SHARIF UNIVERSITY OF TECHNOLOGY
PHYSICS DEPARTMENT

Problem Set 9

Student name: *Ali Fayyaz - 98100967*

Course: *Computational Physcis - (Spring 2023)*
Due date: *May 12, 2023*

Exercise 9.1

First Order Differential Equation
Find the charge on the capacitor of an RC circuit as a function of time, first analytically and then numerically using Euler's algorithm. For different values of h (Euler time steps), find the error of the numerical calculation. Take R and C as 1 for simplicity.

Answer. The differential equation of interest is as follows:

$$V - R\frac{dq}{dt} - \frac{q}{C} = 0$$

where V is the electric potential accross the capacitor, R is the resistance, C the capacitance, and q the charge on the capacitor. The analytical solution is as follows:

$$\frac{dq}{dt} = \frac{CV - q}{RC}$$

$$\int_0^q \frac{dq}{CV - q} = \frac{1}{RC}\int_0^t dt$$

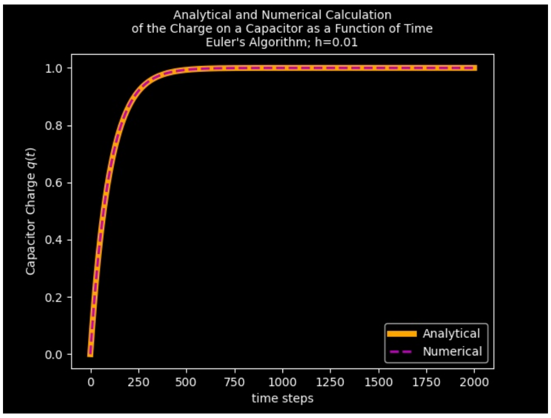
Let $u = CV - q$:

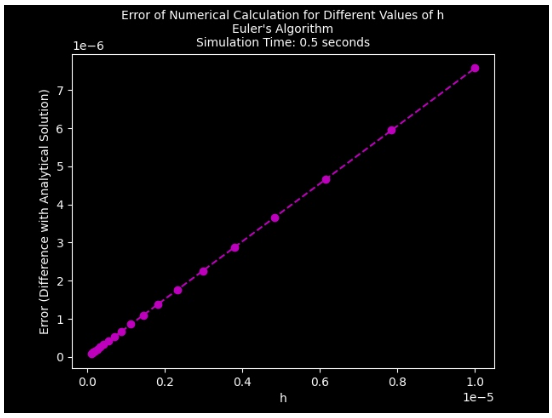
$$-\int_{q=0}^{q=q} \frac{du}{u} = \frac{1}{RC}\int_0^t dt$$

$$ln\left(\frac{CV - q}{CV}\right) = -\frac{1}{RC}t$$

$$q(t) = CV(1 - e^{\frac{-t}{RC}})$$

R, C and V are taken to be reduced unit constants for the numerical calculations. Euler's algorithm, $\frac{dq}{dt} = \dot{q}$ and the analytical answer $q(t)$ were declared as simple functions. With $q(0) = 0$ (initial value of the charge on the capacitor) the simulation took place for 2,000 time steps. Then for 20 different values for the time step, the difference between the analytical and numerical solutions were graphed.





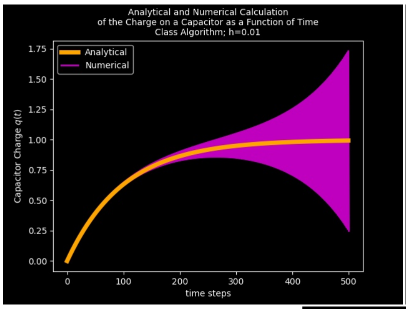
Exercise 9.2

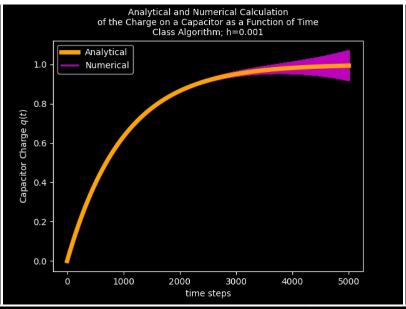
Algorithm Stability

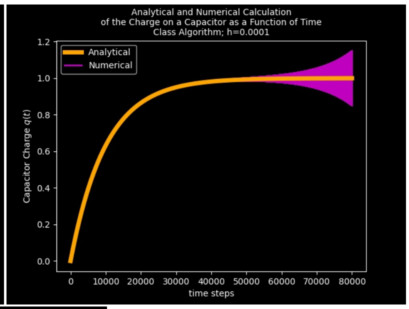
Answer the previous exercise with the algorithm mentioned in the class:

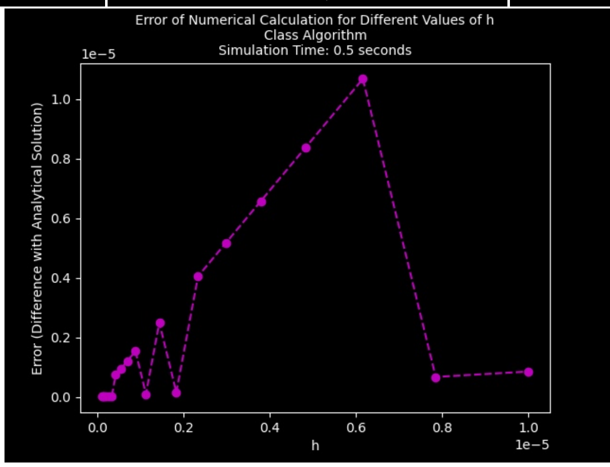
$$y_{n+1} = y_{n-1} + 2f_n h$$

Answer. The algorithm was implemented in a simple function and the results were graphed for different values of h. As expected, the accuracy increses for smaller time steps. Nevertheless, the algorithm is not stable and deviates from the correct, analytical answer as time goes on, the error being unpredictable as the graphs show.









