2016

Constructing and Testing an Inverted, Periodically Driven, Damped Pendulum to Study Chaotic Motion

Andrew Miller

Follow this and additional works at: http://commons.emich.edu/honors

Recommended Citation
Miller, Andrew, "Constructing and Testing an Inverted, Periodically Driven, Damped Pendulum to Study Chaotic Motion" (2016). Senior Honors Theses. 487.
http://commons.emich.edu/honors/487

This Open Access Senior Honors Thesis is brought to you for free and open access by the Honors College at DigitalCommons@EMU. It has been accepted for inclusion in Senior Honors Theses by an authorized administrator of DigitalCommons@EMU. For more information, please contact librir@emich.edu.
Constructing and Testing an Inverted, Periodically Driven, Damped Pendulum to Study Chaotic Motion

Abstract
Following the work of Berger and Nunes Jr., we designed and built an inverted, periodically driven, damped pendulum, which can be a Duffing oscillator, to study the transition from periodic to chaotic motion. We performed magnetic field measurements to characterize the permanent magnets and electromagnets used to drive the pendulum. We also analyzed video recordings of the oscillator in motion to determine the damping coefficient and resonance frequency of the pendulum.

Degree Type
Open Access Senior Honors Thesis

Department
Physics and Astronomy

First Advisor
Ernest Behringer

Second Advisor
J. Marshall Thomsen

Keywords
harmonic, solenoid, disc magnet, tracker, machined, steady-state
CONSTRUCTING AND TESTING AN INVERTED, PERIODICALLY DRIVEN, DAMPED PENDULUM TO STUDY CHAOTIC MOTION

By

Andrew Miller

A Senior Thesis Submitted to the Eastern Michigan University Honors College

in Partial Fulfillment of the Requirements for Graduation with Honors in the Department of Physics and Astronomy

Approved at Ypsilanti, Michigan, on this date April 12, 2011
CONSTRUCTING AND TESTING AN INVERTED, PERIODICALLY DRIVEN, DAMPED PENDULUM TO STUDY CHAOTIC MOTION

By

Andrew Miller

A Senior Thesis Submitted to the

Eastern Michigan University

Honors College

in Partial Fulfillment of the Requirements for Graduation

with Honors in the Department of Physics and Astronomy

Approved at Ypsilanti, Michigan, on this date April 12, 2016
Table of Contents

Abstract

I. Introduction

Fig. 1. The Apparatus of Berger and Nunes, Jr. 3
Fig. 2. EMU version of the apparatus described by Berger and Nunes, Jr 4
Original Design of the Rest Cradles for the Electromagnets 4

II. Apparatus Design and Construction

Fig. 3. Original Design of the Rest Cradles for the Electromagnets 7
Fig. 4. Final Rest Cradle Design 8
Fig. 5. Baseplate Design 9
Fig. 6. Sliding Plate Design 11
Fig. 7. Brass Mass Design 12
Fig. 8. Ruler Clamp Design 13
Fig. 9. Photo of Apparatus 14
Fig. 10. Direction of x 15

III. Characterizing the Permanent Magnets and Electromagnets

Fig. 11. Positions of the Permanent Magnets (Fig. 9 Repeated for the reader) 18
Fig. 12. Vernier Magnetic Field Sensor with Sleeve [10] 19
Fig. 13. Magnetic Field versus Optical Rail Readings for Trials 1 and 2 20
Fig. 14. Magnetic Field versus Trials 1, 2 and Solver 24
Fig. 15. Magnetic Field versus Distance Micrometer Rail Trials 1 and 2 (Elec) 26
Fig. 16. Electromagnet Trial 2 and Solver Fit 28

IV. Characterizing the Motion of the Pendulum

Fig. 17. x Position vs. Time: 0.800 Hz Drive Frequency, Mass Positioned 2 Inches from the Top of the Ruler 30
Fig. 18. Phase Space: 0.800 Hz Drive Frequency, Mass Positioned 2 Inches from the Top of the Ruler 31
Fig. 19. Steady-State Amplitude vs. Drive Frequency 34
Fig. 20. Period of Motion vs Drive Frequency of the Pendulum 35
Fig. 21. Frequency of motion vs. drive frequency 36
V. Modeling the Motion of the Pendulum

Fig. 22. Predicted Evolution of the Deflection Angle of the Pendulum

Fig. 23. Predicted Phase Space Plot of the Pendulum

VI. Summary

Acknowledgements

References

Appendix A. Python code that generates Figs. 13-16

Appendix B. Python code that generates Figs. 17-21

Appendix C. Python code that generates Figs. 22-23
Abstract

Following the work of Berger and Nunes Jr., we designed and built an inverted, periodically driven, damped pendulum, which can be a Duffing oscillator, to study the transition from periodic to chaotic motion. We performed magnetic field measurements to characterize the permanent magnets and electromagnets used to drive the pendulum. We also analyzed video recordings of the oscillator in motion to determine the damping coefficient and resonance frequency of the pendulum.
I. Introduction

What is chaos? The Merriam-Webster dictionary [1] defines chaos as "complete confusion and disorder: a state in which behavior and events are not controlled by anything." In intermediate mechanics the topic of chaos is quickly discussed or completely omitted. There are two defining properties of chaos. The first is known as sensitive dependence on initial conditions. This was observed by Edward Lorenz in 1963 while he was running a computer program to simulate the weather for a 2 month time interval [2]. Such sensitive dependence is observed in computational models of chaotic systems. The models use initial conditions, set by the user, to produce the evolution of the system. The conditions can be set to generate periodic or chaotic behavior. The second form of chaos as described by Leonard A. Smith [3] is noise. Noise is seen in every physical measurement as an uncertainty in that measurement. Smith writes. "Noise gives rise to observational uncertainty, chaos helps us to understand how small uncertainties can become large uncertainties, once we have a model for the noise." Creating chaos takes very sensitive equipment and as an undergraduate there is little opportunity to observe it. This leads to the creation of an apparatus to study and observe chaotic behavior in the undergraduate laboratory.

The paper “A Mechanical Duffing Oscillator for the Undergraduate Laboratory” by J. E. Berger and G. Nunes, Jr. describes an apparatus to study chaotic motion [4]. This apparatus is an inverted, periodically driven, damped oscillator, seen in Fig. 1. The apparatus that we designed and built is very similar to theirs and is shown in Fig. 2.

