UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

PRELIMINARY STUDY ON DETERMINISTIC CHAOTIC SYSTEM FROM PHYSICAL APPROACH

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

LIHUA ZHAO
Norman, Oklahoma
2019

PRELIMINARY STUDY ON DETERMINISTIC CHAOTIC SYSTEM FROM PHYSICAL APPROACH

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Ning Jiang, Chair

Dr. Ronald Barnes

Dr. S. Lakshmivarahan

Acknowledgement

First of all, I would like to express my sincere gratitude to my supervisor, Prof. John N. Jiang, for his guidance, enthusiasm and support throughout this research. I am also deeply grateful of his suggestions and comments in completion of this thesis. To my committees, Prof. Ronald Barnes and Prof. S. Lakshmivarahan, I am grateful for your assistance and instructions during my course study and thesis.

I would also like to thank all my beloved family, especially my wife, for supporting me spiritually throughout writing this thesis and my life in general. Without their love, understanding and support, I could not have finished this thesis.

My journey at the University of Oklahoma is going to the end. When I look back, there is always some people and some moments that I will not forget. I still remember the first day I came to U.S. It is not easy to start a new life in a new country. I would like to thank Dr. Lu Li, Dr. Chen Chen, Dr. Hao Ye, Dr. Yuchao Jiang, Dr. Peter Huang. It was you who made my journey a happy and easy start. I would like to thank my colleagues Dr. Zhihua Cai, Dr. Jijun Qiu, Dr. Binbin Weng, Dr. Hossein Lotfi, Dr. Robert Hinkey, Caleb Chang, Dr. Khosrow Namjou, Dr. Ghulam Hasnain, Dejan Nedin and Leonard Olona, Xinghua Shi and Dr. Joel Key and Prof. Mathew Johnson from physics department. It was nice to work with you. Regardless of success and failure in the past, you have brought me knowledge and experience, showed me a larger world that I have never known. I would also like to thank three principle investigators whom I worked with: Prof. Patrick McCann, Prof. Zhisheng Shi and Prof. Ruiqing Yang.

Within the second stage of my life at the University of Oklahoma, I've enjoyed a lot of new experience and met some new friends. I would like to thank Jeff Moore, Sue Ringus, Cassandra Rigsby, Michael Rath and Joan Kirungi from I-CCEW. Thanks for introducing me a new world. Thanks to my team members Valerie Pham, Cody Degner, Michael Fedell, Wenhua Wang, Morgan Nash, Emma Greene, William Ramataboe, Jordan D'Silva, Tanner Satterthwaite and Jake Manning, it was really wonderful to work with you.

I would like to thank Dr. Nicholas Mould, Dr. Wayne Stewart, Prof. Joseph Havlicek and Prof. Dean Hougen. I've learned a lot from you class.

I would like to thank staff of ECE, Rodney (Clint) Keele, Stephanie Gill, Nicolette Collins, Emily Benton, thanks for your kindness and help during my research and study time. Thanks to Brenda Aflleje from international student office for all of your warmly help.

I would like to thank my lab mates, Dr. Xin Li, Wanghao Fei, Guomin Ji, Gangan Li and Dr. Qiushi Bo for all the time and funs we have had during the past two years.

I would like to thank Ximeng Zu, Zhengguan Li and Yan Liang, it was really a good time of having class with you. In addition, thanks to Ximeng for all of the supports. Specially thanks for Qinqiang Zhu, it was pleasure to study with you during the tough summer time.

Finally, I would like to thank all of my friends: Ruodan Shen, Xiaofang Li, Dr. Fan Chen, Minhua Kang, Dr. Wei Wang, Jian Zhao, Yijia Zhang, Yutong Zheng, Dr. Yiming Xu, Dr. Kangmei Zhao, Yue Ban, Jiazun Li, Zijian Yuan, Qifeng Huang, Dr. Mu Yang, Dr. Lin Lei, Mu Liu, Mengnan Xie. It is nice to meet you all at Norman.

Table of Content

ACKNOWLEDGEMENT ... **IV**

TABLE OF CONTENT ... **VI**

LIST OF TABLES ... **VIII**

LIST OF FIGURES ... **IX**

ABSTRACT ... **XI**

CHAPTER 1: INTRODUCTION ... **1**

1.1 BASIC CONCEPT OF SYSTEM ... 1

1.2 BRIEF HISTORY ON DISCOVERY OF CHAOS ... 3

1.3 BASIC CONCEPT OF CHAOS SYSTEM ... 7

1.4 APPLICATION OF NONLINEAR DYNAMICS ... 11

1.5 MOTIVATION AND ORGANIZATION OF THESIS ... 12

CHAPTER 2: A VIEW ON CHAOTIC SYSTEM ... **14**

2.1 GENERAL METHODS FOR STUDYING CHAOTIC SYSTEM ... 14

2.2 PROPOSITION OF A NEW APPROACH FROM PHYSICS ... 21

CHAPTER 3: INTRODUCTION OF MAGNETIC PENDULUM SYSTEM ... **24**

3.1 BASIC CONCEPT OF THE SYSTEM ... 24

3.2 MODEL OF THE SYSTEM ... 27

CHAPTER 4: PROGRAM OF CHAOTIC MAGNETIC PENDULUM ... **33**

4.1 REQUIREMENTS ANALYSIS ... 33

4.2 CLASS DESIGN ... 36

4.3 CALCULATION ALGORITHM ... 42

4.4 PROGRAM USER INTERFACE 44

4.5 SOME DISCUSSION ABOUT IMPLEMENTATION 45

CHAPTER 5: NUMERICAL CALCULATION AND RESULTS 48

5.1 RESULT FROM THE PROGRAM 48

5.2 MAGNETIC PENDULUM STUDY BY NUMERICAL SIMULATION 50

5.3 ANALYSIS ON SINGLE TRAJECTORY 56

5.4 BASIN ANALYSIS 71

5.5 COMPARE BETWEEN TWO METHOD 78

CHAPTER 6: SUMMARY AND FUTURE WORK 80

REFERENCES 82

List of Tables

Table 1: Summary of 4 different trajectories 68

List of Figures

Figure 1.1: An example of chaotic system: Lorenz attractor 2

Figure 1.1: Henon mapping, a = 1.0, b = 0.3 10

Figure 1.2: Henon mapping, a = 1.4, b = 0.3 10

Figure 2.1: Saddle surface... 15

Figure 2.2: Schema of divergence between two trajectories 18

Figure 3.1: Schema of magnetic pendulum system 25

Figure 3.2: Magnets and pendulum distribution in two-dimensional plane 26

Figure 3.3: Contour plot of potential field of the pendulum system................................ 30

Figure 3.4: Three-dimensional surface plot of potential field 31

Figure 4.1 UML of Magnetic Pendulum program 37

Figure 4.2 Default user interface 44

Figure 5.1 Example of pendulum movement trajectory 48

Figure 5.2 Updated user interface when parameter is changed, k_f = 0.02 49

Figure 5.3 Pendulum trajectory after parameter change.. 50

Figure 5.4: Trajectories plot of fixed points 52

Figure 5.5: Boundary of epsilon neighborhood for fixed point (4.0, 4.0) 53

Figure 5.6: Boundary of epsilon neighborhood for fixed points (4.0, 6.0), (2.27, 3) and (5.73, 3).. 54

Figure 5.7: Individual plot of trajectories for different starting positions 55

Figure 5.8: Combination plot of trajectories for different positions................................ 56

Figure 5.9: Evolution of phase x and y versus time steps.. 57

Figure 5.10: Autocorrelation and partial autocorrelation of phase X and Y 58

Figure 5.11: Evolution of phase Vx and Vy versus time 59

Figure 5.12: Autocorrelation and partial autocorrelation of phase Vx and Vy 60

Figure 5.13: Evolution of phase ax and ay versus time 61

Figure 5.14: Autocorrelation and partial autocorrelation of phase ax and ay.................. 62

Figure 5.15: Power spectrum density of phase X 63

Figure 5.16: Power spectrum density of phase Y 64

Figure 5.17: Evolution of phase x and y for blue trajectory, starting from (0.5, 4.033)... 65

Figure 5.18: Autocorrelation and partial autocorrelation for phase X and Y 67

Figure 5.19: Power spectrum density of phase X 67

Figure 5.20: Power spectrum density of phase Y 68

Figure 5.21: Trajectory separation, reference trajectory: Black Trajectory...................... 69

Figure 5.22: Trajectory separation, reference trajectory: Red Trajectory 71

Figure 5.23: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.5$ and different k_f parameters 74

Figure 5.24: Basin diagram for pendulum system for km = 1, $k_f = 0.2$ and different k_g parameters 76

Figure 5.25: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.2$ and different k_f parameters 77

Figure 5.26: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.2$, $k_f = 0$ 77

Figure 5.27 Basin diagram from program and numerical simulation............................... 79

Abstract

Chaotic phenomenon widely exists in the nonlinear dynamic systems. Such phenomenon is described to be highly related the initial conditions and intrinsic system properties. Traditionally, large effort has been put on evaluation of the existence of chaos, such as Lyapunov exponent or power spectrum methods. In addition, iteration methods are usually employed to explore the states of system. In this work, we will study the chaotic system in a different point of view. The system properties of physics quantities will be considered firstly. Starting from these physics quantities, it is expected to potentially determine the final states of the system. Then a relationship between the initial conditions and final states will be established. The objective of this work is to preliminary study the feasibility of employing such method for predetermine the final state of reappear initial conditions. The main approach is building a strong relationship between each initial conditions and final states within limited computation. Once the relationship is established, we could know the system states distribution and achieve the exploitation on the system. In this work, we will focus on the deterministic chaotic systems that the final states are deterministic and non-periodic.

In order to perform the proof-of-concept, a typical chaotic system, the magnetic pendulum system, will be employed for demonstration. A program written in Java will display the pendulum movement in real-time and also the basin diagram that shows the relationship between initial conditions and final states. All of code for this program are constructed from scratch.

After the implementation of our program, we will compare results from it and that of the separate numerical simulation of the same system. The numerical simulation will be

performed through the commercially available software Mathematica. Further investigation will also be discussed with the results from numerical simulation.

Three conclusions could draw from this work. The first is a proof-of-concept has been established for the opportunity to develop a new approach to study the deterministic chaotic system based on the physical properties methods. The second is the relationship between the initial condition and final states are proved to be work in states predetermination. The third is that relationship between the intrinsic system parameter and the final system states distributions has been found, which could be a guiding line for the similar deterministic chaotic system study in the future. It is also worthy to mention that herein the magnetic pendulum is only an example to testify our approach. And this approach should also be valid for other general deterministic chaotic systems.

In this work, we only perform the preliminary study on this approach. In the future, our study will be extended to more systems and even the system in real world. It is expected such approach would build a new viewpoint on understanding the chaotic system, and a potentially new method to understand data.

Chapter 1: Introduction

Chaotic phenomenon[1-9] refers to the seemingly random irregular motion that occurs in deterministic systems. The behavior of a system described by deterministic theory is characterized by uncertainty, non-repetition and unpredictability. Further research shows that chaos is an inherent characteristic of nonlinear dynamical systems and a common phenomenon in nonlinear systems. Newton's deterministic theory can fully deal with many linear systems, and linear systems are mostly simplified form of nonlinear systems. Therefore, in real life and practical engineering problems, chaos is ubiquitous. In this chapter, we will start from the basic concept of system and introduce chaos from its history, definition and applications.

1.1 Basic Concept of System

Generally, system of any kind of engineer, physics, biology and society, whose state changes with time could be considered as dynamical system. It consists of two factors that are evolution law and initial condition. The former one is the dependency between current and foregone states. The later one is the system state at the beginning. Dynamical system could be divided into two sub-systems, which are deterministic and stochastic system. Deterministic system could be described by time dependent functions, while stochastic system could not, and is only of statistic properties. The later one usually has random initial condition, coefficient changing and external stimulation. Dynamical system could be divided into finite/infinite dimensional system, or continuous/ discrete time system, or even linear/nonlinear system according to different factors.

Linear system is the one changes by the rule of linear combination of its foregone states. On the other hand, nonlinear system is the one which changes in a complex way that could not be decomposed into the linear combination of its foregone states. It is usually described by nonlinear differential equations. Comparing with linear system, nonlinear system appears to be of complicated properties. Firstly, superposition principle of linear system is not valid in nonlinear situation. Secondly, the motion period is not simply determined by the system properties. It is also related to the initial condition. Thirdly, nonlinear system has multiple equilibrium positions and steady motion. System behaviors are related to not only stability of these factors, but also initial condition. Fourthly, response frequency of system is the same in linear system. However, this response is complicated in nonlinear system. Finally, there are only periodical and quasi-periodical motion in linear system. However, there is a complicated motion in nonlinear system, which is the chaotic motion.

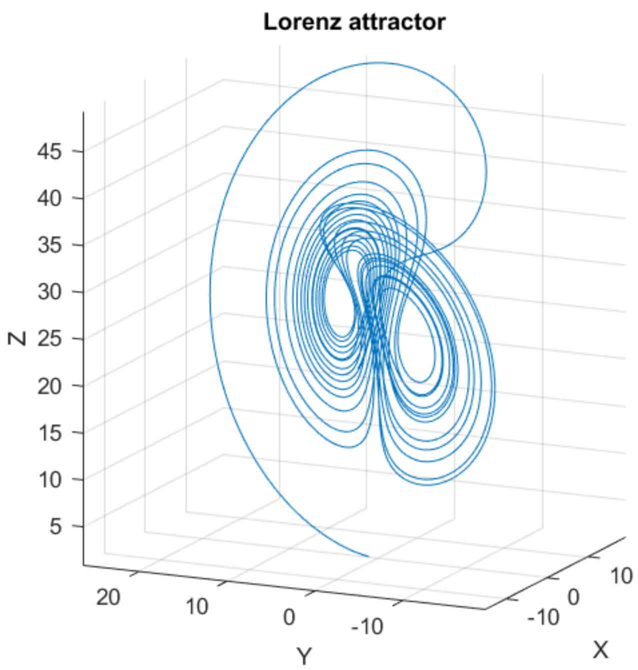


Figure 1.1: An example of chaotic system: Lorenz attractor

Nonlinear dynamics mainly studies the qualitive and quantitive transformation law of motion state in nonlinear system, especially the complex evolution in long duration. Regarding finite dimensional system, the points of interests mainly focus on chaos, bifurcation and fractal. Chaos is a kind of reciprocating non-periodic motion produced by deterministic dynamic system which is very sensitive to initial value and has inherent randomness and long-term prediction impossibility. Bifurcation refers to the qualitative behavior of dynamic systems with the change of system parameters and the qualitative changes. Fractal is a geometric structure without characteristic scale and similarity, which is used to describe the complex geometrical form of fragmentation and irregularities. Before the chaotic phenomenon was widely known, the description of nature was divided into two distinct categories, randomness and certainty. The deterministic system had the deterministic nature. The rise of chaos research has prompted people to pay attention to the problem of finiteness, because random test can only be carried out in a limited time and frequency. Opportunity, cause and effect, determinism and other basic concepts and categories of human understanding of nature need to be re-recognized. The study of nonlinear dynamics has led to the emergence and wide application of a new experimental method, which is numerical experiments.

1.2 Brief History on Discovery of Chaos

Recognition of nonlinear dynamics problem could be traced back to 1673, when C. Huygens studied the behavior of pendulum [1, 8]. He observed two kinds of nonlinear phenomena, which were the isochrone deviation of the single pendulum with large swing and the synchronization of two clocks with approached frequencies. In 1687, Isaac Newton published his great publication “Philosophiae Naturalis Principia Mathematica” [10]. In this work, Newton employed an essential method, the differential equations, for physics problem. While this mathematical tool was not new at that time, it was Newton who firstly formally started using it to describe physics phenomena.

In Newton's method, an object could be treated as a single point of mass, considering in two dimensions, its movement could be described by several equations:

$$x(t) = x(0) + \frac{dx}{dt}t \tag{1}$$

$$y(t) = y(0) + \frac{dy}{dt}t \tag{2}$$

$$\frac{dx}{dt} = v_x(t) = v_x(0) + \frac{d^2x}{dt^2}t \tag{3}$$

$$\frac{dy}{dt} = v_y(t) = v_y(0) + \frac{d^2y}{dt^2}t \tag{4}$$

Therefore:

$$x(t) = x(0) + \left(v_x(0) + \frac{d^2x}{dt^2}t\right)t \tag{5}$$

$$y(t) = y(0) + \left(v_y(0) + \frac{d^2y}{dt^2}t\right)t \tag{6}$$

where x(0), y(0), $v_x(0)$, $v_y(0)$ are called initial conditions. The solution of the two last equations is the trace of object movement.

Now let's consider a system consists of n objects, and there is interaction between any of two objects according to the law of gravitation. The interaction could be quantitively measured by the square reverse of their distance. This is the famous n-body problem. This problem could be generally described as: given the position and current velocity of astronomical objects, trying to estimate their motion status at any moment in the future or past. When n = 2, it is called 2-body problem, or Kepler problem in memory of Kepler, who discovered the rule behind Tycho Brahe's observation and stimulated Newton for the law of gravitation. In 2-body problem, in order to fully study two objects' movement in three-dimensional space, 3 position variables and 3 velocity variables are needed. Therefore, the degrees of freedom are 6. This problem might be really difficult at the first glance; however, it could be solved by introducing the varieties of law of conservation in physics. On the other hand, unfortunately, the 2-body problem is the only solvable problem in n-body problem, which means the solution to the equations could be determined under the initial conditions. When n > 2, regardless of effort has been made, the problem is not analytically solvable, the exact formulas as the solution are still unachieved [1, 8].

In 1889, in order to commemorate the 60th birthday of King Oscar II of Sweden and Norway, a contest was held to produce the best research in celestial mechanics pertaining to the stability of the solar system, which was a particularly relevant n-body problem [6]. The winner was declared to be Henri Poincare. In order to make progress on the problem, Poincare made two simplified assumptions. Firstly, he assumed that the three bodies all moving in a plane. Secondly, he also assumed that two of the bodies were massive and that the third had negligible mass in comparison, so small that it did not affect the motion of the other two. In general, the two large stars would travel in ellipses, but Poincare made another assumption, that the initial conditions were chosen such that the two moved in circles, at constant speed, circling about their combined center of mass. He discovered the crucial ideas of "stable and unstable manifolds", which are special curves in the plane.

These manifolds can cross each other, in so-called homoclinic points. The meaning of these points was unclear to him at that time. Later, he eventually realized that the existence of homoclinic points implied that there was incredibly complicated motion near those points, behavior we now call “chaotic” [1].

In 1890, Poincare published an article “On the equations of dynamic and the three-body problem” [11]. In this work, he established that due to the possibility of homoclinic crossings, no general exact formula exists, beyond Newton’s differential equations, for making predictions of the positions of the three bodies in the future. In 1892, he demonstrated the rationality of the perturbation method, which promoted the research on approximated analytical method of nonlinear dynamics. In 1905, he clearly elucidated the unpredictability caused by sensitivity to initial values. These series of works that became the beginning of the study of nonlinear systems, especially chaotic theory.

Ever since 20s of 20th century, the fundamental difference between linear and nonlinear dynamics has been recognized. G. Duffing and B. van der Pol performed research on typical nonlinear vibration system in 1918 [12] and 1926 [13], respectively, revealed some nonlinear systematic properties, such as secondary harmonic vibration and self-excited vibration. In 1929, A. Andronov built the relation between limit cycle and self-excited vibration, followed by a systematic research on the qualitative features of planar system. During 30s and 40s, N. Krylove, N Bogoliubov and Y Mitropolskii developed the nonlinear dynamics approximated analytical method [14].

The widely research on the chaotic phenomena enhanced the development of nonlinear dynamics. Regarding the physical concept of unpredictability, M. Born and L. Brillouin in 1955 and 1964 [15], respectively, expounded idea of Poincare and suggest uncertainty of classical mechanics due to the instability. On mathematical description of non-periodic, H. Morse introduced symbolic dynamics method in 1921 [16]. S. Smale constructed horseshoe mapping [17]. The generation mechanism of non-cyclical motion of near-integrable conservative system was revealed by A. Kolmogorov in 1954 [18]. Later, his conclusion was strictly proved by V. Arnold and J Moser and known as KAM theorem [19]. Development of computer provided new methods for research on chaos. A series important numerical results proved the existence of chaos, including E. Lorenz's simplified [illegible] [20], 2 degrees of freedom conservative system model of M. Henon and C. Heiles in 1964 [21], forced nonlinear vibration model by Ueda and Qianbo Lin in 1973 and Henon's two-dimension mapping model of strange attractor in 1976 [22]. The concept of strange attractor was proposed by D. Ruelle and F. Takens in 1971 [23]. In 1975, Tianyan Li and J. Yorke tried to give the mathematical definition of chaos on interval mapping [24]. In 1976, R. May's research on complex dynamics of one-dimension mapping made the chaos attract widely attention [25]. Late of 70s, chaos blended with bifurcation and fractal, which made the research of nonlinear dynamics more in-depth and extensive.

1.3 Basic Concept of Chaos System

The first definition of chaos in mathematics was introduced by Li and Yorke in 1975 [24], where the authors established a criterion on the existence of chaos for internal maps. The original expression of their definition was presented through mathematical formulas, which is difficult to read without any background of topology. Herein we will focus on the means of their definition. Considering a continuously self-mapping within a closed interval, there exists a number x satisfies the condition that $f^n(x) = x$, which means an initial value x equals to the final value after n times self-mapping. Herein we called this n the period. Li and Yorke proved that the system should be considered as chaotic if it could satisfy such conditions: first, there is no upper limit for the period n; second, there are countable number of stable orbits, uncountable number of stable trajectories, and unstable trajectories. It is worthy to mention that one obit or trajectory is the subset of self-mapping. More interesting, it has been proved that the period of system could be any integer if there exists period of 3. Hence one could say that a system with period of 3 means chaos.

