PHYS 410-Computational Physics- Project 3

Ali Izadi Rad
Date: Nov. 2016

Pendulum motion

In this project we want to simulate the motion of pendulum in a fixed plane, in presence of driven and damping forces.

1 Introduction

One of the most general form of equitation of motion that we can consider for pendulum which restricted to move in plane can be

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

here m is the mass of pendulum and l represent the length of pendulum. g also is gravity acceleration constant which for sake of simplicity we set them all to one and we work on that special metric system. In order to use RK4 method to solve our differential equation we need to break our equation of motion to two first order differential equation as a following:

$$\begin{aligned} y &= \dot{\theta} \\ \dot{y} &= -\sin\theta - \nu y + A\sin(\omega t) \end{aligned} \tag{2}$$

Notice that by dierentiating the first equation with respect to time, we get an expression which when combined with the second equation gives us the original pendulum equation. In particular, this allows us to regard the phase space of this physical system as the collection of points (θ, v), where $v = l\dot{\theta}$.

We're setting the length of the pendulum to be $l = 1m$, which means that our angular velocity $y = \dot{\theta}$ and linear velocity $v = l\dot{\theta}$ will be the same; sure they have different units, but we simply mention this here and don't worry about it going forward. The distinction will not be important for us. For wroking with RK4 method it's good to consider the system as vector in which:

$$\vec{Y} = (\theta, Y), \qquad \vec{F} = (-\sin\theta - \nu y + A\sin(\omega t)) \tag{3}$$

RK4 Method

We first provide our code, and the comments will follow:

```
% Fourth Order Runge Kutta with a constant step size
function [dynamicallist, times, iter]=RK4(initial,tfinal,nu,w,A,h)
% Definition of Parameters of the Pendulum
%nu = Damping Coefficient
%w = Frequency of Forcing
%A = Amplitude of Forcing
% Initial: vector of initial conditions.
% tfinal is the final time.
%h is the step size.
% Produces the values of the dynamical variables and the times.
% prop: function F evaluating the time derivatives.
    function [B C]=prop(K,t)
        B=K(1,2);
        C=-nu.*K(1,2)-sin(K(1,1))+A*sin(w*t);
    end
%% Initiation
iter=round(tfinal/h);
dynamicallist=zeros(iter+1,length(initial));
dynamicallist(1,:)=initial;
for i=1:iter
    [k11 k12]=prop(dynamicallist(i,:), i*h);
    [k21 k22]=prop(dynamicallist(i,:)+(h/2).*[k11 k12],(i+1/2)*h);
    [k31 k32]=prop(dynamicallist(i,:)+(h/2).*[k21 k22], (i+1/2)*h);
```

```
    [k41 k42]=prop(dynamicallist(i,:)+h.*[k31 k32], (i+1)*h);
    dynamicallist(i+1,:)=dynamicallist(i,:)+(h/6).*([k11 k12]+2.*[k21 k22]+2.*[k31 k32]+[k41 k42]);
end
times= h*(0:iter)';
end
```

In addition to the pendulum parameters, we input an initial vector $[\theta_0, v_0]$ as well as a final time tfinal. Our code outputs the times t from each of the iterations, as well as the variable "dynamicallist" which is a long array with two columns. These two columns correspond to the configuration of the pendulum $\theta(t)$ for each time t as well as the velocity of the pendulum $v(t)$ for each time t. Since our function "prop" above outputs two values (corresponding to the two first-order equations in the system), whenever we evaluate "prop" we collect its outputs into an array like "$[k11k12]$," for example. This is really the only way our code differs from the course code. Next, we provide the elementary code used to plot $\theta(t)$ as well as the phase-space portraits.

```

function Plotter
[A B iter]=RK4([0.2 0],10,10,1,0,1e-3); %The inputs above were just from one example.
L=mod(A(1:(iter+1),1)+pi,2*pi)-pi; %Enforces periodicity of angle \theta
plot(B,L)
%ylim([-pi pi]) %Enforces periodicity of the plots
xlabel('Time (Seconds)')
ylabel('Angle \theta(t) (Radians)')
title('Figure 5: Time Evolution of Configuration of Pendulum (A=0, \nu=10)')
figure;
plot(L,A(1:(iter+1),2))
%xlim([-pi pi]) %Enforces periodicity of the plots
xlabel('Angle \theta (Radians)')
ylabel('Velocity v (m/sec)')
title('Figure 6: Phase Space Portrait of Pendulum (A=0, \nu=10, )')
%The above code shows the general method we use to plot; of course,
%we will slightly change the labels, titles, etc depending on the case
%at hand.
end
```

We point out that our array "L" in the code above, simply takes the angles outputted by RK4, and computes their equivalent value in the range $[-\pi, \pi]$. In addition, when plotting over the angle θ we insist we plot only over the range$[-\pi, \pi]$ such that the periodicity is more evident. Intuitively, we can imagine our time-evolution plots as living on a horizontal cylinder, while the phase-space plots live on a vertical cylinder. The velocity is unconstrained.

2 Unforced Case

Using the fourth-order Runge-Kutta method, we are able to produce plots of the time-evolution of the configuration of the pendulum $\theta(t)$. In addition, we're able to produce plots of the evolution of the system in phase-space (θ, v), (recall, that since we've set the length of the pendulum $l = 1m$, the angular velocity and the linear velocity agree, modulo units). We first study the unforced motion ($A = 0$) with initial conditions $\theta_0 = 0.2$ and $v = 0$ for the three cases $\nu = 1, 5, 10$. We choose $tfinal = 10$.

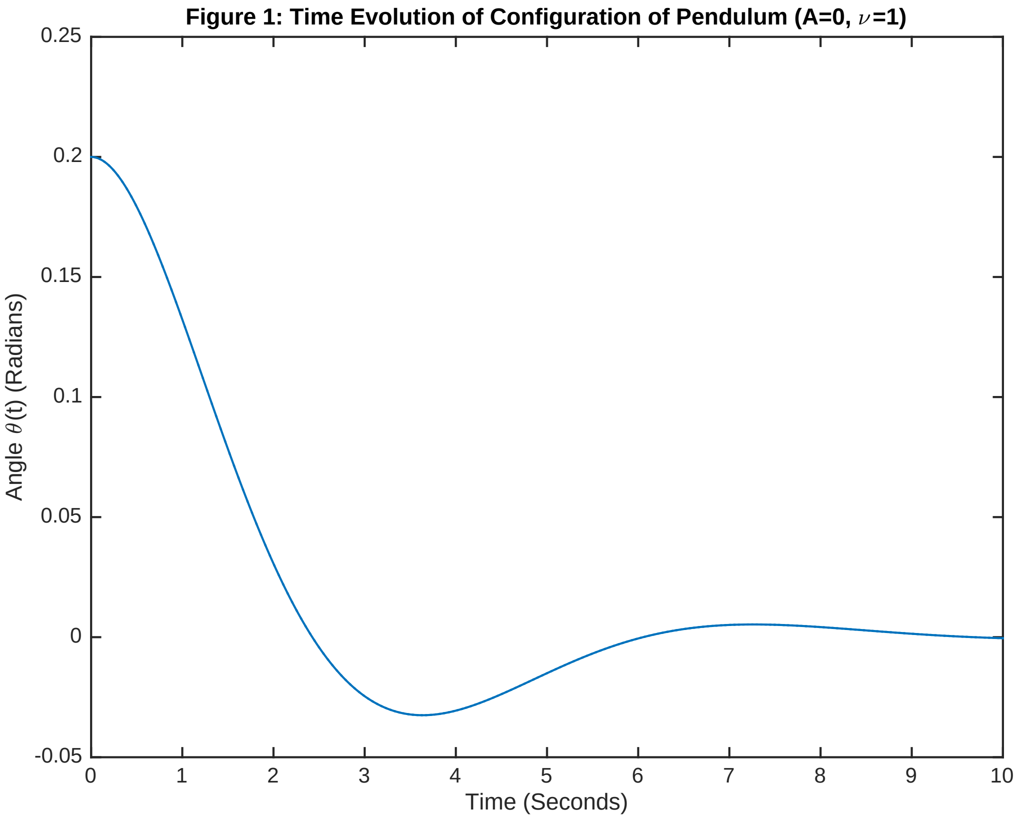


Figure 1: Time Evolution of Configuration of Pendulum (A=0, ν=1)
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
1
2
3
4
5
6
7
8
9
10
Time (Seconds)
Angle θ(t) (Radians)

