Chaos

Mahyar Albalan

403201223

Capacitor Charging

The differential equation governing this problem is as follows:

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

To solve this equation, I will use the desired algorithm:

$$q_{(n+1)} = q_{(n-1)} + 2\delta t(V/R - q_{(n)}/(RC))$$

For simplicity, I used the following initial conditions:

$$Q_0 = 0 \quad ; \quad R = C = 1 \quad ; \quad V = 5$$

With these initial conditions, I used $\Delta t = 1 \times 10^{-6}$ and total simulation time $t = 12$. The theoretical expectation is:

$$Q = CV(1 - exp(-t/(RC))$$

the result of this simulation is as follows:

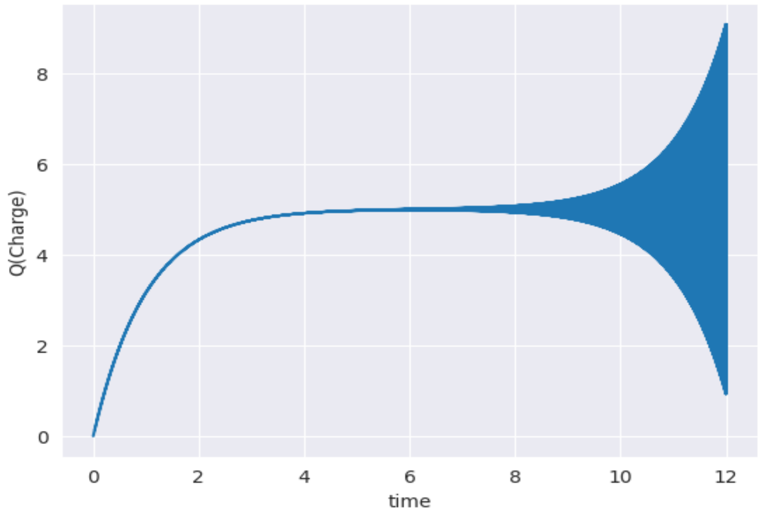

Figure 1: Capacitor's charge vs time

As you can see, at first it converges to the theoretical expectation of 5 (for large t, we expect Q to approach CV), but after getting close to this value, it begins to fluctuate and diverge.

Harmonic Oscillator

the equation governing this problem is as follows:

$$\ddot{x} + \omega^2 x = 0$$

the theoretical solution is:

$$x = Asin(\omega t) + Bcos(\omega t)$$

Now i will solve this question using Euler's method, in this method we convert this second order DE into two first order DEs:

$$\dot{x} = v$$

$$\dot{v} = -\omega^2 x$$

so we can get our solution from:

$$x_{(n+1)} = x_n + \Delta t v_n$$

$$v_{(n+1)} = v_n - \Delta t \omega^2 x$$

i put $\omega = 1$, $x_0 = 1$, $v_0 = 0$,$h = \Delta t = [1e-3, 1e-2, 1e-1, 0.5]$ and total simulation, $t = 100$ for simplicity,For calculating the error, I computed the theoretical value of X at each time step and calculated its MSE from the simulation results. The results are as follows:

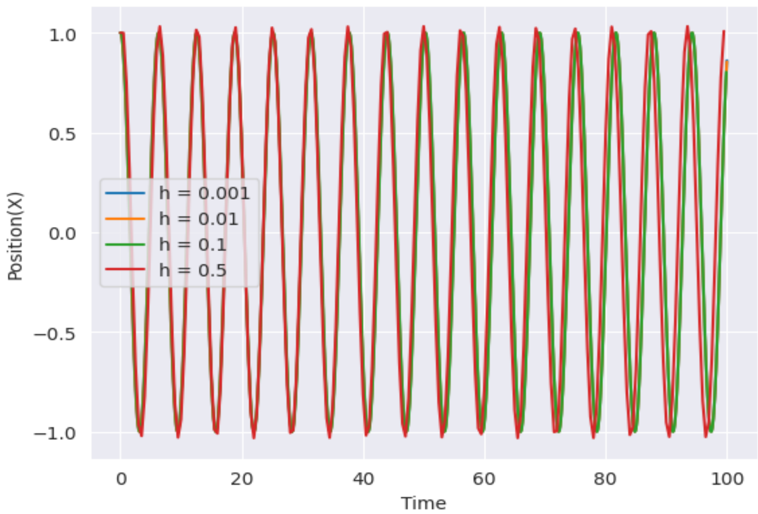

Figure 2: X vs Time with Euler method

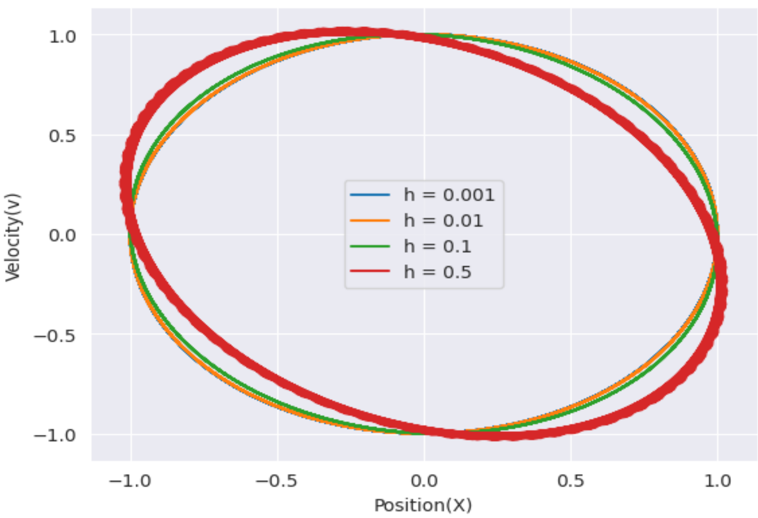

Figure 3: Velocity vs Time with Euler method

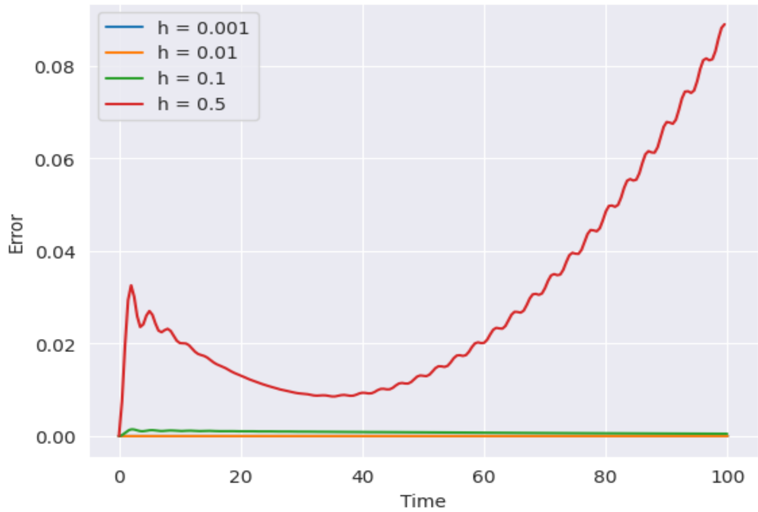

Figure 4: Euler's Method Error

error for $h = 0.5$ is rising but for other hs, errors and simulation results are much better.

Now i will use Euler–Cromer Method for solving this problem, the process is the same as before just a small reordering is needed:

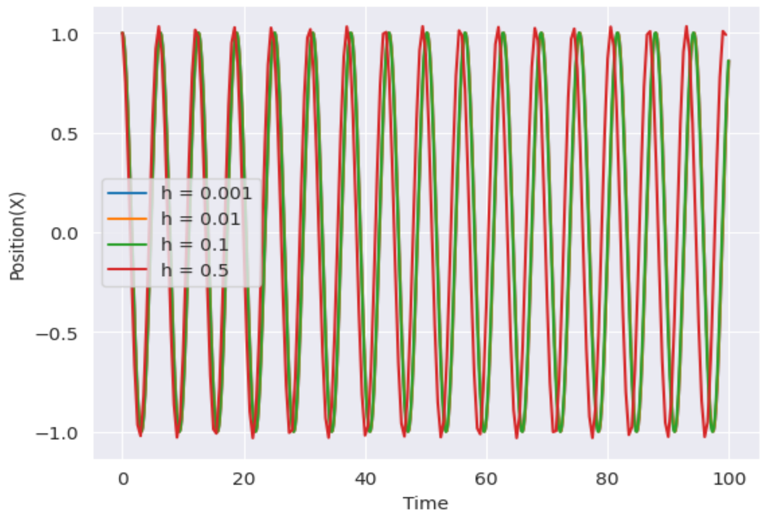

Figure 5: X vs Time with Euler–Cromer method

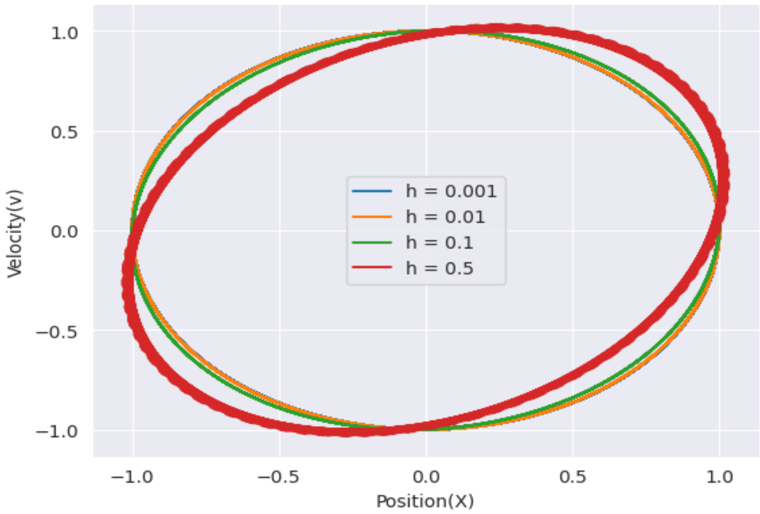

Figure 6: Velocity vs Time with Euler–Cromer method

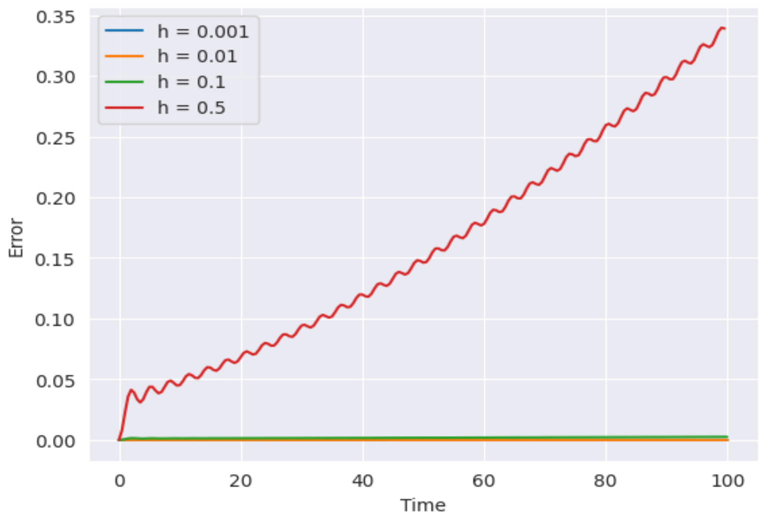

Figure 7: Euler–Cromer Method Error

As you can see, this method's error is lower than the previous one, which makes sense since the previous method had a slight delay compared to this one (a minor reordering fixes this issue).

