and} \quad (2.13)$$

$$\mathbf{a}^{-1} = (s^{-1}, -s^{-2}\mathbf{g}, 2s^{-3}\mathbf{g} \otimes \mathbf{g} - s^{-2}\mathbf{H}), \text{ for } s \neq 0, \quad (2.14)$$

respectively, and unique. Further, the two binary operations in \mathbf{A} can be straightforwardly shown to be commutative and associative both additively and multiplicatively. Lastly, multiplication in \mathbf{A} is distributive over addition. And so, \mathbf{A} is a unitary, associative algebra.

Although a multiplicative inverse does not exist for all automatic differentiation numbers, it does exist for all automatic differentiation numbers whose scalar part (the first component) is non-zero. Recall that this algebra \mathbf{A} is designed to replace real numbers. Since we have no reason to divide a real number by zero, by extension we have no reason to divide an automatic differentiation number $\mathbf{a} \in \mathbf{A}$ by another $\mathbf{b} \in \mathbf{A}$ whose scalar part is zero. Hence, for all intended purposes, the algebra \mathbf{A} can be thought of as an extension of real numbers.

To make this data structure / algebraic structure useful, we need to add or mention a few other items. First, we require the following selection operators: scalar : $\mathbf{A} \rightarrow \mathbb{R}$, gradient : $\mathbf{A} \rightarrow \mathbb{R}^N$, and Hessian : $\mathbf{A} \rightarrow \mathbb{R}^{N \times N}$ as defined by

$$\text{scalar}(s, \mathbf{g}, \mathbf{H}) = s, \quad (2.15)$$

$$\text{gradient}(s, \mathbf{g}, \mathbf{H}) = \mathbf{g}, \text{ and} \quad (2.16)$$

$$\text{Hessian}(s, \mathbf{g}, \mathbf{H}) = \mathbf{H}. \quad (2.17)$$

Next, we note that functions $\phi : \mathbb{R} \rightarrow \mathbb{R}$ can be more efficiently extended as $\tilde{\phi} : \mathbf{A} \rightarrow \mathbf{A}$ by direct definition. Take $\sin(x)$ as an example. Since

$$\nabla \sin(x) = \cos(x) \nabla x \text{ and} \quad (2.18)$$

$$\begin{aligned} \nabla \nabla \sin(x) &= \nabla(\cos(x) \nabla x) \\ &= -\sin(x) \nabla x \otimes \nabla x + \cos(x) \nabla \nabla x, \end{aligned} \quad (2.19)$$

we can define $\sin : \mathbf{A} \rightarrow \mathbf{A}$ as

$$\sin(s, \mathbf{g}, \mathbf{H}) := (\sin(s), \cos(s)\mathbf{g}, -\sin(s)\mathbf{g} \otimes \mathbf{g} + \cos(s)\mathbf{H}). \quad (2.20)$$

This is likely much more efficient than building it up as a power series,

$$\tilde{\mathbf{a}} := (s, \mathbf{g}, \mathbf{H}) \quad (2.21)$$

$$\sin(\tilde{\mathbf{a}}) := \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \tilde{\mathbf{a}}^{2n+1}. \quad (2.22)$$

Lastly, we illustrate how independent variables are initialized under $\mathbf{A}(N)$ via an example: Suppose we are given the Lagrangian $L : \mathbb{R}^4 \rightarrow \mathbb{R}$,

$$L(x, y, \dot{x}, \dot{y}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy, \quad (2.23)$$

where m and g are constants. Since there are four independent variables with values x , y , \dot{x} , and \dot{y} , we will be working in the field $\mathbf{A}(4)$. We then initialize the independent variables in $\mathbf{A}(4)$ as

$$\mathbf{X} = (x, \mathbf{e}_1, \mathbf{0}_{4 \times 4}), \quad (2.24)$$

$$\mathbf{Y} = (y, \mathbf{e}_2, \mathbf{0}_{4 \times 4}), \quad (2.25)$$

$$\dot{\mathbf{X}} = (\dot{x}, \mathbf{e}_3, \mathbf{0}_{4 \times 4}), \text{ and} \quad (2.26)$$

$$\dot{\mathbf{Y}} = (\dot{y}, \mathbf{e}_4, \mathbf{0}_{4 \times 4}), \quad (2.27)$$

where $\mathbf{e}_1 = (1, 0, 0, 0)^T$, $\mathbf{e}_2 = (0, 1, 0, 0)^T$, $\mathbf{e}_3 = (0, 0, 1, 0)^T$, $\mathbf{e}_4 = (0, 0, 0, 1)^T$, and $\mathbf{0}_{4 \times 4}$ is a 4×4 zero matrix. Our extended Lagrangian $\tilde{L} : \mathbf{A}^4 \rightarrow \mathbf{A}$ is then given by

$$\tilde{L}(X, Y, \dot{X}, \dot{Y}) = \frac{1}{2}m(\dot{X}^2 + \dot{Y}^2) - mgY \quad (2.28)$$

and $\nabla \tilde{L}$ and $\nabla \nabla \tilde{L}$ can be obtained via the selection operators.

2.2 Summary

In this chapter, we have provided a formalism for a forward mode of automatic differentiation that can be easily implemented via operator overloading. Operator overloading is really just a fancy way of saying that the same symbol will be used to denote different operators depending on what space it is acting on. For example, the addition symbol $+$ can mean scalar addition, vector addition, matrix addition, among other things. In our case, we would define the addition operator acting on automatic differentiation numbers, i.e. $+ : \mathbf{A} \times \mathbf{A} \rightarrow \mathbf{A}$ and replacing all of the real numbers with their automatic differentiation counterparts. This way, we also get the value of the gradient and Hessian without explicitly programming for them.

Chapter 3

Collocation methods

Given that the numerical solver will be applied in the creation of Poincaré sections, ideally the solver should be symplectic, symmetric (or reversible), implicit, and of high order.

Since the system of interest is Hamiltonian, energy is conserved. Thus, the solver itself must conserve energy to within a small error. As such, the ideal solver would be symplectic, a classification indicating that it conserves area in phase space and also conserves the quadratic first integrals of Hamiltonian systems [3]. It is possible for a solver to be conservative for all practical purposes without actually being symplectic [4]. In fact, symmetric solvers are often, but not always, found to behave as if they were symplectic [3, 5]. Further, it has been shown that no Runge–Kutta method exists that conserves all polynomial invariants of degree higher than two [3]. Hence, while ideal, symplecticity is not absolutely required.

The solver must be of high order and implicit. Together, they provide two main benefits. Higher order usually implies higher accuracy, while implicitness is needed for the method to remain stable for larger timesteps. Also, symmetry or reversibility essentially requires a numerical method to be implicit since explicit methods are those that rely only on information available in the current timestep while ignore the state of the system at a future timestep. The creation of a Poincaré sections requires the time evolution of a system over a very long period of time, which in turn requires many time-steps. The problem that this poses is an insidious one: even if a numerical solver produces a small error each time-step, the total error accumulated can grow very large if the system of interest is nonlinear. Hence, to help mitigate this issue, an implicit, high-order, symmetric, and conservative scheme is desired.

3.1 Collocation method

The collocation method, also known as the pseudospectral method, can be largely summed up as being able to do one thing: estimate the values of the derivative of a function $f : [a, b] \rightarrow \mathbb{R}$ on a set of points $\{x_k\}$, called collocation points, when given the collocation points and f 's corresponding values $\{f(x_k)\}$.

Below, we will see that the collocation method provides us with a differentiation matrix \mathbf{D} that, when acting on a column vector $(f(x_0), \dots, f(x_K))^T$, outputs an estimate for $(f'(x_0), \dots, f'(x_K))^T$, where prime indicates differentiation. This allows us to modify ordinary and partial differentiation equations into a system of, potentially nonlinear, equations.

3.1.1 Differentiation matrix

Consider a grid on $[-1, 1]$ defined by a set of $K + 1$ collocation points $\{x_0, \dots, x_K\}$ such that

$$-1 = x_0 < x_1 < \dots < x_K = 1. \quad (3.1)$$

This provides us with a set of Lagrange interpolating polynomials $l_k : [a, b] \rightarrow \mathbb{R}$, for $k = 1, \dots, K$, defined by

$$l_k(x) = \prod_{n \neq k} \frac{x - x_n}{x_k - x_n}. \quad (3.2)$$

A property of the Lagrange interpolating polynomials is that

$$l_k(x_j) = \delta_{jk}. \quad (3.3)$$

To see this, notice that if $j \neq k$, then

$$\begin{aligned} l_k(x_j) &= \left. \frac{x_j - x_n}{x_k - x_n} \right|_{n=j} \left(\prod_{n \neq j, k} \frac{x_j - x_n}{x_k - x_n} \right) \\ &= 0, \end{aligned} \quad (3.4)$$

but if $j = k$ then

$$\begin{aligned} l_k(x_j) &= l_k(x_k) \\ &= \prod_{n \neq k} \frac{x_k - x_n}{x_k - x_n} \\ &= 1. \end{aligned} \quad (3.5)$$

Since the Weierstrass approximation theorem states that any continuous function $f : [a, b] \rightarrow \mathbb{R}$ can be approximated as closely as desired as a polynomial [6], we approximate f as

$$f(x) \approx \sum_{k=0}^K c_k l_k(x). \quad (3.6)$$

At the collocation points, (3.6) is required to hold exactly. By making use of (3.3), we notice that

$$\begin{aligned} f(x_j) &= \sum_{k=0}^K c_k l_k(x_j) \\ &= \sum_{k=0}^K c_k \delta_{jk} \\ &= c_j \end{aligned} \quad (3.7)$$

and hence

$$f(x) \approx \sum_{k=0}^K f(x_k) l_k(x). \quad (3.8)$$

By taking the derivative of (3.8), we are led to another system of (approximate) equations:

$$\begin{pmatrix} f'(x_0) \\ \vdots \\ f'(x_K) \end{pmatrix} \approx \begin{pmatrix} l'_0(x_0) & \cdots & l'_K(x_0) \\ \vdots & \ddots & \vdots \\ l'_0(x_K) & \cdots & l'_K(x_K) \end{pmatrix} \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_K) \end{pmatrix}, \quad (3.9)$$

and can be more compactly written as

$$\mathbf{f}' \approx \mathbf{D}\mathbf{f} \quad (3.10)$$

where

$$\mathbf{D} := \begin{pmatrix} l'_0(x_0) & \cdots & l'_K(x_0) \\ \vdots & \ddots & \vdots \\ l'_0(x_K) & \cdots & l'_K(x_K) \end{pmatrix}. \quad (3.11)$$

A property to note is that $\det(\mathbf{D}) = 0$. This can be shown by noticing that each of Lagrange interpolating polynomials $\{l_k\}$ is an K^{th} -order polynomial, and so has $(K + 1)$ degrees of freedom. However, each of their derivatives, $\{l'_k\}$, is a polynomial of one degree lower, namely $(K - 1)^{\text{st}}$ -order and with only K degrees of freedom. Since (3.10) is a system

of $(K + 1)$ linear equations, at least one must be linearly dependent on the others. Hence, $\det(\mathbf{D}) = 0$.

