

Math 164 Final Project: The Chaotic Oscillating Magnetic Pendulum

Ian Win

Spring 2006

1 Introduction

1.1 Background

Chaotic behavior is exhibited in many situations, ranging from very complex mechanical systems to everyday occurrences. For instance, chaos can be seen in examples such as species populations, the forced Duffing Equation, or even the weather. In fact, the theory of chaos began in the 1960s when a meteorologist named Edward Lorenz left out a few decimal places when running a trial for his weather simulator.

Although no formal mathematical definition of the word chaos is universally accepted, it can be described as the long-term, aperiodic behavior of a deterministic system which displays sensitive dependence to initial conditions. This idea of chaos is exhibited in the particular system that I chose to study, the chaotic oscillating magnetic pendulum. In this system magnets are used to drive the oscillation of a free moving pendulum with an essentially spherical range of motion.

1.2 The Problem

The pendulum consists of a magnet suspended from a string. The plane under the pendulum contains a distribution of like magnets which, based on their number and placement, should affect the dynamics of the pendulum. The orientation of all the magnets in the system, including the magnet at the end of the pendulum, is the same. For each initial position, the trajectory of the pendulum eventually stabilizes around one of the plane magnets. However, in the interim the motion of the pendulum is chaotic, with basins of attraction for different magnets separated by fractal curves in the plane.

The overall intent of this project is to investigate the dynamics of the chaotic oscillating magnetic pendulum for various numbers of magnets. This objective will be accomplished by various means. First, we will build a model of the magnetic pendulum in order to visually analyze the chaotic behavior of the system. The model, however, is limited in its use because of the fractal nature of the basins of attraction (other than a few of the larger basins, it is impossible to measure accurately enough the initial position of the pendulum, to see if it coincides with theory for most positions in the plane).

A more analytic approach will be used to analyze the chaotic behavior of the system. Programs will be written in MathematicaTM to investigate the chaotic attributes of trajectory behavior. But perhaps more important to understanding the pendulum's chaotic nature, MathematicaTM is used

to plot various basins of attraction for different numbers of magnets, different parameter values, and even different magnetic positions.

1.3 Motivation

There is a larger reason why I chose to take on this specific project. As a group project with Owen Lewis and Jeff Hellrung, I carried out analysis on this very problem. We used several MatlabTM programs to explore this chaotic system in the case of four magnets. Our analysis however, was problematic. To numerically integrate in MatlabTM we used ODE45 to perform the numeric integration. This is a generally good method for most types of problems, but computationally inadequate to deal with the intricacies of the fractal boundaries which surround the basins of attraction in the system. As well, with a symmetric, even positioning of magnets in the plane, the basins should theoretically be identical with respect to quadrants (disregarding color). However, our MatlabTM routines consistently provided us with basins that had a swirling effect throughout the plane, as seen in Figures 1 and 2.

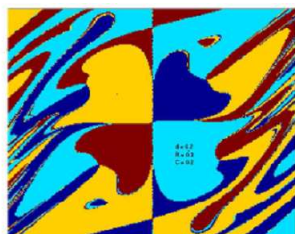


Figure 1: Basins of attraction on $[-4, 4] \times [-4, 4]$ with $R = 0.3, C = 0.2$

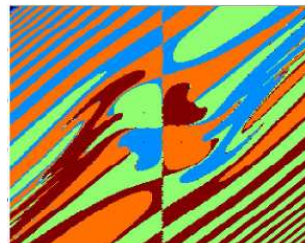


Figure 2: Basins of attraction on $[-8, 8] \times [-8, 8]$ with $R = 0.3, C = 0.2$

While the overall goal behind the project is to investigate the behavior of the pendulum for multiple numbers and positions of magnets and various parameter values, the main motivation behind my choosing the particular problem is to improve upon my prior work in this area and possibly analyze the reasons for the minor shortcomings of the analysis.

MathematicaTM was chosen in order to facilitate the process of determining the best numerical integrator with which to solve the problem. However, a major benefit of applying MathematicaTM to this problem was realized through its ability to represent symbolically as well as numerically.