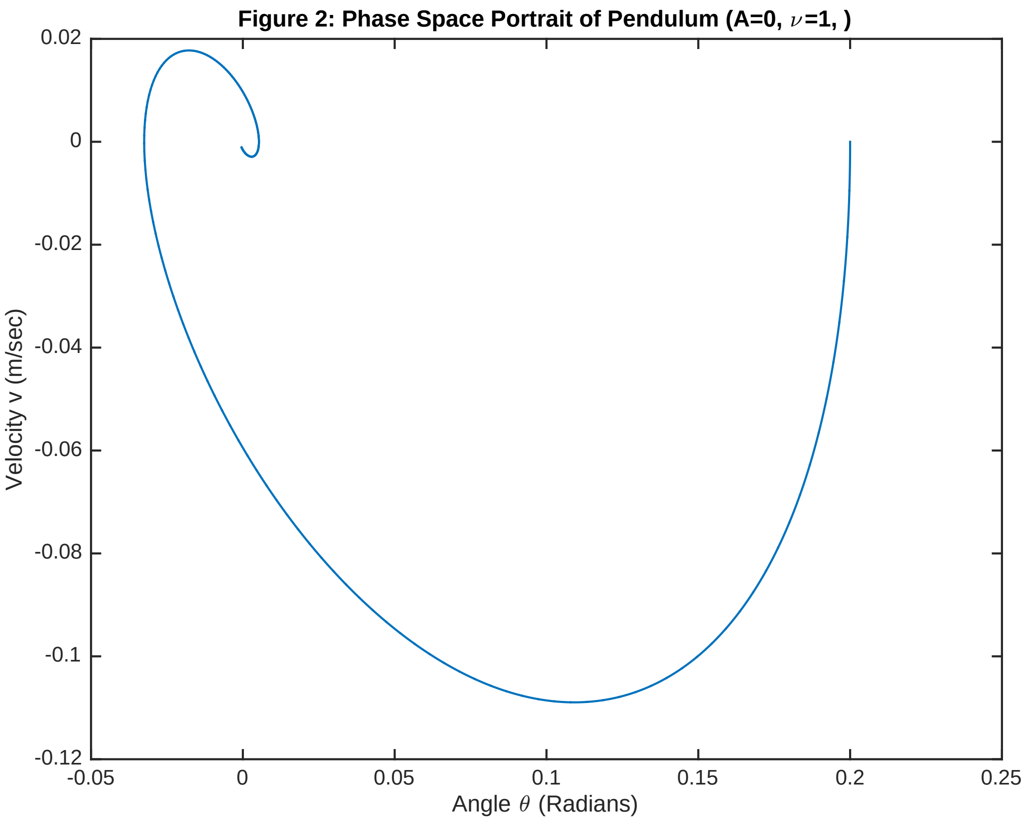


Figure 2: Phase Space Portrait of Pendulum (A=0, ν=1,)
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
Velocity v (m/sec)
-0.05
0
0.05
0.1
0.15
0.2
0.25
Angle θ (Radians)

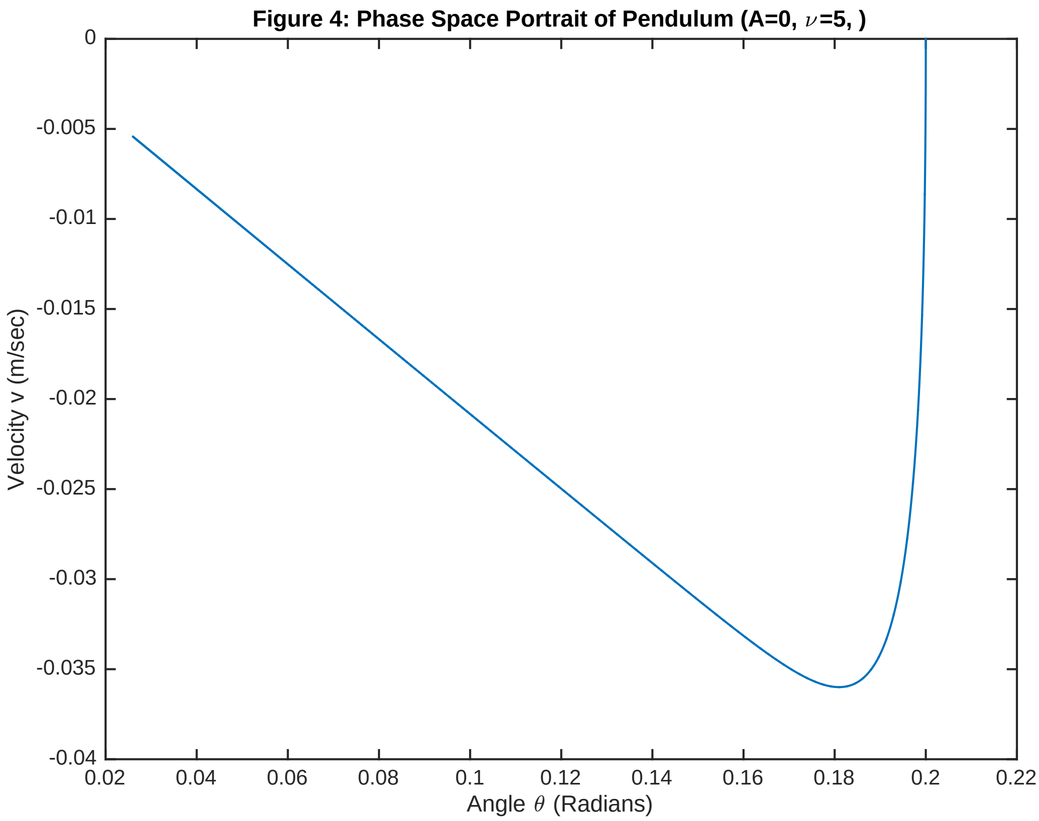
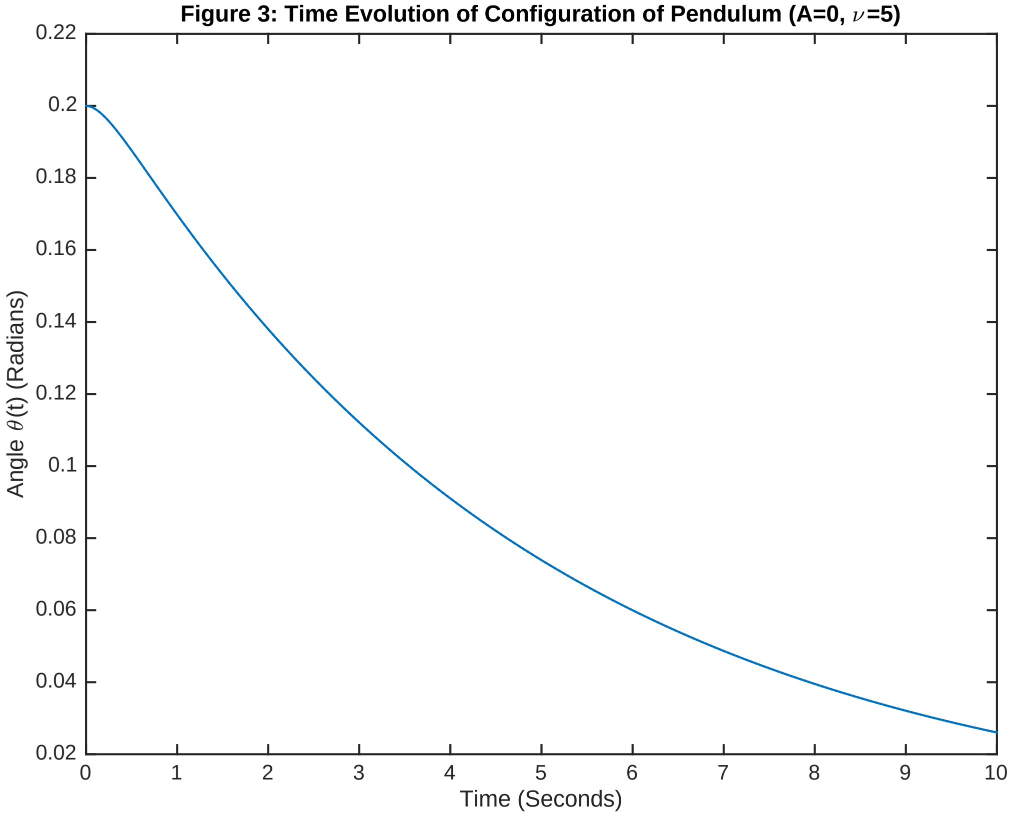


Figure 4: Phase Space Portrait of Pendulum (A=0, ν=5,)
Velocity v (m/sec)
0
-0.005
-0.01
-0.015
-0.02
-0.025
-0.03
-0.035
-0.04
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
Angle θ (Radians)

Figure 3: Time Evolution of Configuration of Pendulum (A=0, ν=5)



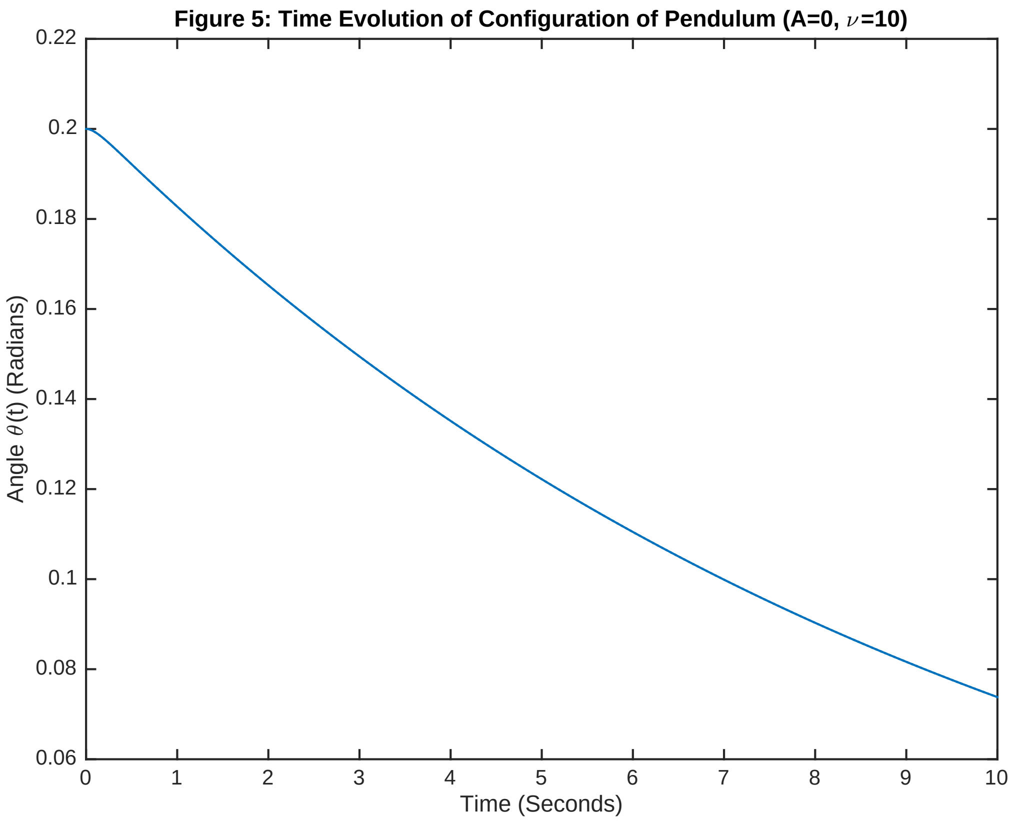


Figure 5: Time Evolution of Configuration of Pendulum (A=0, ν=10)
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
Angle θ(t) (Radians)
0
1
2
3
4
5
6
7
8
9
10
Time (Seconds)