The most widely accepted definition for the chaos was given by Robert L. Devaney [26]. He proposed a chaos system should possess such properties: initial value sensitivity, topology transitivity and density of cycle points. Sensitivity to the initial value means that no matter how close the two points in the space are, the distance between the two can be separated by a larger distance under the action of the remapping. Within nearby area of each point, there should exist a point satisfies such condition. For such a system, any small initial value error could lead to the failure of the computing results after many iterations if a computer is used to calculate its orbit. Topology transitivity means that the neighborhood of any point will "scatter" the entire metric space, which suggests that it is impossible to subdivide or break down into two subsystems that are not affected each other. The density of the periodic point shows that the system has a strong certainty and regularity, which is by no means chaotic, shaped in confusion and actually orderly. In a nutshell, chaotic mapping has three basic elements: unpredictability, non-decomposable, and a regular component. Because of the sensitivity to initial conditions, chaotic systems are unpredictable. Because of topological transitivity, it cannot be subdivided or cannot be broken down into two subsystems that do not interact. However, in this chaotic state, after all, there is a regular component, that is, dense periodic orbital points.

Chaotic system is a branch of nonlinear dynamic system, therefore, most of chaotic system could also be described in the same way [19, 27]. A typical one-dimensional discrete time nonlinear dynamics system is defined by the iteration formula as following:

$$x_{k+1} = \tau(x_k) \tag{7}$$

where $x_k \in V$, k = 0, 1, 2, 3…, is the state of system, $\tau: V \to V$ is a mapping.

With the determined initial condition, we could obtain a series of states for the system by applying this formula. A very simple but widely accepted dynamics system is the Logistic mapping:

$$x_{k+1} = \mu x_k(1 - x_k) \tag{8}$$

where $0 \le \mu \le 4$, and chaos domain is (0, 1).

For two-dimensional systems, the general form could be described as:

$$\begin{cases} x_{k+1} = f_1(x_k, y_k) \\ y_{k+1} = f_2(x_k, y_k) \end{cases} \tag{9}$$

where f_1 and f_2 are mapping functions. So far, a widely applied two-dimensional chaos mapping is the Henon mapping [28]:

$$\begin{cases} x_{k+1} = 1 + y_k - a x_k^2 \\ y_{k+1} = b x_k \end{cases} \tag{10}$$

It has been proved that when $a \in [1.07, 1.4]$, b = 0.3, the Henon mapping is chaotic.

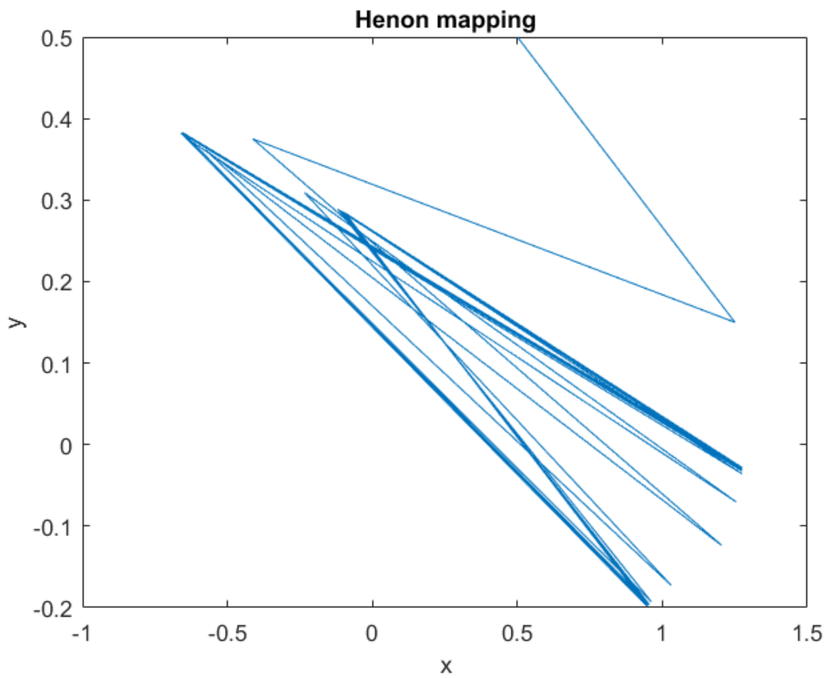


Figure 1.2: Henon mapping, a = 1.0, b = 0.3

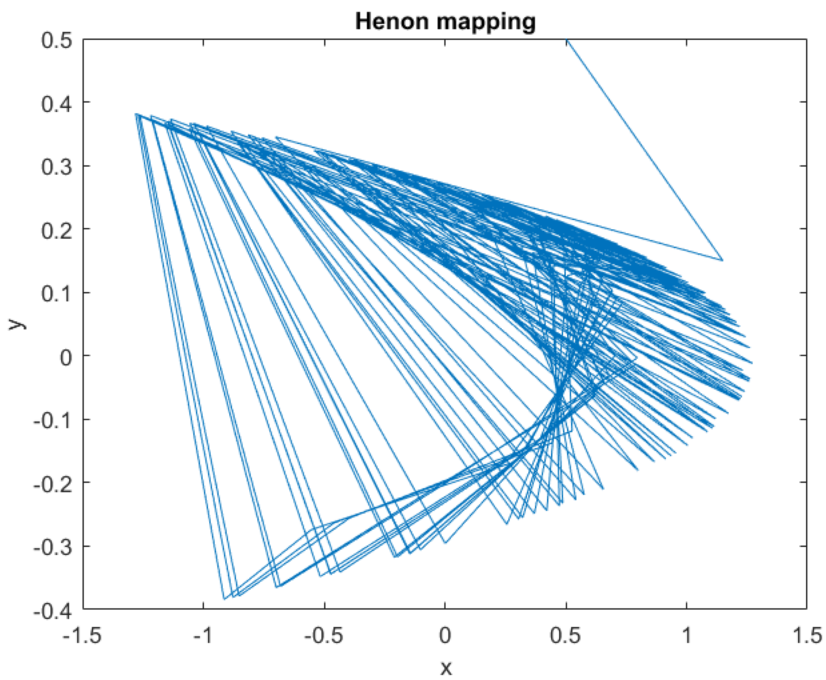


Figure 1.3: Henon mapping, a = 1.4, b = 0.3

1.4 Application of Nonlinear Dynamics

Non-linear factors, which include nonlinear forces such as electric field force, magnetic field force and universal gravitation, kinematic nonlinearity such as normal acceleration and Coriolis acceleration, material nonlinearities such as nonlinear constitutive relation, and geometric nonlinearity such as large elastic deformations, are widely existed in engineering systems [14].

Therefore, most of the problems in engineering practice should be modeled into nonlinear systems. Traditionally, linearization or equivalent linearization is used to process nonlinear systems into linear systems, but only to a certain extent. When the nonlinear factors are very strong, the results obtained by linear theory are not only too large in error, but also cannot explain some practical phenomena. As early as the 1944, the Pioneers of Modern mechanics, T von Kármán, published a review of the article "The engineer grapples with nonlinear problems" [29], which summarized the research results of nonlinear problems of various branches of mechanics at that time, and emphasized the importance of nonlinear problems in engineering. With the development of modern science and technology, the engineering structure is becoming more and more large, high-speed and complicated, so that nonlinear effects must be considered. The rapid development and widespread use of electronic computers, as well as advances in dynamic testing and online data processing technologies, have made it possible to study nonlinear problems in engineering.

Nonlinear dynamics are also playing more and more important role in the study of engineering problems. The importance of nonlinear dynamics in engineering is mainly embodied in the following aspects: nonlinear dynamics show that simple mathematical models may produce complex dynamic behaviors, so they can be applied to nonlinear modeling, prediction and control of time series. Nonlinear dynamics reveal that irregular noise signal may result from low-order deterministic nonlinear systems, thus providing a new way of thinking for noise suppression. The analysis results of nonlinear dynamics for the global and long-term state of the system can be used to study the reliability of numerical simulation results. Nonlinear dynamics also provide new concepts and methods for experimental research, and some characteristic values for identifying mixed motion can be measured in addition to traditional spectrum analysis.

The nonlinear dynamics problems in engineering vary widely, but the solutions are often common. The common premise is to establish a mathematical model of the system. The methods of establishing mathematical models can be divided into two categories. One is theoretical model. From the known principles, conclusions and theorems, the intrinsic dynamics of engineering problems are found through mechanism analysis, and the analytical relationship of related parameters is derived. The other type is the experimental model, which directly identifies the relationship of the parameters involved from the engineering system operation and test data. Based on the mathematical model of the engineering system, the system can be analyzed, simulated, optimized and controlled. Nonlinear dynamics, as a branch of general mechanics, focuses on the analysis of system models, but the experimental modeling of the system is also slightly involved.

1.5 Motivation and Organization of Thesis

In the study related to dynamic system, such as power grid system and financial stock market, possible chaotic phenomenon would be inevitable. Inspired by the previous work, we propose a new approach to study the chaotic systems, which is from the intrinsic system physical properties. In Chapter 2, the fully explanation of our idea and methodology will be introduced. A brief review on the traditional methods are also made. In Chapter 3, the experimental system will be described. The model and details of experimental magnetic pendulum system will be presented. In Chapter 4, a program writing in Java will be started from scratch. The program will be capable of showing basin diagram and the pendulum movement in real time, which will provide a straightforward impression. In Chapter 5, results from our program and numerical simulation will be shown and compared. With data from numerical simulation, a further investigation of the experimental system will be discussed. Chapter 6 summarizes this thesis and talks about some future work.

Chapter 2: A View on Chaotic System

The world around us is full of phenomena that seem irregular and random in both space and time. Exploring the each of these phenomena is almost impossible. Therefore, in addition to the traditional methods to explore the chaotic system, in this work, we introduce a preliminary new approach to view it. In this chapter, we will start with talking about the traditional method to study the chaotic system and then continue with our discussion on the new approach.

2.1 General Methods for Studying Chaotic System

As we have mentioned in Chapter 1, there could be a periodic point in the chaotic system, such point could be described as $x_m = \tau(x_k) = x_k$, where the period of x_k is m-k. There are also some points that remains the same in each iteration, which is $x_{k+1} = \tau(x_k) = x_k$, and k could be any integer. Points of such property are mentioned as fixed point in the system.

There are three types of fixed points based on their properties, which are the sinks, sources and saddles [1]. A sink is a fixed point that attracts an epsilon neighborhood of initial values. A source is a fixed point that repels a neighborhood. And the epsilon neighborhood is the set of points within certain Euclidean distance epsilon. One could understand the sinks and sources as the valleys and peaks. From another point of view, a sink is a stable equilibrium point while a source is unstable equilibrium point. The third kind of fixed point is saddles. It is defined to attract neighborhood in one direction and repel neighborhood in another direction at least, respectively. A saddle exhibits sensitive dependence on initial conditions, as initial values could follow into two directions of totally different properties. Saddles are also considered as unstable fixed points. Saddles are not considered as attractors in the system, however, they still play import roles in the dynamics [1].

In order to study the dynamic systems, it is intensively important and useful to determine the fixed points at the beginning if possible, because the fixed points could indicate the final states of the system. In some system, the fixed points could be estimated by the physical properties or symmetry. Sometimes, numerical methods would be employed, too.

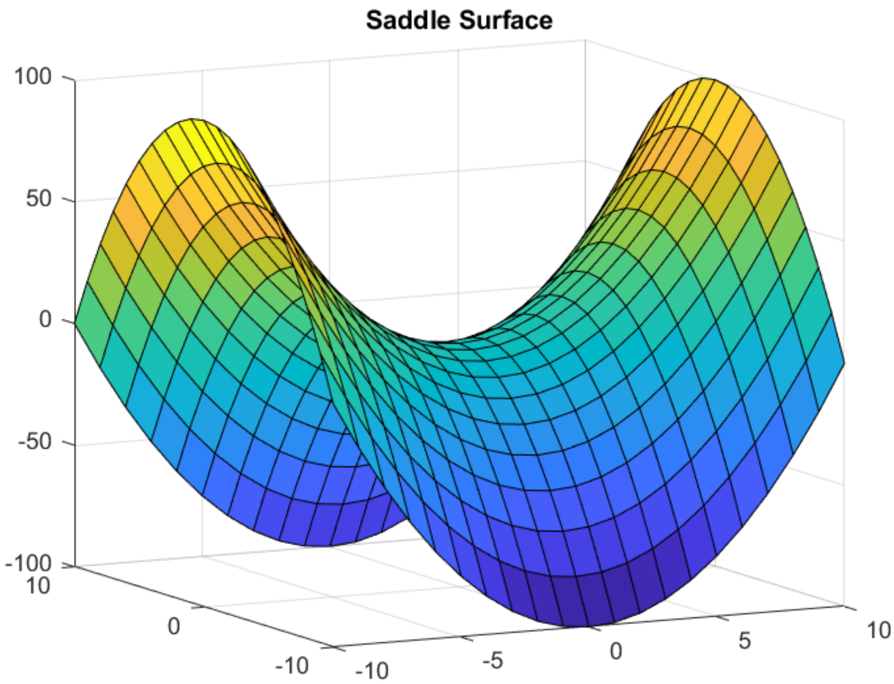


Figure 2.1: Saddle surface

As we have mentioned above, a time series of data by combining the system iteration function and initial conditions could be obtained. Such series of data is mentioned as orbit or trajectory of the system depends on the whether it is closed or not. Orbit could be considered as a special closed trajectory and indicate the existence of periodic points within the system potentially.

Because the trajectories are data of time series, some methods from time series analysis could be also used for chaotic system study or behavior estimate [30-32]. Time series analysis is based on the continuous regularity of objective development, to further speculate on the future development trend by using the historical data of the past through statistical analysis. The past will be continued in the future. This hypothetical premise contains two meanings: one is that there will be no sudden jump change, that the series is moving at a relatively small pace, and the other is that past and current phenomena may indicate the trend of development and change in current and future activities. This determines that the time series analysis method is more significant for short- and near-term prediction in general, but if it extends to the further future, there will be great limitations, resulting in the predicted value deviating from the actual situation and making the decision error.

There are three types of changes in time series data, which are trend, seasonality, randomness. Trend indicates a variable with the progress of time shows a relatively slow and long-term continuous increase or decrease with that the magnitude of change may not be equal. Seasonality is the regularity of alternating appears of peaks and valleys due to the external influences from natural season alternation. Randomness means the random change from individual but statistical properties on the time series data.

In time series analysis, three models are commonly employed if the data is stationary: autoregression AR(p), moving average MA(q) and autoregressive–moving-average ARMA(p, q). AR(p) reflects the influence and effect of the relevant factors on the predicted target only through the historical observation value of time series variables. It is not subject to the hypothetical conditions of independent independence of the model variables. This model can eliminate the difficulties caused by the self-variable selection, multiple collinearity and so on in the common regression prediction method. In MA(q) model, current predicted values are expressed by a linear combination of random interference or predictive errors from various periods in the past. This form can be considered when the hypothetical conditions of AR (p) are not met. ARMA(p, q) is the model of combination AR(p) and MA(q). Usually, in order to perform the time series analysis and forecasting, the first step is to eliminate the trend and seasonality. Then the promising model will be chosen and parameters will be calculated. During this process, autocorrelation, partial autocorrelation calculation could be involved. For model selection, one of the methods Akaike information criterion (AIC) is commonly used.

In order to evaluate the chaotic level of a system, the common method is to check whether two closed point will deviate to each other during the iteration. Such quantitive qualification is defined as Lyapunov numbers and Lyapunov exponents [1, 6, 19]. Lyapunov number presents the average per-step divergence rate of nearby points along the trajectory, and Lyapunov exponents is the natural logarithm of the Lyapunov number. Supposing a dynamics system, there are two trajectories L1 and L2 with the initial conditions as x_0 and $x_0+\Delta x$, as shown in the Figure 2.2:

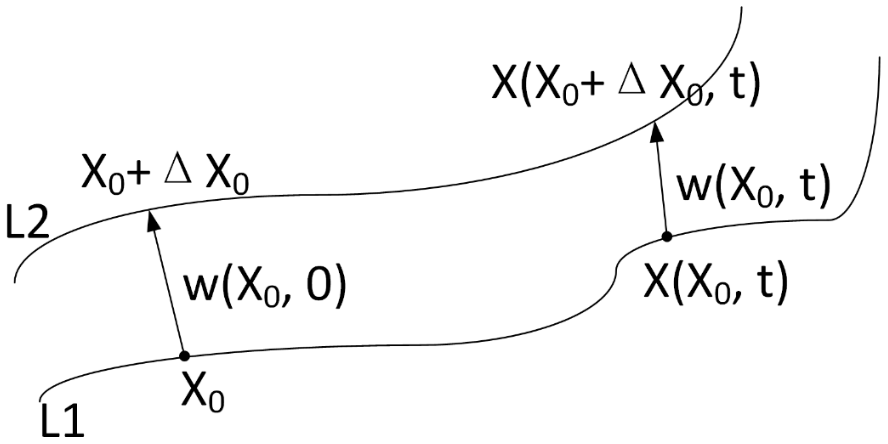


Figure 2.2: Schema of divergence between two trajectories

At time t, the position on L1 and L2 are $x(x_0, t)$ and $x(x_0+\Delta x, t)$, respectively. The distance or separation between the two trajectories at time t is defined as $w(x_0, t) = x(x_0+\Delta x, t) - x(x_0, t)$. The L1 is also called reference trajectory. The Lyapunov exponent λ is given as [6]:

$$\lambda(x_0, w) = \lim_{\substack{t\to\infty \\ w_0\to 0}} \frac{1}{t} ln \frac{|w|}{|w_0|} \quad (11)$$

where $w_0 = w(x_0, 0)$, is the separation between two trajectories at initial time.

For discrete system, the Lyapunov exponent indicates the average divergence or convergence around the reference trajectory, and is defined as [6]:

$$\lambda(x_0, w) = \lim_{w_0\to 0} \frac{1}{t_{max}} ln \frac{|w|}{|w_0|} \quad (12)$$

It is notable that there will be a series of λ in a system, and each of λ could be positive, negative or zero. In practical, it is only necessary to calculate the maximum of these series of λ, the λ_{max}. The property of the system could be reflected by this maximum Lyapunov exponent. One practical method to calculate the maximum Lyapunov exponent is given as [6]:

$$\lambda_{max} = \lim_{m\to\infty} \frac{1}{m\Delta t} \sum_{i=1}^{m} ln \frac{|w_i|}{|w_0|} \tag{13}$$

When $\lambda_{max} < 0$, the system is dissipative or non-conservative, and long duration system behavior does not depend on the initial condition and will converge to equilibrium state. Such system exhibits asymptotic stability; the more negative the exponent, the greater the stability. Super stable fixed points and super stable periodic points have a Lyapunov exponent of $\lambda = -\infty$. This is similar to a critically damped oscillator in that the system heads towards its equilibrium point as quickly as possible [6, 33].

When $\lambda_{max}= 0$, it indicates that the system is in a kind of steady state mode. The system is not sensitive to initial conditions and could show periodic motion. A physical system with this exponent is conservative. Such systems exhibit Lyapunov stability. Take the case of two identical simple harmonic oscillators with different amplitudes. Because the frequency is independent of the amplitude, a phase portrait of the two oscillators would be a pair of concentric circles. The orbits in this situation would maintain a constant separation, like two flecks of dust fixed in place on a rotating record [6, 33].

When $\lambda_{max} > 0$, the system is unstable and chaotic, and it also sensitive to the initial conditions. Nearby points, no matter how close, will diverge to any arbitrary separation. These points are considered to be unstable. For a discrete system, the trajectories will look like snow on a television set. This does not preclude any organization as a pattern may emerge. For a continuous system, the phase space would be a tangled sea of wavy lines. A physical example can be found in Brownian motion. Although the system is deterministic, there is no order to the trajectories that ensues [6, 33].

In addition to the Lyapunov exponents, power spectrum is also used for determining the existence of chaos in the dynamics system. Power spectrum represents the statistical characteristics of stochastic motion processes on various frequency components. In order to describe the randomness of chaos, the spectrum analysis method of stochastic vibration can be used to identify chaos. It is usually assumed that chaos is ergodic [6].

For sampling function x(t) of random signal, power spectrum could be obtained by two method. One is the time average of square of Fourier transformation [6]:

$$\Phi_x(w) = \lim_{T\to\infty} \frac{1}{T} \left| \int_0^T x(t) e^{i\omega t} dt \right|^2 \tag{14}$$

The second method is using the Fourie transformation of autocorrelation [6]:

$$\Phi_x(w) = \int_{-\infty}^{\infty} R_\tau(\tau)\, e^{-i\omega\tau} d\tau \tag{15}$$

where the autocorrelation function $R_\tau(\tau)$ is defined as [6]:

$$R_\tau(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{T/2}^{T/2} x(t)x(t+\tau) dt \tag{16}$$

For discrete dynamics system, one could calculate the autocorrelation, which is also the discrete convolution [6]:

$$c_i = \frac{1}{N} \sum_{j=1}^{N} x_j x_{j+i} \tag{17}$$

And then perform discrete Fourier transformation [6]:

$$p_j = \sum_{j=1}^{N} c_i e^{\frac{2i\pi kj}{N}} \quad (18)$$

where p_j represents the j-th frequency component, that is, the power spectrum of discrete time series. In numerical computation, there is also another method without calculating the autocorrelation function, which is computing the coefficients of discrete Fourier transformation directly.

For periodic motion, there is only one related frequency in the Fourier expansion, and only one of the discrete power spectra is not zero. Therefore, there are discrete spectral lines in the power spectrum only in its motion frequency and its frequency division and frequency multiplication. The power spectrum of a quasi-periodic motion is some discrete lines at several incompatible fundamental frequencies and their superpositions. Chaotic motion is a bounded aperiodic motion, which is a superposition of periodic motions of infinite number of different frequencies, and its power spectrum has the characteristics of random motion. The power spectrum of chaotic motion is continuous spectrum, that is, noise background and wide peak appear.

2.2 Proposition of a new approach from physics

In Chapter 2.1, we have reviewed the general method to study the chaotic system. Usually, in order to evaluate the chaotic system, the analytical solution of the system is required. However, in the practice, it is hardly possible to obtain the analytical solution or even the formula of the system. For instance, the stock market, the only accessible observations are price data while the formula is not achievable. For such system, iterative exploration on data is a difficult task. In addition, exploration could be localized to certain space or degree while the whole picture of the system is not accessed. Especially, it is desired that the final states of the system are known, so that regardless of the initial condition and process, the final states are determined.