2 Derivation of the Governing Equations

(The following derivation was taken from Chaos and Fractals: New Frontiers of Science by Peitgen, Heinz-Otto; Jürgens, Hartmut; and Saupe, Dietmar, Springer-Verlag New York, Inc., 1992.)

2.1 Assumptions

- The length of the pendulum is large compared to the spacing of the magnets. This allows one to assume that the pendulum bob is confined to a plane rather than a sphere.
- The magnets are point attractors positioned in a plane a small distance below the pendulum.

- Magnetic forces follow an inverse squared law; i.e. the force is inversely proportional to the square of the distance.

2.2 Parameters

- (x, y) : the Cartesian coordinate of the pendulum bob
- (x_i, y_i) : the Cartesian coordinates of the magnets
- d : the vertical distance from the pendulum bob to the plane in which the magnets lie.
- R : the friction force coefficient.
- C : the gravitational (spring) force coefficient

2.3 The Model

We define the origin of the Cartesian coordinate system to be the gravitational equilibrium position of the pendulum bob, and specify coordinates of the magnets relative to this origin. The distance between the pendulum bob and magnet i is $\sqrt{(x_i - x)^2 + (y_i - y)^2 + d^2}$. Hence the magnetic force is proportional to $\frac{1}{(x_i - x)^2 + (y_i - y)^2 + d^2}$. However, we must ignore the vertical component of this force, as we assume the pendulum bob is restricted to a plane. Taking this into account, it can be shown that the x and y components of the force are

$$\begin{aligned}\vec{F}_x &= \frac{x_i - x}{((x_i - x)^2 + (y_i - y)^2 + d^2)^{3/2}}, \\ \vec{F}_y &= \frac{y_i - y}{((x_i - x)^2 + (y_i - y)^2 + d^2)^{3/2}}.\end{aligned}$$

The gravitational force that pulls the pendulum bob back to the origin is proportional to the distance from the origin; thus, the x and y components are proportional to $-x$ and $-y$ respectively. The friction force acts in opposition to the direction of motion and is proportional to the velocity (x', y') .

Using Newton's Law, we equate the sum of the above three forces to the acceleration of the mass, and after rearranging, we arrive at our governing equations:

$$\begin{aligned}x'' + Rx' - \sum_i \frac{x_i - x}{((x_i - x)^2 + (y_i - y)^2 + d^2)^{3/2}} + Cx &= 0, \\ y'' + Ry' - \sum_i \frac{y_i - y}{((x_i - x)^2 + (y_i - y)^2 + d^2)^{3/2}} + Cy &= 0.\end{aligned}$$

3 Implementation

To investigate this problem several MathematicaTM notebooks were used. The actual notebooks are found in Appendix A.

3.1 Documentation

3.1.1 Basin.nb: Notebook to produce basin plots

This MathematicaTM notebook allows the user to first specify parameter values and the coordinates of the magnets. It then uses, NDSolve to calculate the solution to the system of equations for a grid of points in the plain, the size of which (and general runtime of the simulation) is governed by the choice of n in the notebook (small $n \rightarrow$ faster, but inaccurate, large $n \rightarrow$ slower, but precise). For each trajectory calculated by NDSolve, the resultant end value is determined to be “closest” to a certain magnet and assigned a color value based on that magnet. The resultant Table of color values formed is then plotted as a RasterArray (in MathematicaTM, an array of values each corresponding to some color). It should be noted that the denominator found with the style option *Hue* should must be greater than or equal to the number of magnets being used in the simulation to obtain interpretable results (i.e. two magnet’s basins will have the same color). It is also important to note that NDSolve outputs an answer in terms of interpolating functions, thus *solution* is a tensor, not a standard vector of numerical values.

3.1.2 Modified_Basins.nb: Notebook to calculate modified basins

This notebook is nearly identical to Basins.nb. The only difference is that a second vector is defined representing the fixed points of the system (these correspond to the Cartesian coordinates of the magnets and the origin). The calculation of minimum distance is then carried out with this new vector which includes the origin, and the basin plot is colored accordingly. Please not that the denominator value within the style option *Hue* must be strictly greater than the number of magnets in order to obtain a sensible picture.