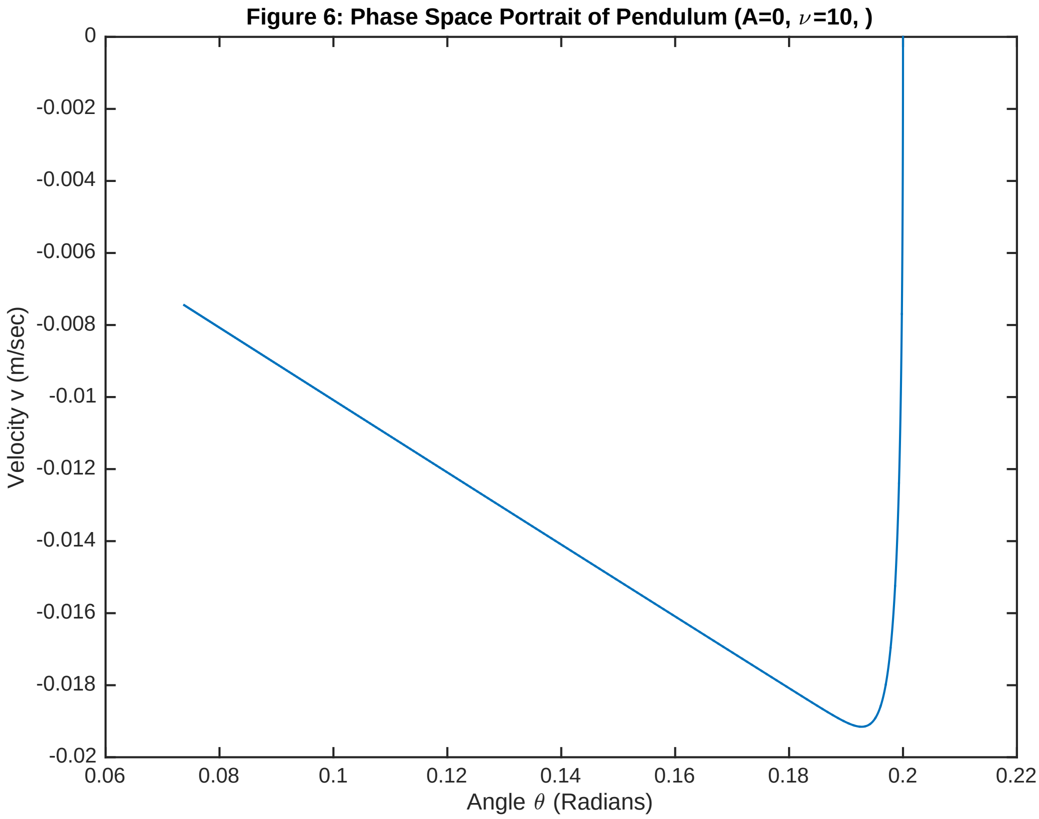


Figure 6: Phase Space Portrait of Pendulum (A=0, ν=10,)
0
-0.002
-0.004
-0.006
-0.008
-0.01
-0.012
-0.014
-0.016
-0.018
-0.02
Velocity v (m/sec)
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
Angle θ (Radians)

We can see from the qualitative behavior of the plots that $\nu = 1$ corresponds to underdamped, $\nu = 5$ corresponds to damped, and $\nu = 10$ corresponds to over-damped. The first one is relatively obvious: $\nu = 1$ is the only one of the three plots that exhibits any sort of oscillatory behavior at all. This is underdamping. Now let's consider the final two cases. Notice that $\theta(t)$ for $\nu = 5$ damps more quickly down to zero than it does for $\nu = 10$. How does this make sense; isn't it backwards? It actually makes good physical sense, recalling that we're starting with zero initial velocity $v0 = 0$ and some non- zero displacement $\theta_0 = 0.2$. Thus, the larger the damping, the harder it is for the pendulum to swing down to $\theta = 0$. In the limit of infinite damping, the pendulum ought to simply sit fixed at $\theta = 0.2$.

Forced Motion

Before producing plots, we need to discuss a method to choose a step size h which yields a desired threshold of accuracy.

The total error in the fourth-order Runge-Kutta method is

$$\epsilon = \frac{\eta}{h} + Ch^5 \tag{4}$$

where the first term is the round off error (η is machine precision) and the second term is the discretization error. The coefficient C is not exactly a constant: it is independent of the step-size h, but it depends on time. Assume the solution we're looking for is $\theta(t)$. We define the quantities u and v by,

$$\theta(t) = u + Ch^5, \qquad \theta(t) = v + C(\frac{h}{2})^5 \tag{5}$$

In other words, to find u we apply Runge-Kutta with a step-size h while to find v we apply Runge- Kutta with a step-size $\frac{h}{2}$. It follows from this that,

$$Ch^5 = \frac{u - v}{1 - 2^{-5}} \approx u - v \tag{6}$$

This gives us a simple way of keeping track of the error in our algorithm: we simply plot $u - v$ as a function of time. The code we used for this is given here

```
function errorAny
[A1 B1 iter1]=RK4([0.2 0], 300, 1/2, 2/3, 0.5, 1e-3); %Run RK4 twice...
[A2 B2 iter2]=RK4([0.2 0], 300, 1/2, 2/3, 0.5, (1e-3)/2); %with different step
    size
L1=mod(A1(1:(iter1+1),1)+pi,2*pi)-pi; %Insist on periodicity
L2p=mod(A2(1:(iter2+1),1)+pi,2*pi)-pi; %Insist on periodicity
L2 =L2p(1:2:length(L2p)); %Take only every other entry from L2p, due to half
    step size
J=abs(L1-L2);
plot(B1, J)
ylim([0 0.00145])
xlabel('Time (seconds)')
ylabel('u-v (Error in RK4)')
title('Figure7 : Error Analysis in RK4 Method')

end
```

To summarize, we run the Runge-Kutta algorithm twice for different step-sizes, and we compute the difference in the results. As we proved above, analytically, this is equal to the discretization error Ch^5. The results are shown in Figures 7 and 8 below for different amplitudes, where we arbitrarily chose $h = 1 \times 10^{-3}$. As we predicted, the error depends on time. The method we take is to note that the highest value the plot reaches is $\approx 3.0 \times 10^{-3}$ for $A = 0.5$. Thus, we can bound the coefficient $|C| < 3$. but the situation for $A = 1.2$ is more different: is in the unstable manifold of the chaotic behaviour, and the error is timestep-sensitive. It grows exponentially over time.

The idea is that because C is independent of h, we can vary h however we please, and we still have this $|C| < \alpha$. We can then argue that we want some level of precision, and solve for the step-size h which provides this. In principle, this should be done each time we vary the amplitude of the pendulum, we show only two examples here. For example, let's say we want precision to 10^{-15}, We then have

$$\alpha h^5 = 10^{-15} \rightarrow h = \alpha^{-1/5} \times 10^{-3} \tag{7}$$

The key is that we've argued that this value roughly gives us precision 10^{-15} even in the worst case scenario since we took the upper bound on the error.

We emphasize that, in principle, this should be done for each amplitude; we've only shown two examples here. We found that$|C| < 1.35$for $A = 1.465$ and $|C| < 0.3$ for $A = 0.5$. We carry out this same method (without providing all the plots) for some of the amplitudes we will be using. We find h_A, the step-sizes for amplitude A to be:

$$h_{0.5} = 5.3 \times 10^{-3}, h_{1.2} = 1.08 \times 10^{-3}, h_{1.35} = 9.8 \times 10^{-4}, h_{1.465} = 9.4 \times 10^{-4} \tag{8}$$

Going forward, we use these adaptive step-sizes in our plots with $A \neq 0$.

We now provide the plots, given that we've defended our choice of h.

The main thing to notice from these last three plots, is that they appear to have a intimate relation to the period of the forcing. Since $\omega = 2/3$, the period of the forcing is $T = 3\pi$. Notice from Figure with$A = 1.35$ that the pendulum also appears to have period roughly 3π. In Figure with $A = 1.44$, it appears that the pendulum completes one of its periods each time the forcing completes two of its periods. It is much less clear, but it turns out that the plot in Figure with $A = 1.465$also completes one of its periods as the forcing completes some larger number of its periods. We will make all of this more precise in the next section, as well as discuss how it relates to Poincare sections, and the interesting phenomena of "chaos".