That \mathbf{D} is singular is to be expected since the inverse of a derivative, i.e. integration, is only unique up to a family of constants. To properly invert a system, an initial condition or integration constant must be provided. Example 1 below demonstrates how to incorporate an initial condition to produce an invertible matrix.

While the method as developed above was for an arbitrary set of collocating points $\{x_k\}$, the version implemented utilizes the Chebyshev–Gauss–Lobatto grid points, given by

$$x_k = -\cos\left(\frac{k\pi}{K}\right), \quad \text{for } k = 0, \dots, K, \quad (3.12)$$

which consists of the endpoints $\{\pm 1\}$ and the values associated with the extrema of the K^{th} Chebyshev polynomial (of the first kind). The n^{th} -Chebyshev polynomial T_n is given by

$$T_n(x) = \cos(n \cos^{-1} x). \quad (3.13)$$

Although this set of collocation points does not lead to a formally symplectic solver, researchers have found that the collocation method with the Chebyshev–Gauss–Lobatto grid points behaves essentially symplectically and exhibits stable long term behavior [4]. Further, the differentiation matrix \mathbf{D} associated with the Chebyshev–Gauss–Lobatto collocation points is known analytically [7] and its components are given by

$$d_{ij} = \begin{cases} -\frac{1+2N^2}{6}, & i = j = 0 \\ \frac{1+2N^2}{6}, & i = j = N \\ \frac{x_j}{2(1-x_j^2)}, & i = j, 0 < j < N \\ \frac{(-1)^{i+j} p_i}{p_j(x_j - x_i)}, & i \neq j \end{cases}, \quad (3.14)$$

where

$$p_i = \begin{cases} 2, & i = 0 \text{ or } i = N \\ 1, & \text{otherwise} \end{cases}. \quad (3.15)$$

3.2 A few examples

In the examples below, the collocation points are denoted by $\mathbf{x} = \{x_k\}$, the differentiation matrix by $\mathbf{D} = (d_{ij})_{0 \leq i,j \leq K}$, and the Lagrange interpolating polynomials by l_k , $k = 1, \dots, K$.

Example 1. (*Initial Value Problem*) Find the $(K - 1)^{st}$ -order polynomial approximation to y that best satisfies

$$y'(x) + \beta y(x) = x^2, \text{ for } x \in [-1, 1] \quad (3.16)$$

subject to initial condition

$$y(-1) = \alpha. \quad (3.17)$$

The system of linear equations representing the above ODE and initial condition is given by

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ d_{10} & d_{11} + \beta_1 & d_{12} & \cdots & d_{1K} \\ d_{20} & d_{21} & d_{22} + \beta_2 & \cdots & d_{1K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{K0} & d_{K1} & d_{K2} & \cdots & d_{KK} + \beta_K \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_K \end{pmatrix} = \begin{pmatrix} \alpha \\ x_1^2 \\ x_2^2 \\ \vdots \\ x_K^2 \end{pmatrix}. \quad (3.18)$$

(3.18) above can be inverted to solve for $\mathbf{y} = (y_0, \dots, y_K)^T$, giving us

$$y(x) \approx \sum_{k=0}^K y_k l_k(x). \quad (3.19)$$

Example 2. (*Boundary Value Problem*) Find the $(K - 1)^{st}$ -order polynomial approximation to y that best satisfies

$$y''(x) = x^2, \text{ for } x \in [-1, 1] \quad (3.20)$$

subject to boundary conditions

$$y(-1) = \alpha \text{ and} \quad (3.21)$$

$$y'(1) = \beta. \quad (3.22)$$

Let $\mathbf{S} := \mathbf{D}^2$. Then the system of linear equations that represents the above problem is given by

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ s_{10} & s_{11} & \cdots & s_{1,K-1} & s_{1K} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{K-1,0} & s_{K-1,1} & \cdots & s_{K-1,K-1} & s_{K-1,K} \\ d_{K0} & d_{K1} & \cdots & d_{K,K-1} & d_{KK} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{K-1} \\ y_K \end{pmatrix} = \begin{pmatrix} \alpha \\ x_1^2 \\ \vdots \\ x_{K-1}^2 \\ \beta \end{pmatrix}, \quad (3.23)$$

where s_{ij} is the $(i, j)^{\text{th}}$ component of \mathbf{S} and d_{ij} is the $(i, j)^{\text{th}}$ component of \mathbf{D} . (3.23) can then be solved for $\mathbf{y} = (y_0, \dots, y_K)^T$. The solution is then approximately

$$y(x) \approx \sum_{k=0}^K y_k l_k(x). \quad (3.24)$$

Example 3. (*Eigenvalue Problem*) Calculate the first N eigenvalues and corresponding eigenfunctions of the time-independent Schrödinger equation

$$\frac{-\hbar^2}{2} \frac{d^2\psi(x)}{dx^2} = E\psi(x) \quad (3.25)$$

on $x \in [-1, 1]$ subject to $\psi(-1) = \psi(1) = 0$.

Let us define the Hamiltonian as

$$\mathbf{H} = \frac{-\hbar^2}{2} \mathbf{D}^2 \quad (3.26)$$

and, using for example the QR algorithm [8], calculate the eigenvalues and eigenvectors of matrix

$$\begin{pmatrix} h_{11} & \cdots & h_{1,K-1} \\ \vdots & \ddots & \vdots \\ h_{K-1,1} & \cdots & h_{K-1,K-1} \end{pmatrix} \quad (3.27)$$

where h_{ij} is the $(i, j)^{\text{th}}$ component of \mathbf{H} . For example, in Matlab and Octave, the eigenvalues and eigenvectors of a matrix A can be calculated by the command

$$[e, v] = \text{eig}(A).$$

Let \mathbf{v} represent the n^{th} eigenvector. Then the associated eigenfunction ψ is approximated by

$$\psi(x) \approx \sum_{k=0}^K v_k l_k(x). \quad (3.28)$$

Note that this solution assumes $N < K$. In practice, $K \geq 3N$ gives good results for the first N eigenvalues and their associated eigenvectors.

3.3 Applying the collocation method to the Euler–Lagrange equations

Let the Lagrangian L of a system be the functional of a path $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^N, t \mapsto \mathbf{x}(t)$, and its derivative $\mathbf{v} := \dot{\mathbf{x}} : \mathbb{R} \rightarrow \mathbb{R}^N, t \mapsto \mathbf{v}(t)$, where $\dot{\mathbf{x}}$ denotes the derivative of \mathbf{x} with respect to time t . The path \mathbf{x} that extremizes the action

$$S(t_1, t_2) = \int_{t_1}^{t_2} L(\mathbf{x}(t), \mathbf{v}(t)) dt \quad (3.29)$$

is given by the solution to the Euler–Lagrange equations,

$$\mathbf{0} = \frac{d}{dt} \frac{\partial L}{\partial \mathbf{v}} - \frac{\partial L}{\partial \mathbf{x}}, \quad (3.30)$$

a system of 2nd-order ODEs.

The goal of this section is to use the collocation method to derive a system of (nonlinear) equations whose solution approximates the solution of the Euler–Lagrange equations.

First, let us define the direct sum \oplus for vectors as

$$\mathbf{x} \oplus \mathbf{y} := \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \quad (3.31)$$

and for matrices as

$$\mathbf{A} \oplus \mathbf{B} := \text{diag}(\mathbf{A}, \mathbf{B}) = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}. \quad (3.32)$$

Also, there are the additional notations useful for taking direct sum over a larger set of indices. For example,

$$\bigoplus_{n=1}^N \mathbf{y}_n := \mathbf{y}_1 \oplus \mathbf{y}_2 \oplus \cdots \oplus \mathbf{y}_N \quad (3.33)$$

and

$$\bigoplus_{n=1}^N \mathbf{A}_n := \mathbf{A}_1 \oplus \mathbf{A}_2 \oplus \cdots \oplus \mathbf{A}_N. \quad (3.34)$$

Let us make the following definitions:

$$\mathbf{X} := \bigoplus_{k=0}^K \mathbf{x}(t_k) \quad (3.35)$$

$$\mathbf{V} := \bigoplus_{k=0}^K \mathbf{v}(t_k). \quad (3.36)$$

Then the gradient of the Lagrangian can be represented as

$$\nabla L(\mathbf{X}, \mathbf{V}) := \begin{pmatrix} \frac{\partial L}{\partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) \\ \frac{\partial L}{\partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) \end{pmatrix} = \frac{\partial L}{\partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) \oplus \frac{\partial L}{\partial \mathbf{V}}(\mathbf{X}, \mathbf{V}), \quad (3.37)$$

where its components are

$$\frac{\partial L}{\partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) = \bigoplus_{k=0}^K \frac{\partial L}{\partial \mathbf{x}}(\mathbf{x}(t_k), \mathbf{v}(t_k)) \text{ and} \quad (3.38)$$

$$\frac{\partial L}{\partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) = \bigoplus_{k=0}^K \frac{\partial L}{\partial \mathbf{v}}(\mathbf{x}(t_k), \mathbf{v}(t_k)). \quad (3.39)$$

Likewise, the Hessian of the Lagrangian can be represented as

$$\nabla \nabla L(\mathbf{X}, \mathbf{V}) := \begin{pmatrix} \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) & \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) \\ \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) & \frac{\partial^2 L}{\partial \mathbf{V} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) \end{pmatrix} \quad (3.40)$$

where its components are

$$\frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) = \bigoplus_{k=0}^K \frac{\partial^2 L}{\partial \mathbf{x} \partial \mathbf{x}}(\mathbf{x}(t_k), \mathbf{v}(t_k)), \quad (3.41)$$

$$\frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) = \bigoplus_{k=0}^K \frac{\partial^2 L}{\partial \mathbf{x} \partial \mathbf{v}}(\mathbf{x}(t_k), \mathbf{v}(t_k)), \text{ and} \quad (3.42)$$

$$\frac{\partial^2 L}{\partial \mathbf{V} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) = \bigoplus_{k=0}^K \frac{\partial^2 L}{\partial \mathbf{v} \partial \mathbf{v}}(\mathbf{x}(t_k), \mathbf{v}(t_k)). \quad (3.43)$$

Further, note that $\dot{\mathbf{x}}$ is now approximated as

$$\dot{\mathbf{X}} \approx (\mathbf{D} \otimes \mathbf{I}_N) \mathbf{X}, \quad (3.44)$$

where \mathbf{I}_N is an $N \times N$ identity matrix and \otimes denotes the tensor product. If \mathbf{A} is a $m \times n$ matrix and \mathbf{B} is an $\alpha \times \beta$ matrix, then

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix} \quad (3.45)$$

is an $m\alpha \times n\beta$ matrix.

