Damped Driven Pendulum and the Road to Chaos

Zachary Warner
PHSX 815 – Spring 2019
University of Kansas
April 29, 2019

1 Introduction

Dynamical systems, such as a simple pendulum, allow for exact predictions of future behavior. Solutions to a chaotic systems, on the other hand, are stringently dependent on the initial conditions of the system. Since many phenomena in nature display chaotic behavior it is useful to model and study such behavior. This is most easily achieved by starting with a "simple" chaotic system. This paper explores the behavior of such a systems by studying numerical solutions of the damped driven pendulum.

The equation of motion for a damped driven pendulum can be written as

$$ml\frac{d^2\theta}{dt^2} + \gamma\frac{d\theta}{dt} + mg\sin\theta = A\cos\omega t \tag{1}$$

where $-\gamma\dot{\theta}$ is the damping force for a damping coefficient $\gamma > 0$, and $A\cos\omega t$ is the driving force with an amplitude A and frequency ω. Equation (1) can be rearranged and simplified so to be written as

$$\frac{d^2\theta}{d\tau^2} + \mu\frac{d\theta}{d\tau} + \sin\theta = a\cos\nu t \tag{2}$$

where $\omega_0 = \sqrt{g/l}$ is initial angular frequency, $\tau = \omega_0 t$ is effective time, $\nu = \omega/\omega_0$ is effective frequency of the driving force, $\mu = \omega_0\gamma/(mg)$ is effective damping coefficient , and $a = A/(mg)$ is effective amplitude. Since equation (2) cannot be solved analytically it must be solved numerically. In order to solve equation (2) numerically we make use of the Runge-Kutta 4th order method (RK4). In order to do this it is usefully to write the 2nd-order ODE as a system of 1st –order ODEs with two variables θ and ν as follows

$$\frac{d\theta}{d\tau} = \nu \tag{3}$$

$$\frac{d\nu}{d\tau} = -\mu\upsilon - \sin\theta + a\cos\nu t\tau\,. \tag{4}$$

We will begin this paper by testing the accuracy of the RK4 method through observing the numerical convergence of a simple pendulum and a pendulum of small oscillations before exploring the chaotic behavior of the system. The chaotic behavior of the damped riven pendulum is achieved by breaking the symmetry of the phase space and then changing the effective damping coefficient to produce period-doubling bifurcation. This chaotic behavior as function of the damping coefficient is then analyzed with graphs of the phase space and the frequency power spectrum $P(v_n) = |F_n|^2$, where F_n are the amplitude of oscillation modes obtained from performing a Fourier transform on the periodic function v_n [1].

2 Transient Motion and Numerical Convergence

We begin by testing our code, Appendix A, for the numerical convergence of the energy relative to the initial energy E / E_0 using the RK4 method for a simple pendulum when $\mu = a = 0$ in equation (1). While the calculation should be done for a sufficient number of oscillations, integration times that are to large violating the symplectic condition of RK4 become an issue. The numerical convergence was found to be optimal for a step size of $h = 0.01T_0$. The numerical convergence was also tested using a constant of motion of the small oscillation $b = \sqrt{\theta^2 + \upsilon^2 / v^2} = b_0$ where $b_0 = a\left[(v\mu)^2 + (1-v^2)^2\right]$ when $(\theta << 1)$ and, again, found to be convergent for $h \sim 0.01$. Both of these convergences can be seen in Fig. 1.