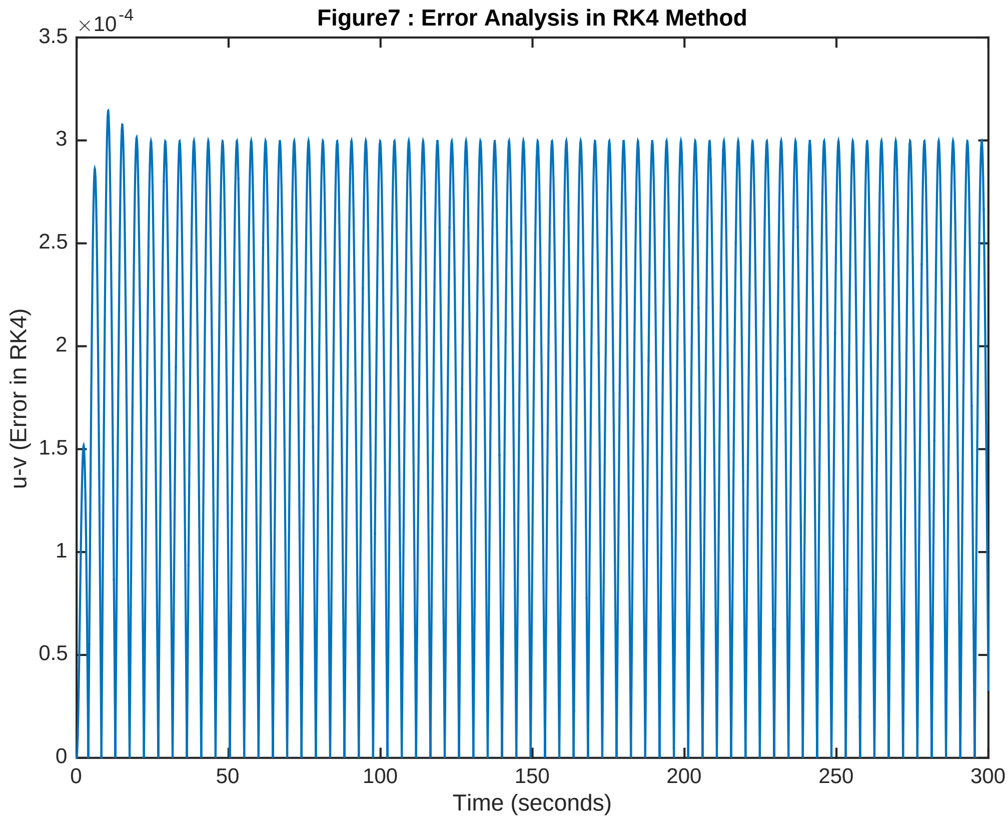


Figure 1: A=0.5, it's seems more normal than A=1.2

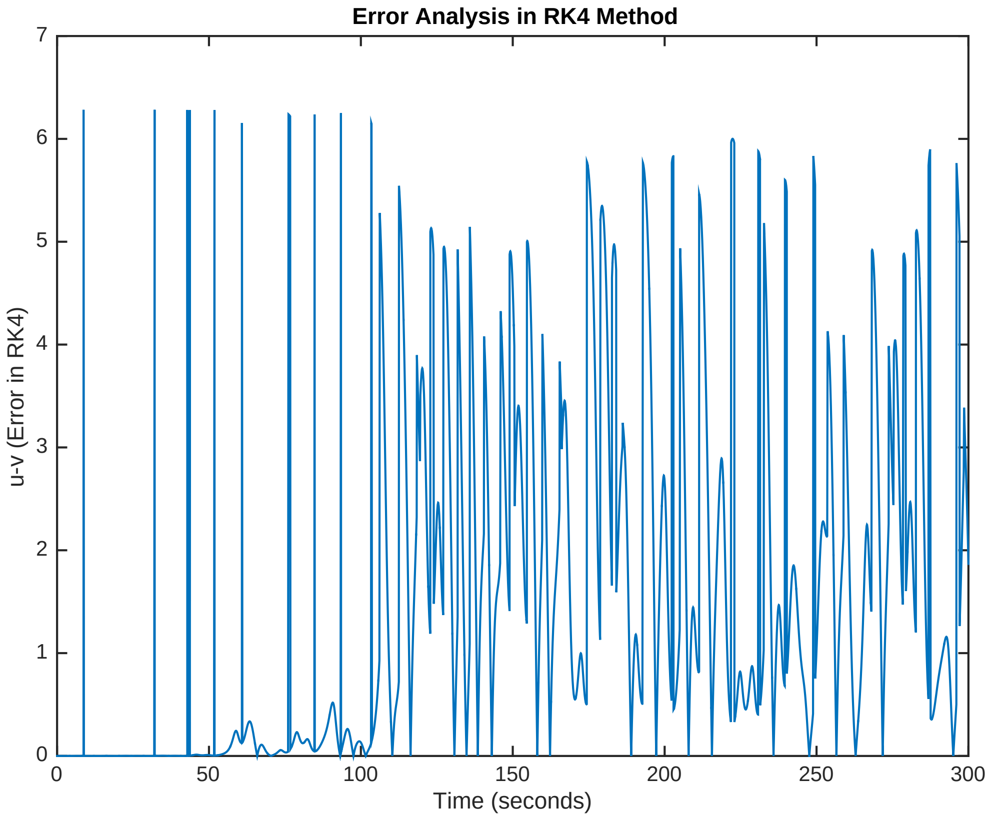


Figure 2: A=1.2, is in the unstable manifold of the chaotic behaviour, and the error is timestep-sensitive. It grows exponentially over time.

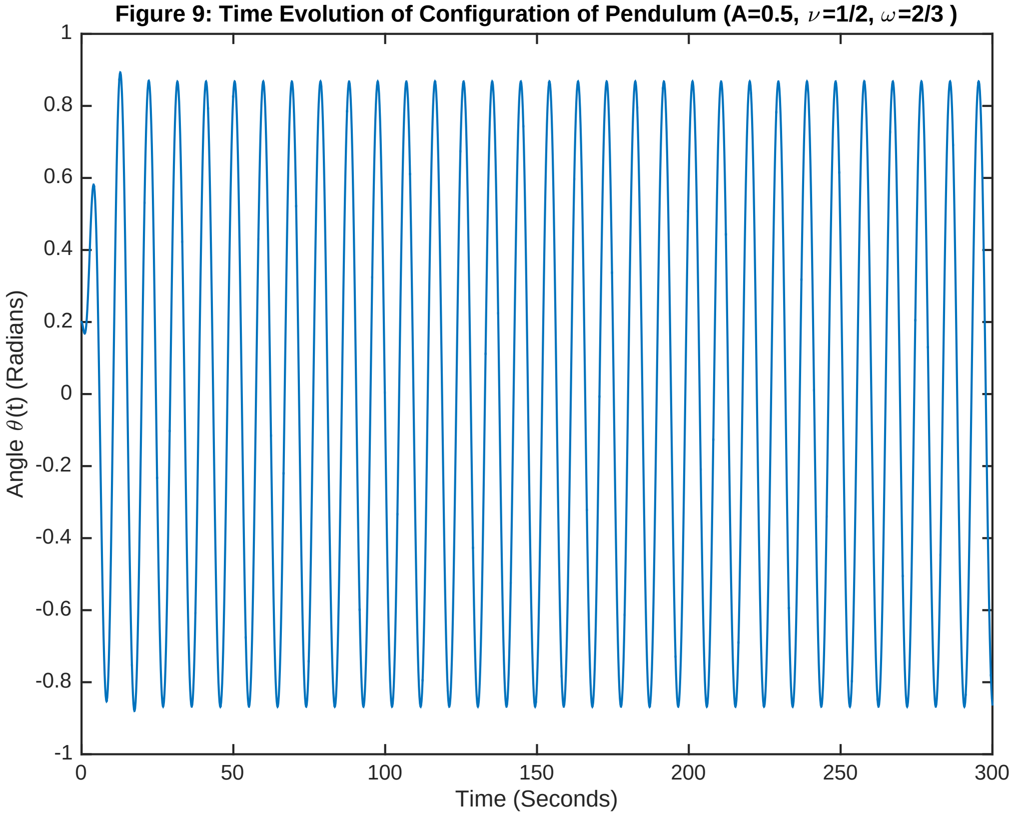


Figure 9: Time Evolution of Configuration of Pendulum (A=0.5, ν=1/2, ω=2/3)
Angle θ(t) (Radians)
Time (Seconds)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
50
100
150
200
250
300