As a system of first-order differential equations, the Euler–Lagrange equations get translated into

$$\mathbf{F}(\mathbf{X}, \mathbf{V}) = \begin{pmatrix} (\mathbf{D} \otimes \mathbf{I}_N) \frac{\partial L}{\partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) - \frac{\partial L}{\partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) \\ (\mathbf{D} \otimes \mathbf{I}_N)\mathbf{X} - \mathbf{V} \end{pmatrix} = \mathbf{0}, \quad (3.46)$$

a system of nonlinear equations with associated Jacobian \mathbf{J} given by

$$\mathbf{J}(\mathbf{X}, \mathbf{V}) = \begin{pmatrix} \mathbf{A}(\mathbf{X}, \mathbf{V}) & \mathbf{B}(\mathbf{X}, \mathbf{V}) \\ (\mathbf{D} \otimes \mathbf{I}_N) & -\mathbf{I}_N \end{pmatrix}, \quad (3.47)$$

where

$$\mathbf{A}(\mathbf{X}, \mathbf{V}) := (\mathbf{D} \otimes \mathbf{I}_N) \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) - \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{X}}(\mathbf{X}, \mathbf{V}) \quad (3.48)$$

and

$$\mathbf{B}(\mathbf{X}, \mathbf{V}) := (\mathbf{D} \otimes \mathbf{I}_N) \frac{\partial^2 L}{\partial \mathbf{V} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}) - \frac{\partial^2 L}{\partial \mathbf{X} \partial \mathbf{V}}(\mathbf{X}, \mathbf{V}). \quad (3.49)$$

Then, given the gradient and Hessian of the Lagrangian and assuming a solution to \mathbf{F} exists, the above system of nonlinear equations (3.46) can be numerically solved iteratively using the Newton–Raphson method.

3.4 A few extra notes

There are two remaining issues to be discussed regarding the collocation method:

1. what is the order of the collocation method; and
2. how to choose an appropriate timestep for a given tolerance level.

3.4.1 Numerical order

Consider once again the IVP (3.16) given above in the example section. Its solution is given by (3.19), a $(K - 1)^{\text{th}}$ -polynomial. As such, its numerical order is straight-forwardly $K - 1$.

To confirm this for the Lagrangian solver (i.e. the collocation method applied directly to the Euler–Lagrange equations with the use of automatic differentiation), we consider Hénon–Heiles model (4.1) with energy $H_0 = 1/13$ and initial conditions

$$(q_1, q_2, \dot{q}_1, \dot{q}_2) = (0, -0.1, 0.3783906541, 0). \quad (3.50)$$

Suppose the numerical order of the method is N , then the error of the method is approximately

$$E = \text{const}(\Delta t)^N, \quad (3.51)$$

where Δt is the timestep taken and the prefactor const is assumed to be independent of the timestep Δt .

Given the solution of the problem using two different timesteps (Δt_1 and Δt_2), we may estimate the order of the numerical method as

$$N = \frac{\log E_1/E_2}{\log \Delta t_1/\Delta t_2}, \quad (3.52)$$

where E_1 is the error of the solution when timestep of Δt_1 was taken and E_2 is the error that of timestep Δt_2 . Here, the $L2$ -norm $\|q_2 - q_{\text{ref}}\|$ is used to estimate the errors E_1 and E_2 .

For $K = 8$, we estimate the numerical order of the Lagrangian solver to be 6.6. For $K = 13$, we estimate the numerical order to be 10.2. It therefore seems to be the case that the integration order is roughly the polynomial degree underlying the method.

3.4.2 Timestep

Given (3.51), a error tolerance level ε , and the estimated error E associated with a given timestep Δt , we may estimate the appropriate new timestep Δt_{new} to be

$$\Delta t_{\text{new}} = P \Delta t \left(\frac{\varepsilon}{E} \right)^{(1/N)}, \quad (3.53)$$

where the prefactor P is usually taken to be 0.8 and acts as a safety factor. The error can be estimated by taking the difference between two solutions using different timesteps.

3.5 Summary

In this chapter, we have derived a new, implicit, and high-order ODE solver designed with Lagrangian systems in mind. It takes as its input the gradient and Hessian of the Lagrangian. When coupled with the automatic differentiation technique of Chapter 2, the user only needs to input the Lagrangian itself. We have thus designed a new tool that allows to user to simulate a Lagrangian system without even knowing the equations of motion!

In the next chapter, we apply this method to a few test cases to confirm its designed-for properties such as near conservation of energy and long term stability.

Chapter 4

Test case: Hénon–Heiles model

Here, we take one of the examples / tests from “Geometric Numerical Integration,” by E. Hairer, C. Lubich, and G. Wanner [3]. Particularly, the Hénon–Heiles model was described as a good test. It requires running the simulation for a long period of time, allowing its energy conservation property to be tested. Energy was even shown to be poorly conserved in the chaotic case by the symplectic Störmer–Verlet method! The explicit Euler method, when applied to it, was shown to turn orderly motion into chaotic ones (Poincaré section shows evidence of chaotic motion when there should not be); and the implicit Euler method was shown to turn chaotic motion into orderly ones (Poincaré section demonstrates less chaotic motion than is the case).

The Hénon–Heiles model is described by the Lagrangian

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\dot{\mathbf{q}}) - U(\mathbf{q}) \quad (4.1)$$

$$T(\dot{\mathbf{q}}) = \frac{1}{2}(\dot{q}_1^2 + \dot{q}_2^2) \quad (4.2)$$

$$U(\mathbf{q}) = \frac{1}{2}(q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3}q_2^3. \quad (4.3)$$

Poincaré sections are produced for $E = 1/12$ (Figure 4.1), which should be nonchaotic, and $E = 1/8$ (Figure 4.2), which should provide numerical evidence of chaos. Our Lagrangian solver applied here is of order $N = 12$, prescribed with a relative error tolerance of 10^{-8} . In all case, energy was well conserved. Even in the chaotic case, where a symplectic solver experienced difficulty, the absolute error in energy remained less than 1.8×10^{-6} . Of course, if needed, the error tolerance level in our Lagrangian solver can always be lowered for higher accuracy. The sections produced by Hairer et al (not shown here) shows no visible differences to ours.

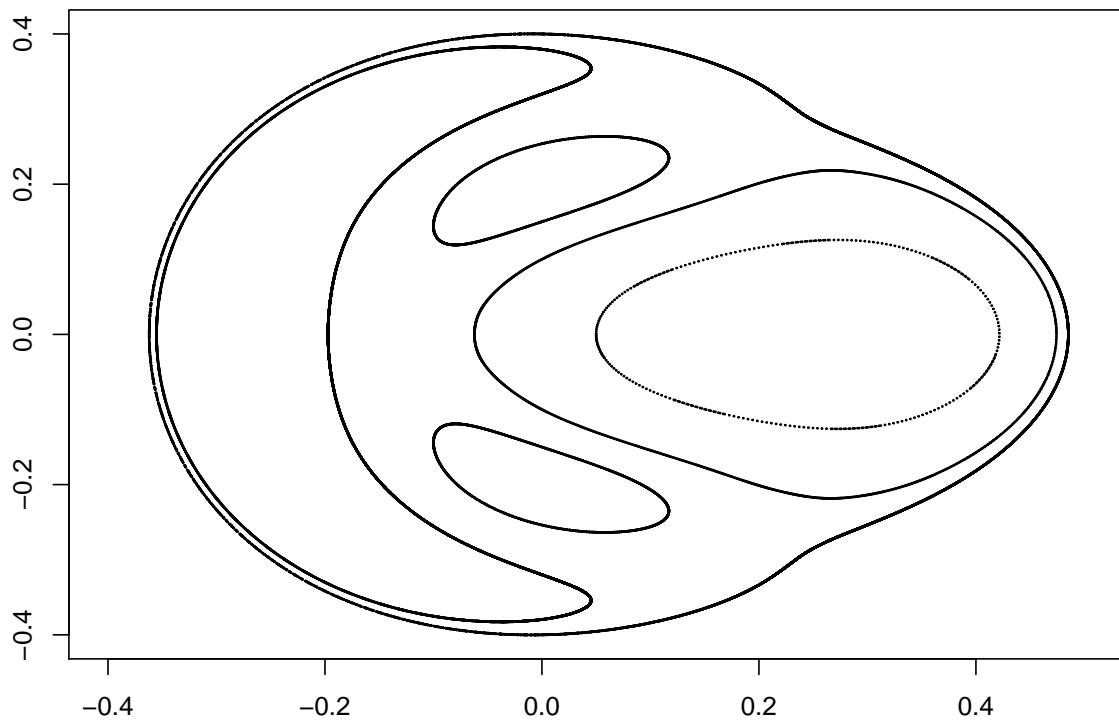


Figure 4.1: The Poincaré section of the Hénon–Heiles model with $E = 1/12$ for $q_1 = 0$ and $\dot{q}_1 > 0$. The x - and y -axes correspond to q_2 and \dot{q}_2 . This is the result of 6 orbits. A relative error tolerance at each timestep of 10^{-8} was prescribed. Energy was conserved to within 6.9×10^{-7} .