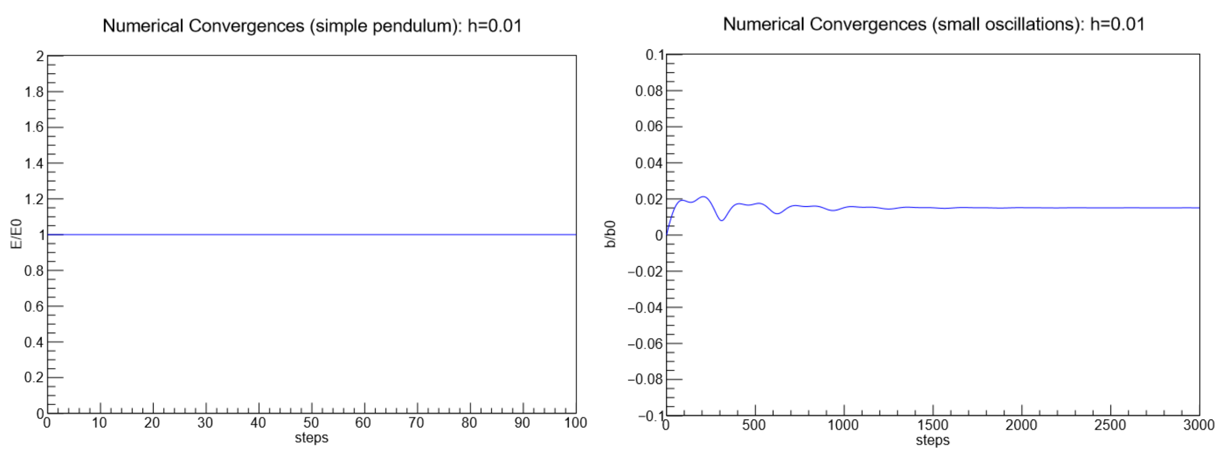

Fig.1 These figures show the numerical convergence of E / E_0 and $b / b_0 - 1$ as a function of number of steps.

With numerical convergence achieved and an appropriate step size chosen we can proceed to the elimination of the transient motion of the system. Due to dissipation, the choice of the initial condition is irrelevant so long as it is in the basin of the attractor [2]. Fig. 2 shows the convergence of the phase space for the system, regardless of the chosen $\upsilon(0)$. Thus, we remove the transient state leaving only the asymptotic state by eliminating the numerical integration of the first few damping time scales that did not contribute to the asymptotic state, Fig. 3. Note that in these figures, and for the duration of this paper, we will be using the parameter values $a = 1.5$ and $v = 2/3$. We also define a parameter $Q = 1/\mu$ as a control parameter for use later on in the study.

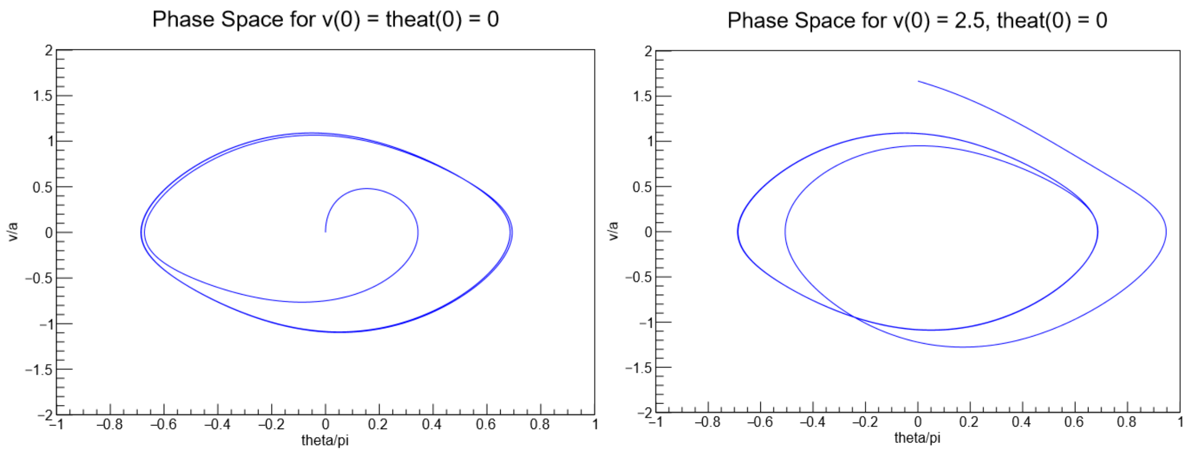

Fig.2 Trajectory of the damped driven pendulum with the transient motion still included.