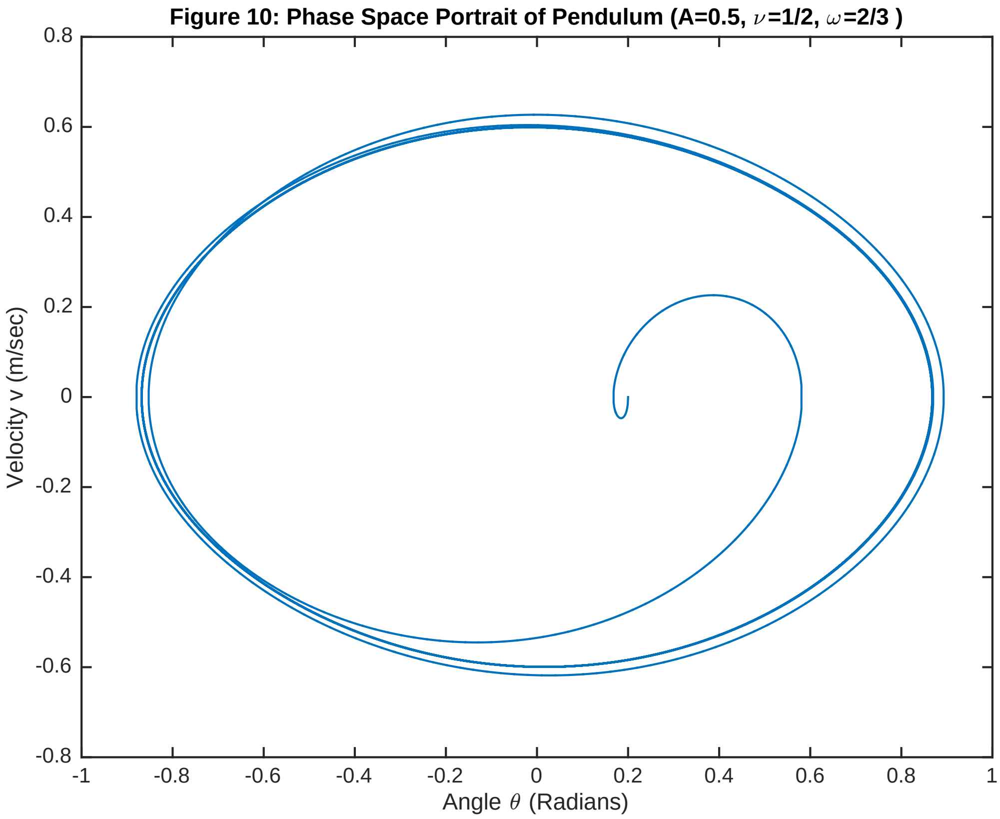
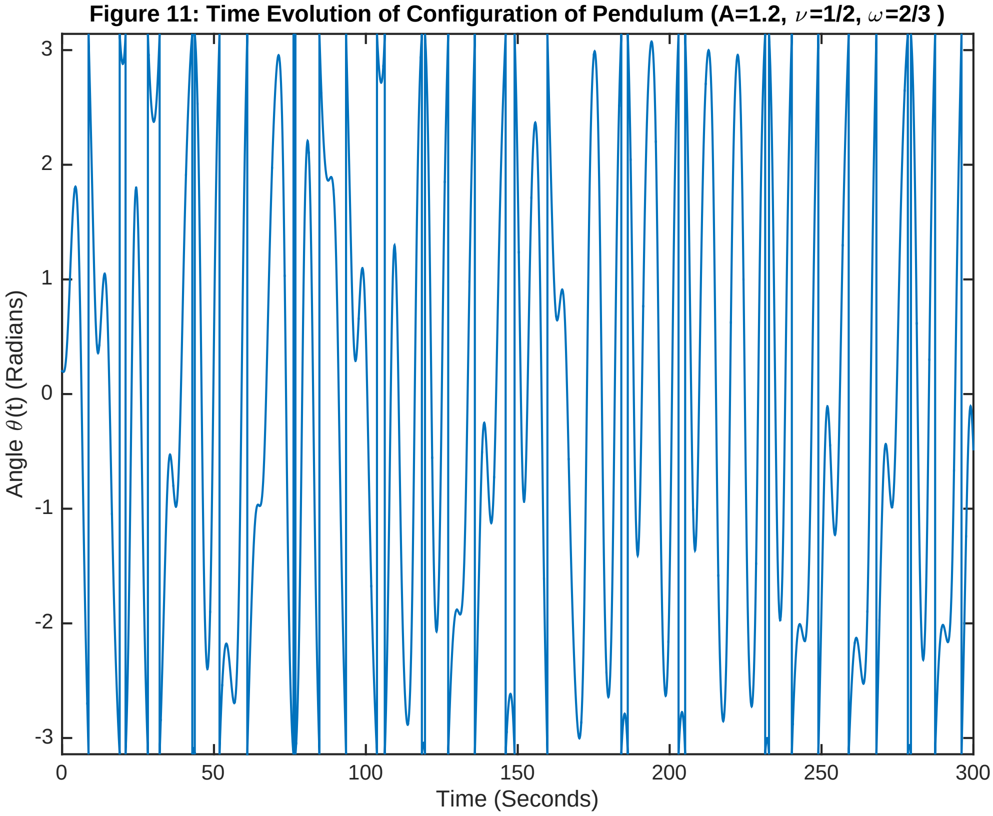


Figure 10: Phase Space Portrait of Pendulum (A=0.5, ν=1/2, ω=2/3)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Velocity v (m/sec)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Angle θ (Radians)

Figure 11: Time Evolution of Configuration of Pendulum (A=1.2, ν=1/2, ω=2/3)



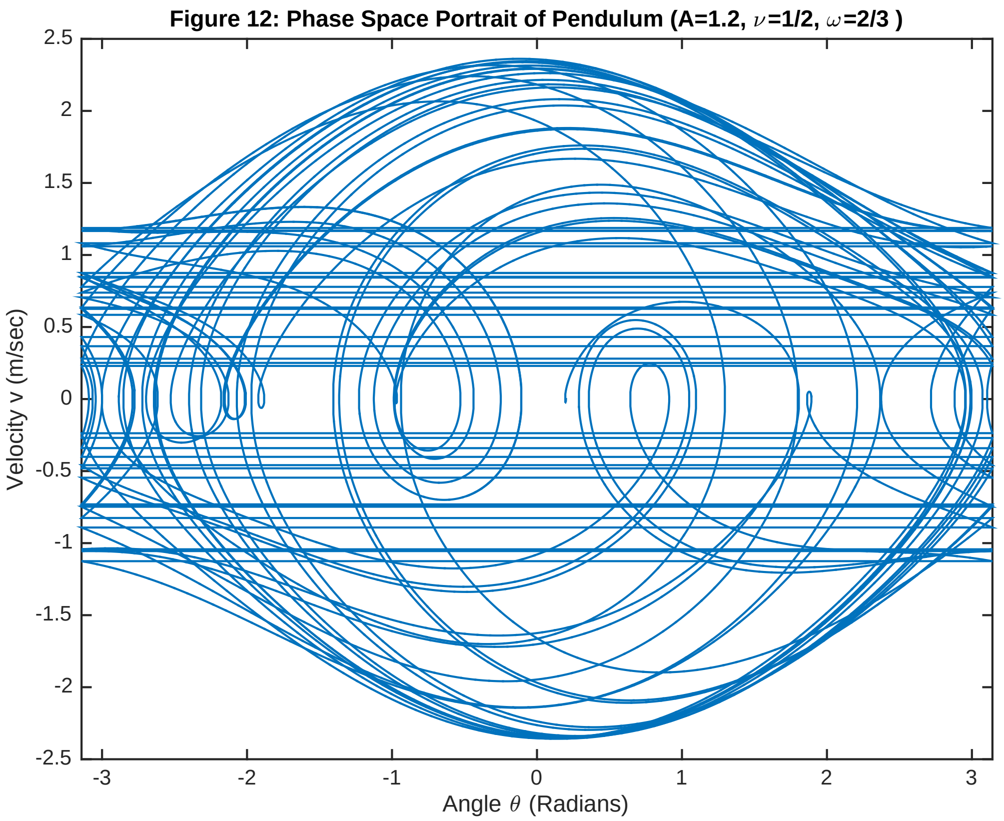


Figure 12: Phase Space Portrait of Pendulum (A=1.2, ν=1/2, ω=2/3)
Velocity v (m/sec)
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-2
-1
0
1
2
3
Angle θ (Radians)

Figure 13: Time Evolution of Configuration of Pendulum (A=1.35, ν=1/2, ω=2/3)

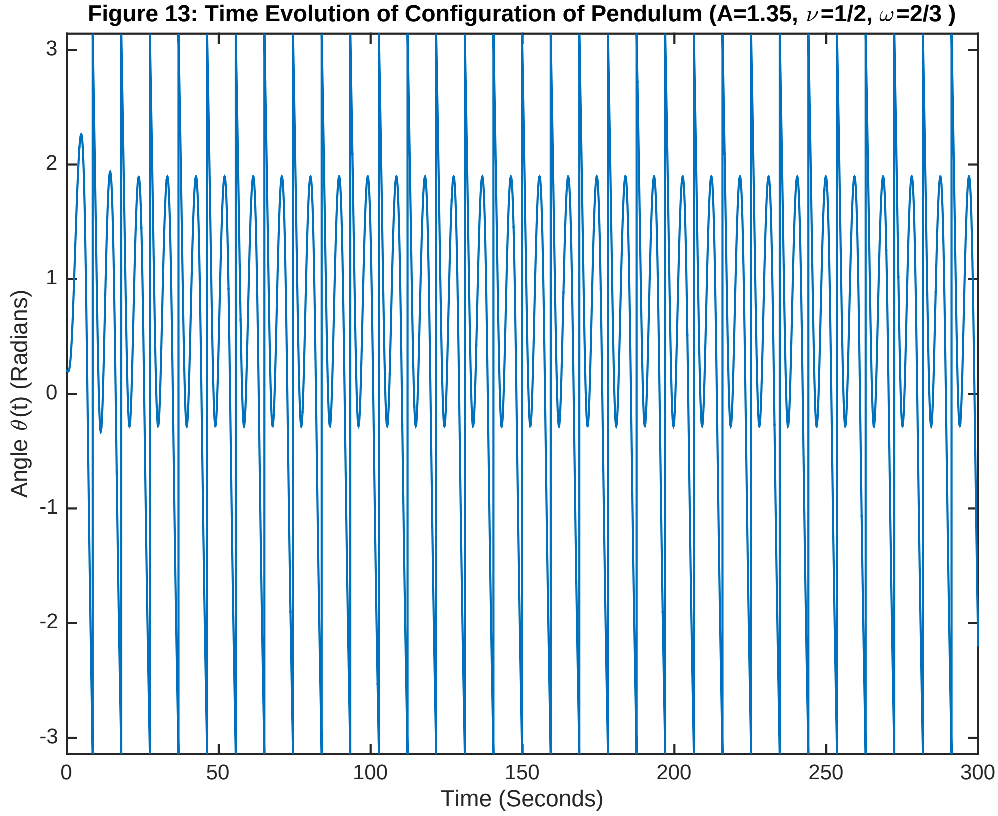


Figure 14: Time Evolution of Configuration of Pendulum (A=1.465, ν=1/2, ω=2/3)