Exercise 9.3

Second Order Differential Equation and Algortihm Comparison

Solve the equation of motion for the simple harmonic oscillator with these algorithms:

1. Euler
2. Euler-Cromer
3. Leapfrog
4. Verlet
5. Velocity Verlet
6. Beeman

Graph position as a function of time, and also the phase space diagram. Discuss the conservation of energy in each algorithm. Do this for different values of h.

Bonus Question: Graph the accuracy of the algorithms as a function of time, for different values of h. Accuracy, here, means the distance of the numerical value with the analytical value in the phase space. Note that the time of simulation must be much larger than the period of oscillation.

Answer. The second order differential equation to be solved is:

$$\frac{d^2x}{dt^2} = \ddot{x} = -\omega^2 x$$

The analytical solution is:

$$\begin{aligned} x(t) &= x_0 \cos(\omega t) \\ \implies v(t) &= \dot{x}(t) = -x_0\omega \sin(\omega t) \\ \implies a(t) &= \ddot{x}(t) = -x_0\omega^2 \cos(\omega t) = -\omega^2 x(t) \end{aligned}$$

The algorithms are as follows (h is the time step, x the position, v the velocity and a the acceleration):

1. **Euler** (Basically two Taylor expansions):

$$\begin{aligned} x_{i+1} &= x_i + v_i h \\ v_{i+1} &= v_i + a_i h \end{aligned}$$

2. **Euler-Cromer**

$$\begin{aligned} v_{i+1} &= v_i + a_i h \\ x_{i+1} &= x_i + v_{i+1} h \end{aligned}$$

3. **Leapfrog**

$$x_{i+1} = x_i + v_i h + \tfrac{1}{2} a_i h^2$$
$$v_{i+1} = v_i + \tfrac{1}{2}(a_i + a_{i+1})h$$

4. **Verlet**

$$x_{i+1} = 2x_i - x_{i-1} + a_i h^2$$
$$v_i = \frac{x_{i+1} - x_{i-1}}{2h}$$

5. **Velocity Verlet**

$$x_{i+1} = x_i + v_i h + \tfrac{1}{2} a_i h^2$$
$$v_{i+1} = v_i + \frac{a_i + a_{i+1}}{2} h$$

