PHYS 325 Assignment 5: Oscillatory Motion and Chaos

Author: A. J. Ogle
Date: May 8th, 2015
PHYS 325 Computational Physics

1 Introduction

Normally, Euler's method suffices for solving equations of motion computationally. For oscillatory motion, Euler's method is simply not enough. As will be later shown, Euler's method does not conserve energy when used to model the motion of a simple pendulum. In this section, the use of two alternate computational methods are explored in the context of simple harmonic motion. These are the Euler-Cromer and Runge-Kutta methods. Both methods have their advantages and disadvantages. For situations in which Euler's method suffices to account for energy conservation, it is appropriate. But for situations in which this is not the case, such as simple harmonic motion (SHO), more computationally expensive methods such as Euler-Cromer and Runge Kutta are required. Accuracy of the model is considered prior to that of computational efficiency.

2 Background

The simple harmonic oscillator is not just a model for the motion of a pendulum on a string, but actually serves as an excellent representation of any stable system with oscillatory motion. This system could be quantum in nature, such as an electron bound to the hydrogen atom, but for this report, the focus will be on the macroscopic SHO. For a SHO, the following forces act on the pendulum:

$$F_\theta = -m * g * sin(\theta) \approx -m * g * \theta \tag{1}$$

This gives the following general form for the SHO:

$$\frac{d * \omega}{dt} = -k * \theta^\alpha \tag{2}$$

For problems 3.1 and 3.2, $k = \frac{g}{l}$ and $\alpha = 1$. In problem 3.4, the case for $\alpha = 3$ will be considered.

The general solution for this differential equation is:

$$\theta = \theta_0 * sin(\Omega * t + \phi), for\ \Omega = \sqrt[2]{\frac{g}{l}} \tag{3}$$

The Euler-Cromer computational method is nearly identical to the Euler method except for one term. Consider the following iterations:

$$\omega_{i+1} = \omega_i - (\frac{g}{l}) * \theta_i * \delta_t \tag{4}$$

$$\theta_{i+1} = \theta_i + \omega_{i+1} * \delta_t \tag{5}$$

$$t_{i+1} = t_i + \delta_t \tag{6}$$

This iteration is identical to the Euler method except for $\omega_{i+1} * \delta_t$ in **??**. The difference is that the new value of ω is used to calculate the new value of θ. Interestingly, this is the only difference between the Euler and Euler-Cromer method, yet it results in conservation of energy for SHO.

In the next few sections, energy conservation will be explored for the SHO model with the Euler, Euler-Cromer, and Runge-Kutta methods.

3 Problem 3.1: Stability of the Euler-Cromer Method

The Euler-Cromer method differs from the Euler method by only one term when calculating the new θ.

If the Euler method is used to model the SHO, energy is not conserved. These are the results shown in **??**. This is because the Euler method is only an approximation, with some amount of error. For the oscillatory motion of the SHO, the Euler method's error compounds too quickly, giving a result which does not agree with energy conservation. All computational methods of modeling physical situations contain some amount of error which compounds after amount of iteration. It is more a matter of how quickly the error compounds which determines the usefulness of the computational model.

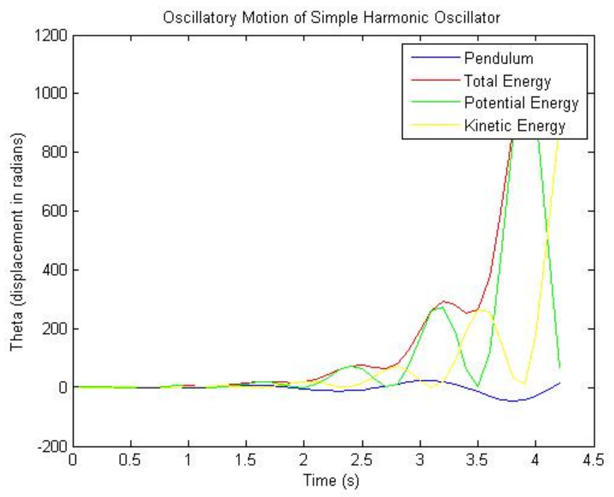

Figure 1: The Euler method is used here to model the SHO. This system is not stable because energy is continually increased for each part of the pendulum's approximated motion.

A better model for the SHO is the Euler-Cromer method, which conserves energy over each cycle of the pendulum's motion.