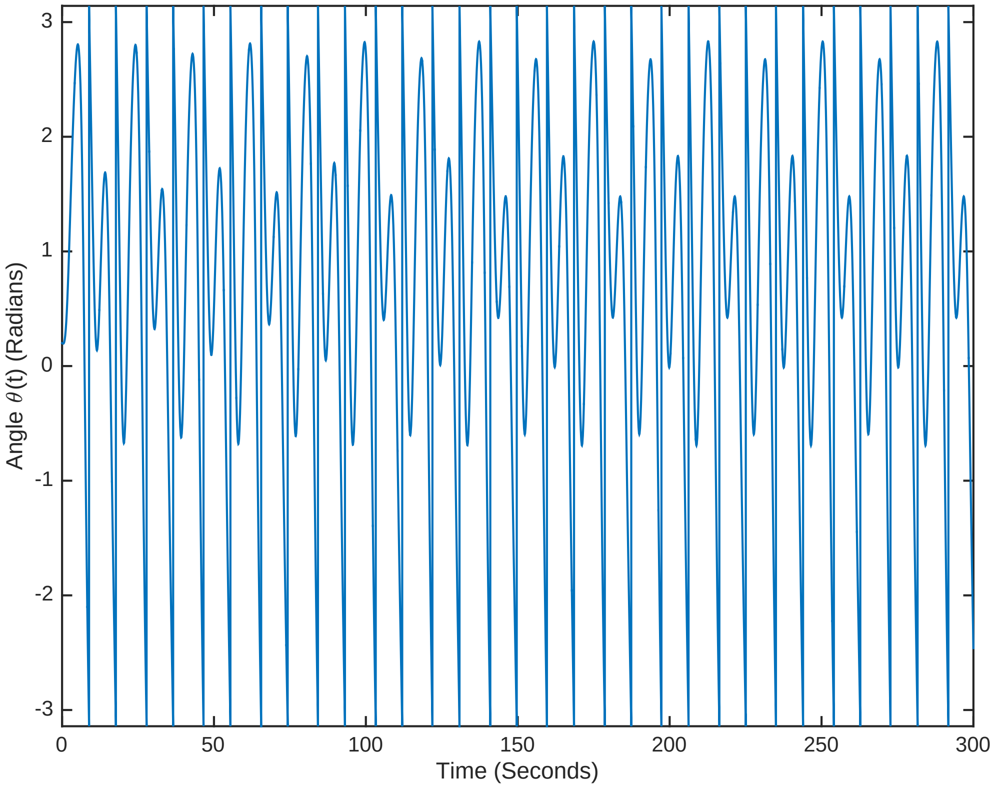
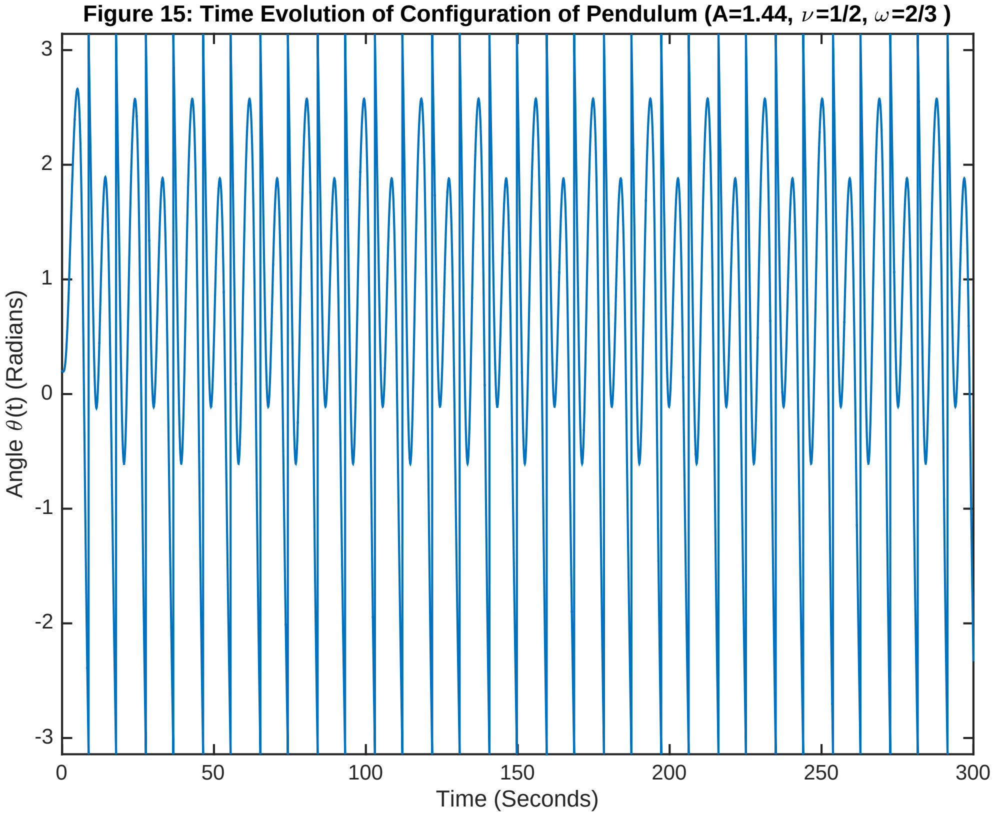


Figure 15: Time Evolution of Configuration of Pendulum (A=1.44, ν=1/2, ω=2/3)



3 Poincare Sections and the Chaotic Non-linear Pendulum

Assume the pendulum oscillates with angular frequency Ω and that the angular driving frequency is ω. If the resulting pendulum motion completes one period in exactly the same time the forcing completes one period, we refer to this as "Period-1 motion". If we were to plot just one point in phase-space for each period of the forcing, the resulting plot would have just a single point. In other words, 1-Period motion would lead to all these points piling up on top of each other. Now, assume the pendulum completes one period every time the forcing completes two periods. We refer to this as "Period-2 motion."

Plotting one point in phase-space for each period of the forcing, would lead to a phase space portrait with just two points; the points would pile up on top of each other in an alternating fashion. Not surprisingly, we define "Period-n motion" by

$$\frac{2\pi}{\Omega} = \frac{2\pi n}{\omega} \tag{9}$$

We want to plot a point in phase-space if and only if the time is some integer multiple of the period of the forcing,

$$t = (\frac{2\pi}{\omega})k, k \in Z \tag{10}$$

If the parameters work out such that the system oscillates with period-n motion, for some n, then we will see precisely n discrete points plotted in phase-space. However, there will exist certain parameters where the resulting phase-space portrait will exhibit "chaos," which means there will be nice, exotic plots as opposed to the simple discrete points. Below we will investigate the parameter space and produce the resulting Poincare sections. To reiterate, we will plot a large number of points (we will choose 5000), one for each period of the forcing. For certain parameters the resulting plots will consist of a small number of discrete points. This corresponds to n-periodic motion described above. For other parameters, we will see chaos.

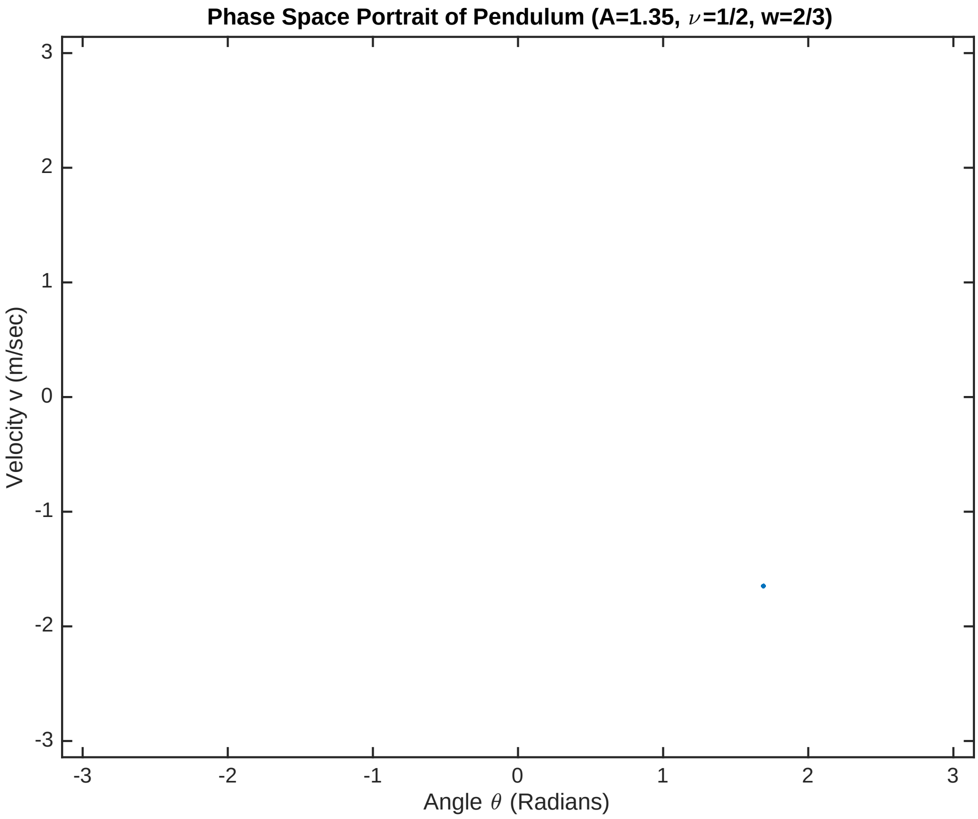
We now provide the code which produces the plots above.