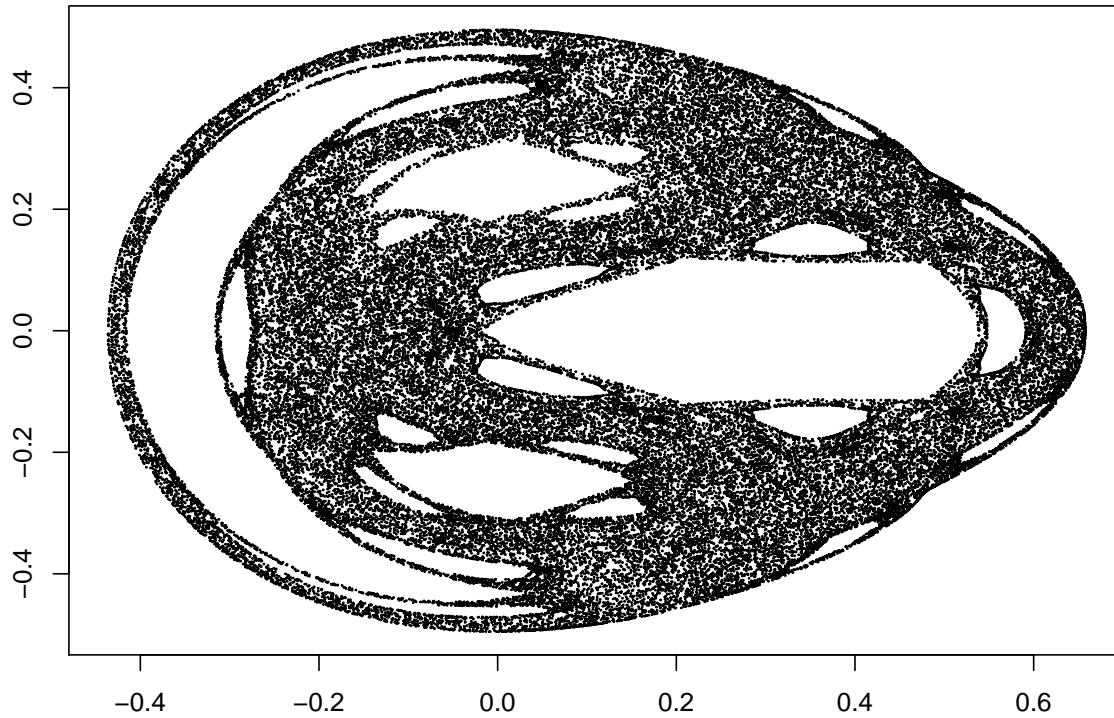


Figure 4.2: The Poincaré section of the Hénon–Heiles model with $E = 1/8$ for $q_1 = 0$ and $\dot{q}_1 > 0$. The x - and y -axes correspond to q_2 and \dot{q}_2 . This is 1 orbit, resulting from running the simulation for $0 \leq t \leq 400,000$. A relative error tolerance at each timestep of 10^{-8} was prescribed. Energy was conserved to within 1.8×10^{-6} .

Chapter 5

A chain of three magnetized, steel balls

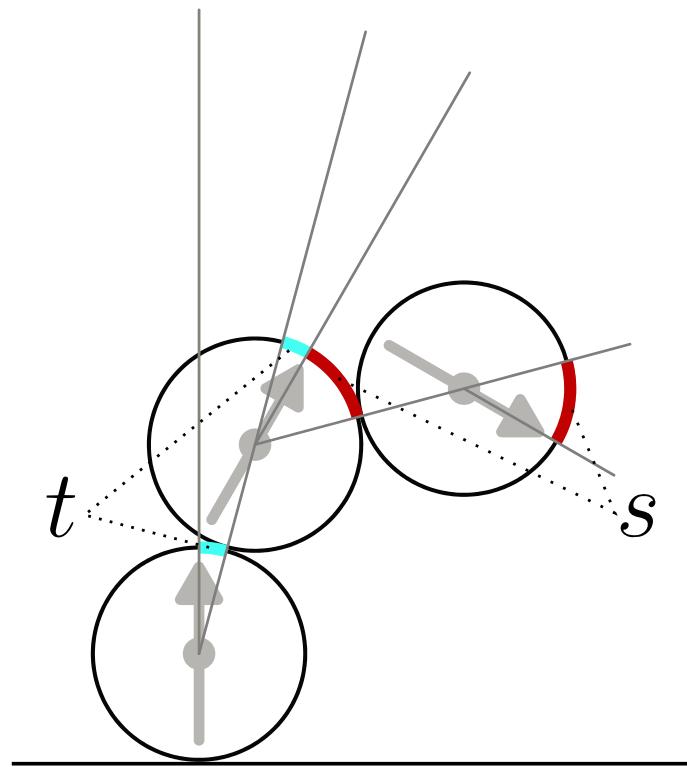


Figure 5.1: Sketch of the system illustrating the two angles t and s .

In this chapter, the Lagrangian is derived for the system depicted by Figure 5.1. Three identical, homogeneously magnetized balls with mass M , diameter d , and magnetic flux density B are stacked to form a chain, with the bottom ball fixed in place. Initially, the magnetic moment of all three balls is pointing upward. Further, neighboring balls are subject to a no-slip condition and we consider the planar problem. Therefore, since the balls are only able to roll on each other, the system has only two degrees of freedom, represented by the angles t and s , see Figure 5.1. The corresponding angular velocities are denoted by $\omega_t = \dot{t}$ and $\omega_s = \dot{s}$.

5.1 Kinematics

We denote the center position of ball i with \mathbf{x}_i , $i = 1, 2, 3$. Since neighboring balls are subject to the no-slip condition we obtain the following relations

$$\mathbf{x}_1 = (0, 0), \quad (5.1)$$

$$\mathbf{x}_2 = d(\sin t, \cos t), \text{ and} \quad (5.2)$$

$$\mathbf{x}_3 = d(\sin t + \sin(2t + s), \cos t + \cos(2t + s)). \quad (5.3)$$

The velocities $\mathbf{v}_i = \dot{\mathbf{x}}_i$ are therefore

$$\mathbf{v}_1 = (0, 0), \quad (5.4)$$

$$\mathbf{v}_2 = d\omega_t(\cos t, -\sin t), \text{ and} \quad (5.5)$$

$$\mathbf{v}_3 = d\omega_t(\cos t, -\sin t) + d(2\omega_t + \omega_s)(\cos(2t + s), -\sin(2t + s)). \quad (5.6)$$

The magnetic dipole moments \mathbf{m}_i , $i = 1, 2, 3$ of the balls contain the orientational information and are given as

$$\mathbf{m}_1 = m(0, 1), \quad (5.7)$$

$$\mathbf{m}_2 = m(\sin(2t), \cos(2t)), \text{ and} \quad (5.8)$$

$$\mathbf{m}_3 = m(\sin(2t + 2s), \cos(2t + 2s)), \quad (5.9)$$

where m is the magnitude of the magnetic dipole moments.

5.2 Kinetic energy

The total kinetic energy is composed of two parts: translational and rotational. Using (5.4–5.6), the translational part simplifies to

$$\begin{aligned} KE_{\text{trans}} &= \frac{1}{2}M \sum_{k=1}^3 \mathbf{v}_k^2 \\ &= \frac{1}{2}M(\mathbf{v}_2^2 + \mathbf{v}_3^2) \\ &= \frac{1}{2}Md^2[2\omega_t^2 + (2\omega_t + \omega_s)^2 + 2\omega_t(2\omega_t + \omega_s)\cos(t + s)]. \end{aligned} \quad (5.10)$$

For the rotational part, note that from (5.7–5.9) we can directly read off the rotational velocities of the balls 2 and 3 as being $\omega_2 = 2\omega_t$ and $\omega_3 = 2\omega_t + 2\omega_s$. The moment of inertia for a solid sphere of mass M and diameter d rotating about its center is

$$I_{\text{sphere}} = \frac{1}{10}Md^2. \quad (5.11)$$

Therefore, the rotational kinetic energy of our system is

$$\begin{aligned} KE_{\text{rot}} &= \frac{1}{2}I_{\text{sphere}}(\omega_2^2 + \omega_3^2) \\ &= I_{\text{sphere}}(4\omega_t^2 + 4\omega_t\omega_s + 2\omega_s^2). \end{aligned} \quad (5.12)$$

The total kinetic energy is thus

$$\begin{aligned} KE &= KE_{\text{trans}} + KE_{\text{rot}} \\ &= Md^2 \left[\left(\frac{17}{5} + 2\cos(t + s) \right) \omega_t^2 + \frac{7}{10}\omega_s^2 + \left(\frac{12}{5} + \cos(t + s) \right) \omega_t\omega_s \right]. \end{aligned} \quad (5.13)$$

5.3 Potential energy

The vector directed from the center of ball j to the center of ball i is given by

$$\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j, \quad (5.14)$$

with $r_{ij} = |\mathbf{r}_{ij}|$ being the associated distance. Since each ball i has a spatially uniform magnetic flux density B , its outer magnetic field is the one of a point dipole and is specified by the magnetic dipole moment \mathbf{m}_i . The total potential energy PE of the system takes into account the dipole-dipole interactions between each pair of balls and the influence of the external gravitational field. With the magnetic constant μ_0 , this energy is given by [9]

$$PE = \frac{\mu_0}{4\pi} \sum_{i < j} \frac{(\mathbf{m}_i \cdot \mathbf{m}_j)r_{ij}^2 - 3(\mathbf{m}_i \cdot \mathbf{r}_{ij})(\mathbf{m}_j \cdot \mathbf{r}_{ij})}{r_{ij}^5} + Mg \mathbf{e} \cdot \sum_i \mathbf{x}_i, \quad (5.15)$$

where $\mathbf{e} = (0, 1)$ denotes the unit vector directed upward antiparallel to the gravitational acceleration, which is of magnitude $g > 0$.

5.4 Dimensionless energies

We now transform the system into a dimensionless form to make it suitable for a numerical investigation. We measure lengths relative to d , times relative to $\sqrt{d/g}$, magnetic dipole moments relative to $\pi Bd^3/6\mu_0$, and energies relative to Mgd . We further introduce the dimensionless parameter

$$\alpha = \frac{B^2}{4\mu_0\rho gd}. \quad (5.16)$$

which represents the competition between magnetic and gravitational forces (ρ is the uniform mass density of the balls).

With (5.13), the dimensionless kinetic energy then reads

$$T = \left(\frac{17}{5} + 2 \cos(t+s) \right) \omega_t^2 + \frac{7}{10} \omega_s^2 + \left(\frac{12}{5} + \cos(t+s) \right) \omega_t \omega_s. \quad (5.17)$$

From (5.1–5.3), (5.7–5.9), and (5.14–5.15) we can calculate the dimensionless potential energy to be

$$U = -\frac{\alpha}{12} [M_{12} + M_{23} + M_{13}] + G_2 + G_3, \quad (5.18)$$

with

$$M_{12} = \cos(2t) + 3,$$

$$M_{23} = \cos(2s) + 3,$$

$$M_{13} = \left[3 \cos\left(\frac{t+3s}{2}\right) \cos\left(\frac{3t+s}{2}\right) - \cos(2t+2s) \right] \Big/ \left[4 \cos^3\left(\frac{t+s}{2}\right) \right],$$

$$G_2 = \cos t,$$

$$G_3 = \cos t + \cos(2t+s).$$

Chapter 6

Results

6.1 Survey of the energy landscape

To recap, the system consists of three magnetized, steel balls stacked on top of each other with the position of the bottom ball (ball 1) fixed and the other balls restricted to planar motion. In this set-up, the magnetic forces act to stabilize the vertical chain of steel balls while gravity pulls on them and acts as a destabilizing force. It is the influence, on Poincaré plots, of this competition between stabilizing and destabilizing forces that we wish to probe.