3.1.3 Trajectories.nb: Notebook to plot trajectories

This MathematicaTM notebook computes a numerical solution to the chaotic pendulum problem for the specified initial condition (x_0, y_0) and plot the trajectory parametrically. Parameters can be varied in the same fashion as Basins.nb. The second part of the notebook will compute and overlay different trajectories, in order to graphically analyze the divergence. This part is essentially identical to the first only it calculates multiple trajectories with NDSolve and then overlays the plots.

3.1.4 Lyapunov.nb: Notebook to plot divergence of nearby trajectories (Lyapunov Exponent)

This notebook is the same as Trajectories.nb up through the calculation of the trajectory. However, two trajectories of interest are specifically calculated with NDSolve. Then the norm of the difference between each of these trajectories is calculated for each time value t . The log of this norm is then plotted against time. If the incline of this graph has an average positive slope then we know by theoretical proof that the trajectories diverge exponentially. Furthermore, with the choice of initial condition, by picking two very close initial conditions as input to NDSolve, one can prove through the exponential divergence of trajectories a sensitive dependence on initial conditions.

3.2 Mathematica™

Mathematica™ was chosen for to numerically solve this problem for a couple of reasons. There are few mathematical programs available with the capability to symbolically compute. The newer versions of Maple™ have this capability. However, Mathematica™ can symbolically compute as well, but it has the much of the analytic power of programs like Matlab™. As well, an added benefit to this is Mathematica's ability to "choose" which method of numerical integration is best suited for solving this particular problem (remember that one of the problems with the previous project was insufficient numerical integration methods).

4 Results

4.1 Chaotic Behavior

First let us simply examine some pictures of trajectories obtained from Mathematica™. Before we go on however, it is important to note a few things. In all cases (except for our first example) we assume that the pendulum swings a constant distance $d = 0.2$ above the plane containing the magnets, which, in most cases, are distributed evenly around a circle of radius 2. Furthermore, in all trials we assume that the pendulum is released from a stationary starting position. That is to say that the initial condition (x_0, y_0, x'_0, y'_0) takes the form $x'_0 = y'_0 = 0$. In Figure 3, we see four different trajectories for a four magnet pendulum.

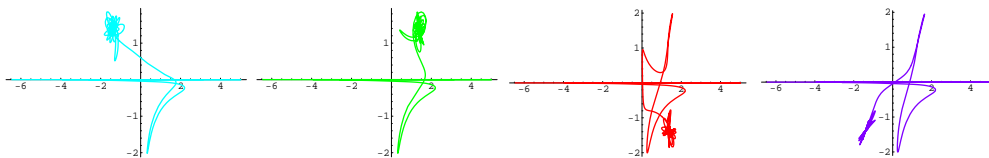


Figure 3: Four Trajectories; $R = .07$ and $C = .2$

These trajectories each have an initial x -coordinate, $x_0 = -6.45843$. Left to right, the initial y -coordinates for each of these trajectories are $y_0 = 0.000351, 0.00035, 0.000345,$ and 0.0003445 respectively. Notice the extremely close proximity of each of the starting positions of these four trajectories. However, they each come to rest over an entirely different magnet. Notice that the pendulum swings in a wildly erratic (highly nonlinear) manner, but quickly settles to an odd decaying oscillation about one of the equilibrium points. While no single trajectory certainly seems to be any easily defined curve, we can determine nothing about the chaotic nature of the system from any one of them alone. However, when we view them together, we can see that the final resting position of the pendulum is highly sensitive to initial conditions (Figure 4).

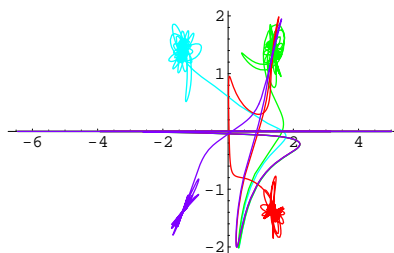


Figure 4: Overlay Plot of Figure 3

To make this qualitatively clear, consider the function $Z(t) = \|x_1(t) - x_2(t)\|$ where $x_1(t)$ and $x_2(t)$ are two solutions corresponding to two different initial values. Then $Z(t)$ is just a real valued function representing the distance between the two trajectories at time t . While $Z(0)$ is very small for the trajectories we will be considering, we find that $Z(t)$ grows exponentially fast as time increases. Let us take for example the first two trajectories calculated in Figure 3 (i.e. cyan and green; $(x_1, y_1) = (-6.45843, 0.000351)$ and $(x_2, y_2) = (-6.45843, 0.00035)$). Below we see a graph of $\log Z(t)$ vs. time for these trajectories.