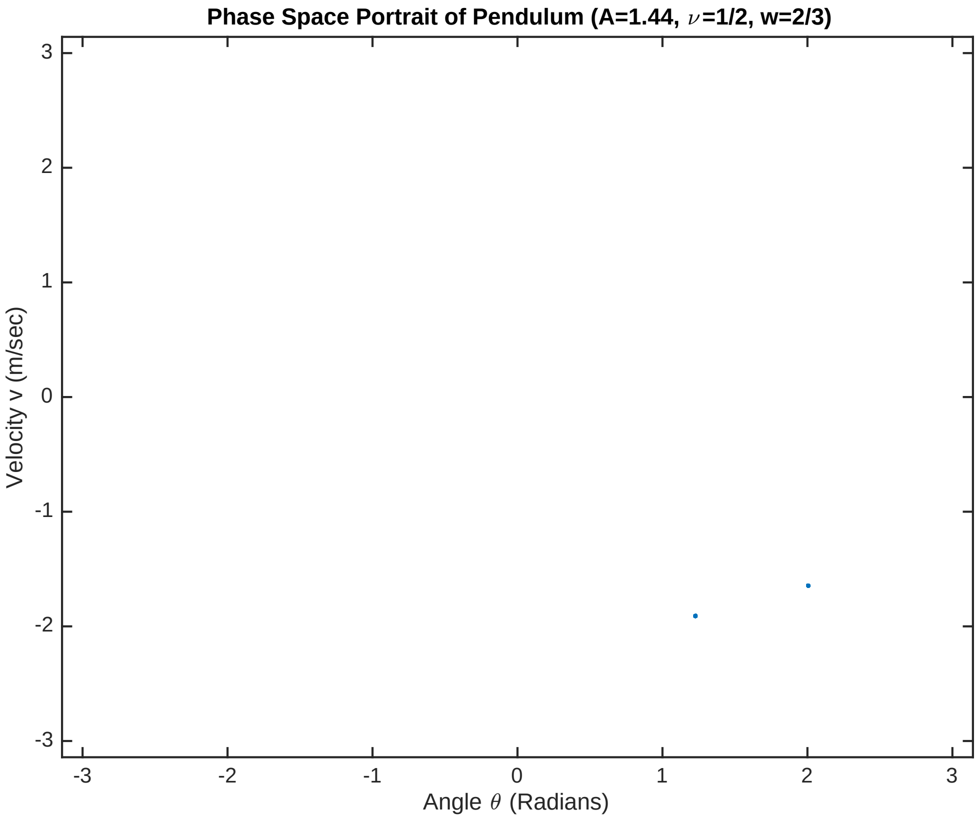
```
function PoincareSection(A,n)
%We fix w=2/3 and nu=1/2; input amplitude A and number of periods n.
T=2*pi/(2/3); %The period of the forcing for w=2/3.
h=T/1000; %Our step size for 1000 points per forcing period.
[B C iter]=RK4([0.2 0], n*T, 1/2, 2/3, A, h);
A1=zeros(1,n+1);
A2=zeros(1,n+1);
for i=1:n+1 A1(i)=A1(i)+B(((i-1)*1000)+1,1);
A2(i)=A2(i)+B(((i-1)*1000)+1,2); end
B1=mod(A1+pi,2*pi)-pi; %Imposes periodicity in [ pi, pi].
plot(B1(6:end),A2(6:end),'.')
%Notice that we don't plot over early entries: this is transient behavior. xlim([ pi pi]) %Imposes periodicity in plot.
xlabel('Angle \theta (Radians)')
ylabel('Velocity v (m/sec)')
title(' Phase Space Portrait of Pendulum (A=1.35, \nu=1/2, w=2/3)')
end
```

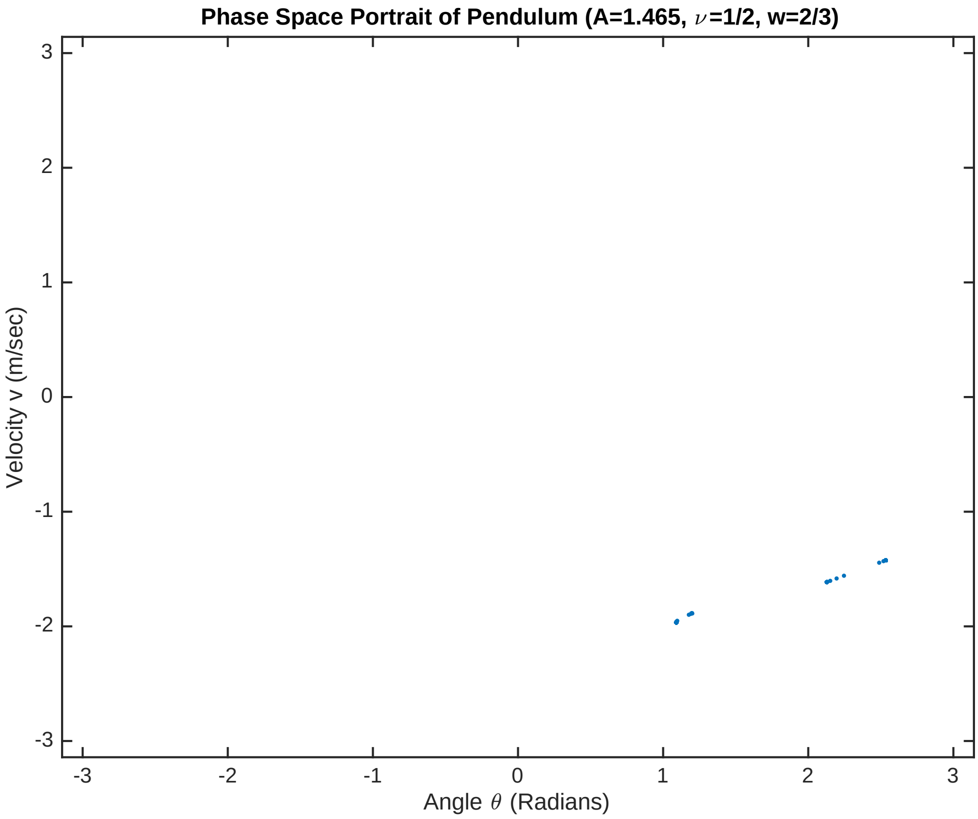
There are a couple of things to point out from the above code. We fix T to be the pendulum period, and take the final time to be nT for a large integer n which we input. In our case, we let $n = 5000$. Choosing h such that we have 1000 steps each period T , we see that the integer multiples of T occur every 1000 entries in our arrays B and C from RK4. Therefore, the only purpose of the for loop is to pick out these specific entries. Finally, we point out that there will be some initial transient behavior. We see this die out very quickly, but we still do not plot over the first 6 entries, so as to leave out the effects of this transience.

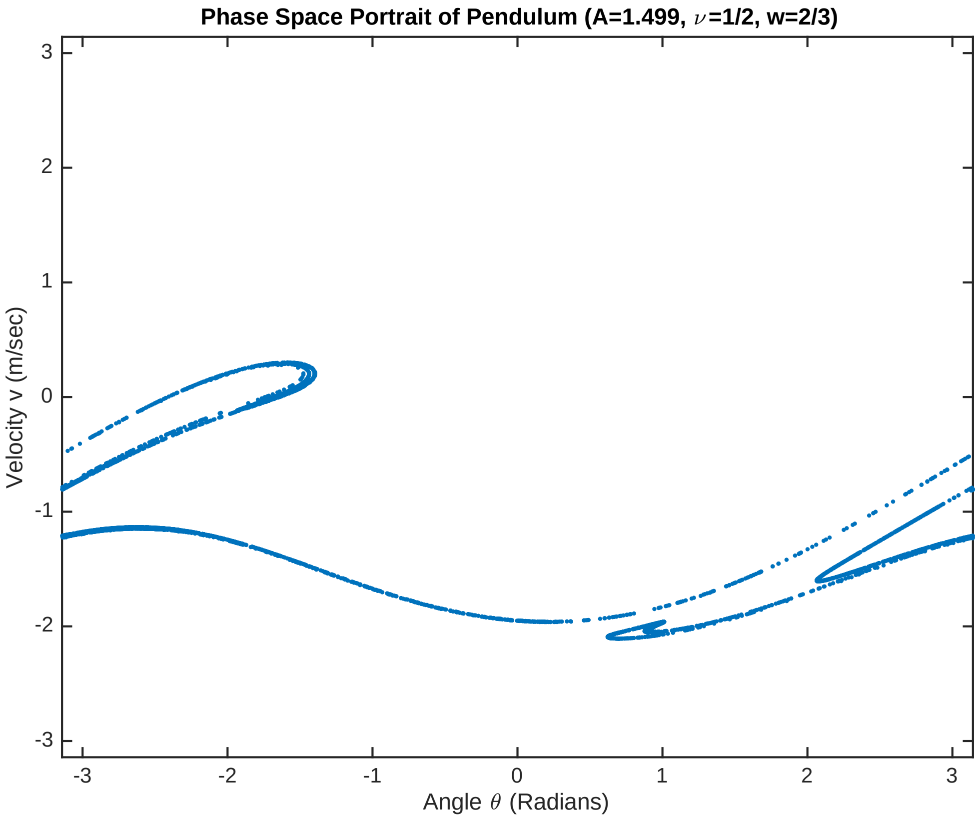
For A = 1.35 there is a single point which means we have 1-periodic motion as described above. For A = 1.44, there are exactly 2 points which means the motion is 2-periodic. For A = 1.47 the motion is 4-periodic.