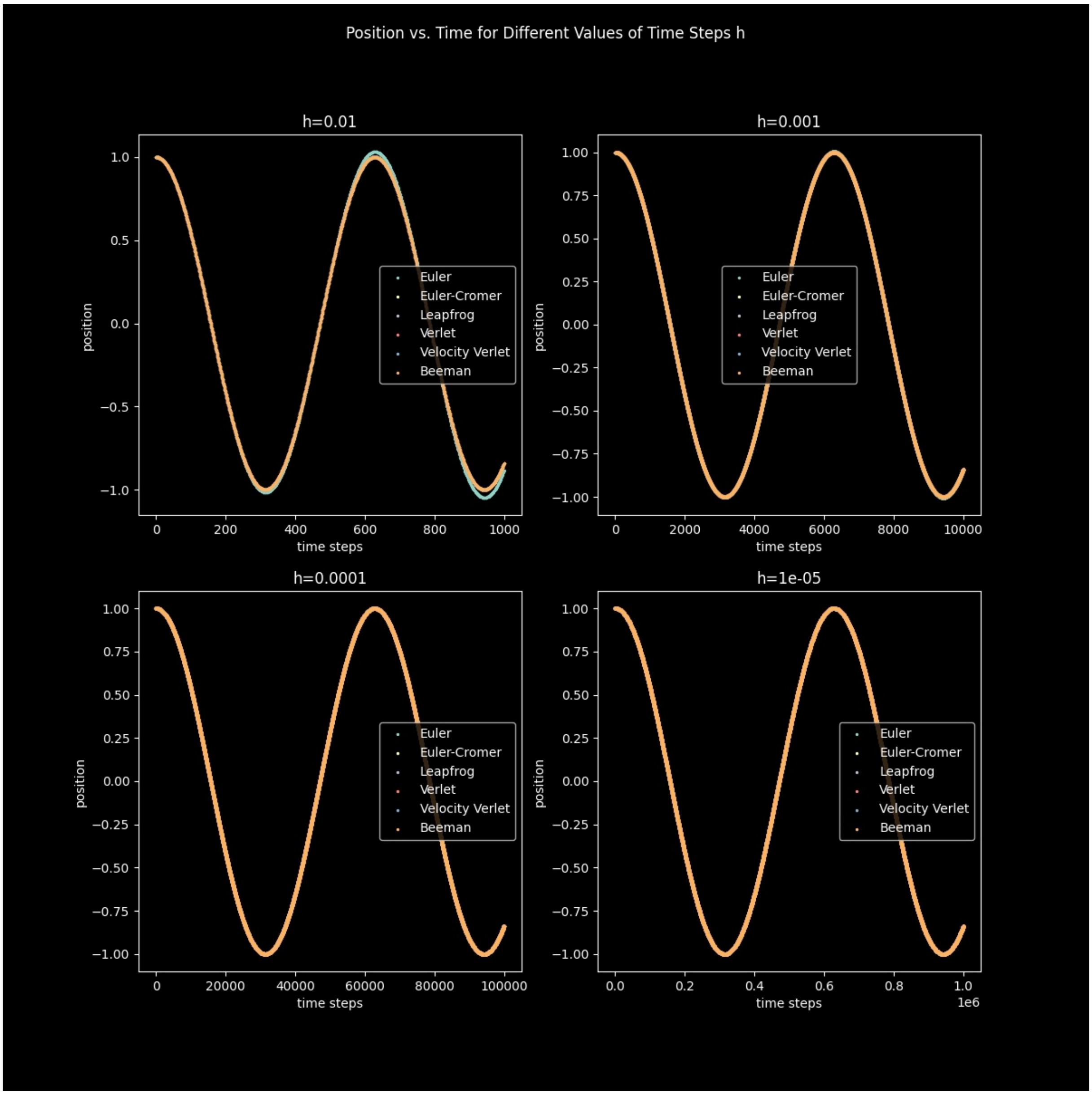
6. **Beeman**

$$x_{i+1} = x_i + v_i h + \tfrac{1}{6}(4a_i - a_{i-1})h^2$$
$$v_{i+1} = v_i + \tfrac{1}{6}(2a_{i+1} + 5a_i - a_{i-1})h$$

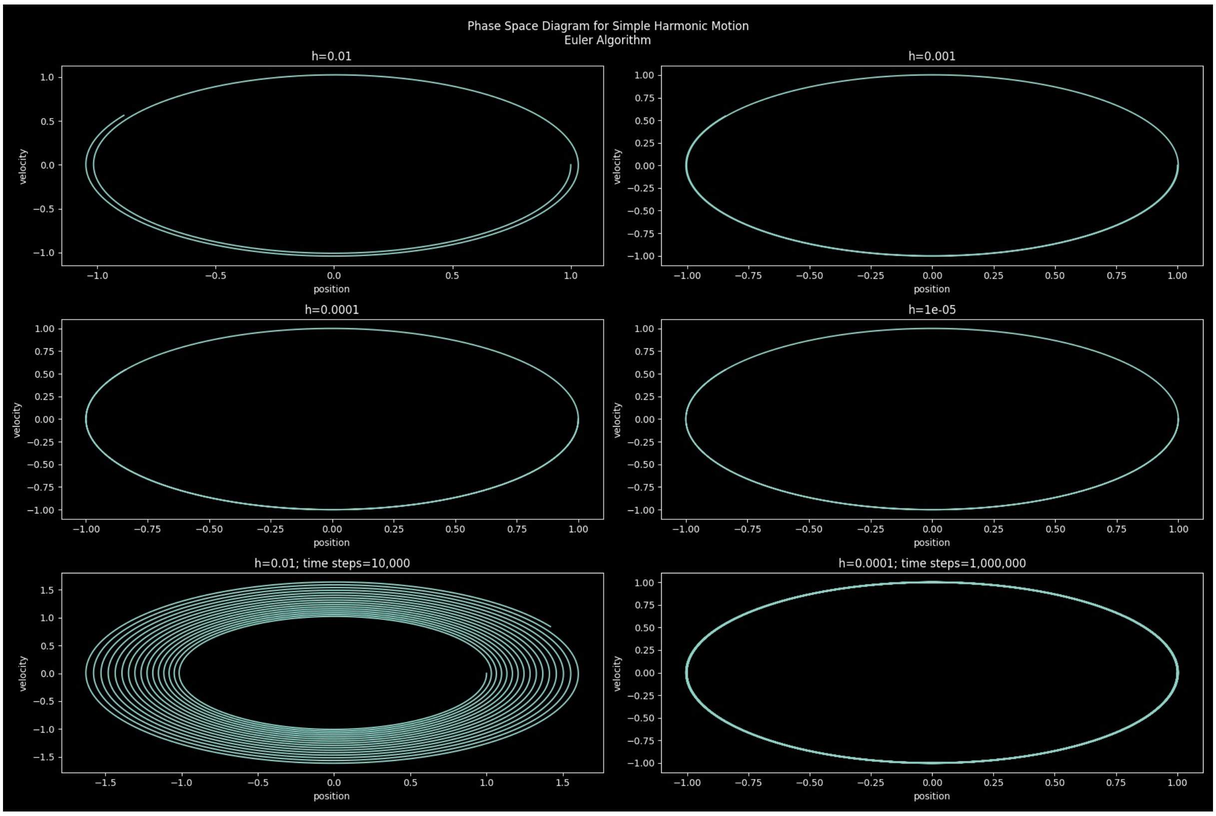
Apart from the function *acc()* for the acceleration based on the analytical solution, six simple functions were defined for the mentioned algorithms, respectively. The input parameters were all the same for these six functions:

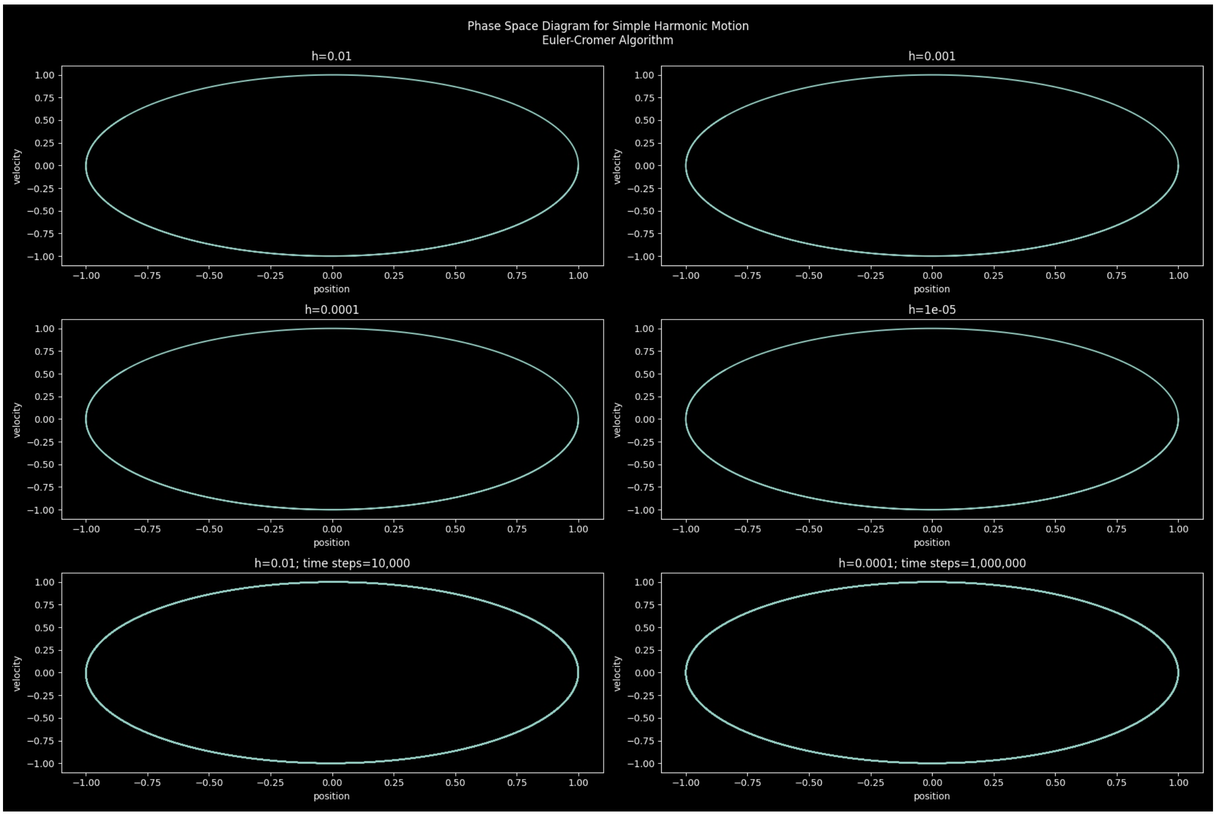
- *x0*: initial position (the value 1 is used in all simulations)
- *acc*: acceleration (second time derivative of x, which for this particular problem is $-\omega^2 x$); the function *acc()* was provided for this paramter, with the default value of $\omega = 1$ in all simulations.
- *h*: time step
- *time*: the length of time of the simulation
- *v0*: initial velocity (the default value 0 is used in all simulations)

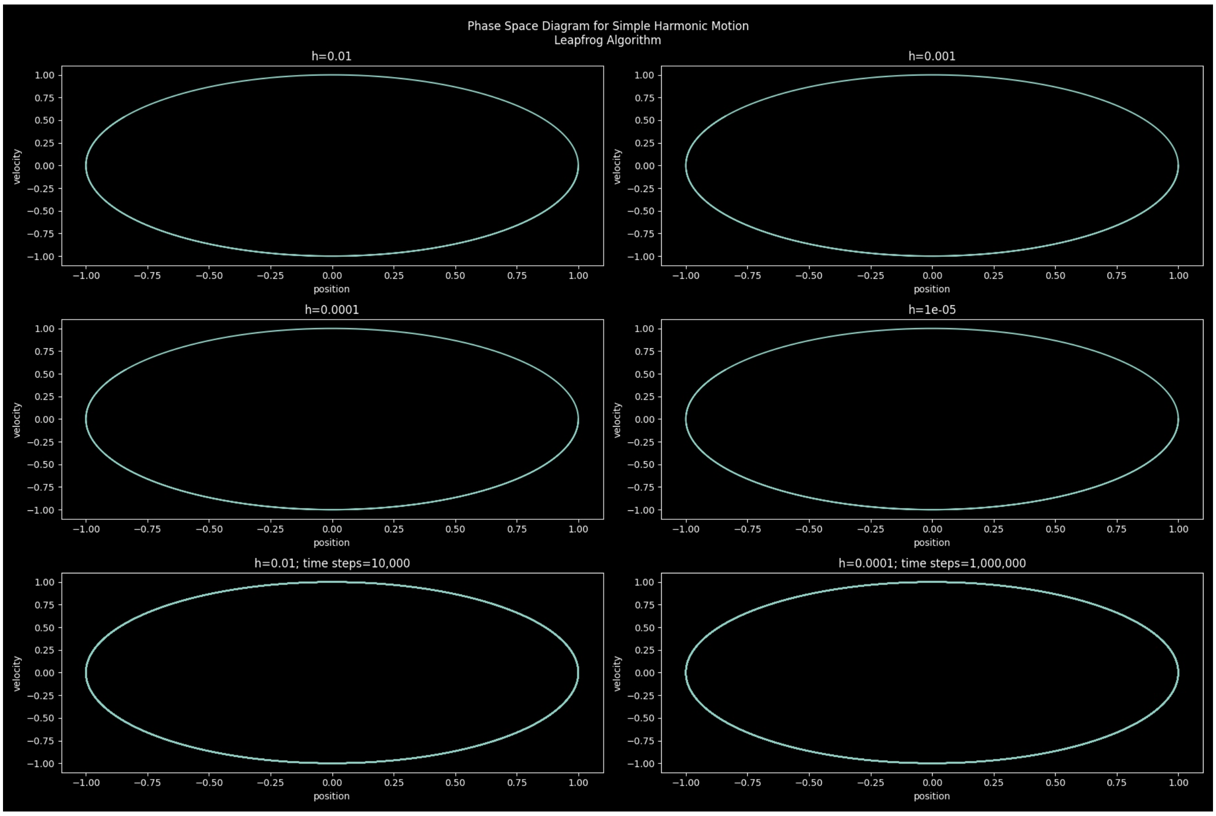
For different values of h, the graph of position vs. time was obtained for the algorithms and graphed together for comparison. As appears, Euler's algorithm quickly becomes inaccurate after a few time steps. Smaller values of h tend to increase accuracy.