The apparatus that we designed and built is very similar to theirs and is shown in Fig. 2.

There are slight differences which do not change the functionality of the apparatus. Our ultimate goal for the apparatus was to replicate the experiments that Berger and Nunes, Jr.
Fig. 1. The Apparatus of Berger and Nunes, Jr. This is Fig. 1 of Ref. [4].
Fig. 2. EMU version of the apparatus described by Berger and Nunes, Jr.
conducted in an attempt to observe chaotic behavior and to provide a lab on chaos for our intermediate mechanics course. Unfortunately, we were not able to generate chaotic behavior with our apparatus. However, we were able to characterize the permanent magnets and electromagnets on the apparatus as well as the motion of the pendulum. A reason for the absence of chaotic behavior may lie in the natural bend of the machinist’s ruler. Three such rulers were ordered from Starrett, all of which contained a natural bend. This bend may be the product of the stamping procedure used to punch out the rulers during manufacturing.

The permanent magnets and electromagnets were characterized using a Vernier magnetic field sensor. This sensor was attached to a sliding rail and the magnetic field strength was measured as a function of sensor position. A model for the permanent magnets was taken from K&J Magnetics [5] as well as Adams Magnetic [6] and a model for the electromagnets was taken from Kulgun [12]. Our data were fit to the models.

The motion of the pendulum was recorded and then analyzed using the program Tracker by Doug Brown [7]. The horizontal (x) velocity and time data were taken from these analyzed videos and fed into a code written by ADM in Python [8] to find the average period over the length of the clip.

The paper is organized as follows: The apparatus design and construction are described in Section II. The characterization of the permanent magnets and electromagnets are described in Section III, the analysis of the motion of the pendulum is described in Section IV, and the creation of a computational model is described in Section V. A summary of the results is given in Section VI.
II. Apparatus Design and Construction

We gathered information from the paper by Berger and Nunes Jr. and set to work designing our own apparatus to recreate their experiments. Our goal with the apparatus was to generate and observe chaotic behavior. Our collaborator, undergraduate student Dustin Pepper, created the original designs for the apparatus and the material was ordered. He then machined almost all of the apparatus before leaving the research team. I then recreated the designs of all of the components of the apparatus using AutoCAD 2013 [9]. I milled the stands for the electromagnets. The cradles used by Berger and Nunes Jr. cradled the cylindrical surface of the electromagnet coils. We simplified our cradle design by placing the cradles on either side of each electromagnet. The cradles can be positioned to accommodate different size solenoids. Our original design for the cradles is shown in Fig. 3 below.

We decided that the top portion of each cradle was not needed. We therefore revised the design by taking those portions away, lowering the second portion of the stand from 1.0 inch to 0.7820 inches, and defining the size divot needed in the second portion to hold the electromagnets at a specified height above the platform supporting the cradles. These revisions are seen in Fig. 4. Figures 5 through 8 are the other components of the apparatus and Fig. 9 is a photo of the apparatus.

As seen in Fig. 5, the baseplate design has holes marked 1 that are for the leveling screws. These screws allow the baseplate plane to be made parallel to the plane of the table on which it is set. The plane of the baseplate is to be parallel with the plan of the table so that the gravitational force is perpendicular to the base. The blind-tapped 4-40 holes are for alignment screws. The sliding plate fits on to the base plate and these alignment screws drive the sliding
Fig. 3. Original design of the rest cradles for the electromagnets.
Fig. 4. Final rest cradle design.
Fig. 5. Baseplate design.
plate in the x direction to center the electromagnets on the ruler. The through holes for ¼" screws enable the screws to fasten the ruler clamp to the baseplate.

As seen in Fig. 6, the sliding plate design has a large rectangular hole in the middle that is for the ruler and ruler clamp. The four horizontal slots are to allow the rest cradles to be fastened to the sliding plate. Screws are inserted in the bottom of these slots and screwed into the bottom of the rest cradles.

The brass mass design shown in Fig. 7 is half of the mass assembly. We machined two identical brass masses and used brass screws to clamp the two together and onto the ruler. The reason we used brass screws is to allow us to treat the mass assembly as a solid block of brass with constant density.

As shown in Fig. 8, the ruler clamp design was also only half of the clamping assembly. Two of these were machined. The blind-tapped single hole allows a ¼"-20 screw to fasten the clamp to the baseplate and the two tapped holes allow two ¼"-20 screws to clamp the ruler.

The ruler clamps, leveling screws, and positioning screw were installed on the baseplate. The rest cradles for the electromagnets were then fastened to the sliding plate and the sliding plate was placed on the base plate. The position of the sliding plate was then adjusted so that the cradles were centered on the ruler clamps. The ruler is then inserted in the ruler clamp and secured. The centers of the permanent magnets are placed one inch above the top edge of the ruler clamp and the electromagnets are placed in their cradles. The brass mass is then screwed into place with the top edge 1.75 ± 0.25 inches from the top of the ruler. The electromagnets are then connected in series to a function generator which allows control of the drive frequency and the amplitude of the applied potential difference that generates the oscillating magnetic field.
Fig. 6. Sliding Plate design.
BRASS MASS

Make: 2
Material: Brass
Scale: 2:1
Tolerance: 0.002" for all dimensions

Author: Andrew Miller
Date: 10-3-13
Contact: amill119@emich.edu

Through hole for 4-40 brass screw
Counter sink hole for 4-40 brass screw
Tapped hole for 4-40 brass screw

Fig. 7. Brass Mass design.
RULER CLAMP

Make: 2
Material: Aluminum
Scale: 1:1
Tolerance: 0.002" for all dimensions

Author: Andrew Miller
Date: 10-3-13
Email: amill119@emich.edu

Fig. 8. Ruler Clamp design.
Fig. 9. Photo of apparatus.
Fig. 10. Direction of $x$
Figure 10 shows the direction of x relative to the permanent magnets.

III. Characterizing the Permanent Magnets and the Electromagnets

We characterized the permanent disc magnets positioned on the machinist ruler by measuring the magnetic field along the cylindrical symmetry axis of the magnets. The two disc magnets attract each other through the ruler and the friction that arises from this interaction holds them in place. Figure 10 below shows the placement of the permanent magnets.