For A = 1.499 and A = 1.50 something very interesting happens The motion becomes chaotic. No longer do we have a bunch of the 5000 total points piling up on top of each other, but rather they spread out nicely to approximate smooth sections. We see that again for A = 1.15 we get chaotic motion, while for A = 1.10 we're back to periodic motion. This hints that the parameter space of A has a very subtle structure with respect to periodicity and chaos. Investigating this space more completely leads to what are called "bifurcation diagrams" which are outside of our scope.

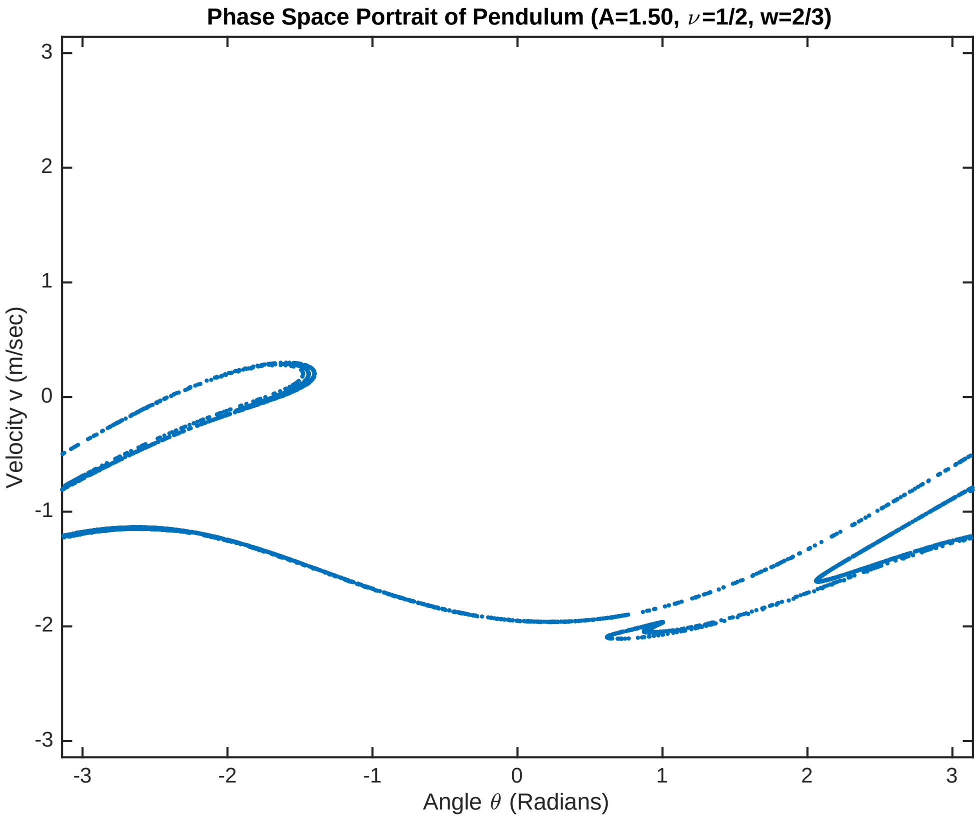

Phase Space Portrait of Pendulum (A=1.35, ν=1/2, w=2/3)
Velocity v (m/sec)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
Angle θ (Radians)

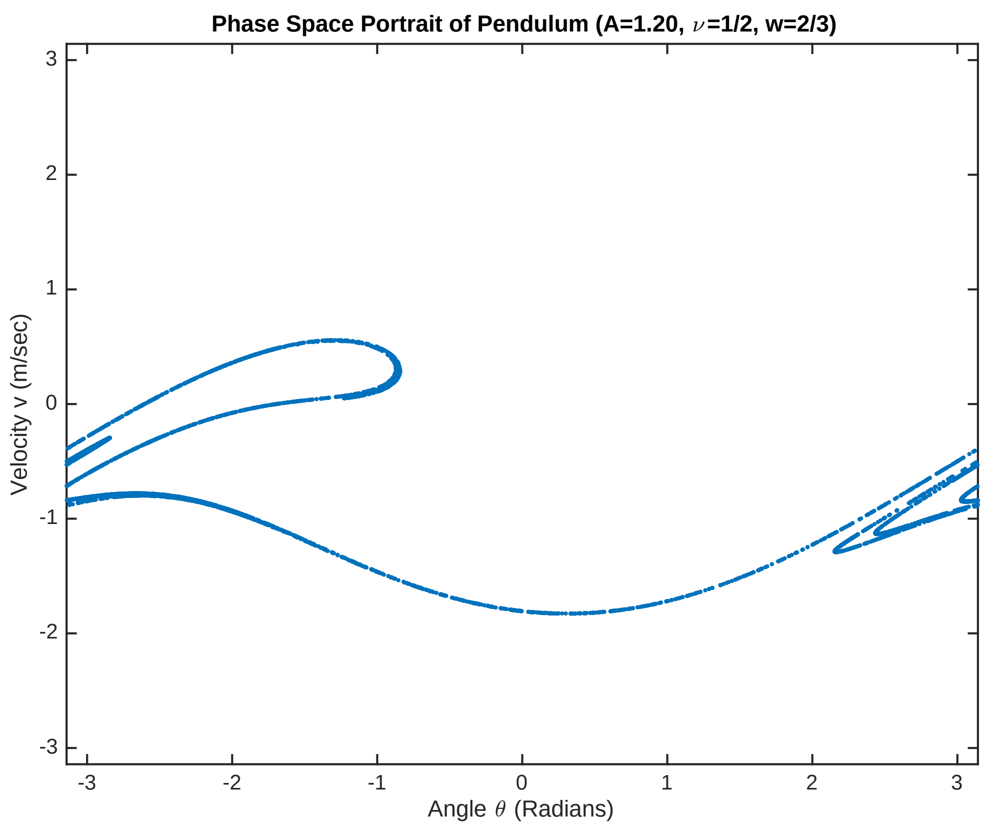

Phase Space Portrait of Pendulum (A=1.44, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

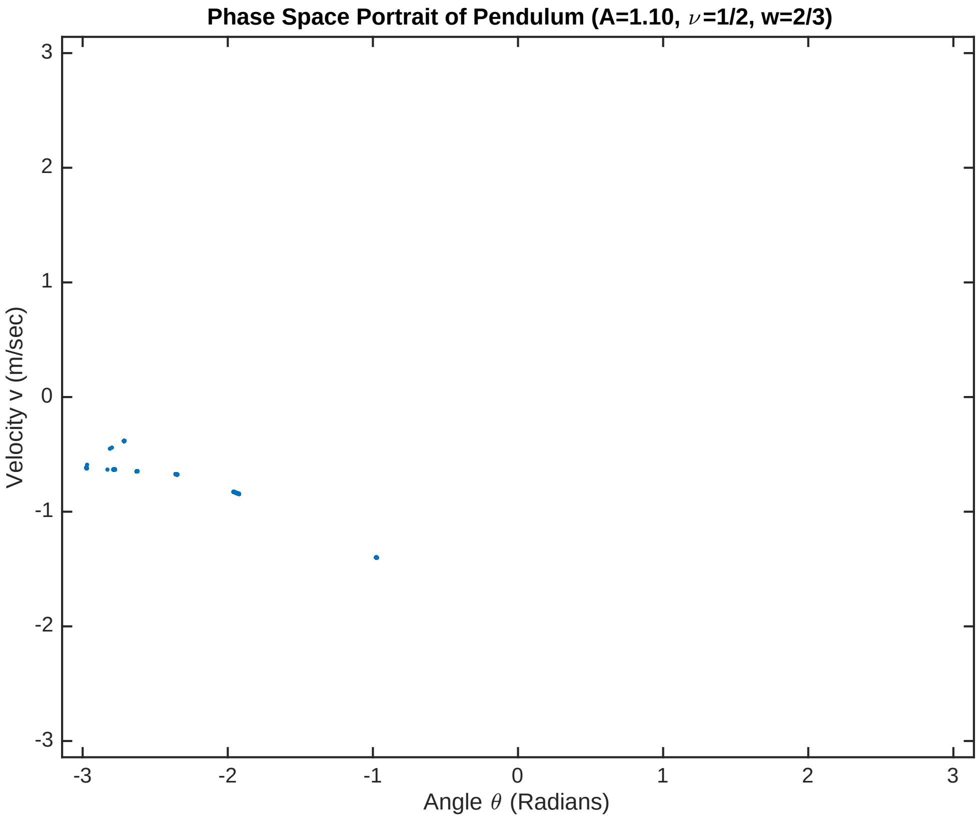

Phase Space Portrait of Pendulum (A=1.465, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3



Phase Space Portrait of Pendulum (A=1.499, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


Phase Space Portrait of Pendulum (A=1.50, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


Phase Space Portrait of Pendulum (A=1.20, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


Phase Space Portrait of Pendulum (A=1.10, ν=1/2, w=2/3)
Velocity v (m/sec)
Angle θ (Radians)
3
2
1
0
-1
-2
-3