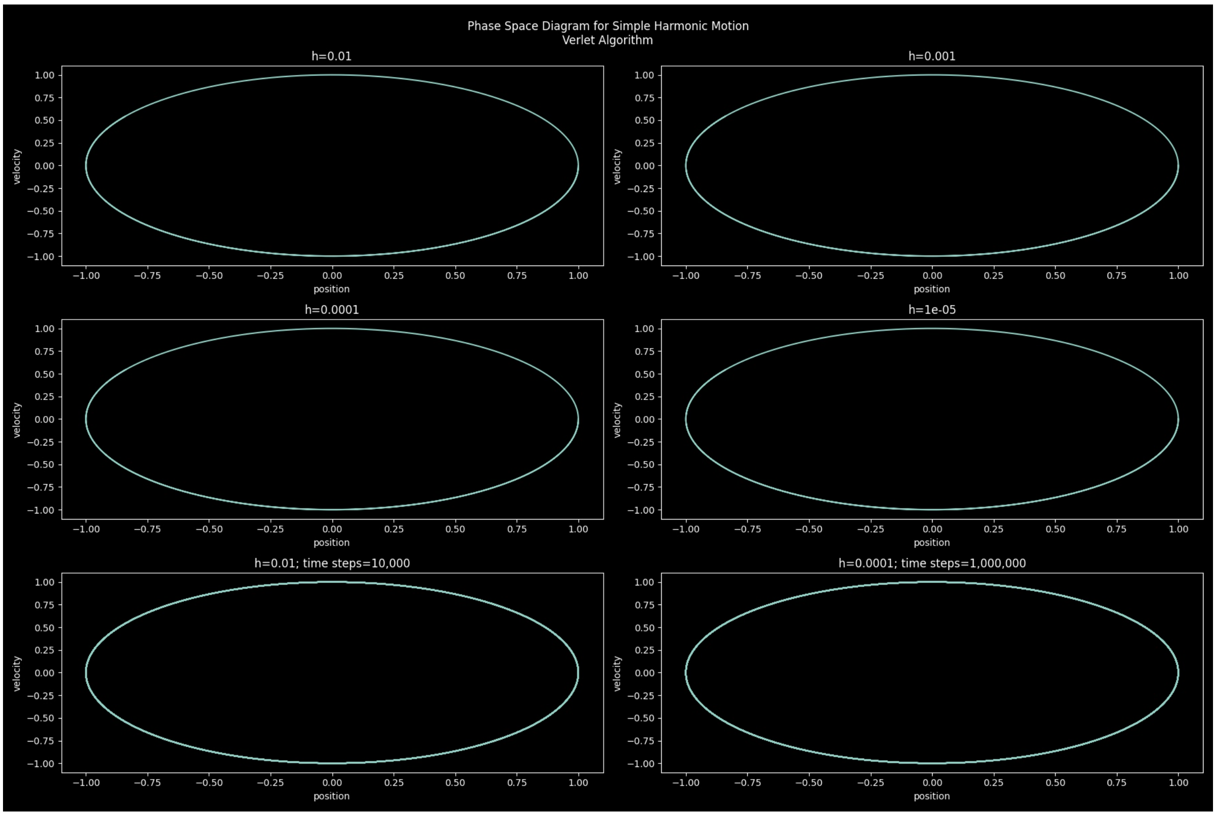


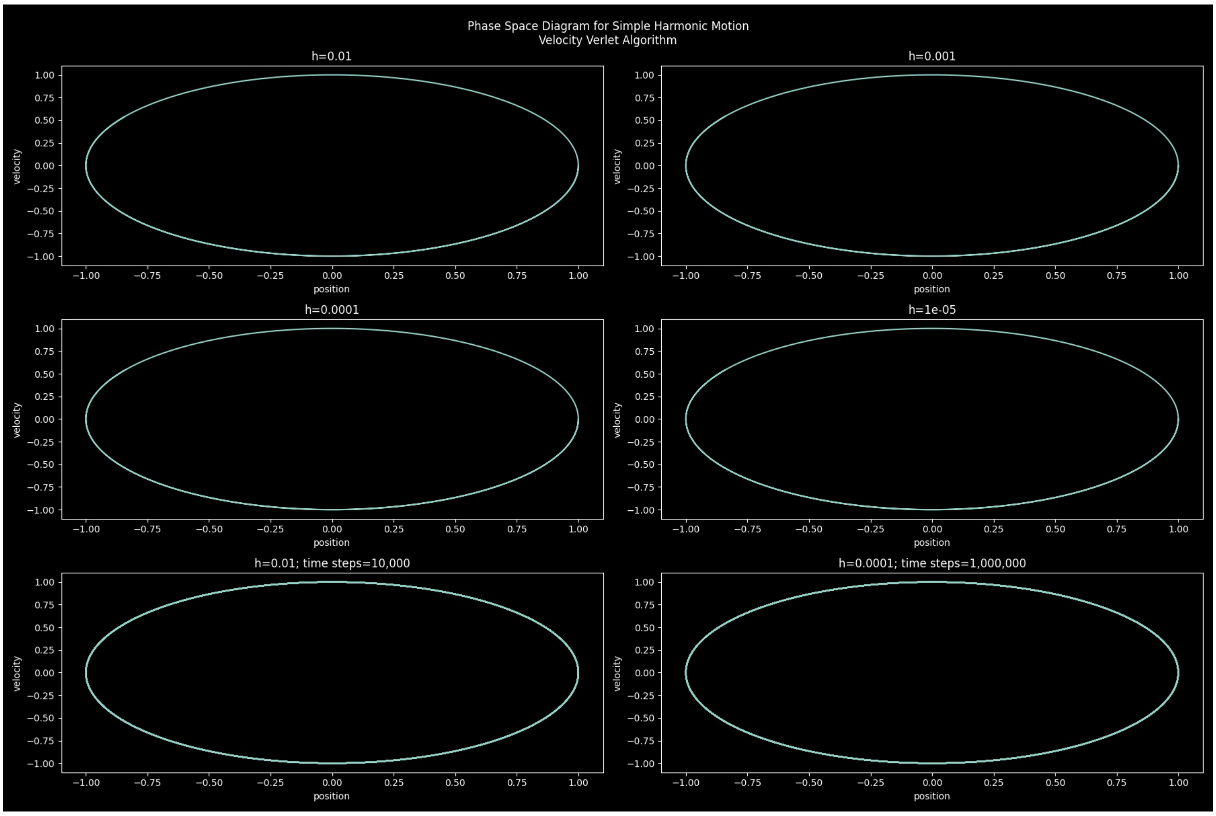
The phase space diagrams were graphed for each algorithm seperately. As suggestive from the figures, energy is conserved in all algorithms but Euler's, in which case it tends to increase. This non-conservation is most evident in $h = 0.01$ as the graph