The magnetic field at the end of one of the disk magnets was measured. An optical rail was used to position a Vernier magnetic field sensor, model MG-BTA, at the same height as the central axial line through the disk magnet. We used a Vernier Logger Pro interface to collect data from the magnetic field sensor as the distance between the sensor and the magnet was varied. The field sensor is encased in a plastic sleeve that keeps the sensor from being damaged.

This plastic portion is counted as the face of the sensor in our experiments. The magnetic field sensor was moved in 1 millimeter increments starting from 0 mm, right up against the disk magnet, to 20 mm away from the magnet. Table 1 below presents our measurements of the field strength versus distance from the face of the magnet.
<table>
<thead>
<tr>
<th>Trial 1</th>
<th>Trial 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x (mm)</td>
<td>δx (mm)</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>2.00</td>
<td>0.50</td>
</tr>
<tr>
<td>3.00</td>
<td>0.50</td>
</tr>
<tr>
<td>4.00</td>
<td>0.50</td>
</tr>
<tr>
<td>5.00</td>
<td>0.50</td>
</tr>
<tr>
<td>6.00</td>
<td>0.50</td>
</tr>
<tr>
<td>7.00</td>
<td>0.50</td>
</tr>
<tr>
<td>8.00</td>
<td>0.50</td>
</tr>
<tr>
<td>9.00</td>
<td>0.50</td>
</tr>
<tr>
<td>10.00</td>
<td>0.50</td>
</tr>
<tr>
<td>11.00</td>
<td>0.50</td>
</tr>
<tr>
<td>12.00</td>
<td>0.50</td>
</tr>
<tr>
<td>13.00</td>
<td>0.50</td>
</tr>
<tr>
<td>14.00</td>
<td>0.50</td>
</tr>
<tr>
<td>15.00</td>
<td>0.50</td>
</tr>
<tr>
<td>16.00</td>
<td>0.50</td>
</tr>
<tr>
<td>17.00</td>
<td>0.50</td>
</tr>
<tr>
<td>18.00</td>
<td>0.50</td>
</tr>
<tr>
<td>19.00</td>
<td>0.50</td>
</tr>
<tr>
<td>20.00</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 1. Magnetic Field measurements versus Optical Rail Readings.
Fig. 11. Positions of the permanent magnets (Fig. 9 Repeated for the reader).
Fig. 12. Vernier Magnetic Field Sensor with sleeve [10].
Fig. 13. Magnetic Field versus Optical Rail Readings for Trials 1 and 2.
The nearly constant magnetic field values for x<15mm were not consistent with an expected power law decrease. We soon realized that the magnetic field sensor was saturated for x < 15 mm. We had to start from a greater distance away to avoid the saturation of the magnetic field sensor.

We next used a Mitutoyo single axis translation stage with manual micrometer control to change the distance between the plastic sleeve of the sensor and the permanent magnet. This new setup allowed for very accurate and minute distance changes along one direction. Two L brackets were installed on the translation stage to hold the magnetic sensor in place. The sensor was again positioned to be in line with the central axis of the disk magnet. We started our measurements so that the face of the sensor was 1 inch from the face of the disc magnet. We moved the sensor toward the magnet in intervals of 0.05 inch. This allowed us to obtain an unsaturated set of measurements of the magnetic field and to find the separation at which the sensor was saturated. Knowing this point is important to avoid recording meaningless data based on the limitations of the sensor. Table 2 contains the data from two trials using the translation stage.

<table>
<thead>
<tr>
<th>Trial 1</th>
<th>Trial 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x (in)</td>
<td>B (mT)</td>
</tr>
<tr>
<td>1.000</td>
<td>2.165</td>
</tr>
<tr>
<td>0.950</td>
<td>2.373</td>
</tr>
<tr>
<td>0.900</td>
<td>2.623</td>
</tr>
<tr>
<td>0.850</td>
<td>2.908</td>
</tr>
<tr>
<td>0.800</td>
<td>3.244</td>
</tr>
<tr>
<td>0.750</td>
<td>3.532</td>
</tr>
<tr>
<td>0.700</td>
<td>4.103</td>
</tr>
<tr>
<td>0.650</td>
<td>4.661</td>
</tr>
<tr>
<td>0.600</td>
<td>5.327</td>
</tr>
</tbody>
</table>

Table 2. Magnetic Field versus Magnet Face – Sensor Face separation.
From this new set of data, we were able to see a decay in the magnetic field as a function of distance. We found an equation on the K&J Magnetics website [3] that describes the magnetic field of a disc magnet along its cylindrical symmetry axis as a function of axial distance \( x \) from the center of the magnet which has thickness \( t \) and radius \( r \):

\[
B_x = \frac{B_r}{2} \left( \frac{t + x}{\sqrt{r^2 + (t + x)^2}} - \frac{x}{\sqrt{r^2 + x^2}} \right)
\]  

(1)

\( B_r \) is a constant called the surface field that is a characteristic of the material from which the magnet is made. The value of \( B_r \) was expected to be in the range of 500-600 mT based on data from McMaster-Carr specifying that the magnet had a maximum pull force of 1.3 lbs, and using the online pull force calculator from Adams Magnetic [6]. We used the Solver function of Microsoft Excel to determine what our experimental value of \( B_r \) was. Table 3 below shows the values from trial 2 used in finding \( B_r \) using Solver.

<table>
<thead>
<tr>
<th>( x ) (in)</th>
<th>Measured ( B ) (mT)</th>
<th>Calculated ( B ) (mT)</th>
<th>( \chi^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>2.134</td>
<td>1.885</td>
<td>0.062</td>
</tr>
<tr>
<td>0.950</td>
<td>2.343</td>
<td>2.118</td>
<td>0.051</td>
</tr>
<tr>
<td>0.900</td>
<td>2.585</td>
<td>2.389</td>
<td>0.038</td>
</tr>
<tr>
<td>0.850</td>
<td>2.863</td>
<td>2.709</td>
<td>0.024</td>
</tr>
<tr>
<td>0.800</td>
<td>3.195</td>
<td>3.086</td>
<td>0.012</td>
</tr>
<tr>
<td>0.750</td>
<td>3.582</td>
<td>3.535</td>
<td>0.002</td>
</tr>
<tr>
<td>0.700</td>
<td>4.039</td>
<td>4.074</td>
<td>0.001</td>
</tr>
<tr>
<td>0.650</td>
<td>4.584</td>
<td>4.724</td>
<td>0.020</td>
</tr>
<tr>
<td>0.600</td>
<td>5.237</td>
<td>5.515</td>
<td>0.077</td>
</tr>
</tbody>
</table>

Table 3. Magnetic Field versus Trial 2 \( x \) Measurements and Fit Values.