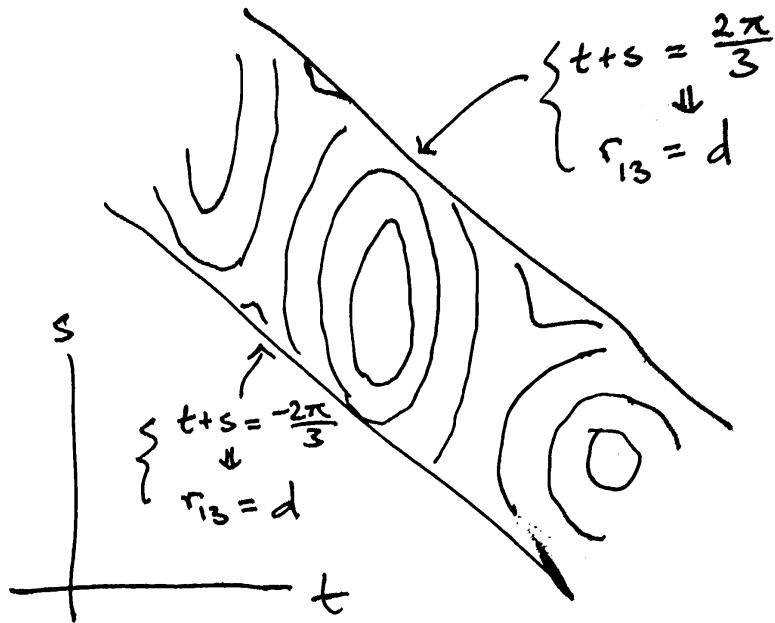


Figure 6.1: Schematic for understanding the energy landscape.

The potential energy landscapes are shown in Figures 6.2–6.11. To understand the plots, first take a look at Figure 6.1. The x - and y -axes correspond to the t and s coordinates, respectively. The upper-right boundary and the lower-left boundary correspond to where $r_{13} = d$, that is, where the first and third balls collide. The model for the three magnetized, steel balls, as described in Chapter 5, does not include any term mimicking contact. As such, the model allows for balls 1 and 3 to overlap and it is not clear, from the potential energy plots, where balls 1 and 3 collide. For this reason, Figures 6.2–6.11 are cut off at the upper-right and lower-left boundaries, which correspond to $r_{13} = d$. For the production of the Poincaré plots, we purposely ignore cases where the system has the total energy necessary to allow balls 1 and 3 to collide.

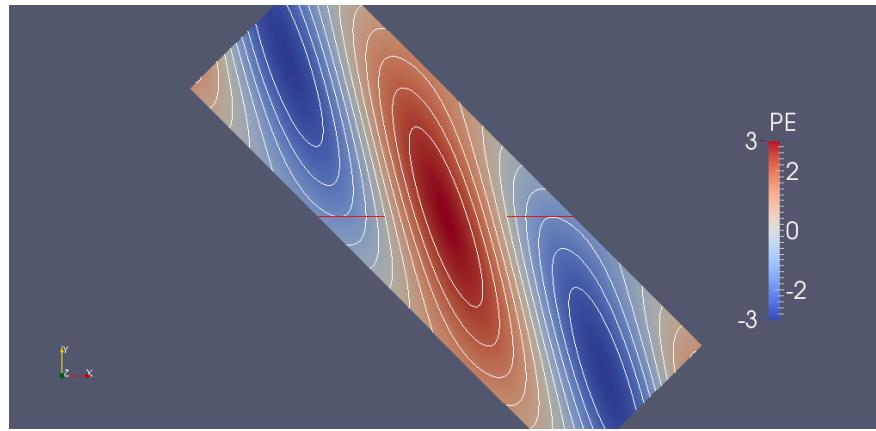


Figure 6.2: Potential energy landscape for $\alpha = 0$. The center of the plot corresponds to $(t, s) = (0, 0)$.

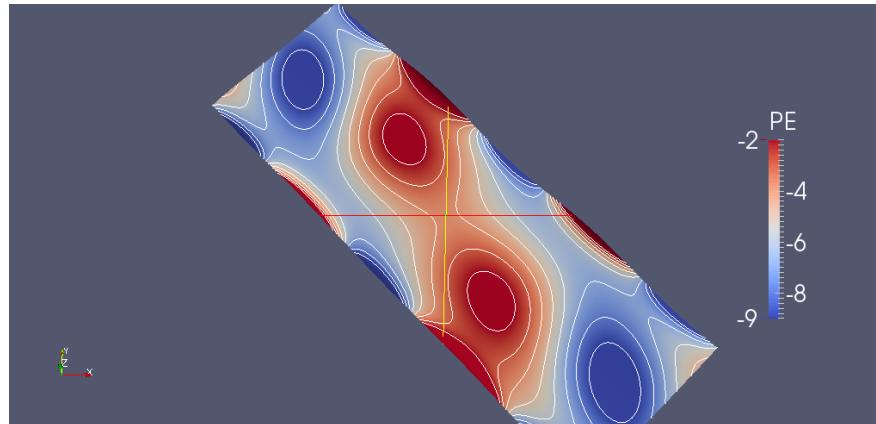


Figure 6.3: Potential energy landscape for $\alpha = 10$. The center of the plot corresponds to $(t, s) = (0, 0)$.

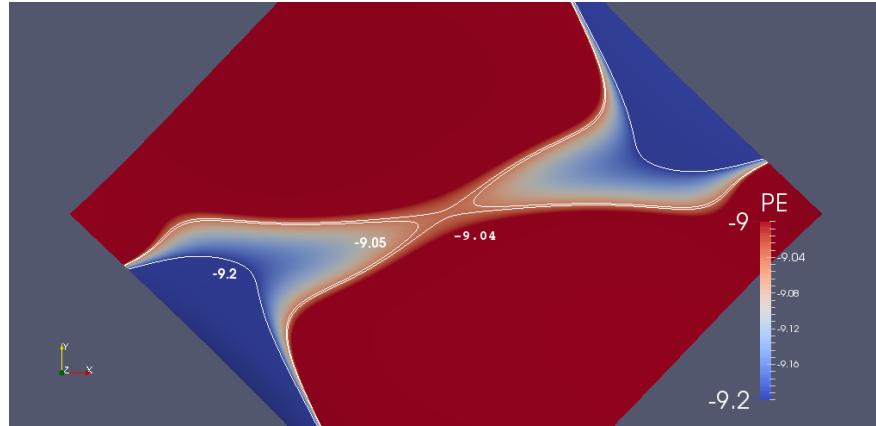


Figure 6.4: Potential energy landscape for $\alpha = 17$. The center of the plot corresponds to $(t, s) = (0, 0)$.

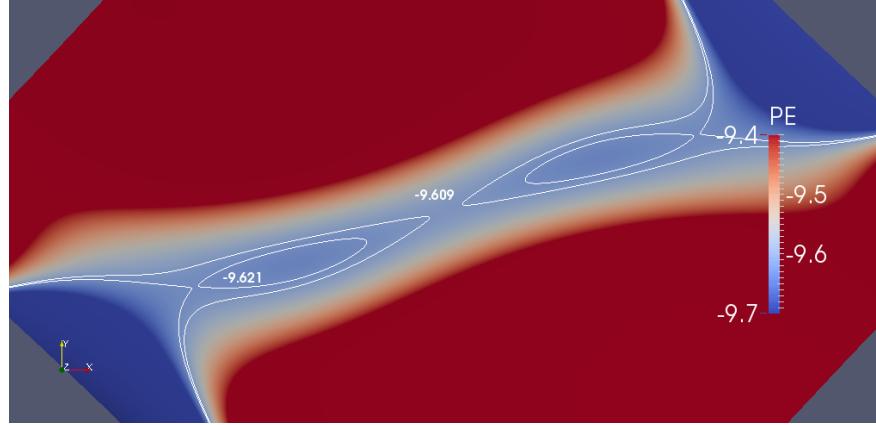


Figure 6.5: Potential energy landscape for $\alpha = 17.8$. The center of the plot corresponds to $(t, s) = (0, 0)$.

In analyzing the pictures, we notice three regimes: $\alpha \lesssim 17.8$, see Figures 6.2–6.5, where no local minimum is accessible from origin without a energy level able of reaching the $r_{13} \leq d$ region; $17.9 \lesssim \alpha \lesssim 18.4$, see Figures 6.6–6.8, where the origin $(t, s) = (0, 0)$ is not a local minimum but sits at the barrier connecting two local minima and is such that the system is able to move between the wells without acquiring sufficient kinetic energy to reach the $r_{13} \leq d$ region; and $18.5 \lesssim \alpha$, see Figures 6.9–6.11, where the origin is located in a local minimum.

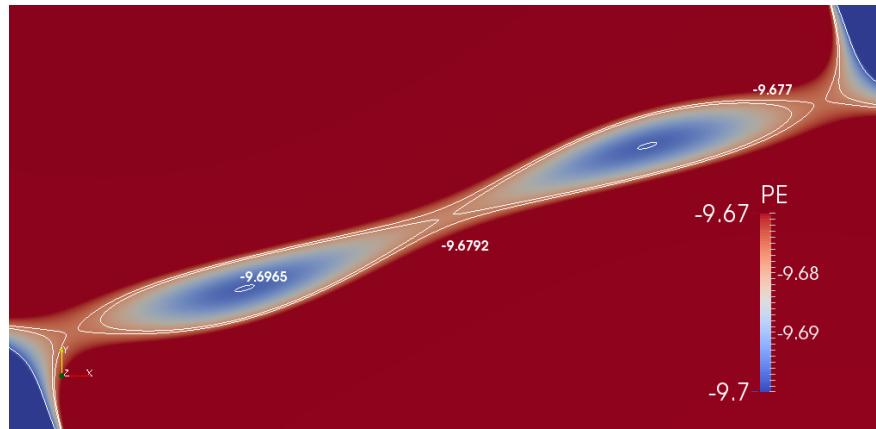


Figure 6.6: Potential energy landscape for $\alpha = 17.9$. The center of the plot corresponds to $(t, s) = (0, 0)$.