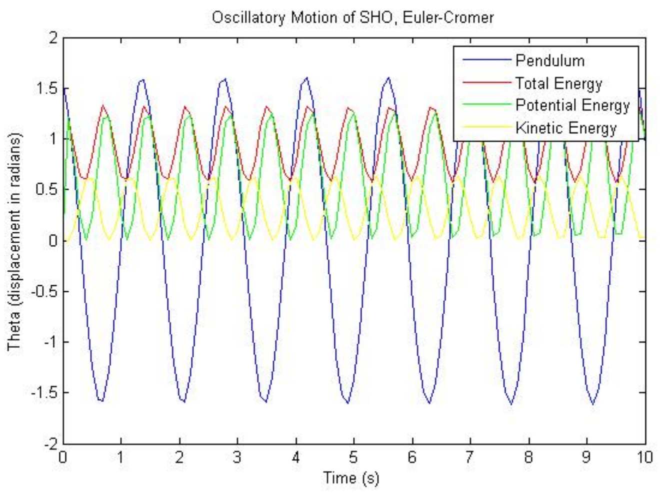

Figure 2: The Euler-Cromer method for modeling the SHO. Notice that while the total energy changes over time, it is conserved over complete cycles of the pendulum's motion.

For this method, shown in **??**, the total energy of the system does not remain constant, but if integrated over the cycles of the pendulum's motion, it can be shown that the total energy of the system is conserved. This is a better model for the SHO, since the energy does not continually add. Nonetheless, there are better methods which maintain constant total energy throughout the motion of the SHO. One of these methods is the Runge-Kutta Method, which is computationally more expensive. In general, it is more important to have a model which represents the physical situation accurately than it is to have a model which has the smallest run time. Run time is sacrificed in place of a more accurate model. The accuracy of this method is considered in the following section.

4 Problem 3.2: Stability of the Runge-Kutta method

The Runge-Kutta method is the most accurate of method for modeling the SHO since it conserves total energy of the system for any time interval. Another useful aspect of the Runge-Kutta method is that it can be used with any order of approximation. For a 1st order Runge-Kutta method, it is equivalent to the Euler method. the issue with the Euler method, however, is that it only considers the initial conditions of a single time step. The Runge-Kutta method is able to approximate the Euler method at the initial part of the time step and then at smaller intervals throughout the time step depending on the order of the Runge-Kutta used. For the scopes of this paper, a Runge-Kutta of order 2 is used.

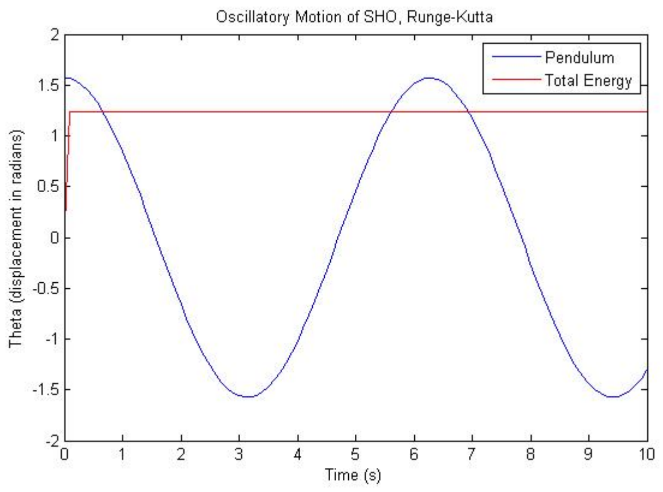

Figure 3: The Runge-Kutta method conserves energy for any time interval. Here, l = 0.5 m and m = 1 kg.

5 Problem 3.4: Harmonic and Anharmonic Oscillators

A harmonic and an anharmonic oscillator differ in their general form of their equation of motion. The very general equation for simple harmonic motion is:

$$\frac{(d^2)x}{dt^2} = -kx^\alpha \tag{7}$$

For a harmonic oscillator, $\alpha = 1$. For an anharmonic oscillator, $\alpha = 3$. The behavior expected of a harmonic oscillator is that the amplitude of the motion does not affect its period. For an anharmonic oscillator, it is expected that the amplitude of the motion does affect its period, and that the period becomes longer as the amplitude is decreased.