The first two columns of Table 3 are the measured data that were obtained using the micrometer rail. The third column is the calculated magnetic field using the fit values of \( B_r \) in Eq. (1). \( \chi^2 \) squared is the square of the difference of the calculated magnetic field and
measured magnetic field. The objective of Solver is to change specified variables in an attempt to reduce chi squared to zero. We allowed Solver to change one variable, \( B_r \), because the thickness and radius of the magnet are known. The fit is close as chi squared is very small but the calculated values are still about 0.3 mT away from the measured. This may be due to the fact that a second magnet was used to hold the first in place. We found a value of 757 mT or 7570 Gauss for \( B_r \) which was a little larger than the estimated range. The results of the Solver iterations and the measured data are presented in Fig. 14.

Here \( x \) is the distance from the face of the permanent magnet to the sleeve of the magnetic field sensor. The distance from the outside of the sensor sleeve to the center of the sensor was measured to be 0.187 inches (0.00475 m). This measurement was then used as an \( x \) offset and was placed into Eq. (1) to account for that distance. The fit shown in Fig. 14 does not match the data. It is possible that the fit because of the second magnet. The two magnets were positioned on the ruler as they would be during operation. The second magnet on the other side of the ruler adds an effective thickness to the overall magnet which is generating the field that the sensor is detecting. This would lead to an increase in thickness in Eq. (1) and an increase in the surface field.

The next task was to characterize the electromagnets. This was done in a similar fashion to that of the permanent magnets in that an electromagnet was set up with the Mitutoyo translation stage. A dc voltage was applied to the electromagnet and the resulting current was calculated using the known resistance of the electromagnet. The first trial used a voltage of 2.50 V and the second used 3.50 V. Table 4 shows recorded values from the two trials with their uncertainties.

Figure 15 shows a plot of the measurements in Table 4.
Fig. 14. Magnetic Field versus x for Trial 1 and Trial 2. Triangles show the Solver fit of Eq. (1) to Trial 2 data.
Table 4. Magnetic Field versus Distance Micrometer Rail with the electromagnet.

<table>
<thead>
<tr>
<th>Trial 1</th>
<th></th>
<th>Trial 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x (in)</td>
<td>δx (in)</td>
<td>B (mT)</td>
<td>δB (mT)</td>
</tr>
<tr>
<td>1.000</td>
<td>0.005</td>
<td>0.797</td>
<td>0.050</td>
</tr>
<tr>
<td>0.950</td>
<td>0.005</td>
<td>0.801</td>
<td>0.050</td>
</tr>
<tr>
<td>0.900</td>
<td>0.005</td>
<td>0.811</td>
<td>0.050</td>
</tr>
<tr>
<td>0.850</td>
<td>0.005</td>
<td>0.839</td>
<td>0.050</td>
</tr>
<tr>
<td>0.800</td>
<td>0.005</td>
<td>0.882</td>
<td>0.050</td>
</tr>
<tr>
<td>0.750</td>
<td>0.005</td>
<td>0.926</td>
<td>0.050</td>
</tr>
<tr>
<td>0.700</td>
<td>0.005</td>
<td>0.979</td>
<td>0.050</td>
</tr>
<tr>
<td>0.650</td>
<td>0.005</td>
<td>1.044</td>
<td>0.050</td>
</tr>
<tr>
<td>0.600</td>
<td>0.005</td>
<td>1.113</td>
<td>0.050</td>
</tr>
<tr>
<td>0.550</td>
<td>0.005</td>
<td>1.196</td>
<td>0.050</td>
</tr>
<tr>
<td>0.500</td>
<td>0.005</td>
<td>1.285</td>
<td>0.050</td>
</tr>
<tr>
<td>0.450</td>
<td>0.005</td>
<td>1.365</td>
<td>0.050</td>
</tr>
<tr>
<td>0.400</td>
<td>0.005</td>
<td>1.503</td>
<td>0.050</td>
</tr>
<tr>
<td>0.350</td>
<td>0.005</td>
<td>1.620</td>
<td>0.050</td>
</tr>
<tr>
<td>0.300</td>
<td>0.005</td>
<td>1.788</td>
<td>0.050</td>
</tr>
<tr>
<td>0.250</td>
<td>0.005</td>
<td>1.944</td>
<td>0.050</td>
</tr>
<tr>
<td>0.200</td>
<td>0.005</td>
<td>2.207</td>
<td>0.050</td>
</tr>
<tr>
<td>0.150</td>
<td>0.005</td>
<td>2.432</td>
<td>0.050</td>
</tr>
<tr>
<td>0.100</td>
<td>0.005</td>
<td>2.763</td>
<td>0.050</td>
</tr>
<tr>
<td>0.050</td>
<td>0.005</td>
<td>2.948</td>
<td>0.050</td>
</tr>
<tr>
<td>0.000</td>
<td>0.005</td>
<td>3.587</td>
<td>0.050</td>
</tr>
</tbody>
</table>
The data then had to be fit so that we could determine an experimental value of the permeability of the ferrite core of the electromagnet. From *Higher Electrical Principles* [11] we
know that the relative permeability of a ferrite core lies anywhere from 200 to 10,000. The model that we fit the data to is seen in Eq. (2) and is derived by [12] using the Biot-Savart Law.

\[
B_z = \frac{\mu_0 l R^2}{8\pi (R^2 + x^2)^{3/2}}
\]  \hspace{1cm} (2)

Eq. (2) was then multiplied by the relative permeability of the ferrite core and applied to more than one loop to arrive at the following.

\[
B_z = \mu_r \frac{\mu_0 l N R^2}{8\pi (R^2 + (x + x_0)^2)^{3/2}}
\]  \hspace{1cm} (3)

The variables in Eq. (3) are as follows: \(\mu_0\) is the permeability of free space, \(i\) is the current in the electromagnet, \(R\) is the radius of the solenoid (0.0310 m), \(N\) is the turns per meter (72712 turns/m), \(x\) is the distance from the center of the solenoid, and \(x_0\) is an offset due to the sleeve of the magnetic field sensor (0.00475 m).