For deterministic chaotic system, in addition to the traditional method to explore the final states, alternatively, we could analysis the system from its physical properties. The final states should be the equilibrium states of the system. These equilibrium states could be physical positions, or other physical quantities, such as pressures, temperatures and etc. Similar to the concept of fixed point in chaotic system, there are also three kinds of the equilibrium states in physics, which are stable equilibrium, unstable equilibrium and neutral equilibrium. Therefore, we could start from these concepts and analysis the basic properties the system, thus, obtain the equilibrium points firstly. Once these points or states are determined, their properties in chaotic system could be investigated by the neighborhood area. Furthermore, instead of lost in the chaotic behavior of the system, we could mark each initial condition of the system by the final states. Hence the distribution of the final states could be achieved.

From such approach, we have setup a relationship between the initial condition and final states, so that once the it has been established, states of any initial condition could be known from beginning without going through the iteration exploration. It is also worthy to mention that such relationship should be setup by the determined system parameters. And it could be reconstructed once system parameters are changed. In reality, such approach could be huge helpful in financial market. For example, for the financial crisis, people are always looking for the relationship between the start or the hint of it.

So, in this work, we will use a typical experimental chaotic system as the preliminary demonstration for such approach. The system we chosen here is the magnetic pendulum system, which involves the effect of magnetic force, gravity force and frictions. In the next chapter, we will start with introducing the basic description and model of the system.

Chapter 3: Introduction of Magnetic Pendulum System

In this chapter, the basic concepts of the magnetic pendulum system will be introduced. Model of the system will be explained, and the symbolic system will be defined. All contents in the following will follow the definition in this chapter.

3.1 Basic Concept of the System

The experimental magnetic pendulum system consists a pendulum suspending from a string. In the plane under the pendulum, there are three magnets distributed equally to the center of the plane. A three-dimensional schematic drawing of the system is shown as Figure 3.1, in which the black ball represents the pendulum and red, yellow and blue spheres are magnets.

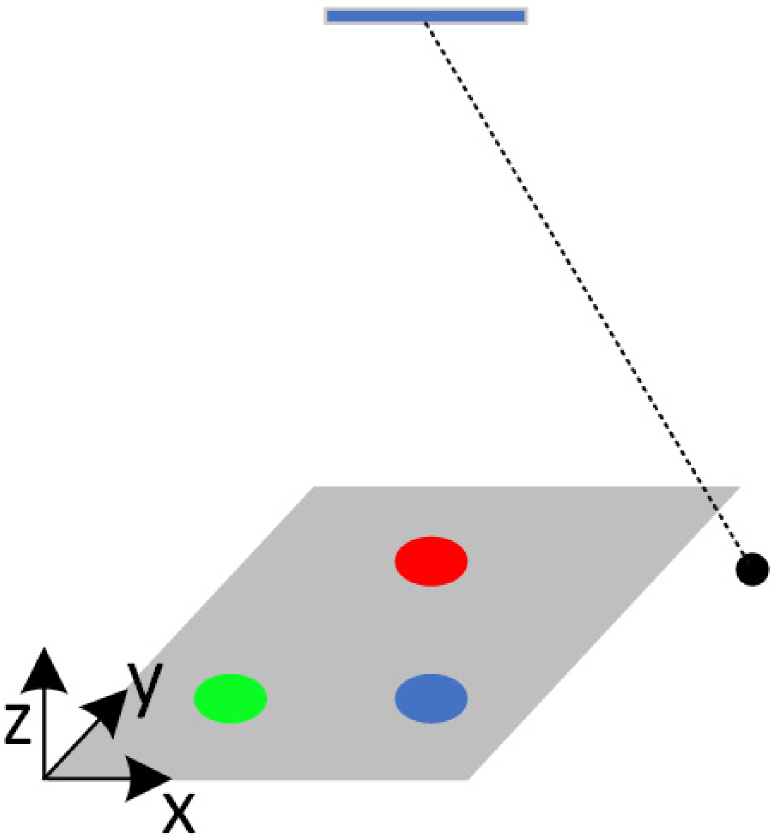


Figure 3.1: Schema of magnetic pendulum system

The distribution of three magnets is shown as Figure 3.2. In Figure 3.2, the O point is the equilibrium point for the pendulum to reach the minimal potential energy.

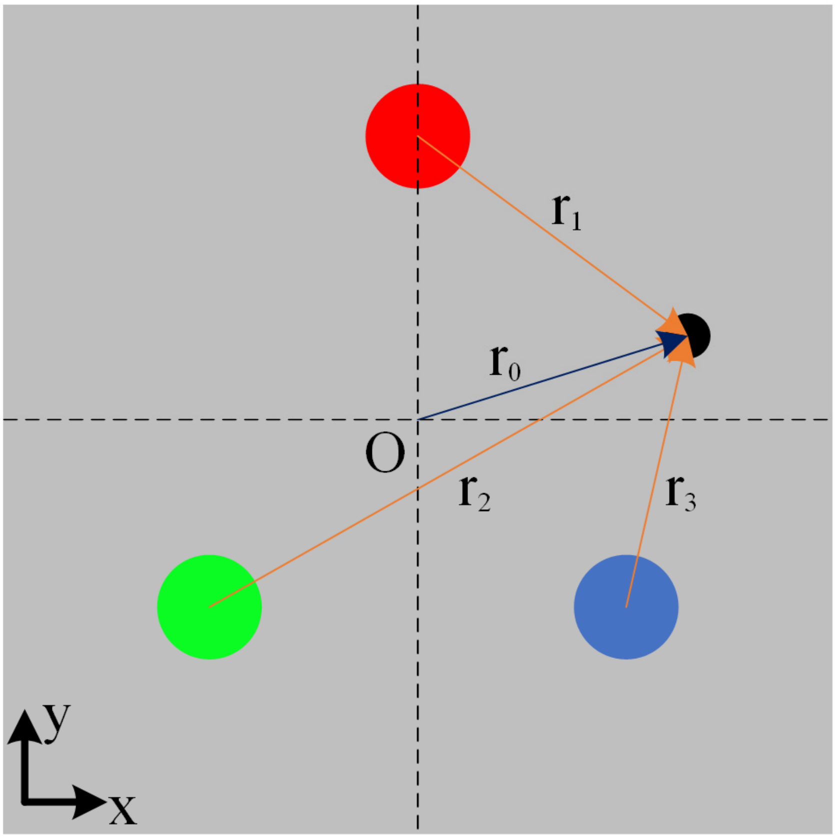


Figure 3.2: Magnets and pendulum distribution in two-dimensional plane

There are some setups about the system:

1. The length of pendulum is very large comparing to the spacing of the magnets. This assumption allows the motion of the pendulum to be considered as in a plane rather than a sphere surface.
2. Although the magnets are presented with some diameters, it is treated as a point source of attraction when calculating the force.
3. Magnetic force follows the inverse squared law; i.e. the force is inversely proportional to the square of the distance.

4. The three magnets have the same magnetic poles; i.e. the N magnetic pole, and the pendulum has the opposite magnetic pole; i.e. the S magnetic pole, so that the pendulum suffer the attraction from every single magnet.
5. There is a small distance h between the plane of magnets and that of the pendulum movement.

As shown in Figure 3.2, the distance in x-y plane from the each of magnet and the O point to the pendulum are marked in the figure by vector r. Here we define them as:

1. r_i: the distance between magnets and pendulum
2. r_0: the distance between O point and pendulum
3. (x_i, y_i): the Cartesian coordinates for each magnet
4. (x, y): the Cartesian coordinates for pendulum
5. (x_O, y_O): the Cartesian coordinates of origin
6. h: the distance between the plane of magnets and that of pendulum

3.2 Model of the System

Now let us analysis the force acting on the pendulum. Supposing the pendulum was moved away from its equilibrium point O, and the current Cartesian coordinated are (x, y), there are three kinds of forces acting on it:

1. The force caused by magnets, as we mentioned above, is inversely proportional to the inversed squared distance, so that the force could be described as:

$$\overrightarrow{F_m} = k_m \frac{\vec{r_i}}{r_i^3} \tag{19}$$

   where F_m is the force, k_m is the strength coefficient for the magnetic force.

The r_i in the x-y plane is $\sqrt{(x_i - x)^2 + (y_i - y)^2}$. However, this could cause a singular point when $x = x_i$ and $y = y_i$, hence as we have mentioned above, a distance h is introduced. Therefore, the equation of r is $\sqrt{(x_i - x)^2 + (y_i - y)^2 + h^2}$. And the magnetic force from single magnet is:

$$F_{m-x} = k_m \frac{x_i - x}{((x_i - x)^2 + (y_i - y)^2 + h^2)^{3/2}} \tag{20}$$

$$F_{m-y} = k_m \frac{y_i - y}{((x_i - x)^2 + (y_i - y)^2 + h^2)^{3/2}} \tag{21}$$

2. The force caused by gravity, and it pulls back of pendulum to the original point O; thus, this force could be described as:

$$\overrightarrow{F_g} = k_g \overrightarrow{r_0} \tag{22}$$

Converting this formula into the Cartesian coordinates, the x and y component of gravity force should be:

$$F_{g-x} = k_g(x_O - x) \tag{23}$$

$$F_{g-y} = k_g(y_O - y) \tag{24}$$

3. Thirdly, we introduced another force here, which is the friction force between the pendulum and the media in which the whole system is. The force acts in the opposite direction of pendulum's motion and is proportional to the velocity of the pendulum. So, the force is described as:

$$\overrightarrow{F_f} = -k_f \vec{v} \tag{25}$$

According to Newton's second law of motion, one could summarize the three kinds of force to the acceleration and mass, herein we assume the mass of the pendulum is 1.

Therefore, we could arrive the final formula for the pendulum as equations (26) and (27). The solutions to these two equations will be the trace of pendulum.

$$x" + k_f x' - k_m \sum_{i=1}^{3} \frac{x_i - x}{((x_i - x)^2 + (y_i - y)^2 + h^2)^{3/2}} - k_g(x_O - x) = 0 \qquad (26)$$

$$y" + k_f y' - k_m \sum_{i=1}^{3} \frac{y_i - y}{((x_i - x)^2 + (y_i - y)^2 + h^2)^{3/2}} - k_g(y_O - y) = 0 \qquad (27)$$

As shown in Figure 3.2, considering a magnetic pendulum system in reality and supposing the lower left corner is the origin point with coordinates (0, 0), and the limit for x and y axis is set to be 8.00. Therefore, the coordinates for O point is (4.00, 4.00), and Cartesian coordinates for red, green and blue magnates are (4.00, 6.00), (2.27, 3.00) and (5.73, 3.00), respectively. The O point is not set to be the origin as (0, 0) although one could expect symmetry in the coordinates for straightforward understanding. We will benefit from this setting in the next chapter when talking about the visualization of chaotic motion. There are three coefficients in the formula, which are magnetic strength coefficient k_m, the gravity coefficient k_g and friction coefficient k_f. For the preliminary calculation, the k_m is set to be 1, k_g is taken as 0.5.

In order to study the formula mentioned above, it is worthy to explore the properties of the calculation domain. According to the description of the system, there are four sources of force located in the domain, which are the gravity equilibrium point O and the positions of the three magnets. Hence the pendulum moves in the complicated potential field from the four sources. Regarding our assumptions, the force acting on the pendulum is reservedly proportional to the square distance. So, the potential filed established by one magnet could described as:

$$V_m = -k_m \frac{1}{r_i} \tag{28}$$

where r_i is the distance between the pendulum and magnet.

And the potential filed formed by the gravity force under our assumption is:

$$V_g = \frac{1}{2} k_g r_o^2 \tag{29}$$

By taking coefficients we have set, the total strength of potential filed could be achieved and shown by the following plot.

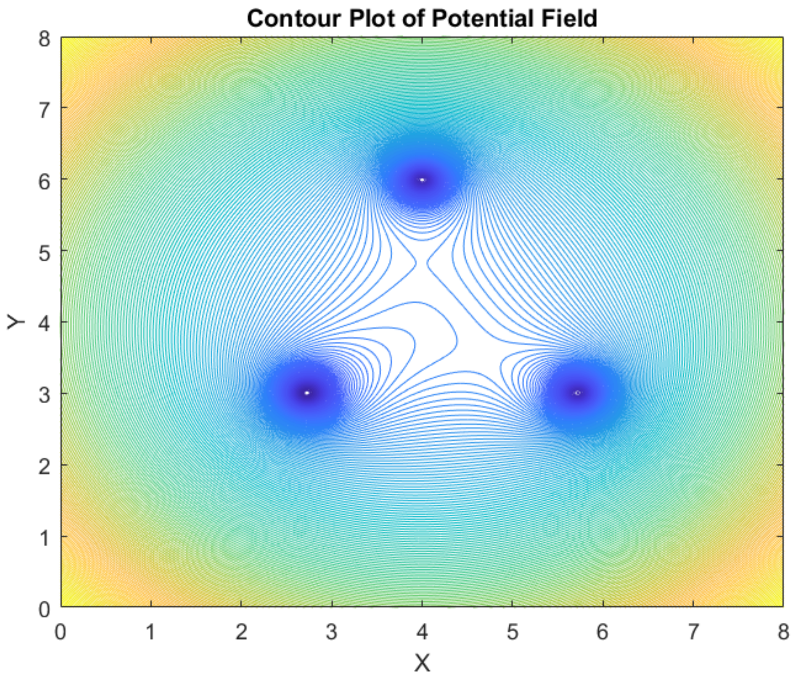


Figure 3.3: Contour plot of potential field of the pendulum system

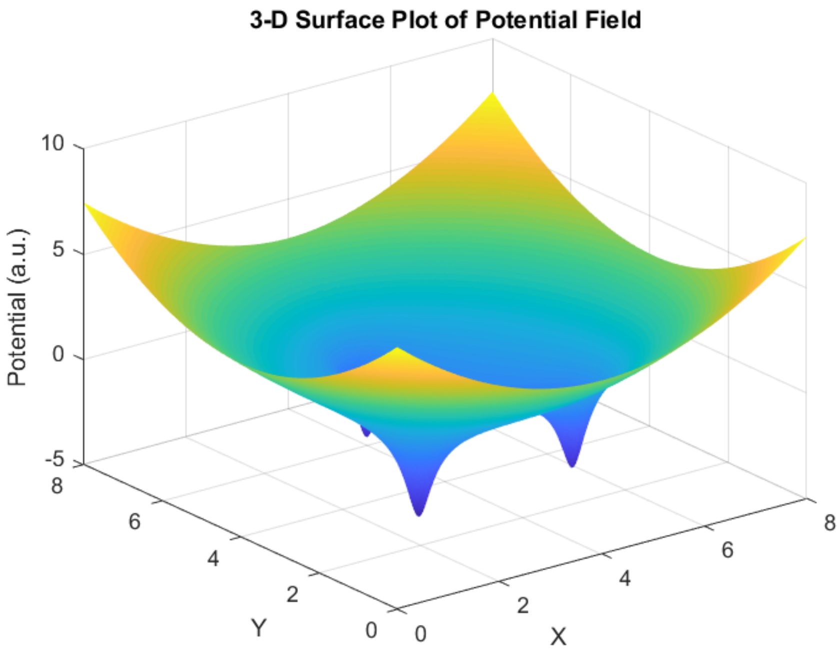


Figure 3.4: Three-dimensional surface plot of potential field

From these two plots, firstly, it is very clear that the whole system is in the potential well, in which any object in the system will move towards the center of the plane. It could also be observed that there are three local minimum potential energy points, which are also the positions of magnets. And these positions should be the stable equilibrium points that any small perturbation to the pendulum at these positions will not move it far away, and the pendulum will finally return to these equilibrium positions. In addition, one could expect that there is another local equilibrium point at the center of the whole plane. However, it is not clear that whether this point is a stable equilibrium point or not.

These equilibrium points are the points of interest for the system, which will be investigated in the following. Furthermore, in the previous calculation for the potential field, the influence from the friction has not been considered. Regarding an object in the field with zero initial speed, it will move to the closet equilibrium points. However, the motion of it could largely change due to the effect from friction, which is also the actual situation.

Before we go further to investigate the system, it is worthy to finish another work first, which is the visualization of the chaotic behavior of the pendulum. It is more straightforward if we could watch the movement of pendulum in real time. Therefore, in the next chapter, we will put our effort on a program first, and then go back the numerical investigation on the properties of the system.

Chapter 4: Program of Chaotic Magnetic Pendulum

In Chapter 3, the magnetic pendulum system has been introduced. In this chapter, we will talk about the implementation of the pendulum movement presentation in real time. A program will be constructed by Java language from scratch. Detailed information about this program will be introduced. All of the source code will be available in an online repository.

4.1 Requirements Analysis

In order to visualize the pendulum motion from starting position to final equilibrium position, a graphic user interface (GUI) is needed. This GUI should consist a window for displaying the pendulum animation and also a control panel window for parameters adjustment. There are mainly two objects in this system, which are the pendulum and the magnets. There also should be a program for the trajectory calculation, and initial calculation should take the values of each parameter k_m, k_g and k_f. The value k_m is the property of magnet, k_g and k_f are set separately through the control panel. All of the three parameters should have default values.

At the finish, this program should be an independent app and could run on different platform, such as Window or Linux operating system (OS). There are two options for such requirement. One is the web app and the other is the desktop app. An alternative choice is the mobile app, which could run on mobile devices, such as smart phones or tablets. However, this program is not designed for public use but lab use initially. Hence mobile app will not be considered here. Regarding the two options above, the web app runs in a web browser. Such implementation usually requires communications between client side and server side. It could be implemented as a fat client-side program that all of the calculation is completed by the program running in the browser. However, we still need a web server to store the program so that anyone who access it could have the program downloaded and loaded into web browser. This process will involve a design for the communication through internet, efficient data transfer and sever setup and so on, which will be a complicated task. In addition, as this is the first time for us to perform such full life cycle software development, quality analysis and debug in the late stages of development over the internet will be a difficult task. Therefore, an app running on the desktop locally will be our choice for the development.

Regarding the desktop app, there are mainly three programming languages for such development, which are C++ [34], Java [35] and Python [36]. They are all Object-oriented programming (OOP) languages and appropriate for such development. Among the three languages, C++ is the most efficient one for running, as the program is compiled into the machine code and runs by the CPU directly. On the other hand, C++ does not specialize in GUI programming but high-density computation. There is not any sufficient build-in library in C++ for GUI programming. One option is the visual C++ Microsoft Foundation Class (MFC), but this library is complex and learning of it will be a long and tough process. The other choice is QT [37], which is a cross-platform framework that is usually used as a graphical toolkit. The most advantage is that this toolkit could run on any platform, desktop computers, laptop, smart phones, tablets with Windows OS, Linux, iOS, Android, and even embedded devices. But according to the feature of C++ language, employment of QT is also a difficult task. The development process will be very slow.

Java language is also popular in industry [35], and employed in many large-scale projects, especially for the service programs running on the servers. Java has many build-in standard libraries for different application. And most of standard libraries have been verified and widely used in industrial projects. One of them is suitable for GUI development, which is the Java Swing library [38]. Java Swing comes with a compatibility for cross-platform, the main disadvantage is its pale and lack of aesthetic appearance. Another disadvantage of Java is its low efficiency. Comparing with C++, Java programs runs in the Java virtual machine (JVM), which brings the high compatibility and low efficiency simultaneously. Many optimizations have been made to JVM to improve its performance. Nowadays, many server applications, middle-wares are written in Java. Its efficiency could not beat C++, however, is still acceptable.

Python is a popular language especially for university education or amateur projects. As a dynamic language, Python is simple and has huge external libraries for development acceleration and simplification. On the other hand, the disadvantage of it is also various. The running efficiency is not high, even slower than Java. This is because the Python does not have type concept, and compiler needs to check all of the variables within the code. This will cost a large amount of time. So, Python is suitable for small project, application of this language in industry for large scale project is very rare. Another reason is the issue of external libraries, as the completeness and security are not strictly validated.

Above all we have briefly summarized and compared three candidate language for the project. Considering the simplification and learning cost, in this work, Java language will be employed for source code implementation. And GUI will be implemented mostly based on the Java Swing standard library.

4.2 Class Design

As Object-oriented programming, the basic unit of coding is class, which is the definitions for the data format and available procedures [39]. From another point of view, a class is a collection of variables and methods, which could be accessed from outside of class (public), or limited to the class internally (private). For the program in this work, the final app is the combination of several classes and all of the functionality is provided by the cooperation among these class.

According to our requirement analysis, there are 14 classes: MainFrame, MainPanel, TracePanel, BasinPanel, ControlPanel, Ball, Magnet, MagnetsCollection, EngineCore, BallEngine, BasinCalEngine, TraceCalEngine, Vect and Toolbox, and 1 initiation class App. Figure 4.1 shows the Unified Modeling Language (UML) diagram of some main classes.