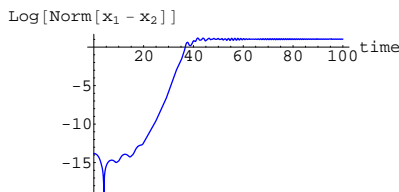


Figure 5: Lyapunov exponent for $(x_1, y_1) = (-6.45843, 0.000351)$ and $(x_2, y_2) = (-6.45843, 0.00035)$

The positive slope of the graph for approximately $t < 40$ indicates a positive Lyapunov exponent, meaning our trajectories diverge exponentially fast, and thus the system exhibits sensitive dependence on initial conditions. $Z(t)$ eventually stops growing when each trajectory settles over its respective pendulum.

4.2 Basins of Attraction

The beauty of the chaotic behavior of the pendulum is best seen through its basins of attraction. Each basin of attraction in the plane has a boundary defined by fractal curves and creates a beautiful picture physically and mathematically. It is important to remember, however, that the model is limited in its use because of the fractal nature of the basins of attraction (other than a few of the larger basins, it is impossible to measure accurately enough the initial position of the pendulum, to see if it coincides with theory for most positions in the plane).

4.2.1 Basins for multiple magnets

Below we see the basins of attraction for various numbers of magnets.

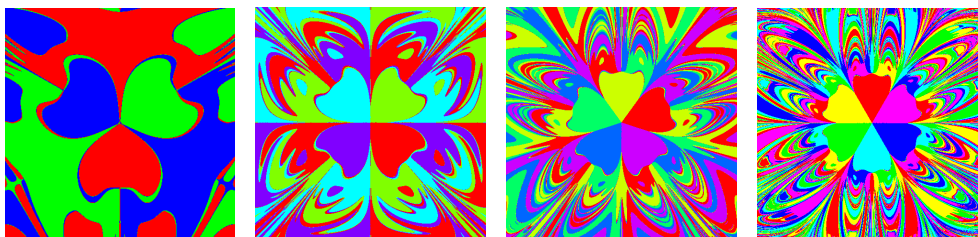


Figure 6: Four Basins $[-6, 6] \times [-6, 6]$ - #magnets = 3, 4, 5, and 6 respectively;
 $R = .2$ and $C = .2$

By looking at these plots, we notice that for the same parameter values, the basins begin to look much more complicated with the addition of more magnets. We can analytically examine this

by plotting the log of the norm of the difference between two divergent trajectories versus time for each number of magnets. In the figure below, we see the graph of the Lyapunov exponent for 3, 4, 5, and 6 magnets, respectively.

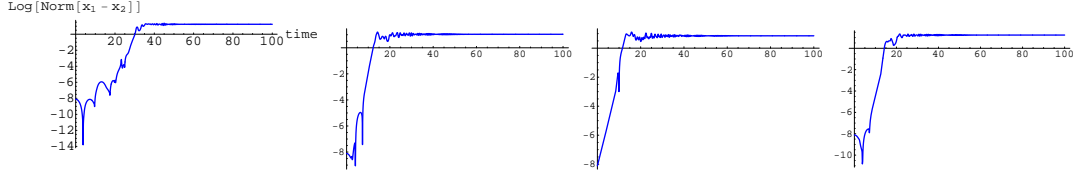


Figure 7: $\log(\|x_1 - x_2\|)$ vs. time for 3, 4, 5, and 6 magnets; For each graph $R = .1$, $C = .2$, $\Delta y_0 = 0$, $\Delta x_0 = 0.0003$.