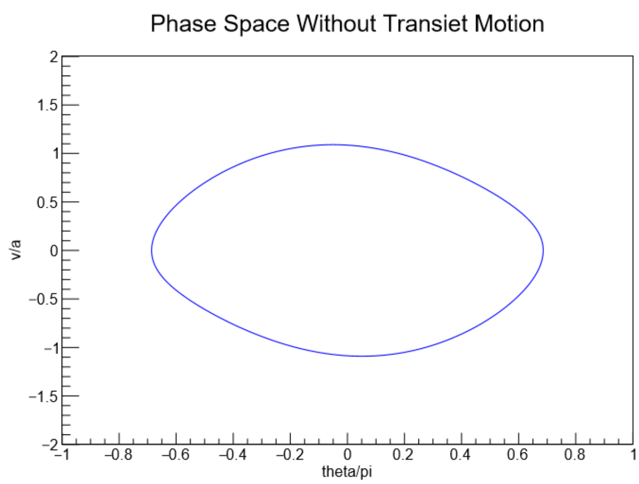

Fig.3 Trajectory of the damped driven pendulum with the transient motion eliminated.

3 Symmetry Breaking and Period-Doubling Bifurcation

As previously mentioned in Section 2 the control parameter Q, which is a function of the damping coefficient, will be used to explore the nonlinear properties of the system exhibited as Q increases. In Fig. 3 $Q = 1.2$ and the system is shown to be symmetrical in phase space, spending equal amount of time in $\theta > 0$ and $\theta < 0$. This symmetry can be broken by decreasing the damping coefficient so that $Q = 1.3$, Fig. 4. It is clear from this figure that the pendulum spends more time in $\theta < 0$. This spontaneous symmetry breaking is a nonlinear phenomenon were two states coexist and both are stable up to the periodic-doubling bifurcation.

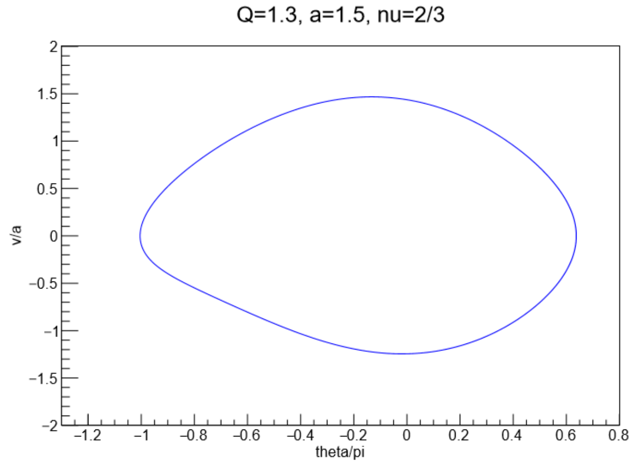

Fig. 4 Trajectory of the damped driven pendulum in the symmetry-breaking state

As we continually increase Q, the symmetry breaking states go through a sequence of period-doubling bifurcations were the period cycle changes from a 1-loop cycle into a 2-loop cycle, the 2-loop cycle changes into a 4-loop cycle, 4 into 8, 8 into 16 and so on until a chaotic state is reached where the frequency spectrum would contain an infinite number of oscillation modes [2]. Fig. 5 shows the breaking of the period cycles as a function of Q starting with a 2-loop cycle, top left.

As mentioned the frequency power spectrum contains the number of oscillation modes of the system. Therefore, we also perform a Fourier transform on $\upsilon(t)$ and calculate the frequency power spectrum as a function of Q, Fig. 6. Moving from top left to bottom right we can see in the figure that the number of oscillation modes change from 1 to 2 and so on following this pattern of splitting the period doubling bifurcation cycle, eventually reaching an infinite (chaotic) number of oscillation modes.