The solid line in Fig. 16 is the Solver fit of Eq. (4) to the experimental data. The fit provided us with a value of the permeability of the ferrite core of 662 which is well within the expected range. The result is not a perfect match and becomes worse closer to the solenoid. This may be because the governing equation has a point where when \(x\) is much less than the length of the solenoid the magnetic field becomes linear and approaches the value of the field inside of the solenoid.
In conclusion, we measured the magnetic field of a permanent disc magnet and of an electromagnet along their cylindrical symmetry axis versus the distance from their centers. Error
bars have been added but are too small to be seen due to the shapes used to show the plotted data positions. We used two methods of data collection, one using an optical rail and the other with a micrometer rail, and one method of data analysis using Solver to find a value for $B_r$ based on our experimental data. We found that for our permanent magnet it had a surface field of 757 mT and that this value was slightly above the expected range. This was likely because of the second permanent magnet used to hold the first one in place.

IV. Characterizing the Motion of the Pendulum

The first approach to characterize the motion of the pendulum was to record a video of the motion with a Casio Exilim FH-25 camera. Before recording video we attached two “crash test dummy markers” to the pendulum mass: one on the lower middle and one on the upper middle portions of the mass. These markers allow “autotracking” of these two points when we analyzed the clips using Tracker software [6]. Knowing where these points were as a function of time allowed us to calculate the angle $\theta$ that the plane of the wide face of the mass made with respect to the vertical. When the angle is 0, the pendulum is vertical. When the angle is $> 0$ the pendulum has rotated clockwise and the brass mass is to the right of the vertical. A plot of the $x$ position versus time is shown in Fig. 17.

One can see from Fig. 17 that as the pendulum oscillates, the value of $x$ changes like a sinusoid with time. The first 100 seconds of the evolution represent the transient of the pendulum. This dies away and gives rise to steady-state motion. In Fig. 18 we see the phase space plot of the above evolution.
Fig. 17. x Position vs. Time: 0.800 Hz drive frequency, mass positioned 2 inches from the top of the ruler.
Fig. 18. Phase Space plot corresponding to Fig. 16. Here, we used 0.800 Hz drive frequency, mass positioned 2 inches from the top of the ruler.

From Fig. 18 one can see that the oval shape is indicative of steady-state motion.
Finally, we also measured the period and steady-state amplitude of the pendulum. Figure 19 shows the steady-state amplitude versus drive frequency plot. This plot cannot be used in full confidence as there are negative values of amplitude which is not possible. The only way that this can be explained lies in the code. The code finds the amplitude by taking the difference between the peak position and the immediately following trough. If the autotracking feature of Tracker has a deviation from the crash test dummy symbol then it is possible for a trough to have a larger positive value than a peak. This would result in a negative value for the amplitude.

This was achieved by analyzing video recordings. We recorded the pendulum motion for 36 different drive frequencies for a length of 5 minutes each. The time and horizontal velocity data were exported from Tracker and placed into a Microsoft Excel spreadsheet. A Python code was written to find the average period of each clip using Enthought Canopy. The way that this code works is as follows. The data is input to Python in a 2 column matrix with as many rows as data found in the Excel file. These two columns are then split up into two different single row arrays called the velocity array and the time array. The code then cuts off the first 7200 frames (4 minutes) of the clip. This ensures that the transient has died away and that the information is now representative of steady-state motion. The next step is to find the time locations of the peaks or the troughs in the position versus time plot using the horizontal velocities. I chose to find the peaks but choosing the troughs would have produced very similar results. The peaks were located by setting up a for-loop that scanned through the velocity data and identified a position peak whenever the sign of the velocity at one time instant was positive and at the next time instant was negative. I figured out the time at which the peak occurred by averaging the two times. These averaged times were placed into their own array. This array, called the temporary time array, is originally as long as the input data arrays and filled with zeros except
for the time instants of peak velocities. This still means that the array is mostly zeros. I next had
to cut off these zeros. This was accomplished by another for-loop in which the temporary time
array is scanned and there is a counter. Every time that the loop finds a value that is not zero, the
counter is increased by 1. This determined the length of the array with usable data. Another
array was then generated of this length and filled with the peak time data called the final time
array. Since the period of motion is the time from peak to peak, I was able to calculate the
period of each oscillation by subtracting the time corresponding to the index i-1 from the time
corresponding to the index i starting at index 1 and going through the length of the array. These
values are then placed into an array called the period array. The values in that array are averaged
and the result is placed in the final array called the master period array. A second master array
called the frequency array holds the drive frequency for each averaged period. This process is
then repeated for each data file and the master period is plotted against the master drive
frequency array. This code yields the plot of period versus drive frequency shown in Fig. 20.

Figure 20 was generated from 36 different trials. The conditions for these trials are all
the same except for the drive frequency. The mass is set 2 inches from the top of the ruler and
the function generator is set at an amplitude of 10 V_{pp}. From these data we can see that the
steady-state period of the pendulum is equal to the inverse of the drive frequency.

Figure 21 is a plot of the drive frequency to the frequency of motion to show that the two
are the same except for low drive frequencies.
Fig. 19. Steady-state amplitude vs. drive frequency.
Fig. 20. Period of Motion vs Drive Frequency of the Pendulum
Fig. 21. Frequency of motion vs. drive frequency.