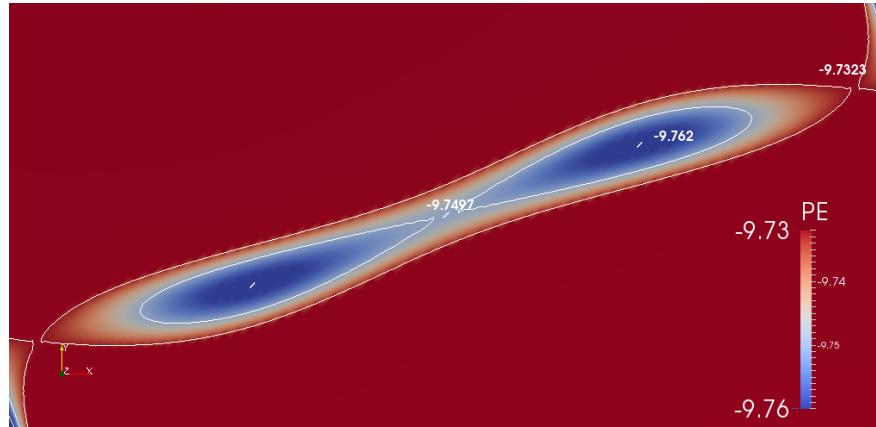


Figure 6.7: Potential energy landscape for $\alpha = 18$. The center of the plot corresponds to $(t, s) = (0, 0)$.

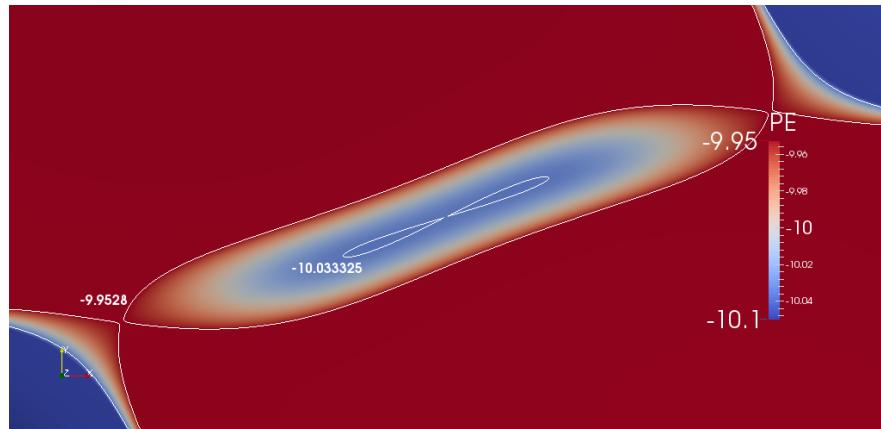


Figure 6.8: Potential energy landscape for $\alpha = 18.4$. The center of the plot corresponds to $(t, s) = (0, 0)$.

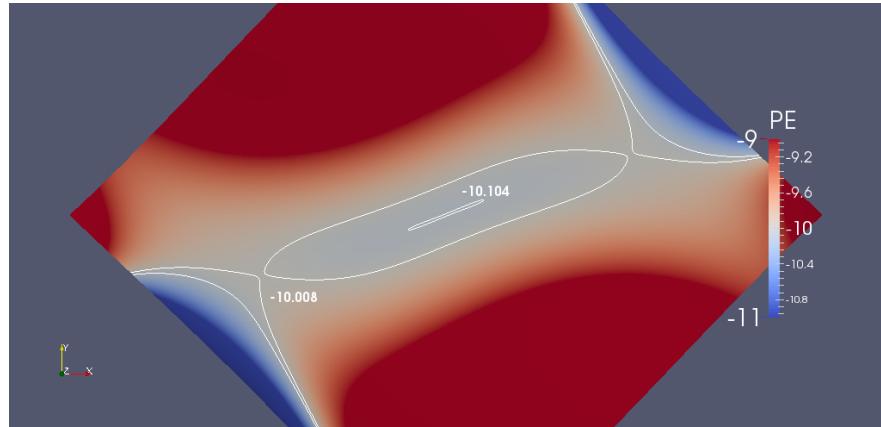


Figure 6.9: Potential energy landscape for $\alpha = 18.5$. The center of the plot corresponds to $(t, s) = (0, 0)$.

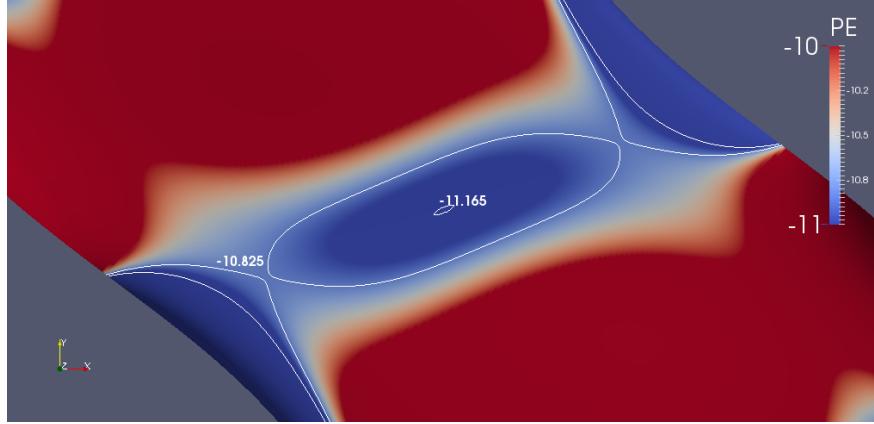


Figure 6.10: Potential energy landscape for $\alpha = 20$. The center of the plot corresponds to $(t, s) = (0, 0)$.

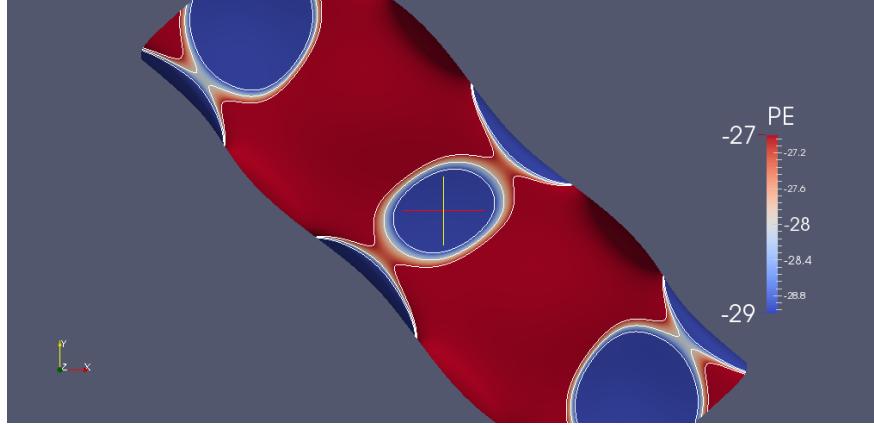


Figure 6.11: Potential energy landscape for $\alpha = 50$. The center of the plot corresponds to $(t, s) = (0, 0)$.

6.2 Results

We do not produce any Poincaré plots with the section defined at $t = 0$ for $\alpha \lesssim 17.8$ since in that regime, gravity dominates and the system is unstable. While there do exist cases where the system can be stable with $\alpha \lesssim 17.8$ (imagine the entire chain rotating like a rod about the center of ball 1 which corresponds to $\omega_t = \text{constant}$ and $\omega_s = 0$), the energy required would mean that the kinetic energy involved dwarfs the potential energy. That would essentially preclude dynamics showcasing the magnetic force and gravitational force competition.

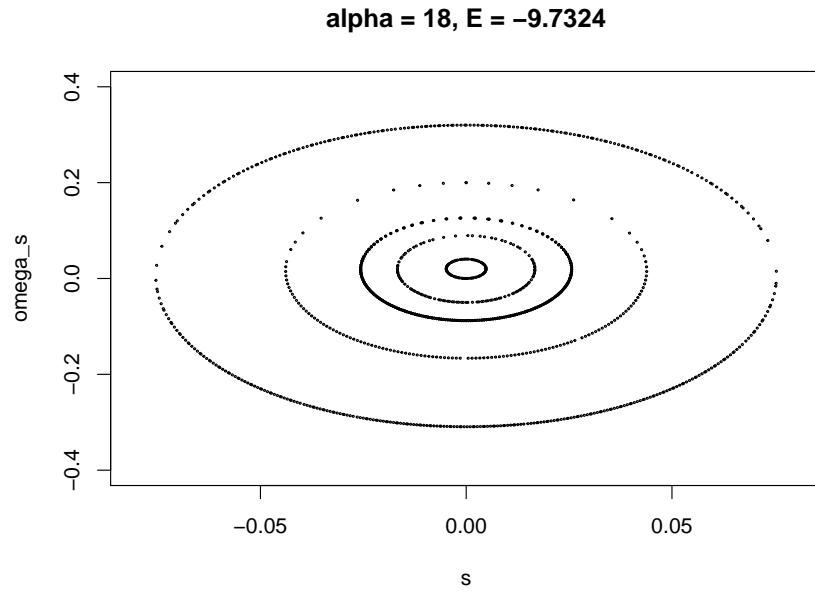


Figure 6.12: Poincaré section for $\alpha = 18$ and $E = -9.7324$, at $t = 0$ and $p_t > 0$, where p_t is the canonical momentum of the coordinate t . The x - and y -axes correspond to s and ω_s , respectively.

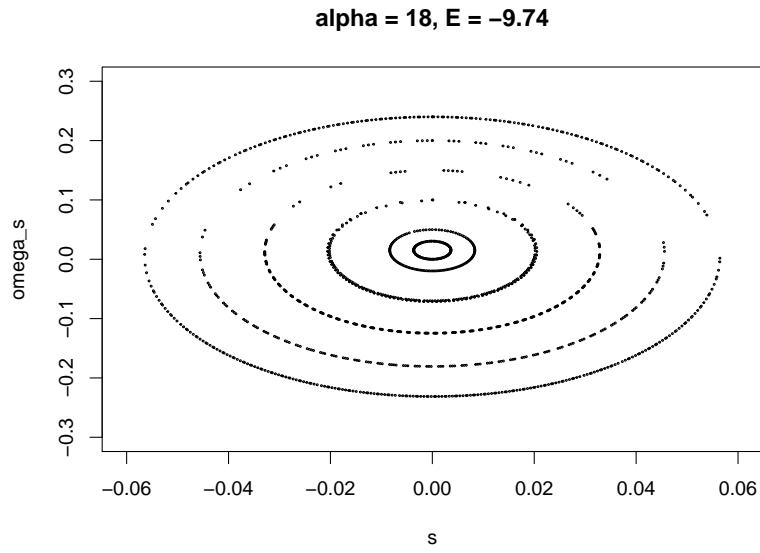


Figure 6.13: Poincaré section for $\alpha = 18$ and $E = -9.74$, at $t = 0$ and $p_t > 0$, where p_t is the canonical momentum of the coordinate t . The x - and y -axes correspond to s and ω_s , respectively.