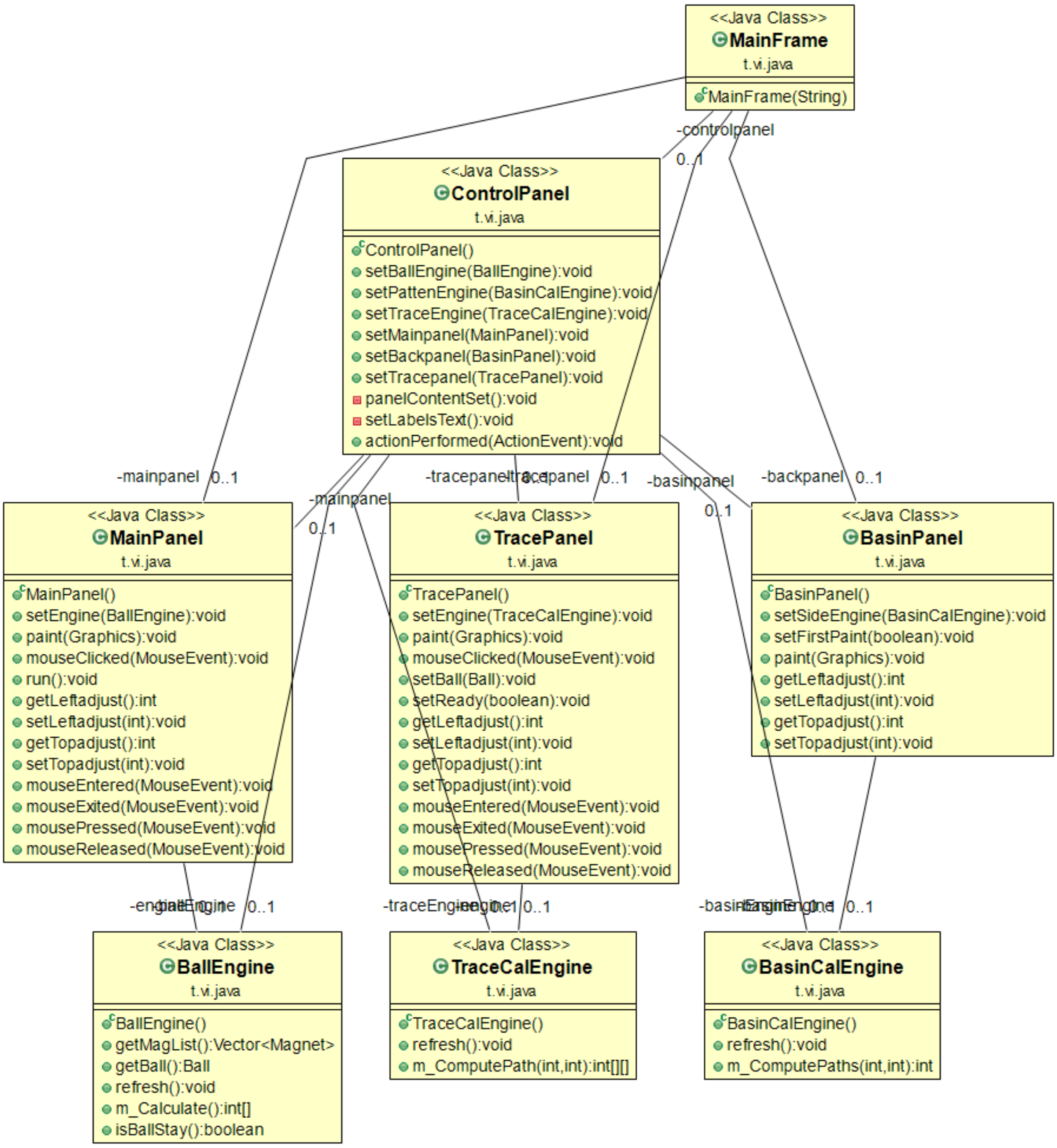


Figure 4.1 UML of Magnetic Pendulum program

MainFrame is the foundation of GUI, according to the Java Swing library. In its framework, MainFrame is the sub-class of JFrame. In the constructor, the size of window should be defined. Any other necessary components are added by calling the method of JFrame, the add(). Another important factor is the layout manager, which determines the layout for the frame and manage the relative position for each section. Herein it is also worthy to mention that the width of frame boarder is different in Windows OS and Mac OS. In order to achieve the same appearance in different operating system, this boarder width shall be detected and adjusted. Default values could not work in various systems.

Panels are two sub-sections in the MainFrame. Usually, definition of panel is not necessary if there is only one window in the MainFrame, as all of the components, such as buttons and menus could be added to the MainFrame directly. However, in this work, as we are going to display animation, a separate window inside the MainFrame is needed. In Java Swing, each separate window shall be defined as a panel. There are 4 panels inside the MainFrame. MainPanel, BasinPanel and TracePanel are used for display and ControlPanel is used for parameters setting and program control. In addition, MainPanel, BasinPanel and TracePanel are set to be in the same position, which means these three panels has to be put into layers. Java Swing provides a JLayeredPane class for such usage. In the implementation, each panel is assigned with an integer number to define its relative position in the layers. The panels in the front need to be set as transparent, so the panels underneath could be visualized. All of these methods have to be declared in the MainFrame.

MainPanel is the sub-class of JPanel. It also has to implement 2 interfaces in Java, the MouseListener and Runnable, and reason for this will be talked about later. In the MainPanel, as an animation will be displayed, the coordinates should be determined first. In Java Swing framework, the origin of the window is the left upper corner of the window with x and y increased when moving in rightward or downward directions, respectively. In the MainPanel, the basic display unit is pixel. One pixel stands for one point in the panel. An object could only move from one pixel to another, therefore the coordinate of position for an object in the panel is always a pair of integers. It is also worthy to mention that the panel board is also involved when calculating the panel width. So, the real position of an object has to be adjusted for boarder width to prevent any distorted or unsatisfied display.

In the final app, we want the initial position of pendulum be determined by the mouse click in the panel, and this is the reason for the MouseListener interface. With the method from it, we could easily add a listener to monitor if a mouse click is made to the panel, and capture to position of the click. This is the power of build-in library, because we could concentrate on the function that we would like to have instead of thinking the low-level program techniques and skill for such tasks. As we have mentioned, an animation will be displayed inside the MainPanel, hence a thread running is needed and it has to implement the Runnable interface. This way is one of two methods in Java for thread, and we will use this way in this work. Thread a common concept for programmers. Herein we will briefly introduce it.

A thread could be considered as a sub-task running inside a program. A program usually has a main thread initiated by default. In this work, anther thread is needed for creating the animation by running the paint method of MainPanel constantly. An animation is fundamentally display of a series of pictures. Hence if we could paint a diagram inside the MainPanel with a bit of changes each time, such as the position of an object, the animation could be achieved finally. And this is the mechanism we are going to apply. With Runnable interface, we could create a thread. In this thread, we will call the method repaint(), and this method will call the method paint() in the MainPanel automatically. With the real-time pendulum position injected into the paint() method, we could eventually see the pendulum moving.

Underneath the Mainpanel, there are TracePanel and BasinPanel. TracePanel displays the pendulum trajectory starting from the position of mouse click. The content showing here is the complementary to the MainPanel as one may lose the whole picture of pendulum's movement when watching animation. BasinPanel display the basin diagram across the whole plane. Basin diagram is defined as the color distribution of each point in the space. And the color is set to be the final destination of each trajectory. For example, a point with a trajectory starting from it and ending at red magnets will be marked as red.

Regarding the ControlPanel, it is also the sub-class of JPanel. From the name, one could obviously realize that this panel is used for the controlling, such as parameters adjustment. In this panel, some text input areas will be employed for value display and input. Some buttons will provide the function for value to be sent to the calculation or reset the app to initial status.

Ball is the class to describe the pendulum. In this work, all of the animation will be displayed in two dimensions. So, it is named as a ball for simplification. There is only one Ball object at running. So most of large-scale applications will employ the singleton rule for such class implementation. Here this rule is not applied as program scale is not very large. However, such rule should be kept in mind in some other projects.

Magnet class stands for the magnets in the system. They could simply be implemented as an array of position values for the calculation. MagnestsCollection class is the wrapper class for Magnet class. Currently we only support 3 magnets in the system so that we could implement it in a simple way by creating 3 magnet instances directly. Such method is easy to implementation but not convenient for future code extension. For example, if we need to add a new feature of increasing the number of magnets, this method would be a bottle neck. Therefore, this class works as preparation for the future feature extension. It takes an argument of integer for the number of magnet and creates certain amount of instance. Engine could know the number of magnets by simply call this class.

EngineCore is the base class for calculation related classes. In this class, the basic and frequently used method and variables will be defined, for instance, the functions for computing the magnetic force, gravity force and frictions. From the programing point of view, such inheritance relations follow the spirit of object-oriented programing, which could save a lot time as there is no need to rewrite the same amount of code in classes with slightly difference. The EngineCore class has 3 sub-classes, which are BallEngine, BaisinCalEngine and TraceCalEngine.

BallEngine class provide the calculation function for the simulation needed in the MainPanel. It collects the initial conditions and parameter value to compute the real-time position of pendulum at each time step. There are two possible approach to perform the calculation. One is limited times simulation. In this approach, the position of pendulum is calculated through limited times iteration. A maximum times value is set at the beginning. The iteration will be terminated when the maximum times is achieved. The other approach is to use the condition. In this approach, there is no limit on the iteration times. It will keep running until some condition is satisfied. So, in this class, the second approach is employed.

The BasinCalEngine is the engine class for calculating the basin diagram for the whole space. This class works with BasinPanel. The main algorithm for calculation each trajectory is the same as BallEngine. In addition, a while loop is performed to iterate all of the points in the plane. There are 801 * 801 pixels in the plane, and so far, our program could not handle such large amount of computation. Therefore, we treat every 5 pixels as 1 point to simplify the computation complexity. And the final basin diagram is acceptable.

TraceCalEngine is the engine class to calculate single trajectory for one starting point, and it works with TracePanel class. Different from the BallEngine that the calculation is terminated by simultaneously check, this class uses fixed iteration times to compute the single trajectory, which is the first approach we have mentioned above. This iteration times is obtained from our experience in the simulation. The benefit of such implementation is data transfer between classes. The whole data of trajectory is packed into an array and sent to tracepanel, so that tracepanel could plot the trajectory easily and efficiently once it is available. The other implementation is using the same method as we have done in the BallEngine, which requires a thread to handle the data transfer in real time and would consume a lot of system resource.

Toolbox class uses various of static variables to store the constants, such as the width and height for MainFrame, panels, and default value for parameters. This is a common practice in the industry for large-scale projects when some key values need to be shared across different teams. Vect class is the utility class, which provides the function of vectors in mathematics. Finally, the App class the initiation class for the whole program, in which the main function is defined. Java will start running from this main function.

4.3 Calculation Algorithm

The real-time position or the trajectory of pendulum could be obtained by solving the equation (26) and (27) in Chapter 3. There are various methods to solve such non-homogenous second order differential equations. In addition, the equations for this system are time dependent. Generally, such equations could be solved numerically by Finite-Difference Time-Domain (FDTD) method. In this work, we could use this approach through two steps. First, we could use the FDTD method and initial condition to solve the trajectory, which is a series of position coordinates, and save this trajectory data. Second, the motion of pendulum is displayed by repainting its position based on the trajectory data. To achieve this goal, a highly efficient algorithm for the FDTD method is required. This will introduce a complicated task and is beyond the scope of this work. Therefore, we will use iteration method, the Beeman's algorithm, which calculates the pendulum position according to the current states.

Beeman's algorithm [40] is designed for numerically integrating second order ordinary differential equations, especially the Newton's equations. Originally, this algorithm was introduced to simulate the molecular dynamics of the system with large number of particles. In this work, as we consider all of pendulum and magnets as point mass and source, they could be treated as particles and Beeman's algorithm could be applied here.

The Beeman's algorithm is given as the following two equations:

$$x(t+\Delta t) = x(t) + v(t)\Delta t + \frac{1}{6}\big(4a(t) - a(t-\Delta t)\big)\Delta t^2 + O(\Delta t^4) \quad (30)$$

$$v(t+\Delta t) = v(t) + \frac{1}{6}\big(2a(t+\Delta t) + 5a(t) - a(t-\Delta t)\big)\Delta t + O(\Delta t^3) \quad (31)$$

where t is the present time, Δt is the time step, x(t) is the position at t, v(t) is the velocity at t, a(t) is the acceleration at t.

4.4 Program User Interface

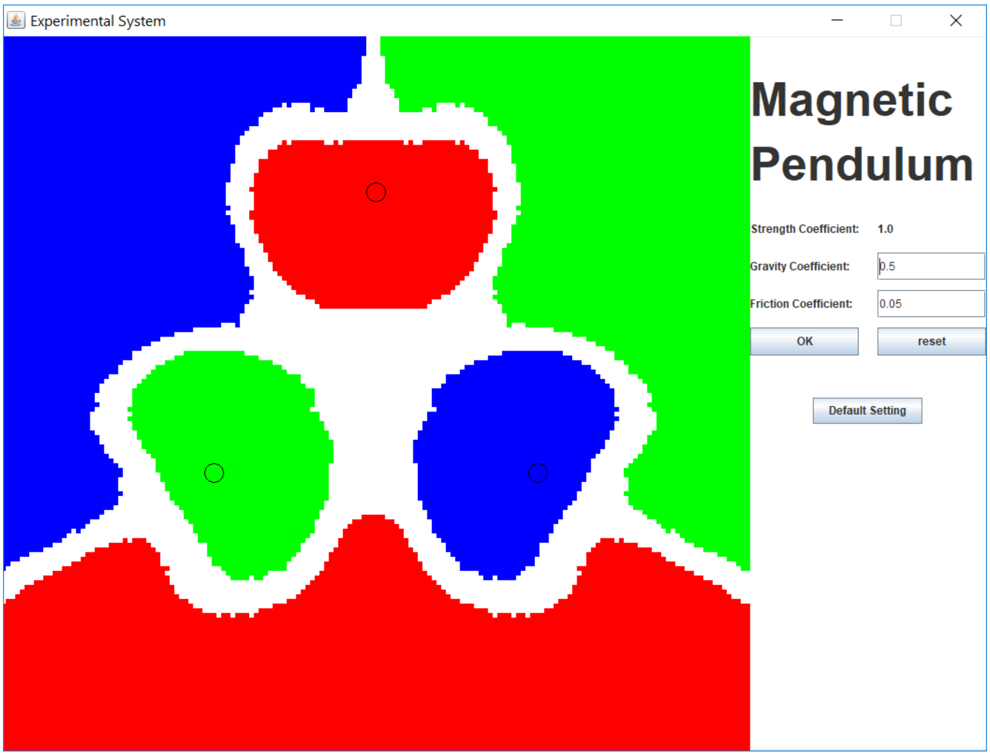


Figure 4.2 Default user interface

Figure 4.2 shows the basic user interface of our program. The section on the left is the MainPanel and BasinPanel. Three black circles mark the position of three magnets. Each magnet is given a color as red, green and blue, which also stands for the final state. The color pattern outside three magnets represents the trajectory final state starting from each initial position. The white color represents the final state is the center of the panel.

The section on the right is the ControlPanel, in which the strength coefficient k_m, the gravity coefficient k_g and friction coefficient k_f are shown. k_g and k_f could modified, and k_m is fixed to be 1.0 in this work. The "OK" button is used for the new parameters setting. A new basin diagram will be calculated when this button is pressed. The "reset" button is used for system status reset during the animation of pendulum, so that one could start a new pendulum movement animation from a new initial position. The “Default Setting” button is used for reset the system to the initial starting status, while k_g is set to be 0.5 and k_f is 0.05.

4.5 Some Discussion about Implementation

The basic framework of this program is implemented through Java Swing packages, by which the data is presented in the form of plotting or animations. And the data is obtained through some classes that perform computation. There are some main obstacles during the implementation of this program are worthy to be mentioned.

First of all, in the equation to calculate the magnetic force, we introduce a parameter h to represent the distance between pendulum and magnets plane. One could benefit this design in two aspects. One is that it matches the physical situation in the practice as the pendulum could not move in the magnets plane. The other one, which is also more important to numerical simulation, is that it overcomes the singular points in the equation. When the positions of pendulum are taken as those points closed to the magnets, the magnetic force could be really large, which would cause a numerical explosion for the computation. Therefore, this value has to be precisely selected. There are two parameters related to the magnetic force for positions closed to magnets, the k_m and h. In this work, we fix k_m to be 1.0, so that h is taken as 40 according to our trial tests. Other method could be fixing h and adjusting parameter k_m. Either way could work for the program in order to prevent the numerical explosion of magnetic forces.

Secondly, the movement of pendulum is implemented through a thread painting a picture of pendulum with an updated position. It has been proven that in order to achieve the animation for human's eye, at least 40 frames (pictures) has to be painted within 1 second. In the program of thread, a while loop is employed to paint the picture constantly. In addition, a time gap is introduced between two iterations to prevent high CPU consumption. According the theory we have mentioned, this time gap should be less than 25 milliseconds. It is also worthy to mention that the position of pendulum is updated every 50ms in this work. Therefore, herein 15ms is taken for a smooth animation of pendulum.

Thirdly, the MainPanel, BasinPanel and TracePanel are in the same position according to the design requirement of this work, and this is implemented through JLayeredPane from Java Swing package. There is a drawback from this implementation that each time the repaint() method is called, the program will automatically perform the paint() method of every panel class. The computation for the MainPanel and TracePanel are less comparing with that of BasinPanel, which could be finished within 1 second, so that they could use real-time calculation method. However, for BasinPanel, it will take at least 5 second to finish a new calculation, which would delay the display of basin diagram and pendulum animation. One potential method to solve this problem is using the buffered image. Herein we employed a simple method that storing the calculated result into an array. This array will be constructed every time the parameters are changed. If there is no parameter changing and only new pendulum animation presenting, the basin diagram will use the data from this array, and this could be finished within 1 seconds, which will not introduce any delay to other parts of program.

So far, we have introduced the design process and detailed information about this program, the source code of this program is available in an online repository:

https://github.com/EntiumZ/Pendulum

Chapter 5: Numerical Calculation and Results

In Chapter 4, we have finished the design and implementation of our program. Some preliminary results could be obtained from this program to show our approach. In addition, we will use another numerical simulation method to perform some further investigation on the magnetic pendulum system. Finally, we will compare the basin result from our program and numerical simulation.

5.1 Result from the Program

Figure 5.1 shows an example of the trace of pendulum starting from an initial position marked as green. According to our definition, the end point of the trajectory should be the green magnet.

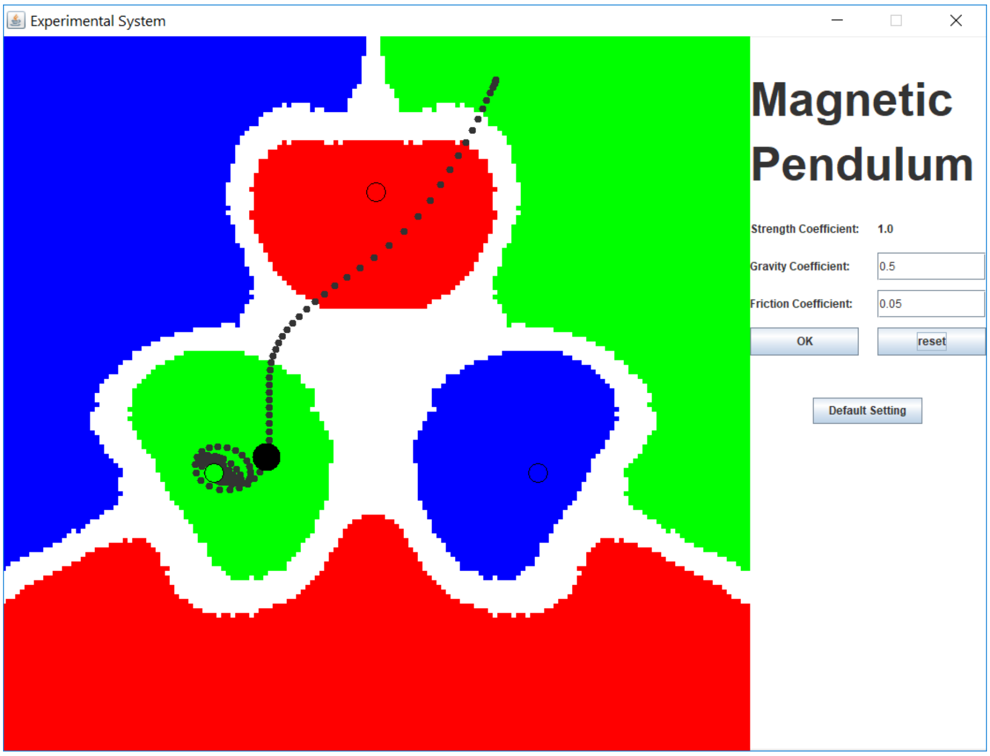


Figure 5.1 Example of pendulum movement trajectory

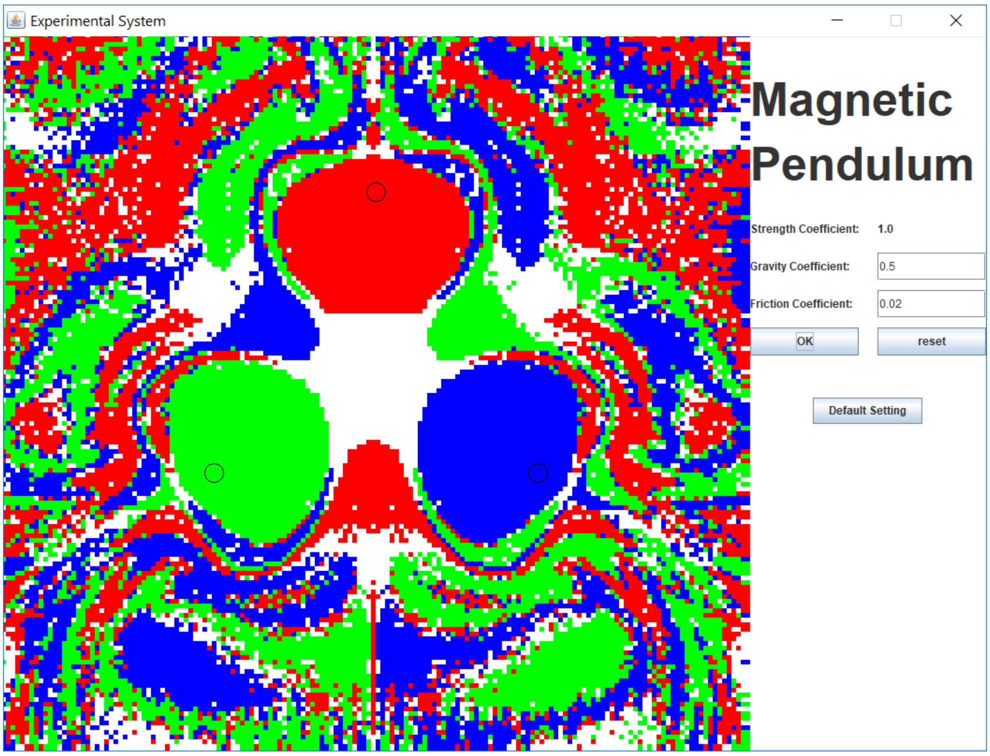


Figure 5.2 Updated user interface when parameter is changed, $k_f = 0.02$

In Figure 5.2 the friction coefficient k_f is set to be 0.02. One could observe that comparing with Figure 5.1, the basin diagram is more complex and states distribution is in a chaotic pattern. However, due the symmetry of final states, the basin diagram is still symmetric. In Figure 5.3, a trace of pendulum stating from a green initial position is shown. It could be seen that the pendulum behavior is more complicated comparing with Figure 5.1. The pendulum is not going to the green magnet directly, but approaching the blue magnet in the middle. However, according to our expectation, the pendulum still ends at the green magnet. So, the regardless of the process in the middle, which could be very complicated, the final state could be known at the beginning.