V. Modeling the Motion of the Pendulum
We also modeled the motion of the pendulum when driven at different frequencies. The code solves the differential equation associated with the motion of the pendulum and then generates a plot of the deflection angle as a function of time and a phase space plot of the pendulum. The differential equation that describes the motion is known as the Duffing equation and is Eq. (2) of [4], given here:

$$\frac{d^2\theta}{dt^2} = -a\theta^3 + b\theta - c\frac{d\theta}{dt} + f\cos(\omega t)$$

(4)

where for the special case of the Duffing equation $a = 1/6$, $b = 1-a/mgl$, $\alpha$ is the torsional spring constant of the pendulum, $c = k/mgl$, $k$ is the damping coefficient, $f$ is the drive amplitude, $\omega$ is $2\pi/\tau$, $\frac{d^2\theta}{dt^2}$ is the angular acceleration of the pendulum, $\frac{d\theta}{dt}$ is the angular velocity of the pendulum, and $\theta$ is the angular position of the pendulum. The evolution of $\theta$ is shown in Fig. 20 for the case in which $b = 0.03$, $c = 0.02$, and $f = 0.002$.

The phase space plot corresponding to Fig. 22 is shown in Fig. 23. This plot shows that in steady-state the phase space plot is an egg shaped oval, which is what we saw in the steady-state phase space plot in Fig 18.
Fig. 22. Predicted evolution of the deflection angle of the pendulum for $a = 1/6$, $b = 0.03$, $c = 0.02$, $f = 0.002$, for initial conditions $\theta(0) = 0$, $\frac{d\theta}{dt}(0) = 0$. 
Fig. 23. Predicted phase space plot of the pendulum for $a = 1/6$, $b = 0.03$, $c = 0.02$, $f = 0.002$, for initial conditions $\theta(0) = 0$, $\frac{d\theta}{dt}(0) = 0$. 

VI Summary
A damped, driven, inverted pendulum was built and characterized. The driving mechanism involved attaching permanent magnets to the pendulum and applying an oscillating driving force to the magnets using a pair of electromagnets. The permanent magnets and electromagnets were both characterized by measuring the magnetic field along their cylindrical symmetry axes. The motion of the pendulum has been recorded and analyzed. Although chaotic behavior was not observed, a comparison of observed motion to predicted motion shows that the pendulum was operating under periodic conditions. These results serve to characterize this apparatus so that it may be used for projects related to intermediate mechanics.

Future work will include the application of strain gauges and an electrical circuit provided by Berger and Nunes, Jr. to measure the deflection of the pendulum. A more accurate analysis of the motion of the pendulum will then be possible.

Acknowledgements

I thank the Department of Physics and Astronomy at Eastern Michigan University for the use of their labs and instruments. I also thank J. E. Berger and G. Nunes, Jr. for the work that they did. I thank Mr. Norbert Vance for his support in the machine shop and Dustin Pepper for all of his work. I also thank Dr. Marshall Thomsen and Dr. Alexandria Oakes for their enduring patience with edits. Lastly, I give a special thanks to Dr. Ernest R. Behringer for all of his help along the way.

References


<http://www.unbeatablesales.com/dhmbta.html>


[12] Magnetic Fields Due to a Solenoid Web Site <http://plasma.kulgan.net/sol_p>
Appendix A.

#
# Plots
#
# This file will generate a plot of the input arrays
#
# Written by:
#
# # Andrew D. Miller
# Department of Physics and Astronomy
# Eastern Michigan University
# Ypsilanti, MI 48197
# (734) 478-7757
# amill19@emich.edu
#

# import the commands needed to make the plot
from pylab import figure,plot,xlim,xlabel,ylim,ylabel,grid,show,title,zeros,legend,errorbar

# import the command needed to make a 1D array, and math functions
from numpy import linspace,sqrt,pi,cos,sin,exp

# Input parameters
d1 = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
D1 = [1,0.95,0.9,0.85,0.8,0.75,0.7,0.65,0.6]
B3 = [2.165,2.373,2.623,2.908,3.244,3.639,4.103,4.661,5.327]
D2 = [1,0.95,0.9,0.85,0.8,0.75,0.7,0.65,0.6]
B4 = [2.134,2.343,2.585,2.863,3.195,3.582,4.039,4.584,5.237]
B4b = [1.885,2.118,2.389,2.709,3.086,3.535,4.074,4.724,5.515]
De1 = [1,0.95,0.9,0.85,0.8,0.75,0.7,0.65,0.6,0.55,0.5,0.45,0.4,0.35,0.3,0.25,0.2,0.15,0.1,0.05,0]
Be1 = [0.797,0.801,0.811,0.839,0.882,0.926,0.979,1.044,1.113,1.196,1.285,1.365,1.503,1.62,1.788,1.944,2.207,2.432,2.763,2.948,3.587]
De2 = [1,0.95,0.9,0.85,0.8,0.75,0.7,0.65,0.6,0.55,0.5,0.45,0.4,0.35,0.3,0.25,0.2,0.15,0.1,0.05,0]
Be2 = [0.901,0.954,0.987,1.032,1.107,1.164,1.25,1.318,1.432,1.54,1.625,1.788,1.952,2.145,2.367,2.627,2.906,3.309,3.771,4.292,4.974]
Be2S =
[1.159, 1.223, 1.291, 1.364, 1.442, 1.526, 1.611, 1.713, 1.822, 2.040, 2.165, 2.299, 2.442, 2.595, 2.758, 2.932, 3.118, 3.315, 3.525, 3.747]
#print D1_correct

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim(-1, 21)

# Label the horizontal axis, with units
xlabel("Distance $x$ [mm]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("Magnetic Field $B$ [mT]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("B Field vs. Optical Rail Reading")

# Generate the plot. The plot symbols will be a green line.
plot(d1, B1, 'rs--', label = "Trial 1")
plot(d2, B2, 'bo:', label = "Trial 2")
errorbar(d1, B1, xerr=0.5, yerr=0.05, fmt='rs')
errorbar(d2, B2, xerr=0.5, yerr=0.05, fmt='bo')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("Distance $x$ [in]", size = 16)

# Define the limits of the vertical axis
ylim()
# Label the vertical axis, with units
ylabel("Magnetic Field $B$ [mT]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("B Field vs. Micrometer Rail Reading and Solver Data")

# Generate the plot. The plot symbols will be a green line.
plot(D1,B3,'rs--',label = "Trial 1")
plot(D2,B4,'bo:',label = "Trial 2")
plot(D2,B4b, 'g^-',label = "Solver")
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim(-0.1,1.1)

# Label the horizontal axis, with units
xlabel("Distance $x$ [in]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("Magnetic Field $B$ [mT]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title