spirals out since the beginning of the simulation. If $h \leq 0.001$, for all the other algorithms, the minimum and maximum of velocity is always -1 and $+1$, respectively.

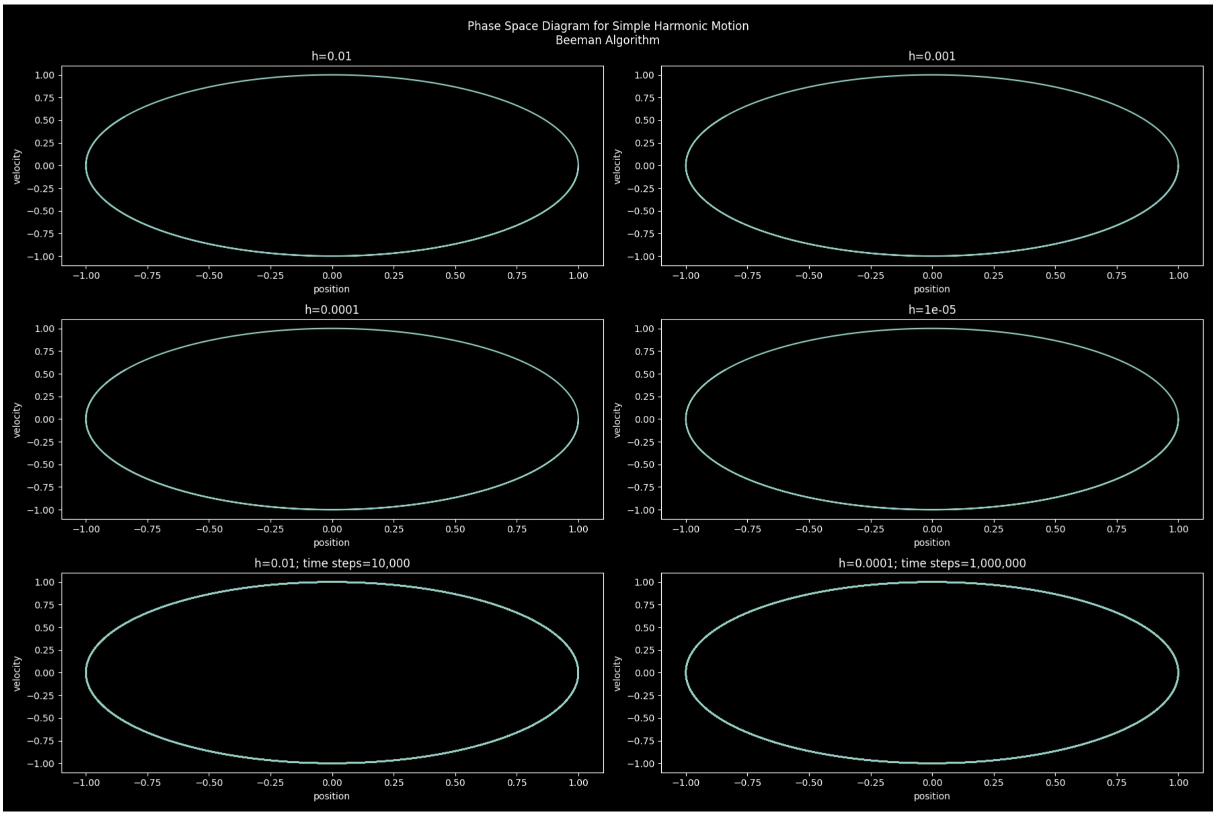

Phase Space Diagram for Simple Harmonic Motion
Euler Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position