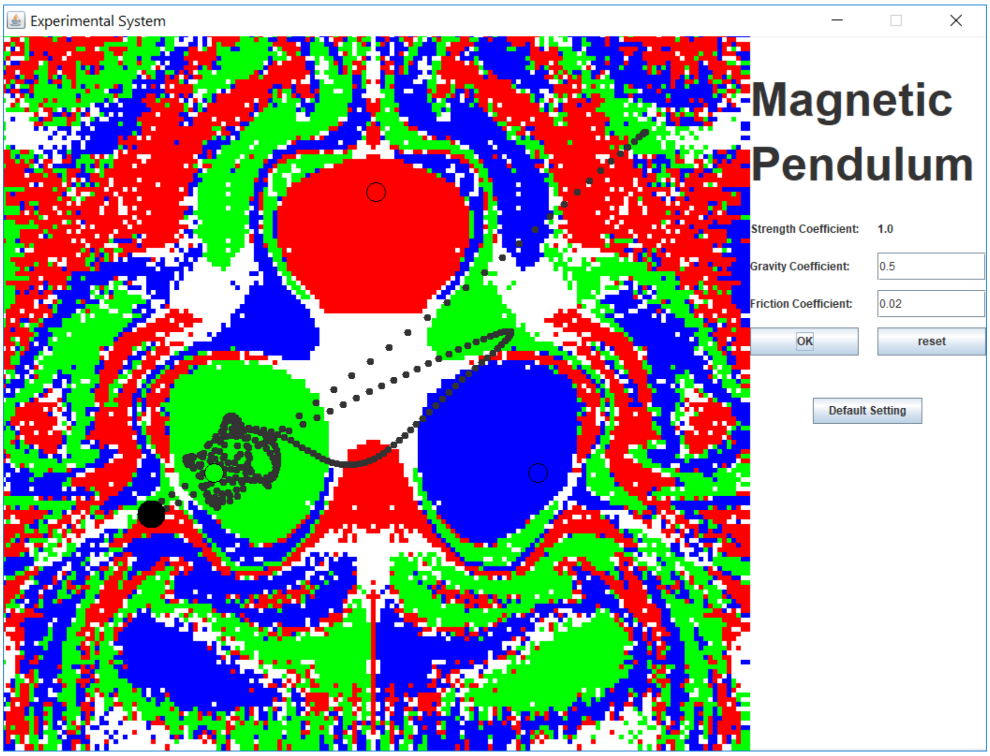


Figure 5.3 Pendulum trajectory after parameter change

5.2 Magnetic Pendulum Study by Numerical Simulation

It has been shown that the formula for the magnetic pendulum system is second order non-homogeneous differential equations. Finding the analytical solution for these equations is a complicated task. In this work, we will employ the commercial simulation software Mathematica to find the solution numerically [41]. Mathematica has different built-in commands for such equations with initial conditions. The mainly employed functions are NDSolve for the numerical solution and ParametricPlot for the results plotting. While most of the input format and standard have been pre-determined by Mathematica, there would not be too much innovation or creation for this process, the code used in this work is mostly inspired by the work mentioned in the Reference [42], however, with different parameters setting. So, the code in this work will be omitted, and detailed example code for reference could be found online [42].

For numerical calculation, the k_m is set to be 1, k_g is taken as 0.5 and k_f is taken as 0.1. The initial condition x_0 and y_0 are starting position for the pendulum and initial velocity of pendulum is taken as 0. Time range t in the NDSolve is set to be 100 at maximum. With these parameters setting, Mathematica could solve the position of pendulum (x(t), y(t)) from start to the maximum time step. The solutions are 2 vectors, that one is for x and the other is for y. Herein we will mention each solution as a trajectory as it is in the other literatures. Figure 5.4 shows 4 trajectories with the starting position setting as the points of interest as mentioned above. It could be seen that trajectories appear to be four steady points, which shows that the pendulum starting from these points with 0 initial velocity will stay at the original points firmly. This matches our analysis from the potential field, that these points are local minimum potential points. Therefore, these 4 points of interest are the fixed points for the system. Pendulum's status with these points as the starting point will not be changed. As the fixed points have been determined, it is also worthy to check if these points are source, sink or saddle, which will be our next discussion. Herein we define a color representation for the trajectories based on the final equilibrium position in all of the phase plot within this work. The red color presents all the trajectories ending at point (4, 6). The green color is for the point (2.27, 3), the blue color is for the point (5,73, 3) and the black color presents the point (4, 4).

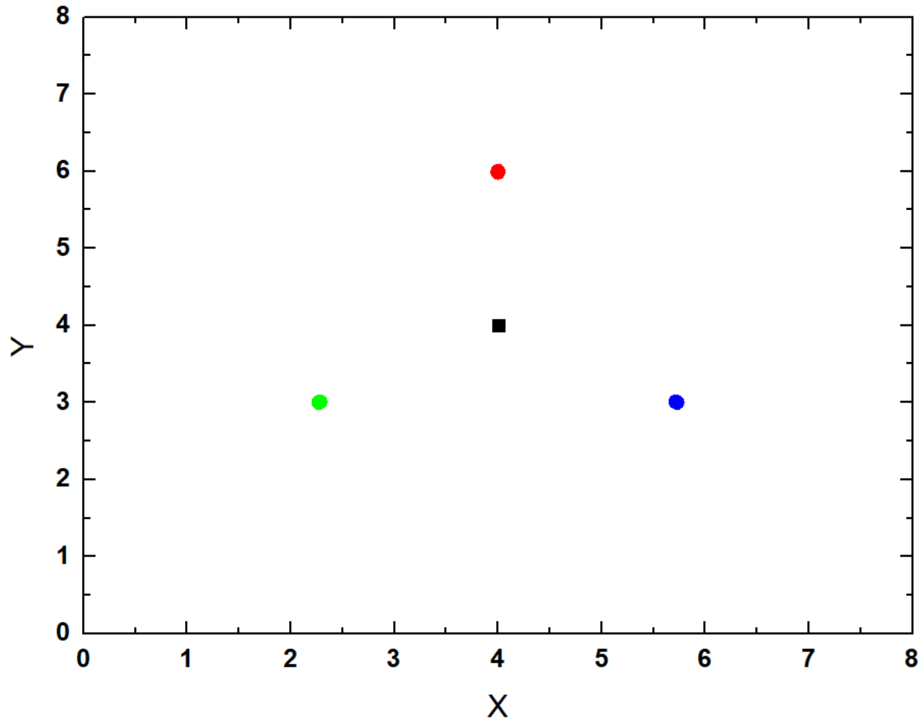


Figure 5.4: Trajectories plot of fixed points

In order to evaluate the property of a fixed point, the neighboring regions around these points should be checked. Herein we used the same method as above by setting the starting points as the ones around the fixed point (4,4) and checking the final position. The Figure 5.5 shows the boundary of area for the pendulum to move back to the fixed point (4, 4), which means the pendulum with any points within the area as the starting position will move back to the fixed point marked as the solid black square. Therefore, this fixed point is a sink or attracting fixed point. The epsilon neighborhood could be calculated by the shortest distance from the fixed point to the boundary, which is 0.52 (a.u.).

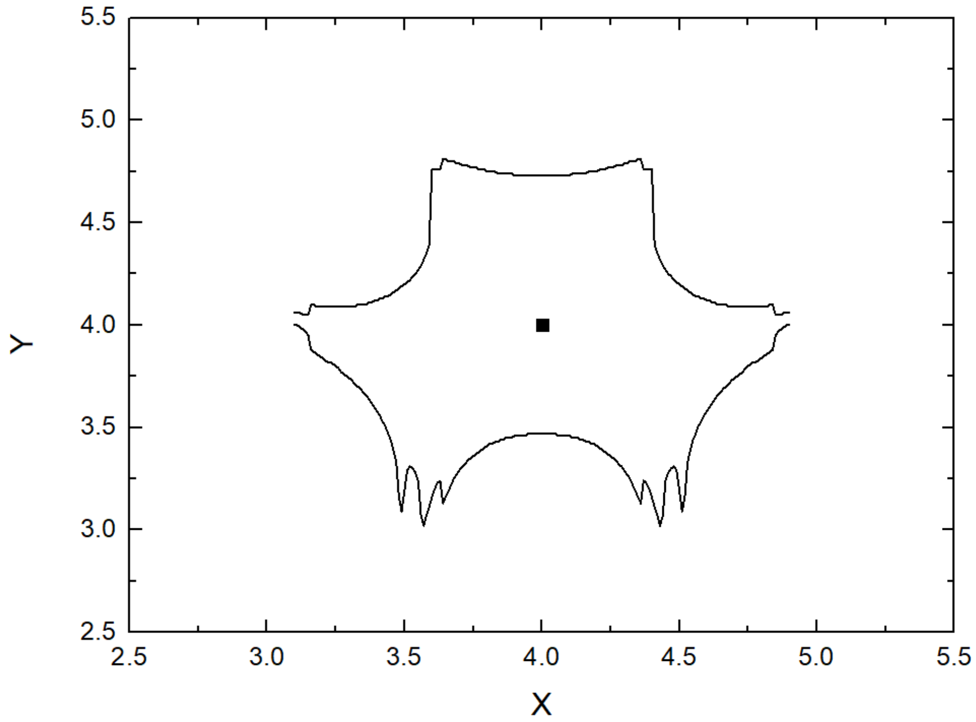


Figure 5.5: Boundary of epsilon neighborhood for fixed point (4.0, 4.0)

The same method could be employed for the other three fixed points. The boundary for them are shown in Figure 5.6. Therefore, all of these fixed points are sinks or attracting fixed points. The red, green and blue solid cycles marked the position of fixed points (4, 6), (2.27, 3) and (5.73, 3), respectively. And the epsilon neighborhood for them are 0.4(a.u.), 0.405(a.u.) and 0.405(a.u.), respectively.

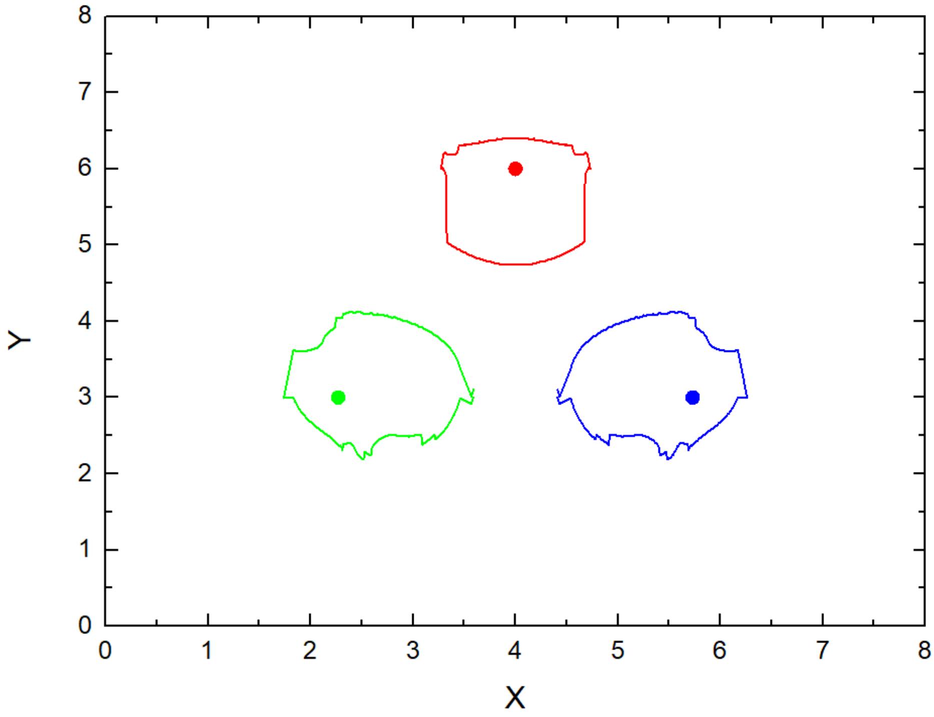
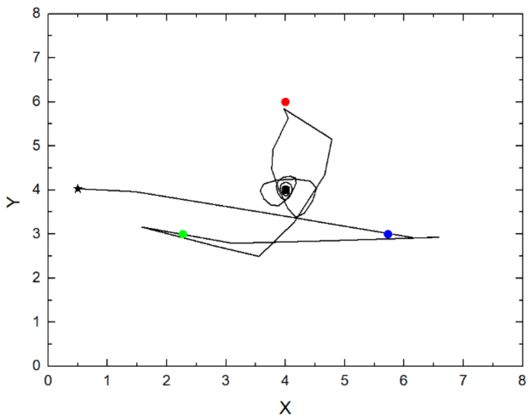
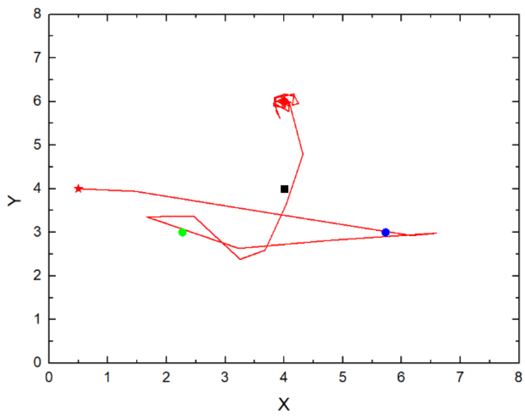


Figure 5.6: Boundary of epsilon neighborhood for fixed points (4.0, 6.0), (2.27, 3) and (5.73, 3)

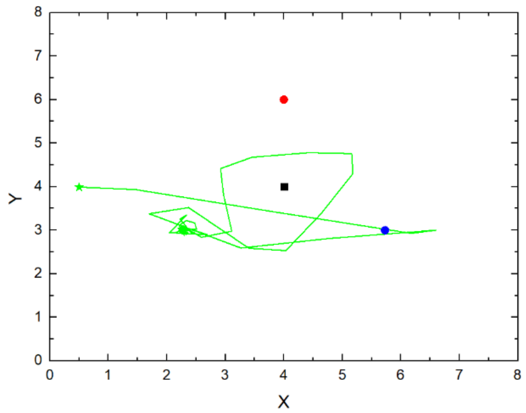
Now let's investigate some other points that are far away from the sinks. The figures below show the pendulum trajectories from 4 different starting points (marked as stars). And the starting positions are (0.5, 4.03) for the black line, (0.5, 4) for the red line, (0.5, 3.99) for the green line, and (0.5, 4.05) for the blue line. The color of lines also indicates the final equilibrium position of the pendulum as we have mentioned above. From each single trajectory, one could observe that the pendulum could approach or even pass the multiple equilibrium positions, however, it will settle down to only one position. This does not comply with the analysis we have made based on the potential field, and could be attributed to the introduction of friction. In addition, from each single trajectory, it is not easy to define any type of curve or determine any behavior of the chaotic system. On the other hand, by putting all of them together in Figure 5.8, it is very clear that the final resting position is highly sensitive to the initial conditions.



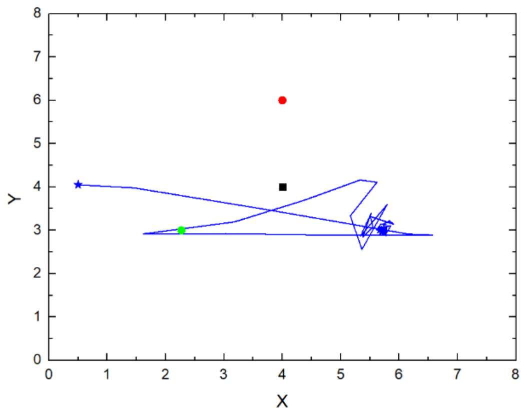
(a): Trajectory for staring position (0.5, 4.03), marked by black star



(b): Trajectory for staring position (0.5, 4.031), marked by red star



(c): Trajectory for staring position (0.5, 4.034), marked by green star



(d): Trajectory for staring position (0.5, 4.033), marked by blue star

Figure 5.7: Individual plot of trajectories for different starting positions

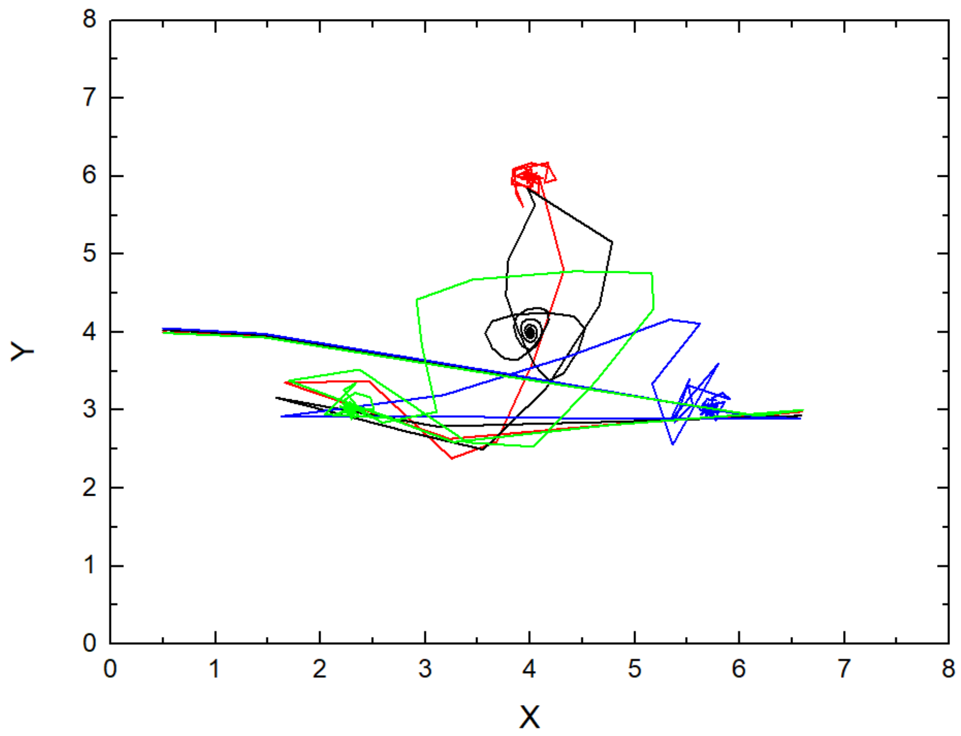


Figure 5.8: Combination plot of trajectories for different positions

So far, it has been shown that the pendulum behavior is sensitive to the initial position. In addition, the final status of pendulum is not a single one but a set, or a collection of positions. These features are all indicate that the system is a chaotic system. However, more investigations should be made to determine the properties of the system, especially on each trajectory.

5.3 Analysis on Single Trajectory

Regarding the trajectory starting from the position (0.5, 4.03), an oscillation on the phase x and y could be observed from Figure 5.9, if they are plotted as time series. This could be attributed to the forces acting on the pendulum, as these forces are always pointing to the center and each magnet, which are also the sinks. Hence, the pendulum could not escape far away from these sinks and will always move towards them potentially. Figure 5.10 shows the autocorrelation and partial autocorrelation calculation on the phase x and y. These plots suggest there are a moving average and auto regression properties within the data, according to the basic time series concepts.