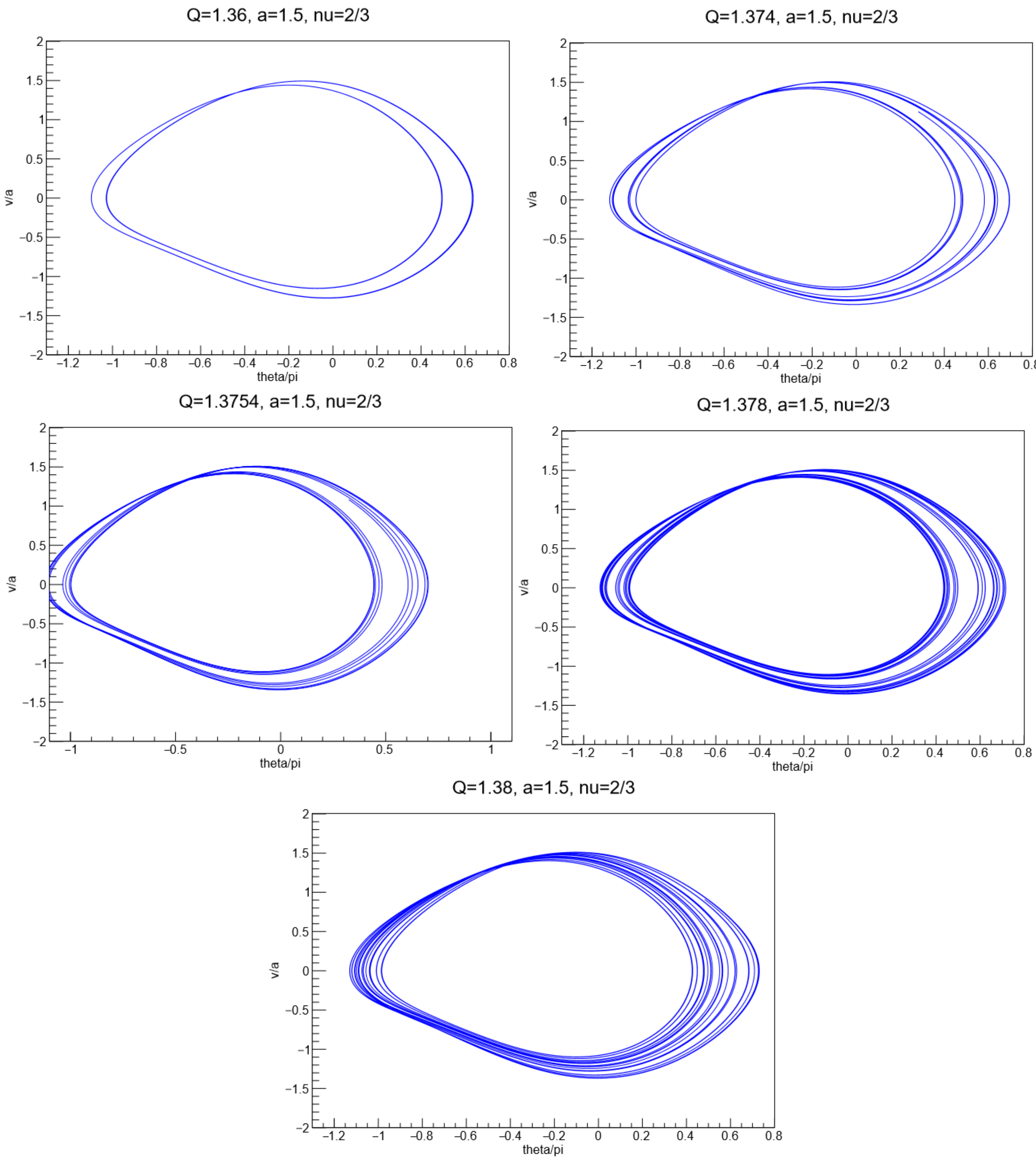

Fig. 5 Period-doubling bifurcations of the limit cycle from period-2 to chaotic orbit.