# Generate the plot. The plot symbols will be a green line.
plot(Del,Bel,'rs--',label = "Trial 1: V = 2.50 V")
plot(D2,B2,'bo:',label = "Trial 2: V = 3.50 V")
plot(D2,B2S,'go-',label = "Solver Trial 2")

errorbar(Del,Bel, xerr=0.005, yerr=0.05, fmt='rs')
errorbar(D2,B2, xerr=0.005, yerr=0.05, fmt='bo')
legend(loc=1)
# Show the plot
show()
Appendix B. (Repeated 36 times for the different files used)

# Chaotic Pendulum
#
# This code computes the average period of data files that it imports
# and plots them against their respective drive frequencies.
# It also generates the evolution and phase space plots of
# the data for a drive frequency of 0.8 Hz
# Lastly, it finds the angular position of the pendulum in each file
# and uses that to produce a plot of the average amplitude versus
# drive frequency
#
# Written by:
#
# Andrew D. Miller
# Department of Physics and Astronomy
# Eastern Michigan University
# Ypsilanti, MI 48197
# (734) 478-7757
# Amill119@emich.edu
#
# Some imported functions are not used
#
from scipy.fftpack import fft, fftshift
from numpy import linspace, genfromtxt, abs, max, min, log10, arctan, pi
from pylab import plot, xlim, xlabel, ylim, ylabel, grid, show, figure, title, zeros, legend
files = 36
MasterPeriod = zeros(files)
MasterAmp = zeros(files)
# Import the text file which is filled with time and V values
signal = genfromtxt("2016-29-01_0.2_10_02_11.csv", delimiter=',')
# print "len(signal) = ",len(signal)
# print signal

time1 = zeros(len(signal))
vel1 = zeros(len(signal))
frequency = zeros(files)
x_top = zeros(len(signal))
y_top = zeros(len(signal))
x_bot = zeros(len(signal))
y_bot = zeros(len(signal))
time2 = zeros(len(signal))
vx_top = zeros(len(signal))
vx_bot = zeros(len(signal))
# print len(signal)
w = 0
for i in range(0, len(signal)):
    time1[i] = signal[i][0]
    vxl[i] = signal[i][1]
    time2[i] = signal[i][2]
    x_top[i] = signal[i][3]
    y_top[i] = signal[i][4]
    x_bot[i] = signal[i][5]
    y_bot[i] = signal[i][6]
    vx_top[i] = signal[i][7]
    vx_bot[i] = signal[i][8]
    #print x top
    if x_top[i] != 0:
        w = w + 1
        #print w
        x_top2 = zeros(w)
        y_top2 = zeros(w)
        x_bot2 = zeros(w)
        y_bot2 = zeros(w)
        time3 = zeros(w)
        vx_top2 = zeros(w)
        vx_bot2 = zeros(w)
        Theta_rad = zeros(w)
        Theta_deg = zeros(w)
        for i in range(0, w):
            x_top2[i] = x_top[i]
            y_top2[i] = y_top[i]
            x_bot2[i] = x_bot[i]
            y_bot2[i] = y_bot[i]
            time3[i] = time2[i]
            vx_top2[i] = vx_top[i]
            vx_bot2[i] = vx_bot[i]
            #print x_top2[i]

for i in range(0, w):
    Theta_rad[i] = pi / 2 - arctan(abs(y_top2[i] - y_bot2[i]) / abs(x_top2[i] - x_bot2[i]))
    Theta_deg[i] = (180 * Theta_rad[i]) / pi

    #print Theta_rad[0]
    #print Theta_deg[0]
    #print pi
    #print len(Theta_rad)
    #print len(time3)
    #print time1
    #print vxl
    #print x1
    #print time2[2]
#print x_top[2]
#print y_top[2]
#print x_bot[2]
#print y_bot[2]
#print vx_top[2]
#print vx_bot[2]

peaks = zeros(w)
peaktime = zeros(w)
troughs = zeros(w)
troughtime = zeros(w)
Amplitude = zeros(w)
r=0
l=0
for i in range(0,w-2):
if Theta_rad[i]<Theta_rad[i+1] and Theta_rad[i+1]>Theta_rad[i+2]:
    peaks[i] = x_top[i]
    peaktime[i] = time3[i]
if Theta_rad[i]>Theta_rad[i+1] and Theta_rad[i+1]<Theta_rad[i+2]:
    troughs[i] = x_top[i]
    troughtime[i] = time3[i]

d = 0
for i in range(0,len(peaks)-1):
    if peaks[i]!=0:
        d=d+l
c = 0
for i in range(0,len(troughs)-1):
    if troughs[i]!=0:
        c=c+l
peaks2 = zeros(d)
troughs2 = zeros(c)
p=0
for i in range(0,len(peaks)-1):
    if peaks[i]!=0:
        peaks2[p] = peaks[i]
p=p+1
q=0
for i in range(0,len(troughs)-1):
    if troughs[i]!=0:
        troughs2[q] = troughs[i]
q=q+1
lengtharray = [len(peaks2),len(troughs2)]
f = min(lengtharray)
for i in range(0,f):
    Amplitude[i] = peaks2[i]-troughs2[i]
    #print Amplitude[i]

Steadytime = zeros(len(time1)-7200)
SteadyV = zeros(len(vxl)-7200)

for i in range(0,len(Steadytime)):
    Steadytime[i] = time1[i+7200]
    SteadyV[i] = vxl[i+7200]

timetemp1 = zeros(len(signal))

for i in range(0,len(SteadyV)):
    if SteadyV[i]>0 and SteadyV[i+1]<0:
        timetemp1[i] = (Steadytime[i] + Steadytime[i+1])*0.5
        #print timetemp1[i]
        #print i
        #print timetemp1[1]

j = 0
for i in range(0,len(signal)):
    if timetemp1[i]!=0:
        j=j+1
        #print timetemp1[i]

finaltime1 = zeros(j)
k=0
for i in range(0,len(signal)):
    if timetemp1[i]!=0:
        finaltime1[k] = timetemp1[i]
        k=k+1
        #print finaltime

        #print finaltime1

period1 = zeros(j)

for i in range(0,j-1):
    period1[i] = finaltime1[i+1]-finaltime1[i]
    #print period1[i]