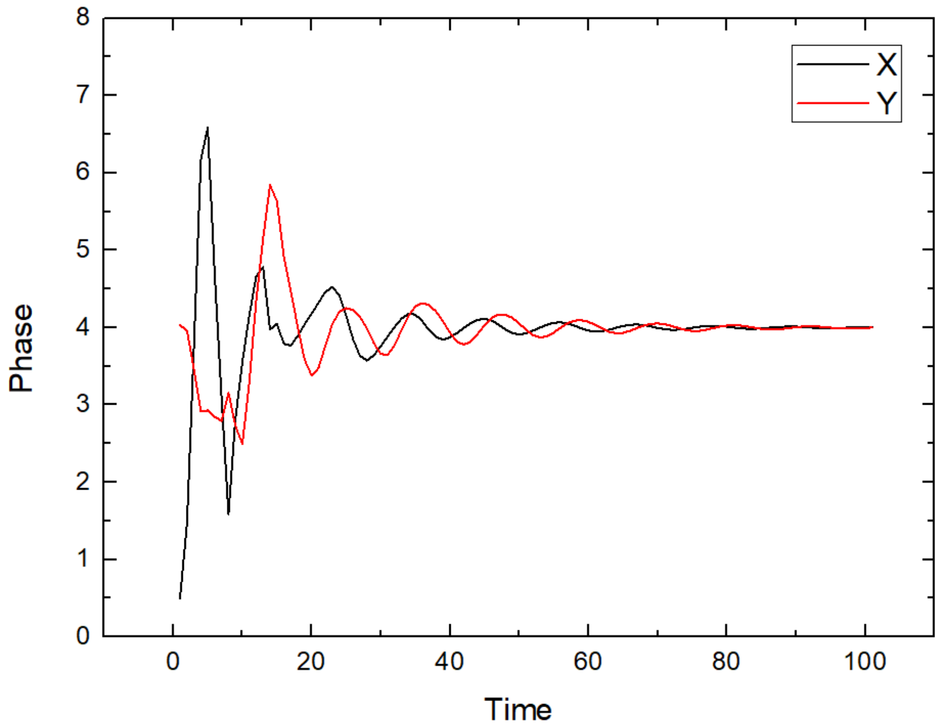
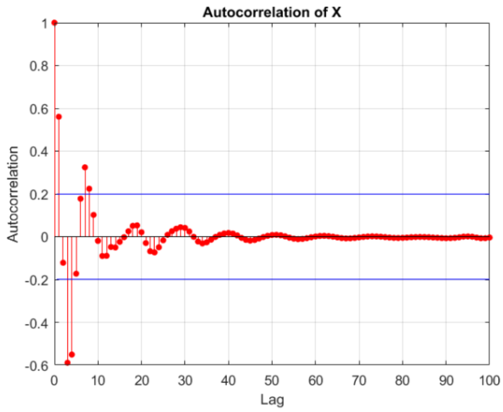
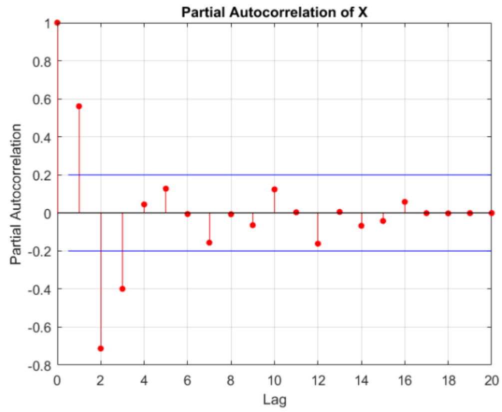


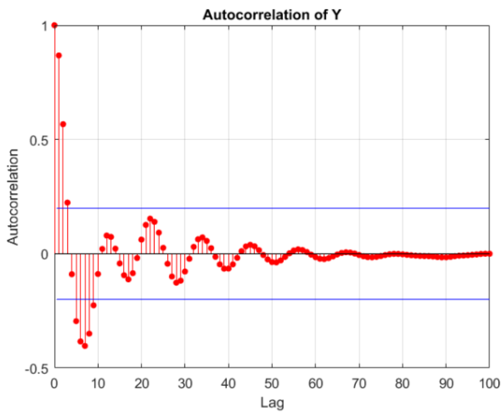
Figure 5.9: Evolution of phase x and y versus time steps



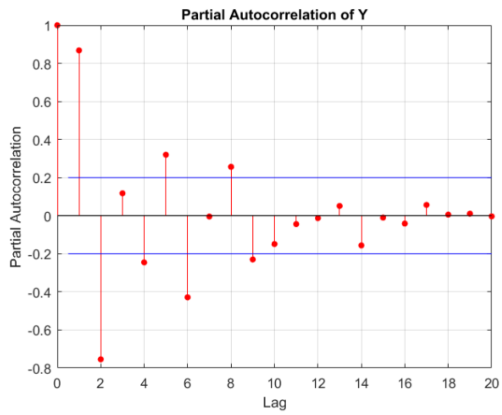
(a): Autocorrelation plot for phase X



(b): Partial autocorrelation plot for phase X



(c): Autocorrelation plot for phase Y



(d): Partial autocorrelation plot for phase Y

Figure 5.10: Autocorrelation and partial autocorrelation of phase X and Y

It is worthy to mention that these calculations are performed on the raw data of x and y without reducing any seasonality or trend. In order to reduce the trend of original phase data, we could create a new time series from x(t) - x(t-1). On the other hand, according to the physics meaning of x, the position of pendulum, the difference computation on that just equals the pendulum velocity. Furtherly, computation of difference on velocity could give us the acceleration. Both of velocity (v) and acceleration (a) could easily obtained from numerical simulation. Figure 5.11 and 5.13 show the velocity and acceleration of pendulum for phase x and y, respectively. The autocorrelation and partial autocorrelation are also shown.

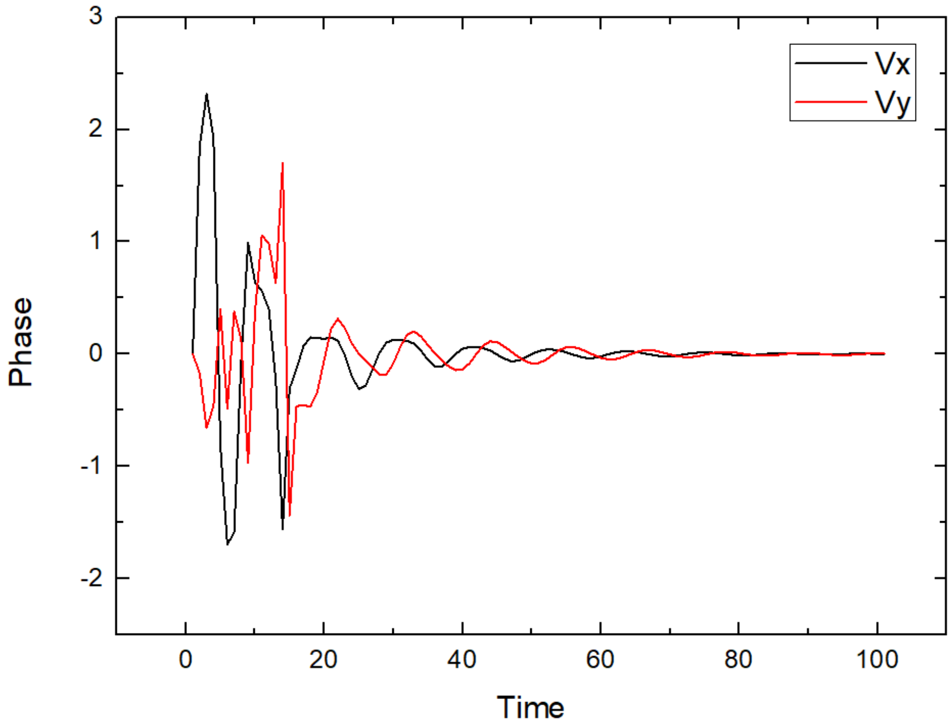
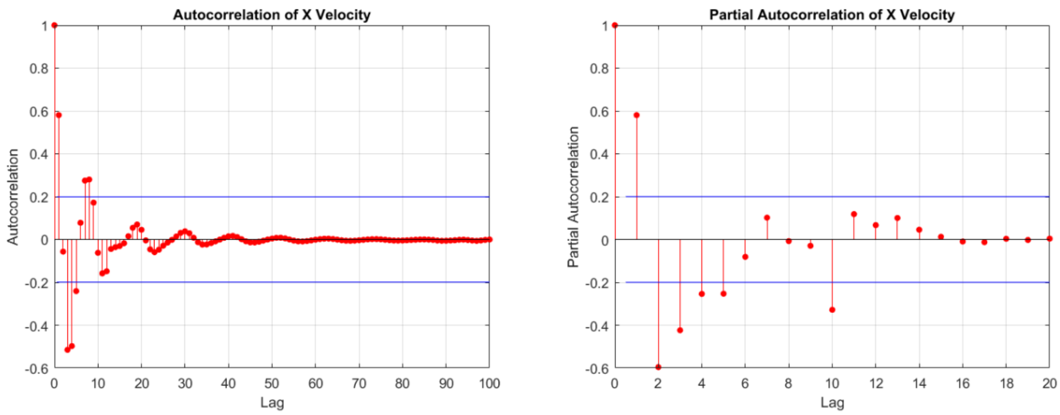
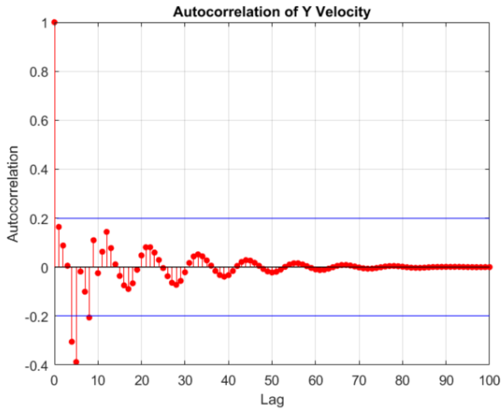


Figure 5.11: Evolution of phase Vx and Vy versus time

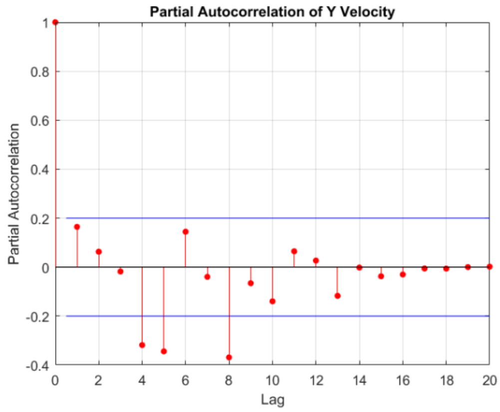


(a): Autocorrelation plot for phase Vx

(b): Partial autocorrelation plot for phase Vx



(c): Autocorrelation plot for phase Vy



(d): Partial autocorrelation plot for phase Vy

Figure 5.12: Autocorrelation and partial autocorrelation of phase Vx and Vy

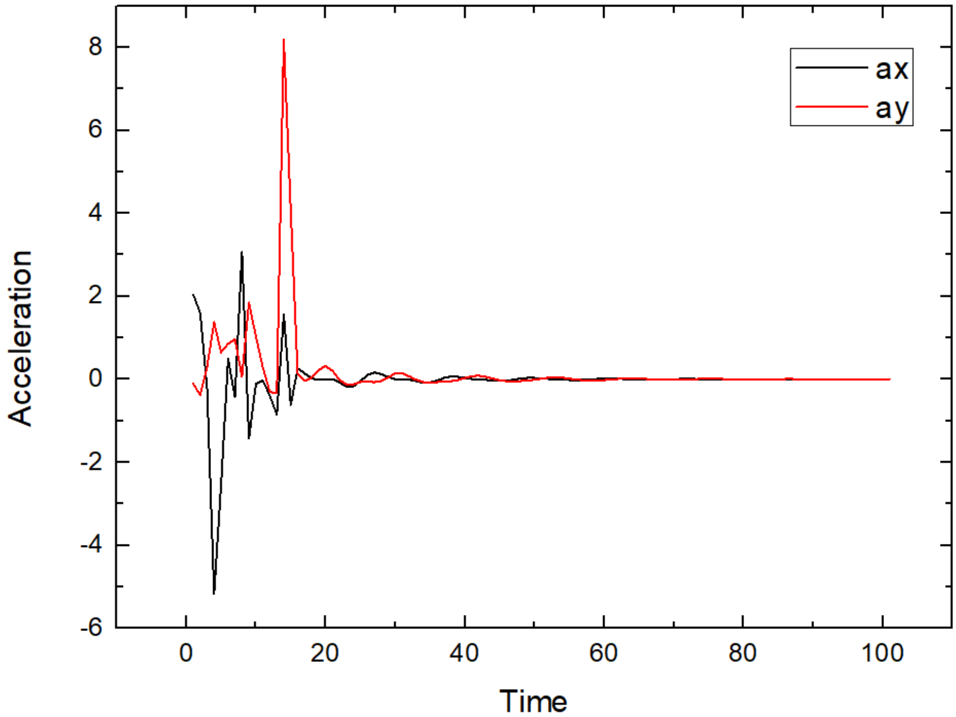
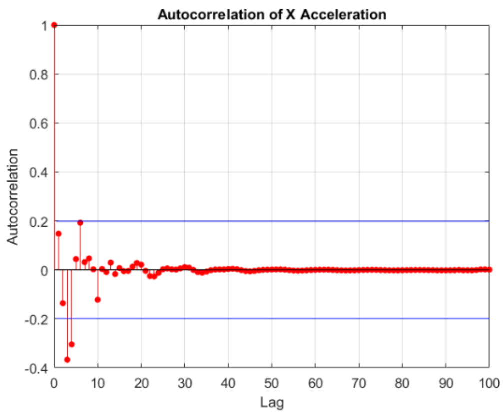
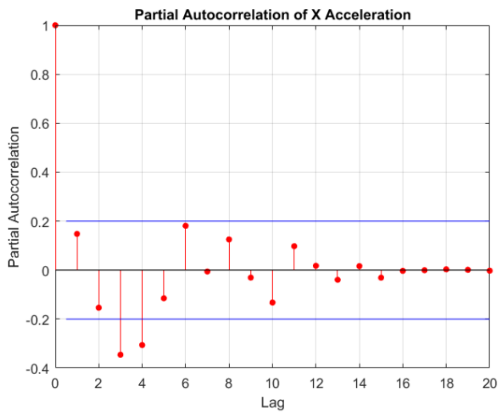


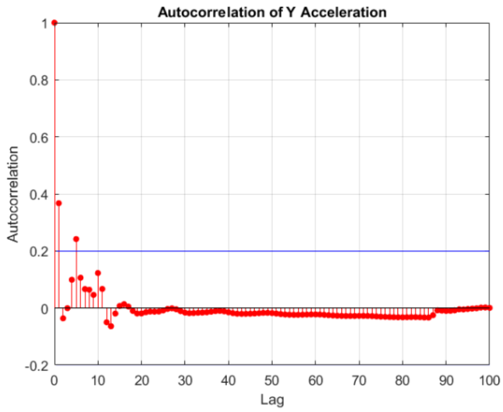
Figure 5.13: Evolution of phase ax and ay versus time



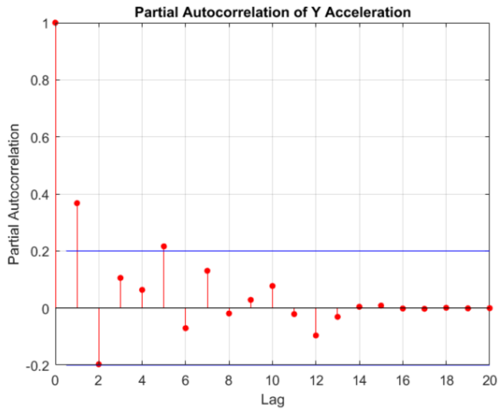
(a): Autocorrelation plot for phase ay



(b): Partial autocorrelation plot for phase ay



(c): Autocorrelation plot for phase ay



(d): Partial autocorrelation plot for phase ay

Figure 5.14: Autocorrelation and partial autocorrelation of phase ax and ay

From these plots, we could find that there are strong autocorrelations in each phases of pendulum. And the reason of this could be implemented in several points. Firstly, the acceleration of pendulum depends on its position of each time step. This could be understood from the differential equation mentioned in Chapter 3. Secondly, this acceleration determines the variation quantity of velocity. The initial velocity of pendulum is set to be 0. So, velocities thereafter fully depend on its value of previous time step and the acceleration. Thirdly, the velocity, which is also the variation amount of position, determines the next pendulum position. When the pendulum moves to a new position, this cycle of relationship will start again. It is also worthy to mention that, as one of force caused by the friction, which relates to the velocity, there should also be some cross correlation between the velocity and acceleration. This could be confirmed by the correlation calculation between them. For instance, the correlations are between v_x and a_x, and for vy and ay are -0.2628 and 0.1795, respectively.

Another power tool of evaluating such behavior is the power spectrum. As is has been mentioned, a chaotic motion usually comes with a broad band power spectrum. This is also the case for the pendulum in our system. The Figure 5.15 and 5.16 shows the power spectrum plot for phase x and y. It could be seen that motion of pendulum mainly consists of direct and low frequency components. In addition, the spectrum appears to be really broad, which matches our expectation. Both of the spectrum of x and y confirms that there is a chaos motion of pendulum in the two directions.

Figure 5.15: Power spectrum density of phase X

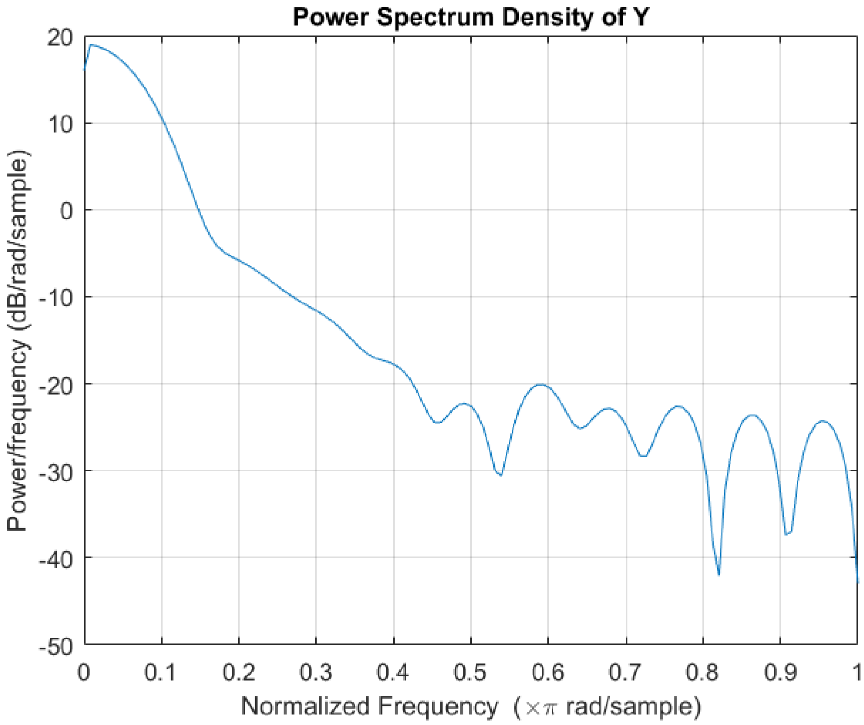


Figure 5.16: Power spectrum density of phase Y

So far, we have studied the properties of one trajectory, which is the one starting from (0.5, 4.03) and ending to the center of simulation plane (4.0, 4.0). In order to fully check the properties of the system, at least a second trajectory needs to be investigated. Herein we take the trajectory starting from (0.5, 4.033). This trajectory finally converges to the blue magnet (5.73, 3.0), which is also shown in the Figure 5.8 as the blue lines. The plots of phase x and y versus the time are shown in the Figure 5.17.

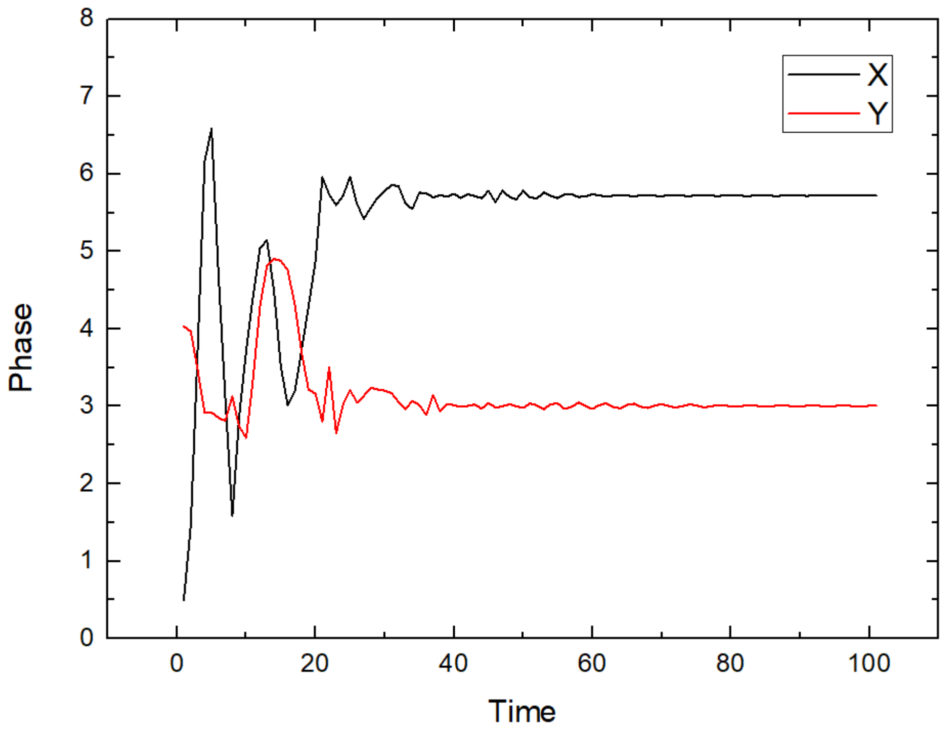
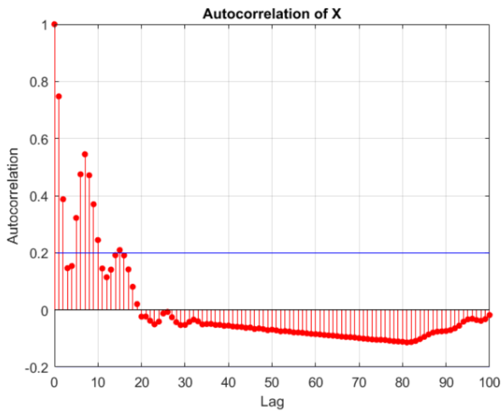
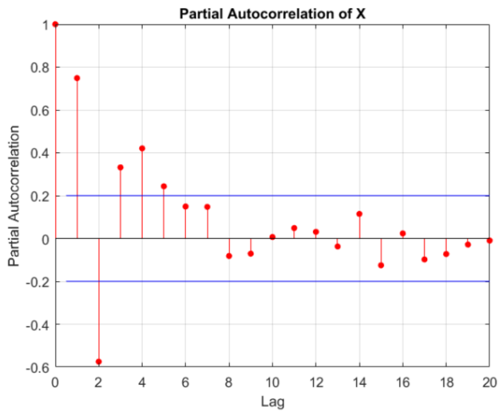


Figure 5.17: Evolution of phase x and y for blue trajectory, starting from (0.5, 4.033)

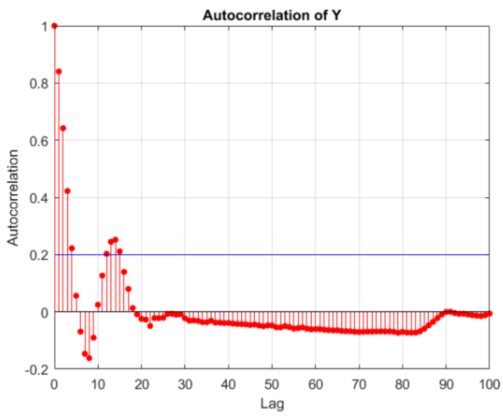
Different from the trajectory of (0.5, 4.03), this one does not show strong oscillation with periodicity within the evolution of phases. There is only some weak periodicity in the first half part of the time series. With the motion evolving, especially after the 30th time step, the pendulum becomes in the small vibration around the equilibrium position. This behavior could be attributed to the difference in final equilibrium positions. After the 30th time step, the pendulum is in low speed and comes into the near field of blue magnets. While the equilibrium position for this trajectory is one of the magnets, so the forces from the other two magnets are most likely to be counterbalanced. The pendulum will suffer a large magnetic force from the blue magnet, and a relatively small gravity force that behaves as a perturbation. Hence the vibration is formed. In the case of trajectory (0.5, 4.03), the final equilibrium position is the center of the plane. The pendulum suffers a really small gravity force and a relatively large magnetic forces from all of the three magnets. Hence the pendulum is in a noticeable periodic reciprocate motion around the equilibrium position. In Figure 5.18, the autocorrelation and partial autocorrelation of phase x and y are also studied. Similar to the trajectory of (0.5, 4.03), time series of the trajectory of (0.5, 4.033) also shows strong moving average and auto regression properties, which meets our expectation again.