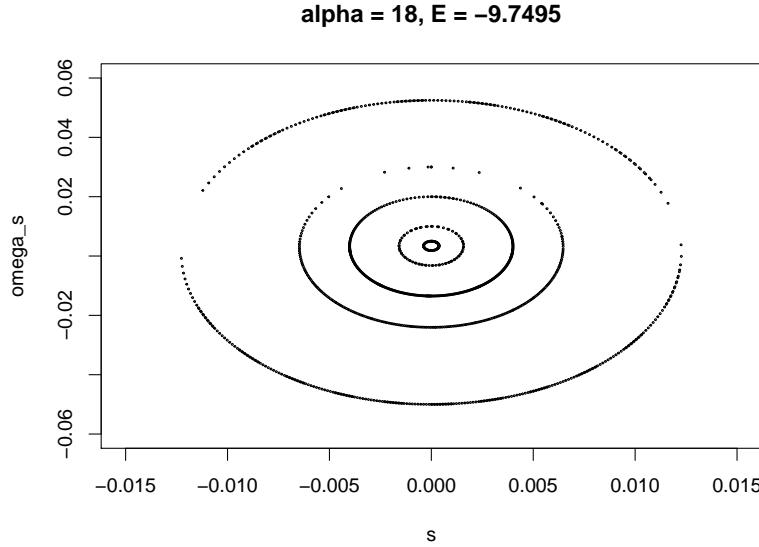


Figure 6.14: Poincaré section for $\alpha = 18$ and $E = -9.7495$, at $t = 0$ and $p_t > 0$, where p_t is the canonical momentum of the coordinate t . The x - and y -axes correspond to s and ω_s , respectively.

For $17.9 \lesssim \alpha \lesssim 18.4$, we find an interesting interplay between the magnetic and gravitational forces. To investigate this interplay, we look specifically at the case where $\alpha = 18$, see Figures 6.12–6.14, assuming its dynamics to be representative of the others in the regime. From the plot of the potential energy, Figure 6.7, we see that there are two numbers important here: the energy where the region $r_{13} \leq d$ becomes energetically accessible, $E_{\max} \approx -9.7323$; and the energy level of the barrier separating the two potential wells, $E_{\text{barrier}} \approx -9.7497$. Only systems with total energy between E_{barrier} and E_{\max} is capable of traversing between the two wells while avoiding acquiring enough kinetic energy for the first and third balls to collide. We produce three Poincaré plots: one with $E = -9.7324 \lesssim E_{\max}$; one with $E = -9.7495 \gtrsim E_{\text{barrier}}$; and one with $E = -9.74$, approximately the average of E_{\max} and E_{barrier} . These three plots span the entirety of energy levels of interest. In all three, the dynamics of t and s over time for $\alpha = 18$ and $E = -9.74$ is shown in Figures 6.15–6.16.

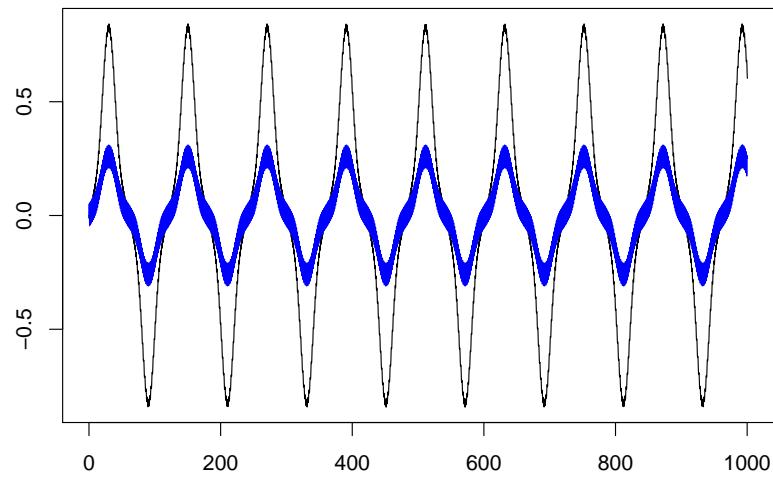


Figure 6.15: A plot of t (black) and s (blue) vs time for $\alpha = 18$ and $E = -9.74$, with intial conditions $t = 0$, $s = 0$, $\omega_s = 0.24$, and ω_t such that $E = -9.74$ and $\omega_t > 0$.

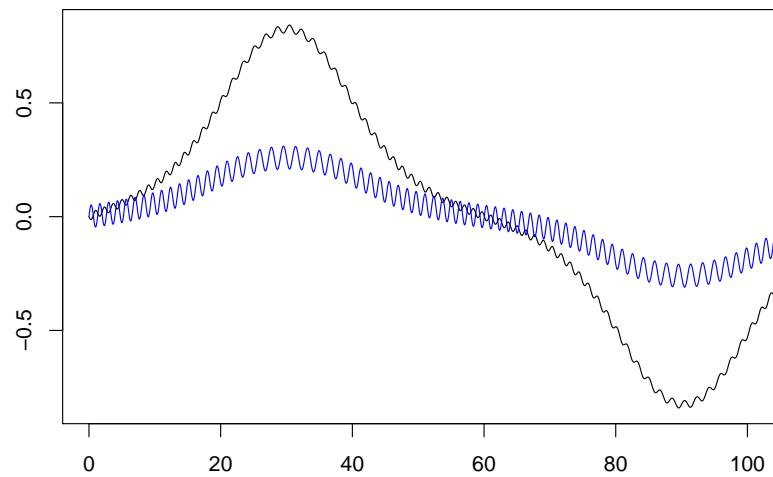


Figure 6.16: A closer view of the preview plot. Again, t is black and s is blue. $\alpha = 18$ and $E = -9.74$, with intial conditions $t = 0$, $s = 0$, $\omega_s = 0.24$, and ω_t such that $E = -9.74$ and $\omega_t > 0$.

For $18.5 \lesssim \alpha$, the stabilizing, magnetic force dominates and the system acquires a local minimum. For large enough kinetic energies, the system can be kicked out of the local minimum. However, as that would require the system to acquire sufficient energy to access the $r_{13} \leq 1$ region, we explicitly ignore such cases as the kinetic energy required would dwarf the potential energy, preventing any interplay between the magnetic and gravitational forces from showing up clearly. For this regime, we look at $\alpha = 50$, see Figure 6.17. No sign of chaos was detected here either.

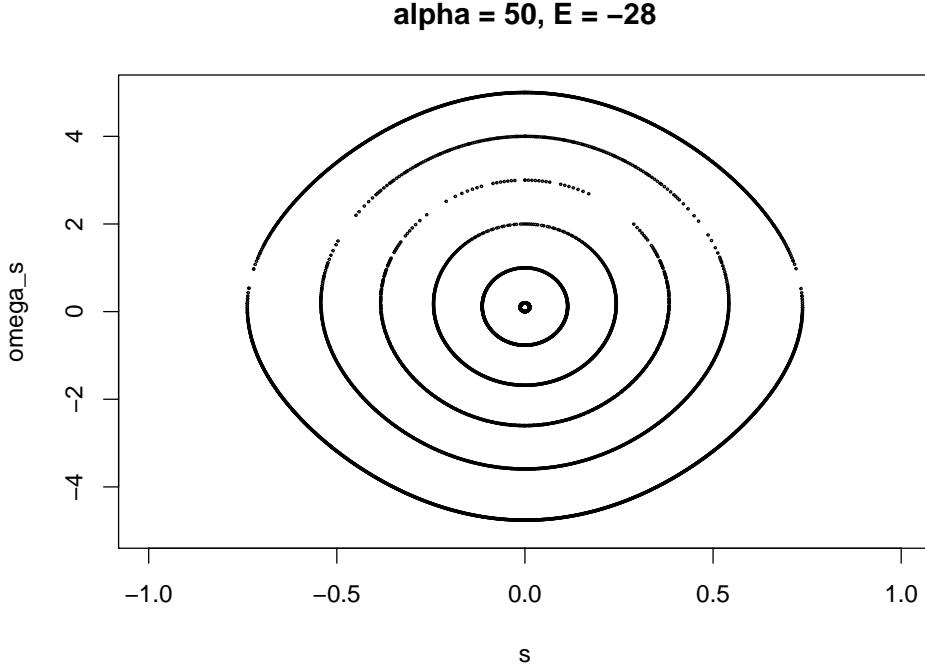


Figure 6.17: Poincaré section for $\alpha = 50$ and $E = -28$, at $t = 0$ and $p_t > 0$, where p_t is the canonical momentum of the coordinate t . The x - and y -axes correspond to s and ω_s , respectively.

6.3 Discussion and conclusion

For all α and energies we analyzed, the resulting Poincaré plots show no sign of chaos. This might be perhaps puzzling, giving that the related double pendulum system is both chaotic and a trove of interesting dynamics.

Here, we offer a potential reason the lack of interesting dynamics: the system here is highly constrained; and its range of motion, highly restricted. For the purpose of exploring

the interplay between the stabilizing, magnetic force and the destabilizing, gravitational force and to avoid balls 1 and 3 from colliding, we were restricted to exploring cases where energy is relatively low. To support this claim, we additionally analyzed the system with initial conditions $(t_0, s_0) = (\pi, -\pi)$. In this case, instead of the balls stacked on top of each other in a vertical chain, the balls hang from each other in a vertical chain, again with the position of ball 1 fixed. A Poincaré plot of this scenario, for $\alpha = 10$ and $E = -6.5$, is shown in Figures 6.18–6.20. The plot clearly demonstrates evidence of chaos despite, or because of, the lack of competition between stabilizing and destabilizing forces; both the magnetic force and gravitational force are stabilizing here.