Phase Space Diagram for Simple Harmonic Motion
Euler-Cromer Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position


Phase Space Diagram for Simple Harmonic Motion
Leapfrog Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position


Phase Space Diagram for Simple Harmonic Motion
Verlet Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position


Phase Space Diagram for Simple Harmonic Motion
Velocity Verlet Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position


Phase Space Diagram for Simple Harmonic Motion
Beeman Algorithm
h=0.01
h=0.001
h=0.0001
h=1e-05
h=0.01; time steps=10,000
h=0.0001; time steps=1,000,000
velocity
position

Checking Conservation of Energy with Min and Max of Data; $h = 0.01$

	xs_euler_cromer	vs_euler_cromer	xs_leapfrog	vs_leapfrog	xs_verlet	vs_verlet	xs_velocity_verlet	vs_velocity_verlet	xs_beeman	vs_beeman	xs_euler	vs_euler
min	-1.000012	-1.000012	-0.999999	-0.999987	-1.000012	-0.999999	-0.999999	-0.999987	-0.999999	-1.000004	-1.048264	-1.040063
max	1.000011	1.000010	1.000000	0.999985	1.000011	0.999997	1.000000	0.999985	1.000000	1.000001	1.031926	1.023851

Checking Conservation of Energy with Min and Max of Data; $h = 0.001$

	xs_euler_cromer	vs_euler_cromer	xs_leapfrog	vs_leapfrog	xs_verlet	vs_verlet	xs_velocity_verlet	vs_velocity_verlet	xs_beeman	vs_beeman	xs_euler	vs_euler
min	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.004724	-1.003935
max	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.003147	1.002359

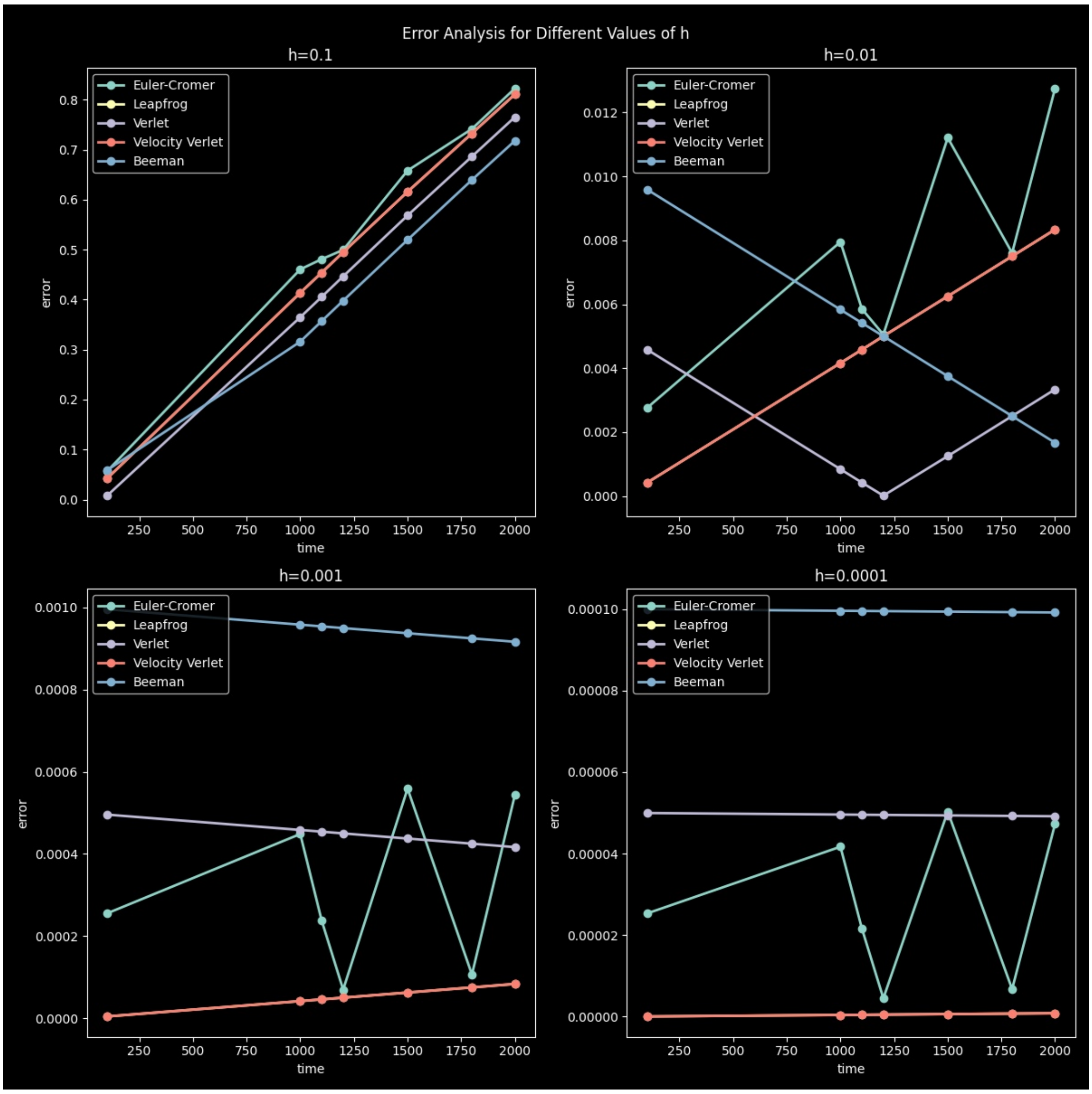
Checking Conservation of Energy with Min and Max of Data; $h = 0.0001$