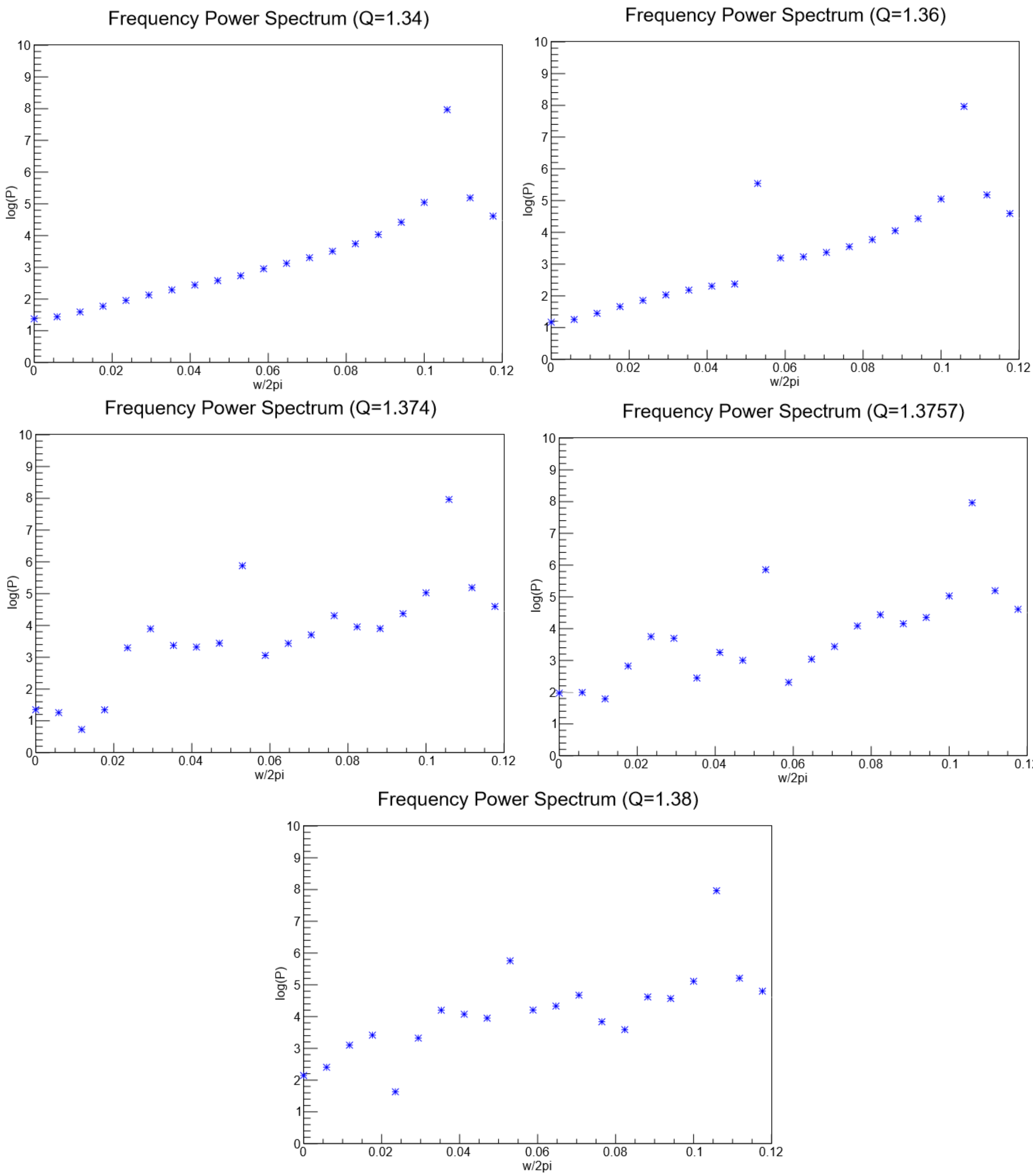

Fig. 6 Frequency power spectrum of the limit cycles from period-2 to chaotic orbit.