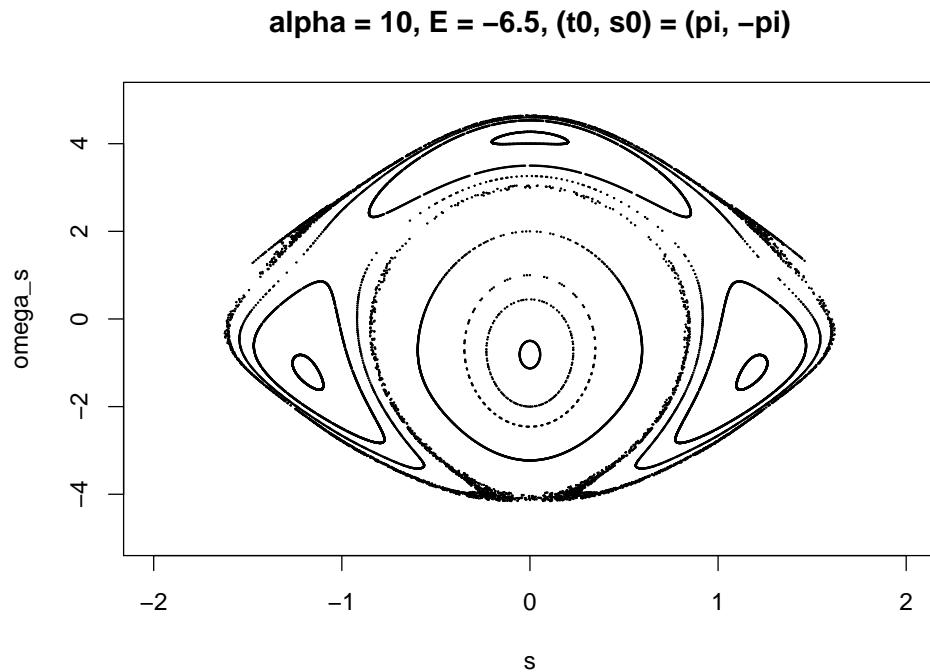


Figure 6.18: Poincaré section for $\alpha = 10$ and $E = -6.5$, at $t = \pi$ and $p_t > 0$, where p_t is the canonical momentum of the coordinate t . The x - and y -axes correspond to s and ω_s , respectively. All trajectories here start at $(t_0, s_0) = (\pi, -\pi)$.

alpha = 10, E = -6.5, (t0, s0) = (pi, -pi)

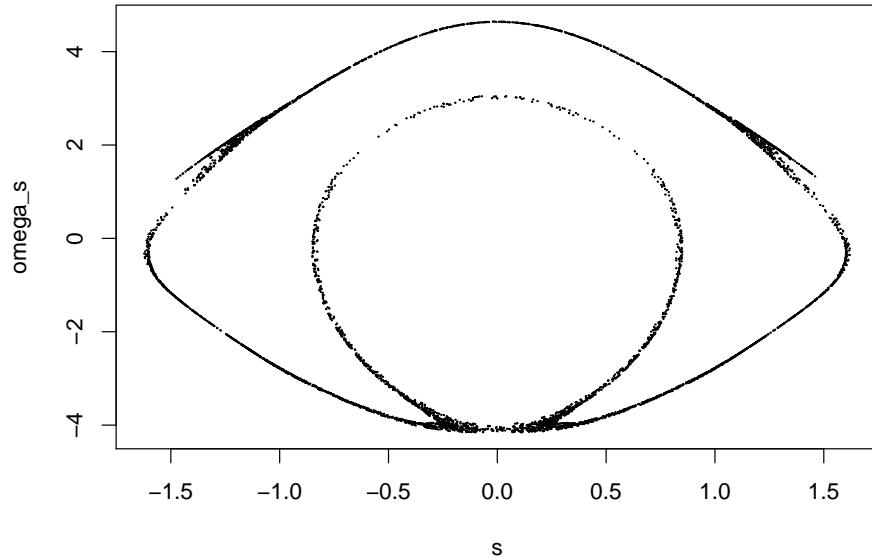


Figure 6.19: Isolated the trajectory with evidence of chaos from Figure 6.18.

alpha = 10, E = -6.5, (t0, s0) = (pi, -pi)

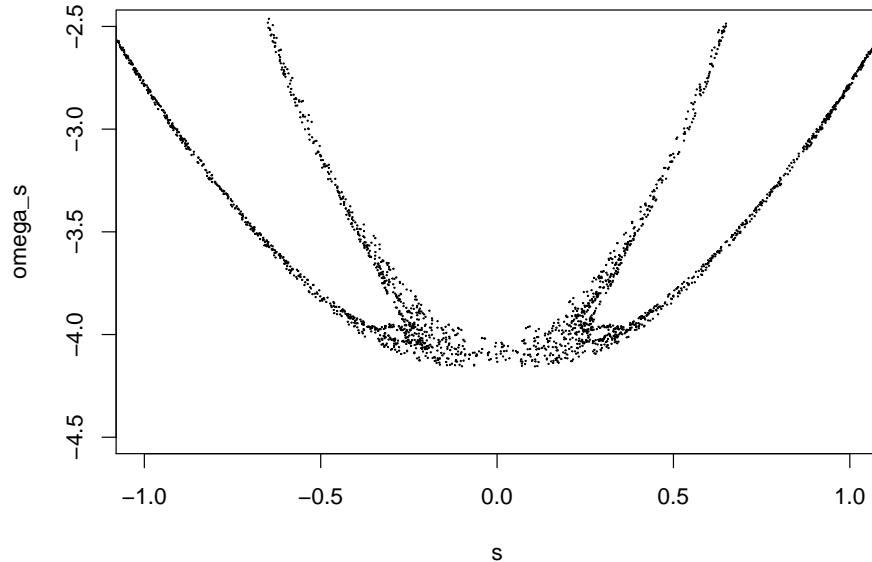


Figure 6.20: A further upclose view of the trajectory with evidence of chaos from Figure 6.18.

While it is not yet clear as to why this system is devoid of chaos, we believe that is also because the range of motion of the steel balls system is inherently restricted. Even if we were to allow balls 1 and 3 to overlap, we cannot allow their centers to overlap (unlike how the bottom point mass of the double pendulum is allowed to overlap with the origin). This is because the magnetic potential energy from the interaction of balls 1 and 3 is not unique when the separation distance is zero.

To see this, notice that the point $r_{13} = 0$ can be reached in a multitude of ways, including from the sides ($t = 0$ and $s \rightarrow \pm\pi$) and from the top or bottom ($s = 0$ and $t \rightarrow \pm\pi$). While in both cases, when r_{13} nears zero, the magnetic moment of ball 3 is essentially the same and pointed upward, the magnetic flux density they feel is different. Approaching from the sides, the magnetic moment of ball 3 is anti-parallel to the magnetic flux density resulting from ball 1; from the top or bottom, they are parallel and aligned. As it approaches from the sides, ball 3 feels an ever greater repulsive force but would feel an ever greater attractive force if it approaches from the top or bottom. Hence, the magnetic potential energy resulting from their interaction is not unique for the limiting case of r_{13} approaching zero.

As such, this leads us to further hypothesize that an analogous system with magnetized spheres of differing radii capable of passing through each other can, like the double pendulum, produce chaotic motion.

Chapter 7

Summary

For this thesis, we have achieved the multiple goals. We

1. designed, implemented, and tested a novel type of numerical solver that we named Lagrangian solvers,
2. gave a forward-mode automatic differentiation method an algebraic structure by formulating it as an associative algebra and implemented the result into our Lagrangian solver, and
3. applied our new Lagrangian solver to producing Poincaré cuts for a system of magnetized, steel balls.

By applying our implementation of automatic differentiation to our Lagrangian solver, we have produced a method that takes a single function, namely the Lagrangian, as an input. This eliminates the need for the user to derive the associated Euler–Lagrange equation and manipulate it to be in a canonical form suitable for the usual ODE solvers such as the various Runge–Kutta methods.

After the designing and building of our Lagrangian solver, we then applied it to analyzing a system of magnetized, steel balls. Since the system shows marked resemblance to the double pendulum system, we wished to understand what impact the added interplay between stabilizing and destabilizing forces would have. Disappointingly, we found that, in this particular system, the interplay between stabilizing and destabilizing forces did not lead to any particularly interesting dynamics. Instead, we found that the regime where the interplay exists to be nonchaotic. Upon further investigation, it was understood that the regime where the interplay exists restricts the motion of the system.

The work done in this thesis can be furthered in a few directions. First, the Lagrangian solver can be extended to allow for holonomic and non-holonomic constraints since the basis of the Lagrangian solver is the collocation method and differential algebraic equations (DAEs) can be formulated naturally with the collocation method. Second, the forward mode implementation of automatic differentiation here can be replaced with a reverse mode implementation as that can be far more efficient when dealing with scalar functions such as Lagrangians. Third, given that our solver only requires the Lagrangian be given and not the associated system of ODEs (or DAEs), it should be relatively easy to apply our solver to systems with Lagrangians too complicated or bothersome to consider otherwise. Fourth and last, the Lagrangian solver can be rewritten as a system of 2^{nd} -order ODEs rather than the system of 1^{st} -order ODEs (Euler–Lagrange equations supplemented by $d\boldsymbol{x}/dt = \boldsymbol{v}$). That would half the number of equations and shrink the Jacobian matrix to a quarter of its size, and so would likely represent massive gains in efficiency.

Bibliography

- [1] A. Griewank and A. Walther, *Evaluating Derivatives*, SIAM, Philadelphia, 2008.
- [2] U. Naumann, *The Art of Differentiation Computer Programs*, SIAM, Philadelphia, 2012.
- [3] E. Hairer, C. Lubich, and G. Warner, *Geometric Numerical Integration*, 2nd ed., Springer-Verlag Berlin Heidelberg, New York, 2000.
- [4] N. Kanyamee and Z. Zhang, *Comparison of a spectral collocation method and symplectic methods for Hamiltonian systems*, International Journal of Numerical Analysis and Modeling **8** (2011), no. 1, 86–104.
- [5] E. Faou, E. Hairer, and TL. Pham, *Energy conservation with non-symplectic methods: Examples and counter-examples*, Bit Numer Math **44** (2004), no. 4.
- [6] G. Meinardus, *Approximation of Functions: Theory and Numerical Methods*, Springer-Verlag Berlin Heidelberg, Berlin, 1967.
- [7] J. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover Publications, New York, 2000.
- [8] Éric Walter, *Numerical Methods and Optimization: A Consumer Guide*, Springer International, Cham, 2014.
- [9] J. Schönke and E. Fried, *Stability of vertical magnetic chains*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **473** (2017), no. 2198, available at <http://rspa.royalsocietypublishing.org/content/473/2198/20160703.full.pdf>.