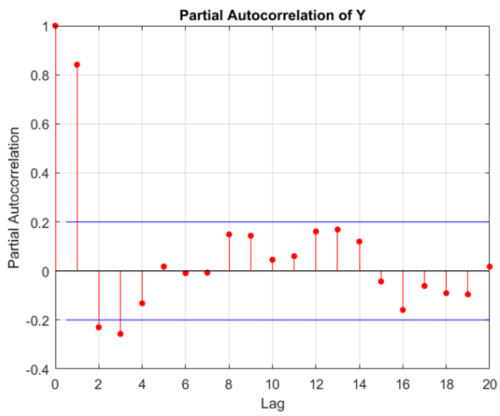
(a): Autocorrelation plot for phase X



(b): Partial autocorrelation plot for phase X



(c): Autocorrelation plot for phase Y



(d): Partial autocorrelation plot for phase Y

Figure 5.18: Autocorrelation and partial autocorrelation for phase X and Y

The Figure 5.19 and 5.20 show the power spectrum of trajectory (0.5, 4.033). Its property is also quite similar to the previous one. There are also strong direct and low frequency component. And the spectrum widely distributed across the whole frequency range.

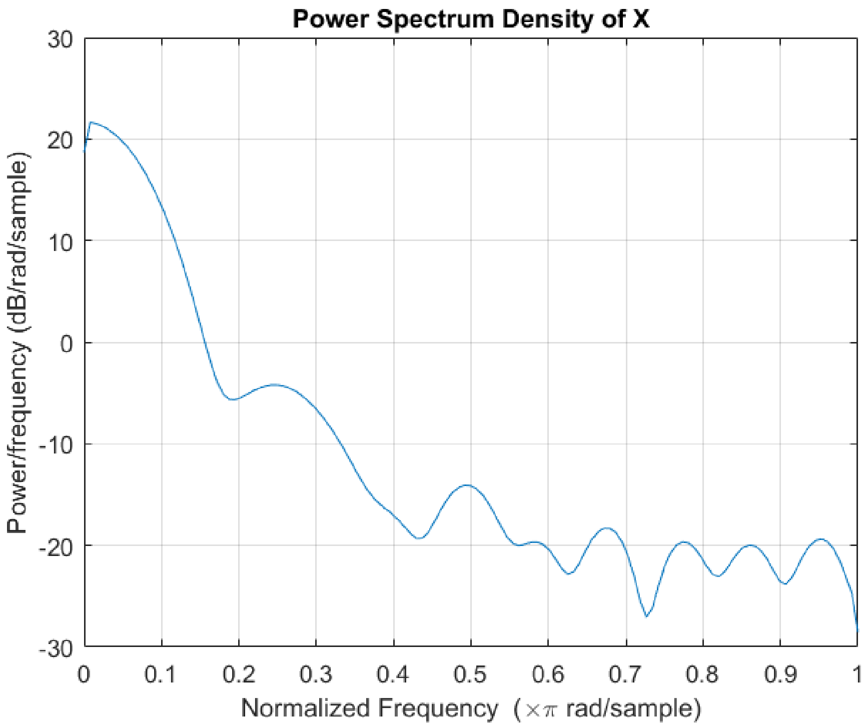


Figure 5.19: Power spectrum density of phase X

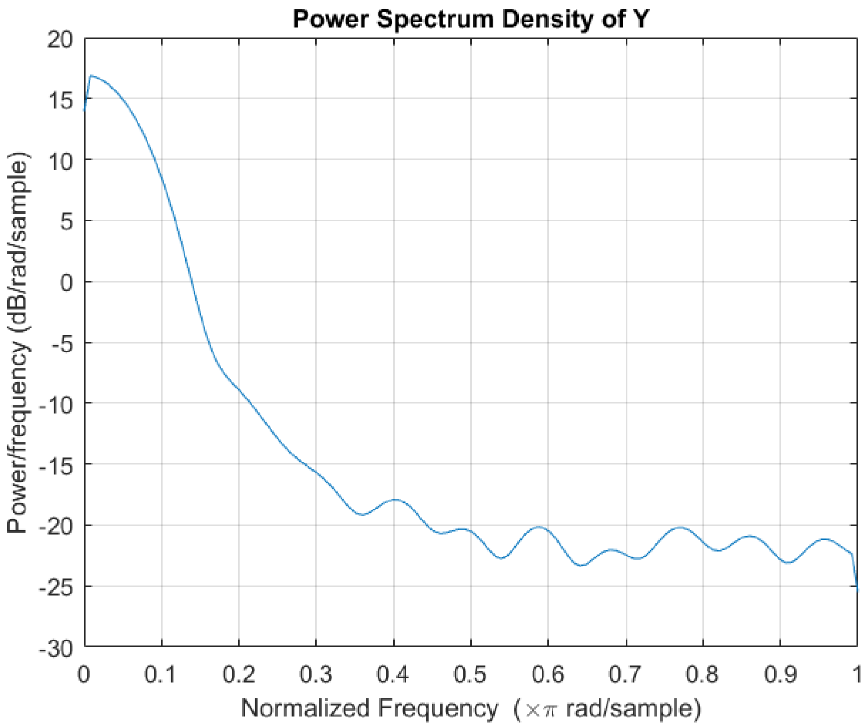


Figure 5.20: Power spectrum density of phase Y

So far, we have investigated the relation between position, velocity and acceleration. All these studies are performed in two separate dimensions, i.e. in the x and y directions only. From the motion equation of pendulum, we could notice that the force, even for one dimension, is determined by both position x and y. The table below summarizes 4 trajectories and the correlation coefficients between x and y.

Table 1: Summary of 4 different trajectories

Index	Trajectory Start Position	Equilibrium Position	Color
1	(0.5, 4.03)	(4.0, 4.0)	Black
2	(0.5, 4.031)	(4.0, 6.0)	Red
3	(0.5, 4.033)	(5.73, 3.0)	Blue

4	(0.5, 4.034)	(2.27, 3.0)	Green

We have studied several pendulum trajectories individually to show the existence of chaos. In order to numerically describe this chaos, the Lyapunov exponent is usually employed. Herein we will also use this method for the pendulum system. It is worthy to mention that we will use the norm of separation between different trajectories alternatively to extrapolate the Lyapunov exponent in this work. This is because the Lyapunov exponent, according to its definition, is the separation when time tends to infinity, which is very difficult in this work as we are using the numerical simulation method, and time is finite. Figure 5.21 shows the plots of Norm of separation between the black trajectory and the other red, green and blue trajectories, and the black trajectory is the reference.

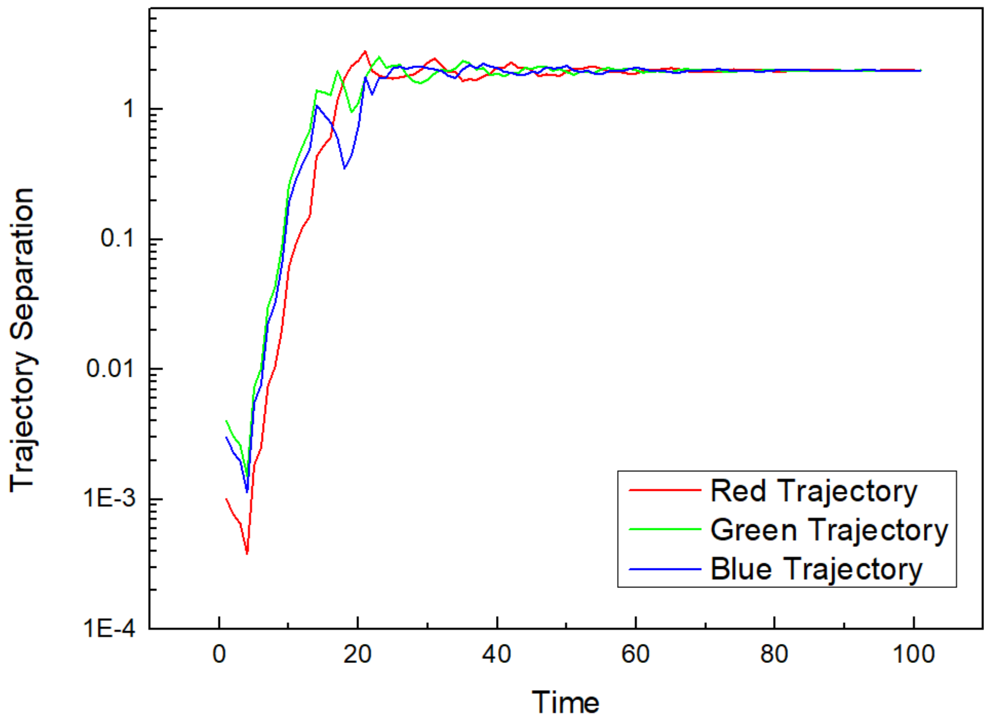


Figure 5.21: Trajectory separation, reference trajectory: Black Trajectory

The starting positions we chosen here are very closed to each other, so, it could be seen that the distances between trajectories at the beginning are quite small. With the time evolving, the differences between the black trajectory and other three increase exponentially. It means that trajectory from two nearby positions, will diverge to different separated positions, which indicates that the system is chaotic. The final separations of three lines appear to be the same value. This is reasonable, as the distances from final positions of red, green and blue trajectory are the same. In this case, we have confirmed that the final positions of the three trajectories are different. However, in general, one might deduce a wrong conclusion that the final positions are the same. Therefore, the such method is powerful to determine whether chaos is the system behavior, but it is still quite limited to show the final system status, especially when there are multiple equilibrium states. From the equation (13), the maximum Lyapunov exponent could be calculated. The values of λ_{max} for red, green and blue trajectories are 6.8931, 5.6970 and 5.8951, respectively. As we have mentioned in the beginning of the chapter, these positive values confirm the system is chaotic.

Figure 5.22 studied the trajectory separations between the red trajectory and black, green and blue ones, and the red one is the reference. It is notable that there are 2 values when time reaches maximum. This is quite straightforward as the distances from red magnet to the black one or green and blue one is different. And values of max for black, green and blue trajectories are 6.8931, 6.4220 and 6.7547, respectively.

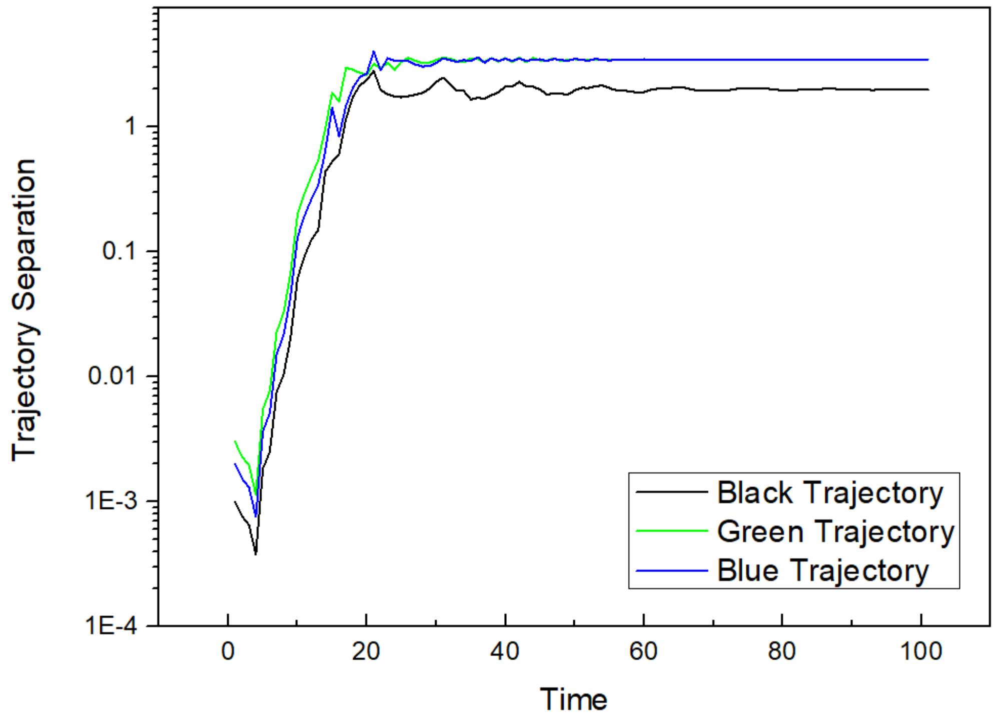
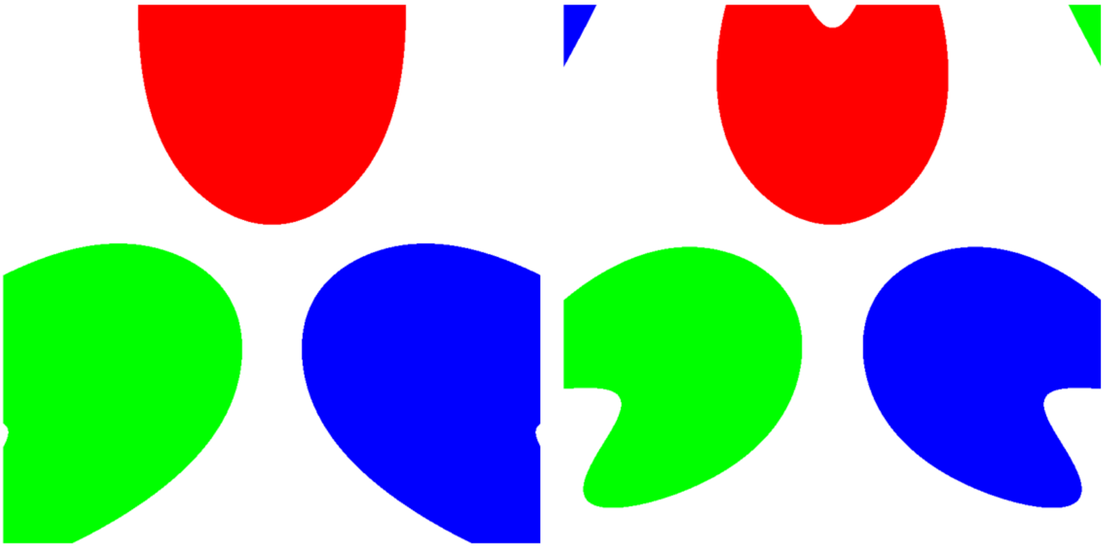


Figure 5.22: Trajectory separation, reference trajectory: Red Trajectory

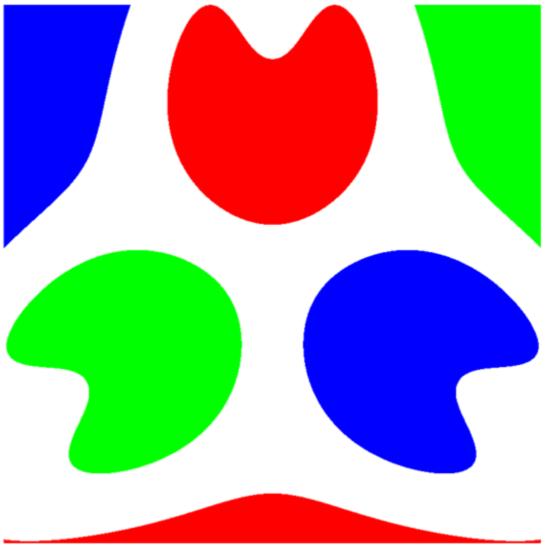
5.4 Basin Analysis

We have studied the properties of several trajectories and confirm that the system is chaotic. However, states distribution in this chaotic system is still not very clear. In addition, in the discussion above, we fixed the values of k_m, k_g or k_f. Their influences on the system is also difficult to be reflected by single or several trajectories analysis. Therefore, it is worthy to simulate the final states for the whole plane. This calculation is very time consuming, however, it could provide us a better and clearer picture for the behavior of the system.

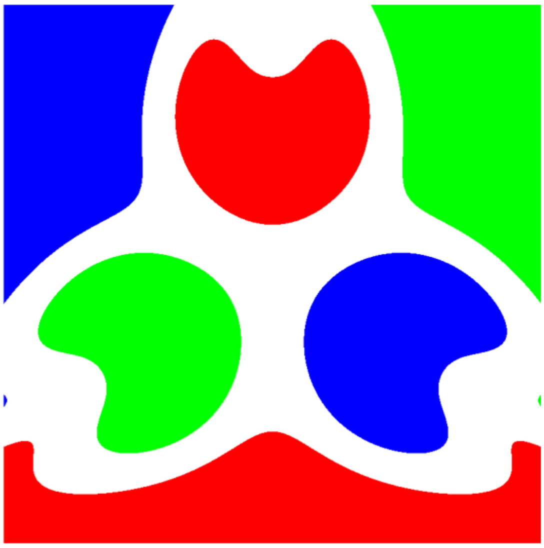
In order to reducing the simulation complexity, the maximum time step is still set to be 100. The starting position of pendulum will be set starting from (0, 0) with a space step 0.01 in both directions and finally reaches to (8.0, 8.0). So, there will be 641601 trajectories in total. The state of each trajectory will be referred to its final equilibrium position, and marked on the starting position of the trajectory. For instance, the mentioned black trajectory, which is the one from (0.5, 4.03), its final equilibrium position is (4.0, 4.0). So, the position (0.5, 4.03) will be marked as white in the basin plot. The red, green or blue colors refer to trajectory equilibrium to red, green or blue magnet, respectively. Regarding the influence from different parameters, herein we will fix the k_m to 1, and change the value of kg and k_f, in order to simplify the discussion. The basin diagrams from different values of k_f are shown in the Figure 5.23. The value of k_f is decreased from 1.0 to 0.1. k_g is taken as 0.5.