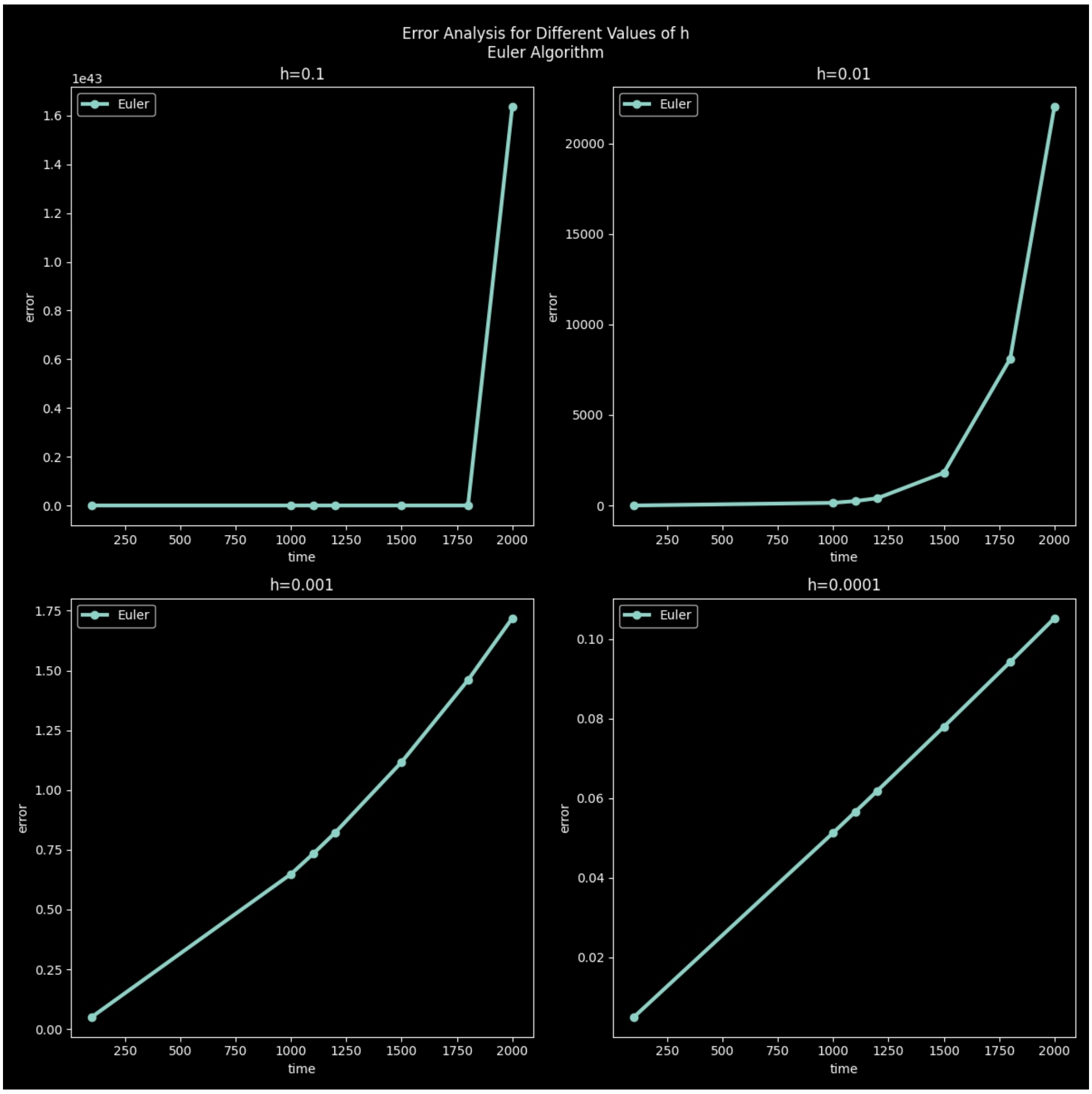
	xs_euler_cromer	vs_euler_cromer	xs_leapfrog	vs_leapfrog	xs_verlet	vs_verlet	xs_velocity_verlet	vs_velocity_verlet	xs_beeman	vs_beeman	xs_euler	vs_euler
min	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000471	-1.000393
max	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.000314	1.000236

Checking Conservation of Energy with Min and Max of Data; $h = 0.00001$

	xs_euler_cromer	vs_euler_cromer	xs_leapfrog	vs_leapfrog	xs_verlet	vs_verlet	xs_velocity_verlet	vs_velocity_verlet	xs_beeman	vs_beeman	xs_euler	vs_euler
min	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.000047	-1.000039
max	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.000031	1.000024

For the bonus question, the error was defined as the Pythagorean distance of the numerical value and the analytical value in the phase space. Due to extermely high errors for the Euler algorithm, it was graphed seperately. It is evident from the figures that Velocity Verlet is the most stable algorithm, and has the smallest deviations from the analytical solution in longer runs, for small time steps.





Sources:
1. Physics Libretexts: RC Circuits
2. Wikipedia: Numerical methods for ordinary differential equations