While the results of this paper were able to show that for $\alpha = 1$, the period did not depend on amplitude, this was also proven for the case of $\alpha = 3$. These results are not as expected, and further examination of the simulation used to obtain the results is required.

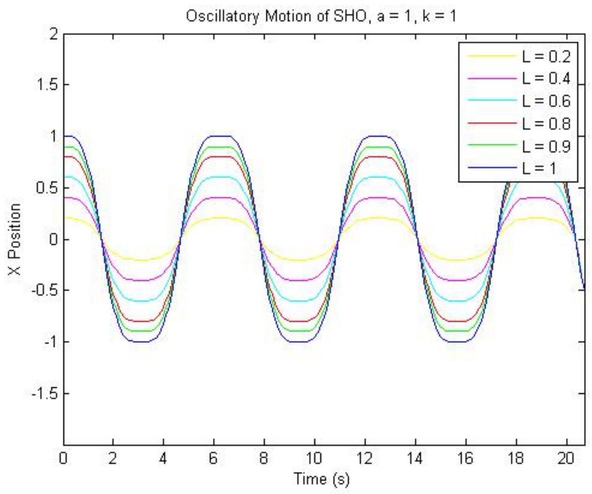

Figure 4: A harmonic oscillator with different values for the amplitude, determined by the variable l. Here, $\alpha = 1$ and l is varied from $1 - 0.2$. As can be seen, the different amplitudes do not result in differing periods for the oscillator. This is as expected.

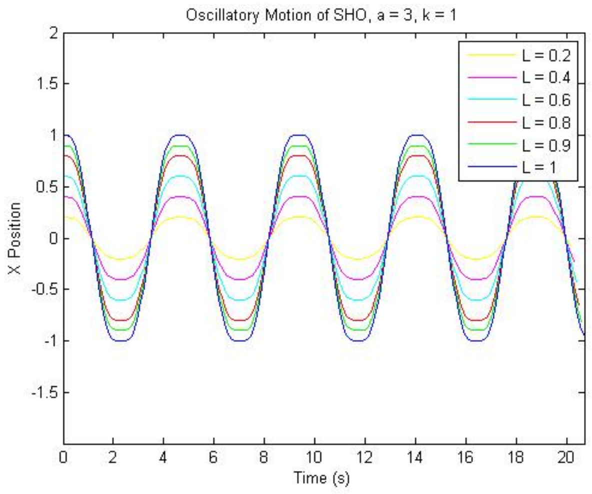

Figure 5: An anharmonic oscillator with $\alpha = 3$. The expected result is that the oscillator's period now depends on the amplitude, with a longer period for smaller amplitudes. This was not the case, and must have something to do with how the simulation itself was programmed.

6 Conclusions

Overall, although the Runge-Kutta method has a longer computational time, it is more accurate in representing the motion of the SHO and conserves energy over any time interval. For a faster computation, the Euler-Cromer method can be used, but energy is only conserved over specific time intervals. For accuracy that can be scaled easily, the Runge-Kutta method is preferred, since it can be used to any order desired in order to produce a representative model. For simulations used in this paper, a Runge-Kutta method of order 2 was utilized.

References

[1] Giordano, Nicholas J., and Hisao Nakanishi. ”3. Oscillatory Motion and Chaos” Computational Physics. Upper Saddle River, NJ: Pearson/Prentice Hall, 2006. N. pag. Print.