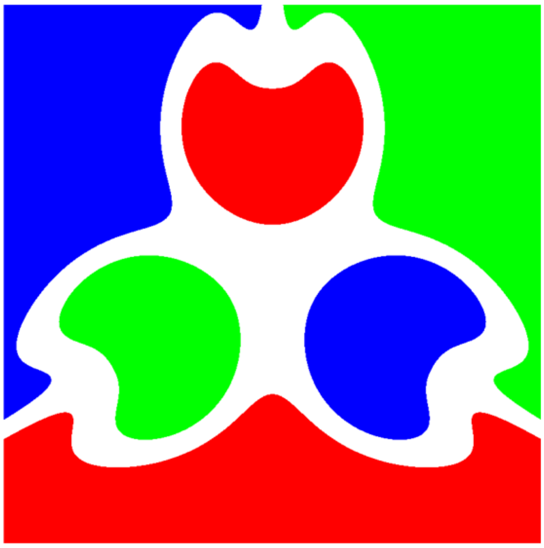
(a): Basin diagram, $k_f = 1.0$ (b): Basin diagram, $k_f = 0.9$



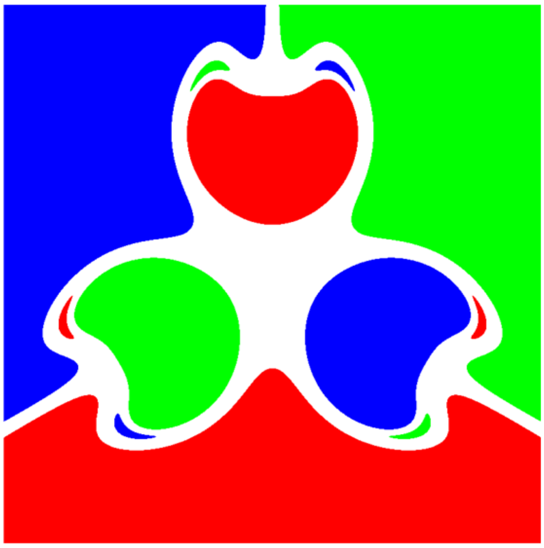
(c): Basin diagram, $k_f = 0.8$



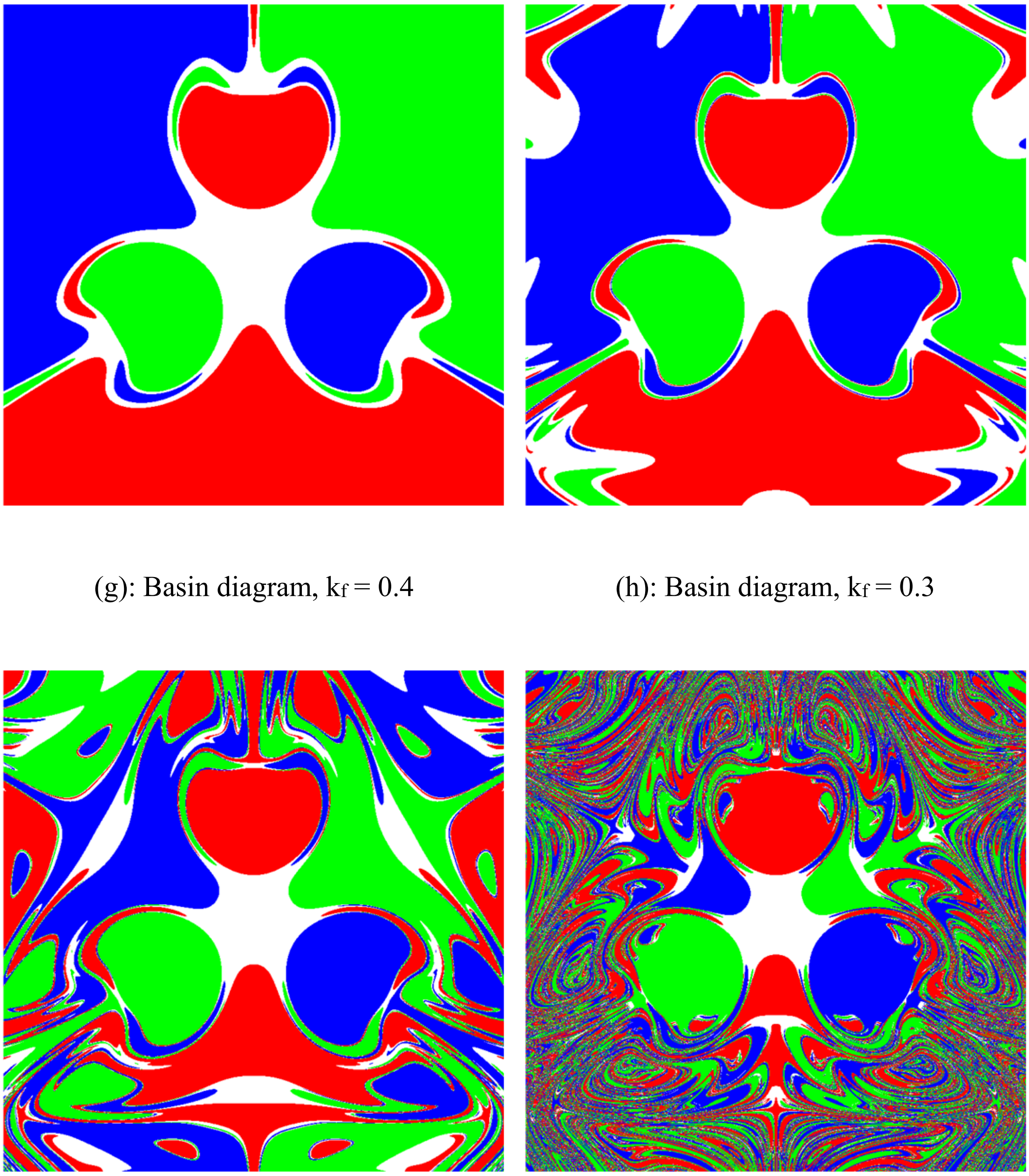
(d): Basin diagram, $k_f = 0.7$



(e): Basin diagram, $k_f = 0.6$



(f): Basin diagram, $k_f = 0.5$

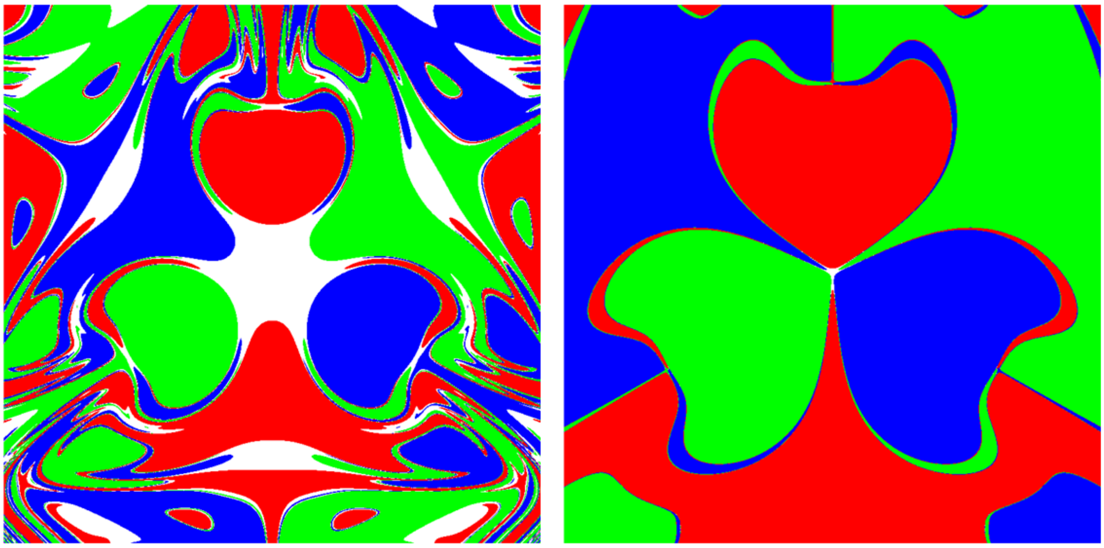


(g): Basin diagram, $k_f = 0.4$ (h): Basin diagram, $k_f = 0.3$

(i): Basin diagram, $k_f = 0.2$ (j): Basin diagram, $k_f = 0.1$

Figure 5.23: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.5$ and different k_f parameters

It could be seen that when k_f is 1.0, the whole plane consists 4 clear areas for each equilibrium position. With the decreasing of k_f, those equilibrium area (white area) for the center position and closed to the plane boarder become divisive. Some equilibrium zone for the magnet positions appears. Ultimately, when k_f is larger than 0.5, the pattern is very orderly. The boundary for each equilibrium area is well determined. The only difference is for different k_f, the area for each equilibrium position changes, due to the value change impact on different force. When k_f decreases to 0.4, an interesting phenomenon appears that equilibrium zones for other two magnets emerge at the edge of equilibrium area. With the further decreasing of k_f, these zones become larger, and the equilibrium areas closed to plane boarder become divisive. With the further decreasing of k_f, the whole plane begins to show the chaotic pattern, and finally, when k_f is 0.1, the chaos is formed eventually. From this investigation, we could observe that the formation of chaos in the system is highly related to the parameter k_f, which the friction parameter between pendulum and the air. However, a question is easily to be raised that whether the chaos is related to the k_f only. Hence, we keep the k_f as constant 0.2, and changes k_g. The result is shown in Figure 5.24.

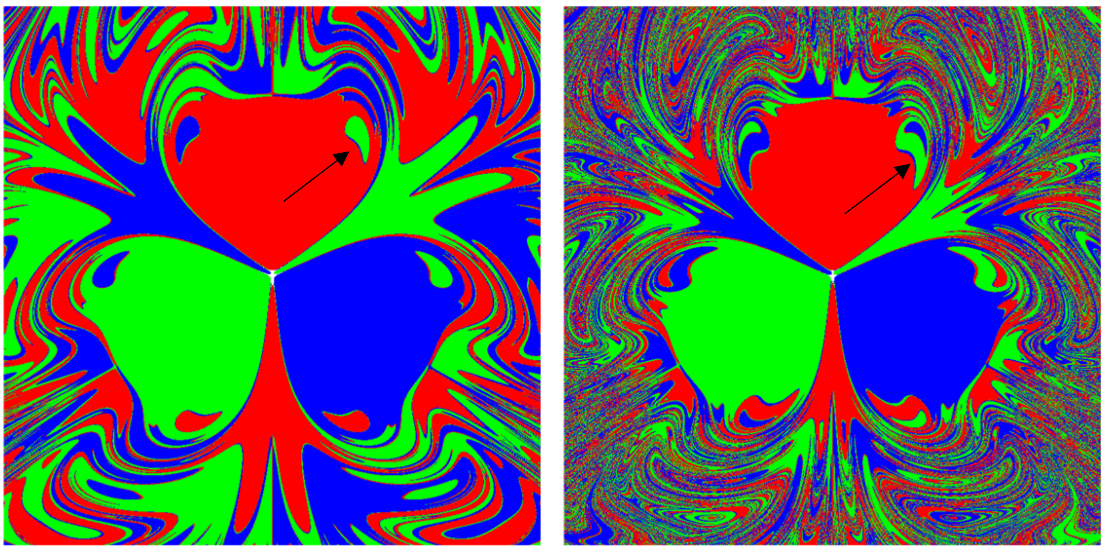


(a): Basin diagram, $k_g = 0.5$ (b): Basin diagram, $k_g = 0.2$

Figure 5.24: Basin diagram for pendulum system for km = 1, $k_f = 0.2$ and different k_g parameters

From Figure 5.24, one could observe that for the same k_f, the system behavior is also related to k_g parameter. For k_g is 0.5, the system begins to show the chaotic behavior. But when k_g is 0.2, the system only starts to become divisive. And the chaotic behavior is still far away according to our experience from previous investigation. It is also interesting to find that the equilibrium area for the center position becomes smaller when k_g decreases. This is reasonable, because when k_g decreases, the force from gravity should decrease, and its impact on the pendulum is also reduced.

However, on the other hand, k_f is still playing an important role here. When we further decrease k_f to 0.1 and keep kg as 0.2. A huge difference to the Figure 5.24(b) appears that the system becomes quite chaotic, as shown in Figure 5.25(a). In addition, it could be found that k_f is responsible for splitting the equilibrium zones or areas, as shown in Figure 5.25(b).



(a): Basin diagram, $k_f = 0.1$ (b): Basin diagram, $k_f = 0.06$

Figure 5.25: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.2$ and different k_f parameters

From Figure 5.25(a) to Figure 5.25(b), the k_f decreases from 0.1 to 0.06, and the chaos becomes more series. Taking the area marked by the black arrow in the figures as an example, the area is determined as green at the first, but becomes larger and divisive with the decreasing of k_f. An equilibrium area for another state emerges. Those areas closed to the plane board also reflects the same phenomenon. With this rule, in extreme condition, when the k_f is taken as 0, the system is extremely chaotic, which is shown as Figure 5.26.

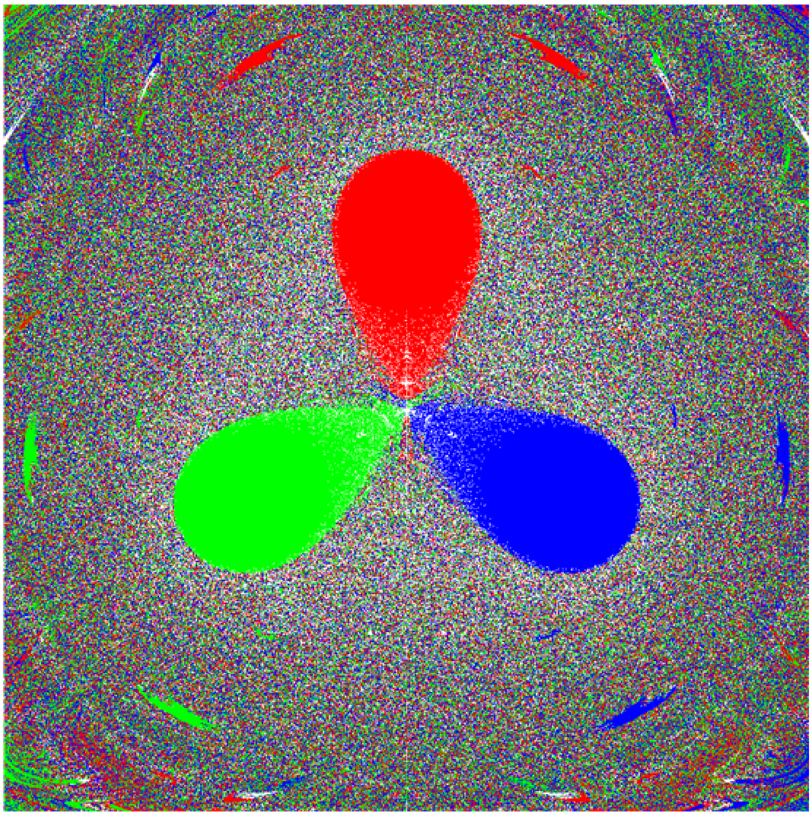
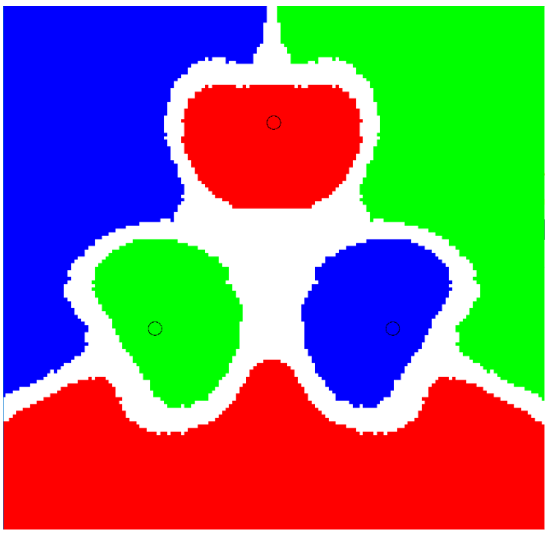


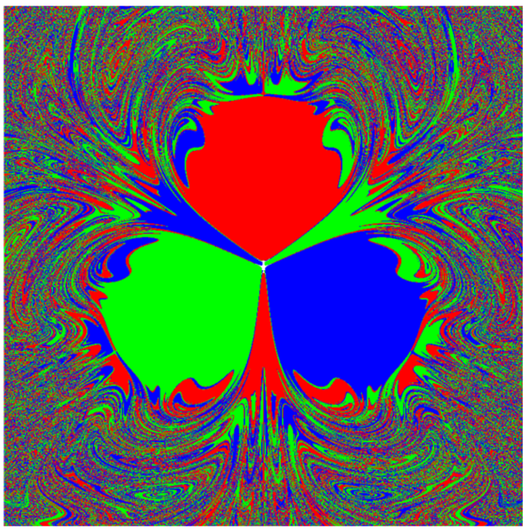
Figure 5.26: Basin diagram for pendulum system for $k_m = 1$, $k_g = 0.2$, $k_f = 0$

From Figure 5.26, we could see that the system is in a total chaotic status, except for the areas closed to the magnets positions. It is very difficult to observe any rule or pattern in most area of the plane. The information from the system is reduced. It is also worthy to mention that the status for each point in the plane is from numerical simulation of limited time step. In the practical situation, this result could be different, however, the system will still be chaotic. In this simulation, we keep each parameter as constant. In reality, especially for k_f, it could the function with time or positions, and the situation could be more chaotic.

From the above discussion, we could see that the chaos of system originally comes from the numerical relation between various parameters. The absolute value for individual parameter could influent the system, however, the relative difference is the key to trigger the chaotic phenomenon. km and k_g mainly determine the basic boundary for each equilibrium area and k_f introduce the level of chaos. Herein a practical and relatively accurate method is expected to describe the level of chaos, which will be our work in the future.

5.5 Compare Between Two Method





(a): Basin diagram from program, k_f = 0.05, k_g = 0.5

(b): Basin diagram from numerical simulation, k_f = 0.05, k_g = 0.5

Figure 5.27 Basin diagram from program and numerical simulation

In Figure 5.27, one could see that the basin diagram from our program and numerical simulation, we attribute this to the difference between the algorithm we used and that of Mathematica. In addition, in our program, in order to reduce the simulation complexity, we reduced the computation precision. This could be another reason for the difference. However, our program also shows the same rule of changing as the numerical simulation when parameters are changed, which has shown that our program, regardless of precision, is self-consistent. In the future work, we will improve our program with other algorithms, and refactor some program framework to achieve higher precision.

Chapter 6: Summary and Future Work

In this work, we started with a brief review of system concepts and chaotic system history, which is the main content of Chapter 1. Chaos is the complicated temporal behavior of simple systems. According to this definition, chaos is a type of motion, or more generally a type of temporal evolution and dynamics. The world around us is full of such phenomena that seem irregular and random in both space and time. Exploring the origin of these phenomena is usually a hopeless task due to the large number of elements involved; therefore, instead of the traditional method, the system, especially the deterministic chaotic system, could be studied from its physical properties. Thus, from these properties, one could potentially determine the intrinsic states of the system. A relationship between the initial condition and final states could be achieved by the tradition iteration method. Once this relationship is established, we could perform the predetermination on the system through the relationship, which is the key sprit of this work and proposed in Chapter 2, after a brief review of traditional method of studying chaotic system. In Chapter 3, we introduced the concepts and model our experimental system, the magnetic pendulum system. The symbolic system is also established in this chapter. In Chapter 4, we designed and implemented a computer program for visualization of pendulum movement in the magnetic and gravity field. The friction is also considered. In Chapter 5, in addition to the results from our program, we used another numerical simulation method to investigate the magnetic pendulum system. From the basin diagram of our program and that of numerical simulation, we have proved that the state distribution could be used as the relationship of initial conditions and final states. One could predetermine the final state of any initial positions in the system without considering the possible complicated middle process. In addition, from our study, we build the links between the basin diagram and system parameters. Those parameters related to the final states, such as k_m and k_g, would change the effect area of each fixed points, while the parameter related to the middle process would have a large impact on the state distribution.

For future work, we would keep improving our algorithm and program framework, so that a more precise basin diagram could be achieved. So far, in our work, we only support 3 magnets. In later work, we will improve our program to support multiple magnets. In addition, which is also more important, we will employ other chaotic systems to validate our preliminary approach. It is also expected that our approach could be used on data coming from real world, such as stock markets. Due the huge quantity of data, some new technology, such as methods from data science, is also necessary in the future.

References

[1]. K. T. Alligood, T. D. Sauer and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems* 1997 Berlin, Heidelberg: Springer Berlin Heidelberg

[2]. D. K. Arrowsmith and C. M. Place, *An introduction to dynamical systems* 2001 Cambridge University Press

[3]. F. Diacu, *Celestial encounters : the origins of chaos and stability* 1996 Princeton, N.J. : Princeton University Press

[4]. P. G. Drazin, *Nonlinear systems* 2002 Cambridge University Press

[5]. J. Gleick, *Chaos : making a new science* 1988 New York, N.Y., U.S.A. : Penguin

[6]. L. C. a. Y. Liu, *Nonlinear Dynamics* 2000 Shanghai Jiaotong University Press

[7]. E. N. Lorenz, *The essence of chaos* 1993 Seattle : University of Washington Press

[8]. H.-O. Peitgen, H. Jürgens and D. Saupe, *Chaos and Fractals: New Frontiers of Science* 2004 New York, NY: Springer New York

[9]. I. Prigogine, *Order out of chaos : man's new dialogue with nature* 1984 New York : Bantam Books

[10]. I. Newton, *Philosophiae naturalis principia mathematica* 1687 Londini: J. Societatis Regiae ac Typis J. Streater

[11]. H. Poincaré and B. D. Popp, *The three-body problem and the equations of dynamics : Poincaré's foundational work on dynamical systems theory* 2017

[12]. G. Duffing, *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung* 1918 Vieweg

[13]. B. van der Pol, *LXXXVIII. On "relaxation-oscillations".* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1926. **2**(11): p. 978-992.

[14]. S. H. Strogatz, *Nonlinear dynamics and chaos with applications to physics, biology, chemistry, and engineering* 2015 Westview Press

[15]. L. Brillouin, *Scientific Uncertainty, and Information.* 2015.

[16]. H. M. Morse, *Recurrent geodesics on a surface of negative curvature* 1947

[17]. S. Smale, P. M. Pardalos and T. M. Rassias, *Essays in mathematics and its applications : in honor of Stephen Smale's 80th birthday.* 2012.

[18]. A. Kolmogorov, *On the conservation of conditionally periodic motions under small perturbation of the Hamiltonian.* Journal, Year. **98**(527): p. 2-3.

[19]. T. Tél, *Chaotic dynamics : an introduction based on classical mechanics* 2006 Cambridge : Cambridge University Press

[20]. E. N. Lorenz, *The mechanics of vacillation.* Journal of the atmospheric sciences, 1963. **20**(5): p. 448-465.

[21]. M. Henon, Heiles, Carl, *The Applicability of the Third Integral of Motion: Some Numerical Experiments.* Astron.J., 1964. **69**: p. 73-79.

[22]. M. Henon, *A two-dimensional mapping with a strange attractor.* Comm. Math. Phys., 1976. **50**(1): p. 69-77.

[23]. D. Ruelle, *Chaotic Evolution and Strange Attractors* 1989 Cambridge University Press

[24]. B. R. Hunt and J. A. Yorke, *The theory of chaotic attractors* 2011 Springer

[25]. R. M. May, *Simple mathematical models with very complicated dynamics.* Nature, 1976. **261**(5560): p. 459-467.

[26]. R. L. Devaney, *An introduction to chaotic dynamical systems.* 2018.

[27]. S. Wiggins, *Introduction to applied nonlinear dynamical systems and chaos* 2003 Springer
[28]. M. Benedicks, Carleson,L., *The dynamics of the Henon map.* Annals of Mathematics, 1991. **133**(1).
[29]. T. Von Kármán, *The engineer grapples with nonlinear problems* 1944 Durand Reprinting Committee, California Institute of Technology
[30]. P. J. Brockwell and R. A. Davis, *Time series : theory and methods* 2013 Springer
[31]. J. D. Hamilton, *Time series analysis* 2012 Levant Books
[32]. W. A. Fuller, *Introduction to Statistical Time Series.* 2009.
[33]. *The Chaos Hypertextbook* https://hypertextbook.com/chaos/
[34]. S. B. Lippman, J. Lajoie and B. E. Moo, *C++ Primer* 2012 Pearson Education
[35]. H. Schildt, *Java : A Beginner's Guide, Sixth Edition.* 2014.
[36]. A. Martelli, *Python in a Nutshell* 2017 O'Reilly Media, Inc, USA
[37]. *QT* https://www.qt.io/
[38]. H. Schildt, *Swing : a beginner's guide* 2007 McGraw-Hill
[39]. B. Eckel, *Thinking in Java.* 2015.
[40]. D. Beeman, *Some multistep methods for use in molecular dynamics calculations.* Journal of Computational Physics, 1976. **20**(2): p. 130-139.
[41]. *WOLFRAM MATHEMATICA* https://www.wolfram.com/mathematica/
[42]. *Ian Win* https://www.math.hmc.edu/~dyong/math164/2006/win/finalreport.pdf