From Figure 7, we can see that as we increase the number of magnets, we see an increase in the slope of the graphs on the interval $t = [0, 20]$. While this is visually apparent, it may still be slightly analytically unclear. While there is a method for calculating the the largest Lyapunov exponent of a system, it is not examined in this paper. For the purpose of our analysis, the visual estimation of the systems' Lyapunov exponents is adequate.

4.2.2 Basins for varied parameter values

The diversity of these basins does not stop at a variation of the number of magnets. We can cause changes in basins of attraction through varying each parameter, from the friction coefficient to the positions of the magnets. Below we see a few basins for three magnets, each with a different friction force coefficient.

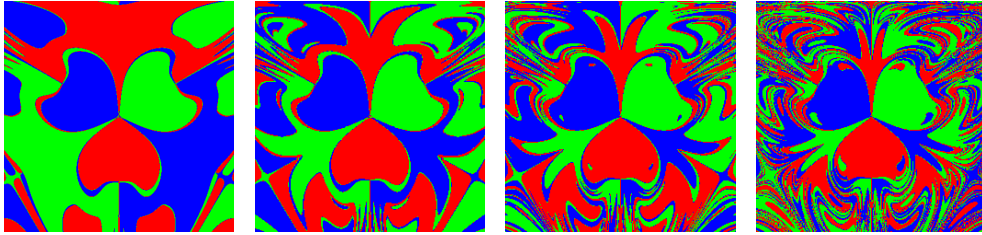


Figure 8: Basins for 3 magnets, $C = .2$: 1) $R = .2$, 2) $R = .15$, 3) $R = .125$, and 4) $R = .1$, respectively

The basins become much more chaotic as we decrease our friction force coefficient. We can look at this qualitatively in terms of a difference in Lyapunov exponents.

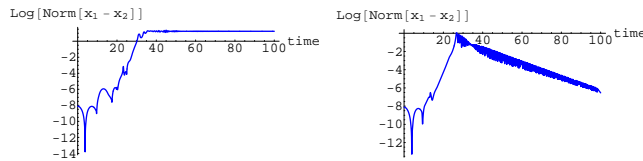


Figure 7: $\log(\|x_1 - x_2\|)$ vs. time for $(x_1, y_1) = (4.0003, 8.5)$ and $(x_2, y_2) = (4, 8.5)$: left- $R = .1$, right- $R = .15$

The exact same trajectories which diverge to different magnets for a value of $R = .1$ converge to the same magnet for $R = .15$. This clearly demonstrates an increase in the chaotic behavior of the system as the friction force coefficient decreases (which coincides with theory and logic). Next we see a few basin plots for 6 magnets, each with a different friction force coefficient.

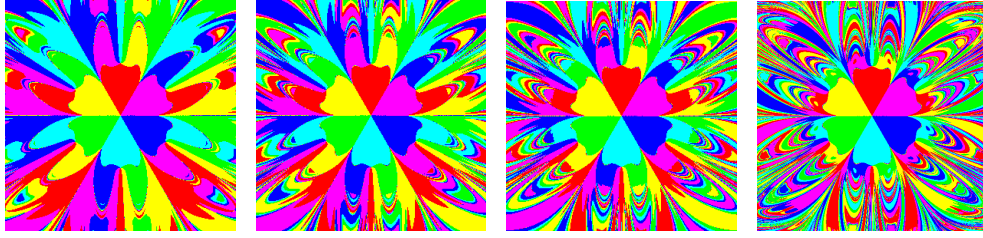


Figure 8: Basins for 6 magnets, $C = .2$: 1) $R = .3$, 2) $R = .275$, 3) $R = .25$, and 4) $R = .2$, respectively

There are a couple of important things to notice here. First, notice that to obtain similar pictures (chaotically speaking) to those of the 3 magnet basins, we had to raise the value of the friction coefficient. The qualitative reason for this is the same as before, adding magnets increases chaos. Second, notice here that the friction force coefficient value where larger basins are overtaken by their more chaotic boundaries is approximately double that in the case of three basins. While this presents an odd coincidence, more investigation would need to be done to fully quantify such a relation.