Lastly, we look at a bifurcation diagram which plots one point of the orbit as a function of Q. This allows us to easily study the bifurcation of a periodic orbit as a function of the damping applied to the system. Fig. 7 is a bifurcation diagram of our system showing the first three period-doubling bifurcations of the limit cycle.

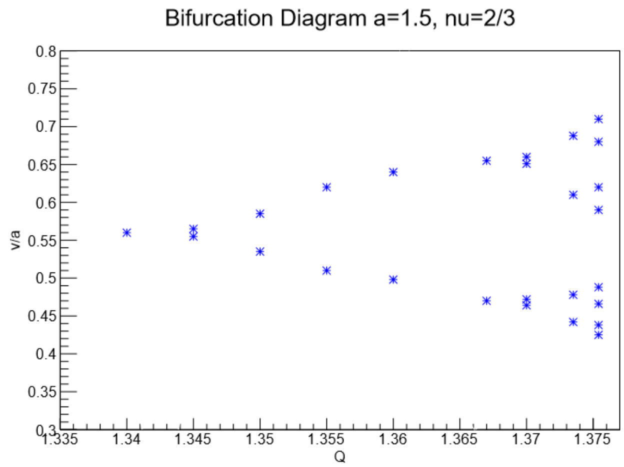

Fig. 7 Bifurcation diagram showing the first three period-doubling bifurcations of the limit cycle.

4 Conclusion

We studied the chaotic behavior of a damped driven pendulum by analyzing the numerical solutions of equation (2) produced by the RK4 method. To explore our system we looked at graphs of the phase space and observed the period-doubling bifurcation of the limit cycle as the dampening of the system was decreased (increase in Q). We also analyzed the frequency power spectrum of the pendulum as a function of Q . In both cases we found the found that as we continually increase Q the symmetry breaking states of the system go through a continual sequence of period-doubling bifurcations where the period cycle changes from a 1-loop cycle into a 2-loop cycle, the 2-loop cycle changes into a 4-loop cycle, 4 into 8, 8 into 16 and so on until a chaotic state is reached where the frequency spectrum contains an infinite number of oscillation modes and the system becomes stringently dependent upon the chosen initial conditions.

References

[1] Shi, Jack J., *Lecture Notes For Computational Physics: Chapter 2*. 2019.
[2] Shi, Jack J., *Lecture Notes For Computational Physics: Chapter 3*. 2019.

Appendix A

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cmath>
#include <stdlib.h>
#include <TGraph.h>
#include <TCanvas.h>
#include <TAxis.h>

using namespace std;


//Zach Warner
//Project III
/*This program solves the damped driven pendulum using RK4 method and does the following:
(1) Calculates the first three perieod-doubling bifurcation. Plot the bifurcation in parameter space
and in phase space.
(2) calculate the frequency power spectrum for the obrit
(3) Calculate and plot a chaoitic orbit and its frequency spectrum
*/


double pi=atan(1.0)*4.0;
double pi2=atan(1.0)*8.0;


double f1(double v, double theta,double t,double Q)
{
  double mu = 1/Q;
  double a = 1.5;
  double nu = 2.0/3.0;
 return -mu*v-sin(theta)+a*cos(nu*t);
}

int main()
{

  //fourier transform
  double N=17000; //number of steps
  double t_ft_graph[40000];
  double y_graph[40000];
  double P_graph[39999], w_graph[39999];
  double t = 0;
  double A[39999],B[39999],P[39999],w[39999],f[39999];
```

```
double temp,tempA,tempB,tempf, tempw;