# print sum(period1)/len(period1)

MasterPeriod[0] = sum(period1)/len(period1)
frequency[0] = 0.2
MasterAmp[0] = sum(Amplitude)/len(Amplitude)

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim(0.0, 4.0)

# Label the horizontal axis, with units
xlabel("Drive Frequency [Hz]", size = 16)

# Define the limits of the vertical axis
ylim(0.0, 6.5)

# Label the vertical axis, with units
ylabel("Frequency of Motion [Hz]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Frequency of Motion vs. Drive Frequency")

# Generate the plot. The plot symbols will be a green line.
plot(frequency, 1.0/MasterPeriod, 'ro', label = "Data")
#plot(y,x, label = "1/\theta")
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("Time [s]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("$\theta$ [rad]", size = 16)

# Make a grid on the plot
grid(True)
# Generate the title
title("Angular Position vs. Time f = 0.4 Hz")

# Generate the plot. The plot symbols will be a green line.
plot(time3, Theta_rad,'go-')
legend(loc=1)
# Show the plot
show()
# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("\theta [rad]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("\omega [rad/s]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Phase Space f = 0.4 Hz")

# Generate the plot. The plot symbols will be a green line.
plot(Theta_rad,Theta_rad/time3,'go-')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("Drive Frequency [Hz]", size = 16)

# Define the limits of the vertical axis
ylim()
# Label the vertical axis, with units
ylabel("Amplitude [m]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Resonance Curve")

# Generate the plot. The plot symbols will be a green line.
plot(frequency, MasterAmp, 'go')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("x Position [cm]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("dx/dt [emfs]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Phase Space")

# Generate the plot. The plot symbols will be a green line.
plot(x_top2, vx_top2, 'g-')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()
# Label the horizontal axis, with units
xlabel("Time [s]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("s Position [cm]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Evolution")

# Generate the plot. The plot symbols will be a green line.
plot(time3,x_top2,'g-')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("Time [s]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("x Position [cm]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Evolution f = 0.8 Hz")

# Generate the plot. The plot symbols will be a green line.
plot(time_long,x_long,'g-')
legend(loc=1)
# Show the plot
show()
```python
# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("x Position [cm]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("x Velocity [cm/s]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Phase Space f = 0.8 Hz")

# Generate the plot. The plot symbols will be a green line.
plot(x_long,vx_long,'g-')
legend(loc=1)
# Show the plot
show()

# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("Time [s]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("x Position [cm]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Steady Evolution f = 0.8 Hz")
```
# Generate the plot. The plot symbols will be a green line.
plot(time_steady,x_top_steady,'g-')
legend(loc=1)
# Show the plot
show()
# Start a new figure
figure()
# Define the limits of the horizontal axis
xlim()

# Label the horizontal axis, with units
xlabel("x Position [cm]", size = 16)

# Define the limits of the vertical axis
ylim()

# Label the vertical axis, with units
ylabel("x Velocity [emfs]", size = 16)

# Make a grid on the plot
grid(True)

# Generate the title
title("Steady Phase Space f = 0.8 Hz")

# Generate the plot. The plot symbols will be a green line.
plot(x_top_steady,vx_top_steady,'g-')
legend(loc=1)
# Show the plot
show()
Appendix C.

# Duffing_Oscillator_evolution.py
# Computes the numerical solution of
# d2theta/dt2 = b*theta - a*theta**3 - c*dtheta/dt -r*cos(omega*t)
# with user-supplied initial conditions
# obtained using the fourth-order Runge-Kutta algorithm
# with constant step size
#
# Written by:
#
# Ernest R. Behringer
# Department of Physics and Astronomy
# Eastern Michigan University
# Ypsilanti, MI 48197
# (734) 487-8799
# ebehringe@emich.edu
#
# 20140513 by ERB starting with M. Newman's odesim.py code
#
from math import cos,sqrt,pi
from numpy import array ,arange
from pylab import figure,plot,xlim,ylim,xlabel,ylabel,grid,show,title

# Specify Duffing oscillator parameters
a = 1.0/6.0
b = 0.03
c = 0.02
f = 0.002 # Scaled drive amplitude
freq = 0.19 # drive frequency [Hz]
tscale = 0.17 # Time scale [s]
thetamax = 0.7 # maximum angular displacement [rad]

# Here are the derivatives
def derivs(r,t):
    theta = r[0]
    thetap = r[1]
    dtheta = thetap
    dddtheta = -a*theta**3 + b*theta - c*thetap + f*cos(2.0*pi*freq*tscale*t)
    return array([dtheta,ddtheta],float)
# Specify initial conditions
theta0 = sqrt(6.0*b)
dthetadt0 = 0.0

# Calculate the numerical solution using
# fourth-order Runge-Kutta algorithm
t1 = 0.0 # initial scaled time
t2 = 120.0/tscale # final scaled time
N = 4800 # number of time steps
h = (t2-t1)/N # time step size

tpoints = arange(t1,t2,h)
thetapoints = []
dthetadtpoints = []

r = array([theta0,dthetadt0],float)
for t in tpoints:
    thetapoints.append(r[0])
    dthetadtpoints.append(r[1])
k1 = h*derivs(r,t)
k2 = h*derivs(r+0.5*k1,t+0.5*h)
k3 = h*derivs(r+0.5*k2,t+0.5*h)
k4 = h*derivs(r+k3,t+h)
r += (k1+2*k2+2*k3+k4)/6

# Generate the evolution plot
figure()
plot(tpoints,thetapoints,"b-")
xlim(t1,t2)
ylim(-thetamax,thetamax)
xlabel("Scaled time \(t\) [\text{s}]",fontsize=16)
ylabel("Angular Position \(\theta\) [\text{rad}]",fontsize=16)
title("Evolution with Scaled Drive Frequency = 0.002")
grid(True)
show()

# Generate the phase space plot
figure()
plot(thetapoints,dthetadtpoints,"g-")
xlim(-thetamax,thetamax)
xlabel("Angular Position \(\theta\) [\text{rad}]",fontsize=16)
ylim(-0.08,0.08)
ylabel("Angular Velocity \(d\theta/dt\) [\text{rad/s}]",fontsize=16)
title("Phase Space with Scaled Drive Frequency = 0.002")
grid(True)
show()