However, R is not the only parameter which we can vary. We can vary the gravity coefficient, the height of the pendulum bob, and even the positions of the magnets. By varying the gravity coefficient and the height of the pendulum bob, we see a very interesting effect. If C or d is raised enough, the unstable fixed point at the origin becomes stable. In the following figure, we see the effect of increasing C on the 4 magnet system.

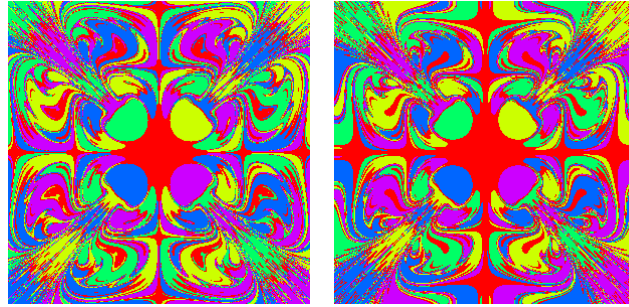


Figure 9: Basins for 4 magnets, $R = .2$: left - $C = .6$, right - $C = .7$

In these figures, the red basin represents trajectories which are pulled down to the origin. It is clearly shown from the figure that as we increase the gravity force coefficient, the origin becomes a stronger stable fixed point. Theoretically, we can imagine this as the force of gravity eventually overpowering the force of the magnets. As well, a similar effect is seen when increasing the height of the pendulum bob; as the distance between between the bob and the magnets increases, the force from each magnet grows smaller until the force of gravity is large enough to predominate parts of the plane.

Finally, let us see some unusual basins, achieved through odd magnet placements in the plane. In the following diagram each of the basins is created with parameter values of $R = .2$ and $C = .2$.

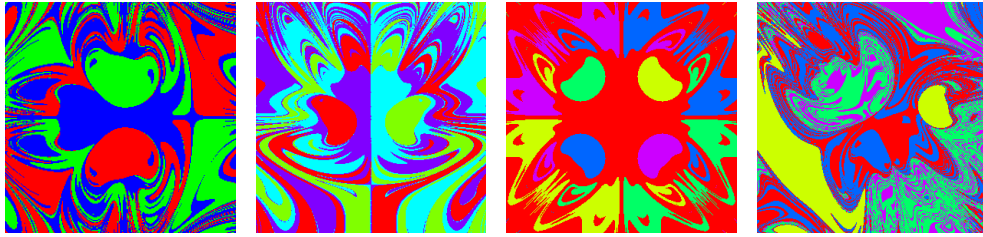


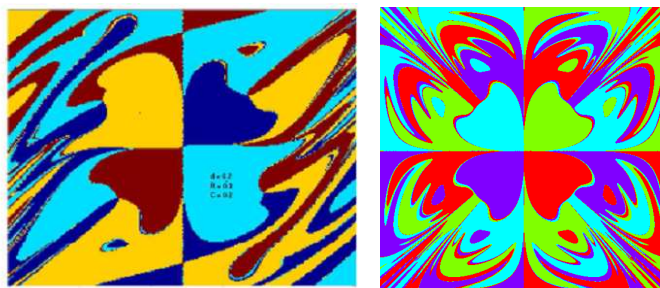
Figure 10: Basins for uneven magnet positions

In the last basin we see how a fairly “random” placement of magnets in the plane causes the basins to interact. The yellow basin is a solitary magnet and thus has very clearly defined basins. The red and the blue basins are placed very close to each other, in a side by side fashion, and we can see their basins begin to smush together. Then there are the green and purple basins, which are placed one nearly directly behind the other from the origin. We can see how the two basins have become almost intertwined in their location in the plain. It seems that magnet placement has as much of an effect on the location of the basins as does the interaction with other magnets, but this seems like a fairly hard theory to test. Then in the third basin plot, we have the normal distribution of four magnets in the plain, and then the addition of a fifth magnet at the origin. It is important to notice the similarity here between the gravity basins and this basin plot. It seems to resemble a very structured gravity basin plot, which makes sense, because we are essentially introducing the same stable fixed point which gravity does, but leaving a relatively strict parameter set, so the basins maintain a more structured form and do not succumb to dispersion as in the case of most extreme parameter variance.