//rk4
int i,count;
double x0,t0,xt,a,mu,nu,v0,E0,b0,Q,v_bi,theta;
double h;// step size RK4
double x_RK4[5],v_RK4[5];
double h2,t1,t2;
double n; // number of steps for RK4
double x_graph[30001],v_graph[30001],t_graph[30001],steps[30001],E_graph[30001];
double v_update,x_update;
double x[30001];
double v[30001],vft[30001];
double b[30001];
double x_bi[27],Q_bi[27];
double temp_t,temp_x,temp_v;
double k1,k2,k3,k4,l1,l2,l3,l4; //k-values to be used in the RK4 method

 Q=1.38;
 x0 = 0;   // Initial angle
 t0 = 0;   // Initial time
 v0 = 0;  // Initial angular velocity
 a=1.5;
 mu=1/Q;
 nu=2.0/3.0;
 t1=0;
 t2=300;
 h=.01;
 n=(t2-t1)/h;
 b0=a*sqrt(pow(nu,2)*pow(mu,2)+pow((1-pow(nu,2)),2));

 h2 = 0.5*h;  // used in RK4 method


 x[0] = x0;//inital conditions for angle
 v[0] = v0;//i.c. angular velocity
 temp_t=0;


  for(int j=0;j<n;j++)   // RK 4th order method
  {
       temp_v=v[j];
       temp_x=x[j];
       steps[j]=j;
```

```
        if(temp_t>130) //eliminating transient motion
        {
                i=j-13000;
                 vft[i]=temp_v;
                 x_graph[i]=x[j]/pi;
                 v_graph[i]=v[j]/1.5;

        }

        k1=h*v[j];
        l1=h*f1(temp_v,temp_x,temp_t,Q);
         k2=h*(v[j]+.5*l1);
        l2=h*f1(temp_v+.5*l1,temp_x+.5*k1,temp_t+h2,Q);
         k3=h*(v[j]+.5*l2);
        l3=h*f1(temp_v+.5*l2,temp_x+.5*k2,temp_t+h2,Q);
        k4=h*(v[j]+l3);
        l4=h*f1(temp_v+l3,temp_x+k3,temp_t+h,Q);
        x[j+1]=(x[j]+(k1+2*k2+2*k3+k4)/6);
        v[j+1]=(v[j]+(l1+2*l2+2*l3+l4)/6);

        t_graph[j]=temp_t;
        temp_t=temp_t+h;

      }


   for(int n = 0; n<N-1; n++)//forward Fourier transform
   {
    tempA=0;
    tempB=0;
    tempw=(pi2*n)/N;

    for(int k=0; k<N-1; k++)
{

  temp = tempw*k;
  tempA = tempA + v_graph[k]*cos(temp);
  tempB = tempB - v_graph[k]*sin(temp);
}

    A[n] = tempA;
    B[n] = tempB;
    P[n] = pow(tempA,2) + pow(tempB,2);
    P_graph[n]=log10(P[n]);
    w_graph[n] = tempw/(h*pi2);
```

```
    }

 //This section is for graphing

   TGraph *gr1 = new TGraph(N-1,w_graph,P_graph);
   TAxis *axis = gr1->GetXaxis();

   // gr1->Draw("AC");//Draws forward historesis loop
   axis->SetLimits(0,100); //x-axis
   gr1->GetHistogram()->SetMaximum(10);//y-axis
   gr1->GetHistogram()->SetMinimum(0);
   gr1->GetXaxis()->SetTitle("w/2pi");
   gr1->GetYaxis()->SetTitle("log(P)");
   gr1->GetXaxis()->CenterTitle();
   gr1->GetYaxis()->CenterTitle();
   gr1->SetTitle("Frequency Power Spectrum (Q=1.38)");
   gr1->SetLineColor(0);
   gr1->SetMarkerColor(4);
   gr1->Draw("AC*");

  return 0;

}
```