7 MATLAB code

```
%% Problem 3.1: Investigating the Euler-Cromer Method for SHO
clear all

%initial conditions
m = 1; %mass of pendulum
l = 0.5; %length of the wire
g = 9.8; %force of gravity
E_e(1) = 0;
E_r(1) = 0;
K_e(1) = 0;
P_e(1) = 0;
w(1) = 0; %initial angular velocity
dt = 0.1;
ft = 100;
t = [0:dt:ft+dt]; %initial element in time array
theta(1) = pi/2; %initial angle of pendulum
theta_r(1) = pi/2;

for i = 1:length(t)
    %calculating the motion of the pendulum itself
    w(i+1) = w(i)-(g/l)*theta(i)*dt;
    theta(i+1) = theta(i) + w(i)*dt;
    t(i+1) = t(i) + dt;

    %calculating energy for over the entire loop
    K_e(i+1) = (1/2)*m*(l^2)*(w(i)^2)*dt;
    P_e(i+1) = m*g*l*(theta(i)^2)*dt;
    E_e(i+1) = K_e(i+1) + P_e(i+1);
    E_e(i+2) = E_e(i+1);
end

%plot(t, theta, 'r');
hold on;

for i = 1:length(t)
    %calculating the motion of the pendulum itself
    theta_p = theta_r(i) + (1/2)*w(i)*dt;
    w_p = w(i) - (1/2)*theta_r(i)*dt;
    theta_r(i+1) = theta_r(i) + w_p*dt;
    w(i+1) = w(i) - theta_p*dt;
    t(i+1) = t(i) + dt;

    %calculating energy for over the entire loop
    %K(i+1) = (1/2)*m*(l^2)*(w(i)^2)*dt;
    %P(i+1) = m*g*l*(theta(i)^2)*dt;
    E_r(i+1) = (1/2)*(w(i)^2 + theta(i)^2);
```

```
end

%plot(t, theta_r, 'b');
hold on;
plot(t, E_r,'g');
hold on
plot(t, E_e, 'y');
axis([0, 10, -10, 10]);
title('Oscillatory Motion of SHO, Runge-Kutta')
xlabel('Time (s)')
ylabel('Theta (displacement in radians)')
legend('Euler-Cromer','Runge-Kutta','Energy of Runge-Kutta','Energy of Euler-Cromer')

%%Problem 3.2
%% Problem 3.1: Investigating the Euler-Cromer Method for SHO
clear all

%initial conditions
m = 1; %mass of pendulum
l = 0.5; %length of the wire
g = 9.8; %force of gravity
E(1) = 0;
K(1) = 0;
P(1) = 0;
w(1) = 0; %initial angular velocity
dt = 0.1;
ft = 10;
t = [0:dt:ft+dt]; %initial element in time array
theta(1) = pi/2; %initial angle of pendulum

for i = 1:length(t)
    %calculating the motion of the pendulum itself
    w(i+1) = w(i)-(g/l)*theta(i)*dt;
    theta(i+1) = theta(i) + w(i+1)*dt;
    t(i+1) = t(i) + dt;

    %calculating energy for over the entire loop
    K(i+1) = (1/2)*m*(l^2)*(w(i)^2)*dt;
    P(i+1) = m*g*l*(theta(i)^2)*dt;
    E(i+1) = K(i+1) + P(i+1);

end

plot(t, theta);
hold on;
plot(t, E,'r');
title('Oscillatory Motion of Simple Harmonic Oscillator')
xlabel('Time (s)')
ylabel('Theta (displacement in radians)')
plot(t, P, 'g');
plot(t, K, 'y');
legend('Pendulum','Total Energy','Potential Energy','Kinetic Energy');

%% Problem 3.4
clear all

%initial conditions
a = 3;
m = 1; %mass of pendulum
l = [0.2,0.4,0.6,0.8,0.9,1]; %length of the wire
color = ['y','m','c','r','g','b']; %color spectrum
g = 9.8; %force of gravity
k = 1;
E(1) = 0;
K(1) = 0;
P(1) = 0;
w(1) = 0; %initial angular velocity
dt = 0.1;
ft = 20;
t = [0:dt:ft+dt]; %initial element in time array
theta(1) = pi/2; %initial angle of pendulum
for j = 1:length(l)

x(1) = l(j)*sin(theta(1));

for i = 1:length(t)
    %calculating the motion of the pendulum itself
    w(i+1) = w(i)-k*(theta(i)^a)*dt;
    theta(i+1) = theta(i) + w(i+1)*dt;
    t(i+1) = t(i) + dt;
    x(i+1) = l(j)*sin(theta(i+1));

    %calculating energy for over the entire loop
    %K(i+1) = (1/2)*m*(l^2)*(w(i)^2)*dt;
    %P(i+1) = m*g*l*(theta(i)^2)*dt;
    %E(i+1) = K(i+1) + P(i+1);
end

c = color(j);
plot(t,x,c);
hold on
end

min_ax = min(x)-1;
max_ax = max(x)+1;
axis([0, t(end), min_ax, max_ax]);
legend('L = 0.2','L = 0.4','L = 0.6','L = 0.8','L = 0.9','L = 1');
title('Oscillatory Motion of SHO, a = 1, k = 1')
xlabel('Time (s)')
ylabel('X Position')
```