In the first two basin plots, 3 and 4 magnets, respectively, are placed in an even distribution around a half circle in the plane whose center is located at the origin (considering a full circle). Notice how the uneven location of magnetic force causes a swirl in the plane in the direction of the force. This is a very enlightening result. As I mentioned earlier, when I initially started this project, the basins achieved through MatlabTM had a swirling effect in the plot (see Figure 1). The basins in Figure 10 help us understand this numerical result. MatlabTM represents everything numerically (i.e. float, double precision), so in defining the positions of the magnets, MatlabTM can only represent these coordinates within a certain precision. Thus if the magnets are not represented as the exact same value in MatlabTM, there will be a bias in the influence of the magnetic force in some fashion, that will alter the shape of the basins of attraction. Theoretically, the basins should be producing identical images per each quadrant (not considering color). Thus we can attribute the original error in basin calculation to the inaccuracy of numerical approximation.

5 Conclusion

Mathematica™ and the coupling of symbolic and numeric computation were effectively able to produce theoretically sound basins of attraction, rectifying the problem of the previous computational work done on this problem.



From the simulations run, it can be seen that the dynamics of the “chaotic oscillating magnetic pendulum” do indeed satisfy sufficient conditions for chaotic behavior. Upon discovering a positive Lyapunov exponent for the system, I can assert that nearby trajectories in the plane will diverge at an exponential rate, showing very sensitive dependence to initial pendulum placement. Similarly, no periodic behavior was witnessed in any trial. As we increase the number of magnets, the chaotic behavior of the system becomes more extreme. As well, varying the parameters can have interesting effects on the basins of attraction and divergence of trajectories, even causing the origin to become the predominant stable fixed point. If I were to continue this experiment I would like to investigate slightly perturbed magnet positions in the plane (i.e. almost evenly placed around circle) in order to truly gauge whether or not the original basin pictures can be attributed to inaccuracies in numerical calculation. As well, I would like to actually numerically (or analytically) calculate a largest Lyapunov exponent for each number of magnets to qualify the change in chaotic behavior of the system by adding or subtracting magnets. As well, the difference in chaotic behavior between symmetric and non symmetric positioning of magnets could be qualified.

Appendix A Mathematica™ Code

Appendix A.1 Basin.nb

```
n = 250; d = 0.2; R = 0.15; g = 0.2; mags = {{Sqrt[3], 1}, {-Sqrt[3], 1}, {0, -2}};
f[mag_] := (d^2 + (mag[[1]] - x[t])^2 + (mag[[2]] - y[t])^2)^1.5;
solution =
  NDSolve[{x''[t] == Plus @@ Map[(#[[1]] - x[t])/f[#] &, mags] - g x[t] - R x'[t],
    y''[t] == Plus @@ Map[(#[[2]] - y[t])/f[#] &, mags] - g y[t] - R y'[t],
    x[0] == x1, x'[0] == 0, y[0] == y1, y'[0] == 0}, {x, y}, {t, 0, 100},
    MaxSteps -> 200000];
Show[Graphics[
  RasterArray[
    Table[final = {x[100], y[100]} /. solution[[1]];
      radii = Map[(final - #).(final - #) &, mags]; r = Min[radii];
      Hue[Position[radii, r][[1, 1]]/3], {y1, -5.0, 5.0, 10.0/n},
      {x1, -5.0, 5.0, 10.0/n}]]], AspectRatio -> 1];
```

Appendix A.2 Modified_Basin.nb

```
n = 250; d = 0.2; R = 0.15; g = 0.2; mags = {{Sqrt[2], Sqrt[2]}, {-Sqrt[2],
  Sqrt[2]}, {-Sqrt[2], -Sqrt[2]}, {Sqrt[2], -Sqrt[2]}};
fixedpoints = {{Sqrt[2], Sqrt[2]}, {-Sqrt[2],
  Sqrt[2]}, {-Sqrt[2], -Sqrt[2]}, {Sqrt[2], -Sqrt[2]}, {0, 0}}
f[mag_] := (d^2 + (mag[[1]] - x[t])^2 + (mag[[2]] - y[t])^2)^1.5;
solution =
  NDSolve[{x''[t] == Plus @@ Map[(#[[1]] - x[t])/f[#] &, mags] - g x[t] - R x'[t],
    y''[t] == Plus @@ Map[(#[[2]] - y[t])/f[#] &, mags] - g y[t] - R y'[t],
    x[0] == x1, x'[0] == 0, y[0] == y1, y'[0] == 0}, {x, y}, {t, 0, 100},
    MaxSteps -> 200000];
Show[Graphics[
  RasterArray[
    Table[final = {x[100], y[100]} /. solution[[1]];
      radii = Map[(final - #).(final - #) &, fixedpoints]; r = Min[radii];
      Hue[Position[radii, r][[1, 1]]/5], {y1, -5.0, 5.0, 10.0/n},
      {x1, -5.0, 5.0, 10.0/n}]]], AspectRatio -> 1];
```

Appendix A.3 Trajectories.nb

```
d = 0.25; g = 0.2; R = 0.07; mags = {{Sqrt[3], 1}, {-Sqrt[3], 1}, {0, -2}};
f[mag_] := (d^2 + (mag[[1]] - x[t])^2 + (mag[[2]] - y[t])^2)^1.5;
solution =
  NDSolve[{x''[t] ==
    Plus @@ Map[(#[[1]] - x[t])/f[#] &, mags] - g x[t] - R x'[t],
    y''[t] ==
    Plus @@ Map[(#[[2]] - y[t])/f[#] &, mags] - g y[t] - R y'[t],
    x[0] == -6.458, x'[0] == 0, y[0] == 2.967, y'[0] == 0}, {x, y}, {t, 0,
```

```

100}, MaxSteps -> 200000];
plotsol = {x[t], y[t]} /. solution
ParametricPlot[Evaluate[plotsol], {t, 0, 100}, PlotStyle -> RGBColor[0, 0, 1],
PlotRange -> All]
ParametricPlot[Evaluate[plotsol], {t, 0, 100}]

```

Appendix A.4 Lyapunov.nb

```

d = 0.2; g = 0.2; R = 0.1; mags = {{Sqrt[2], Sqrt[2]}, {-Sqrt[2],
Sqrt[2]}, {-Sqrt[2], -Sqrt[2]}, {Sqrt[2], -Sqrt[2]}};
f[mag_] := (d^2 + (mag[[1]] - x[t])^2 + (mag[[2]] - y[t])^2)^1.5;
solution1 =
NDSolve[{x''[t] ==
Plus @@ Map[(#[[1]] - x[t])/f[#] &, mags] - g x[t] - R x'[t],
y''[t] ==
Plus @@ Map[(#[[2]] - y[t])/f[#] &, mags] - g y[t] - R y'[t],
x[0] == 4, x'[0] == 0, y[0] == 4, y'[0] == 0}, {x, y}, {t, 0, 100},
MaxSteps -> 200000];
solution2 =
NDSolve[{x''[t] ==
Plus @@ Map[(#[[1]] - x[t])/f[#] &, mags] - g x[t] - R x'[t],
y''[t] ==
Plus @@ Map[(#[[2]] - y[t])/f[#] &, mags] - g y[t] - R y'[t],
x[0] == 4.0003, x'[0] == 0, y[0] == 4, y'[0] == 0}, {x, y}, {t, 0,
100}, MaxSteps -> 200000];
plotsol1 = {x[t], y[t]} /. solution1;
plotsol2 = {x[t], y[t]} /. solution2;
diff = Norm[plotsol1[[1]] - plotsol2[[1]]];
(*fin = Sqrt[diff.diff];*)
Plot[Evaluate[Log[diff]], {t, 0, 100}, PlotStyle -> {RGBColor[0, 0, 1]},
PlotRange -> All]

```

Appendix B References

Jürgens, Hartmut, Peitgen, Heinz-Otto, and Saupe, Dietmar.
Chaos and Fractals: New Frontiers of Science. Springer-Verlag,
New York, Inc., 1992.

<http://www.sas.upenn.edu/~uak/>