Now i will use Frog leap method,the results for this method are as follows:(i dont mention this algorithm's updating role)

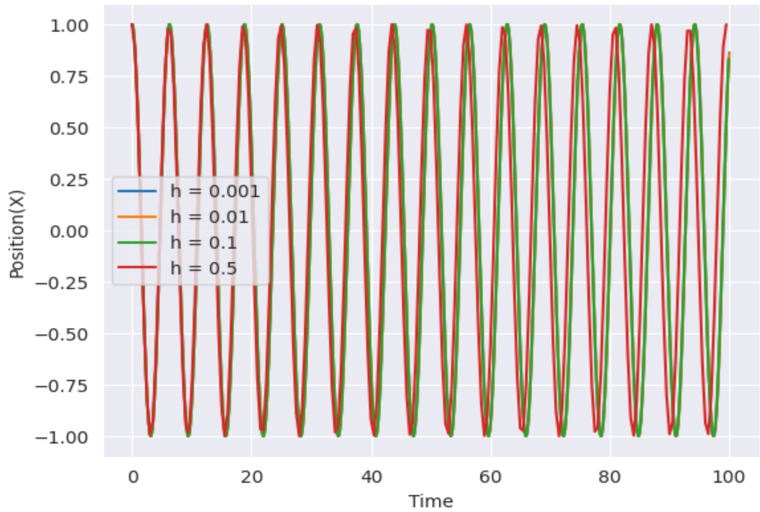

Figure 8: X vs Time with Frog leap method

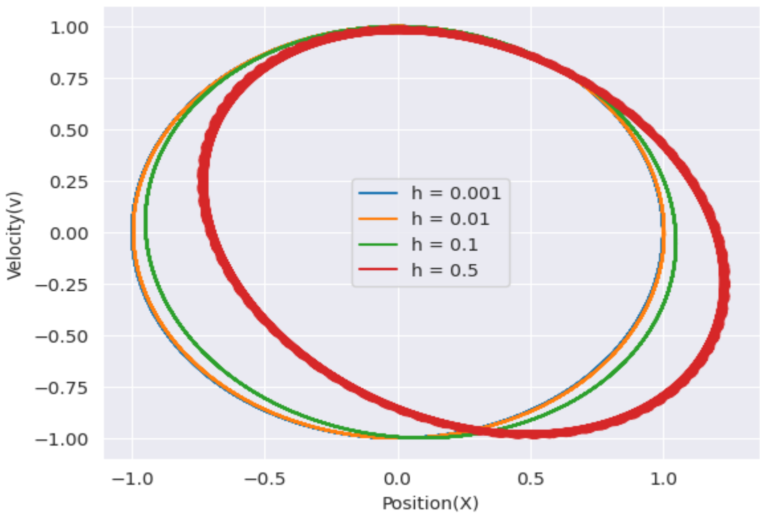

Figure 9: Velocity vs Time with Frog leap method

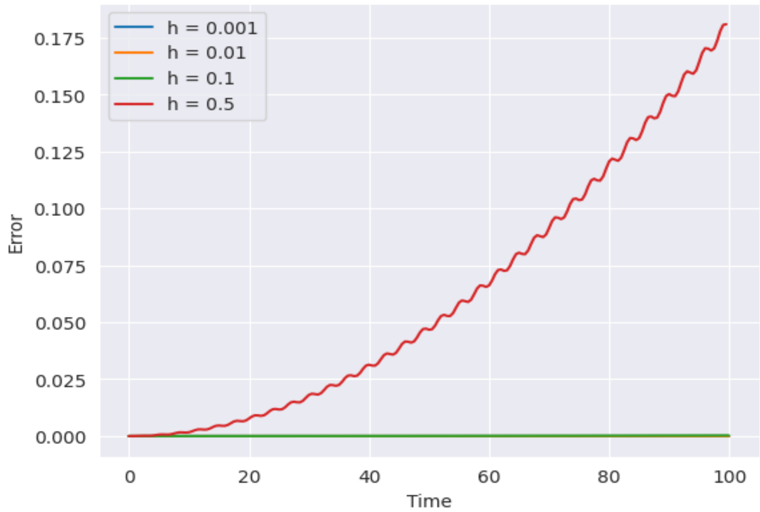

Figure 10: Frog leap Method Error

This method's error is less than that of previous methods. For smaller h, its error is approximately an order of magnitude lower than previous methods.

now i will use Verlet algorithm, the results of this algorithm are as follows:

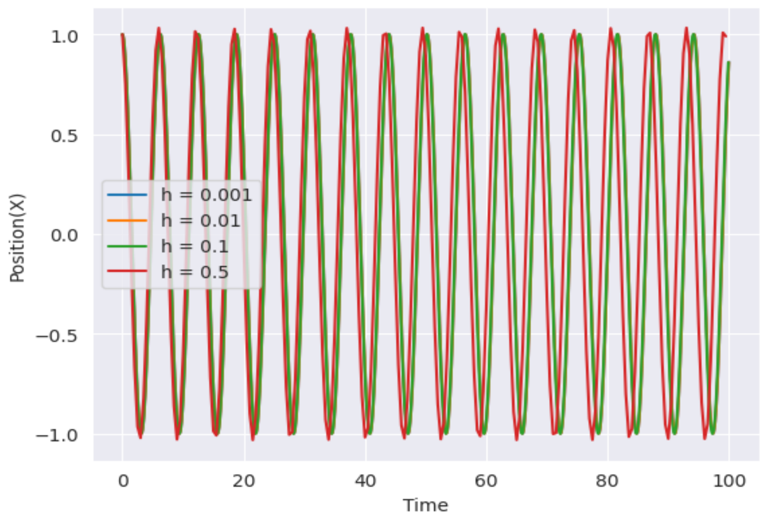

Figure 11: X vs Time with Verlet method

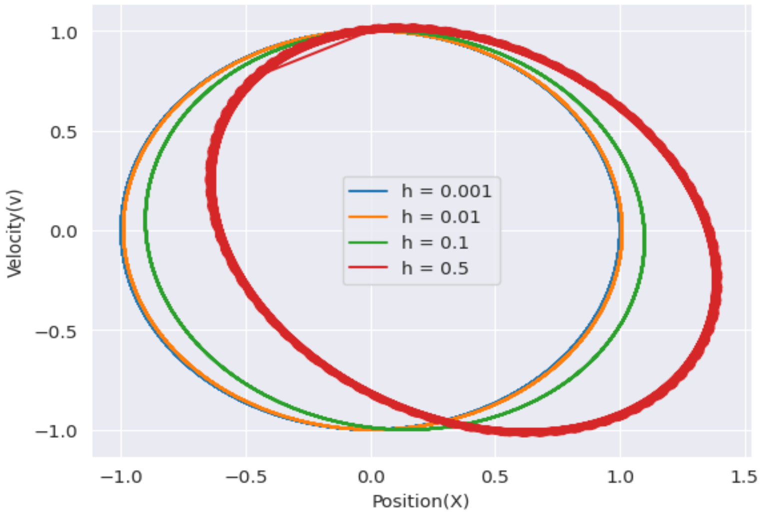

Figure 12: Velocity vs Time with Verlet method

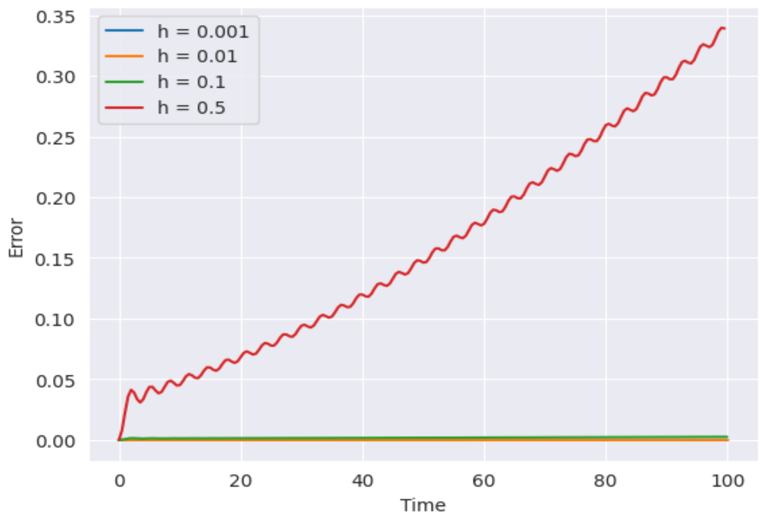

Figure 13: Verlet Method Error

now i will use velocity verlet method:(i dont mention their updating rule here again!)

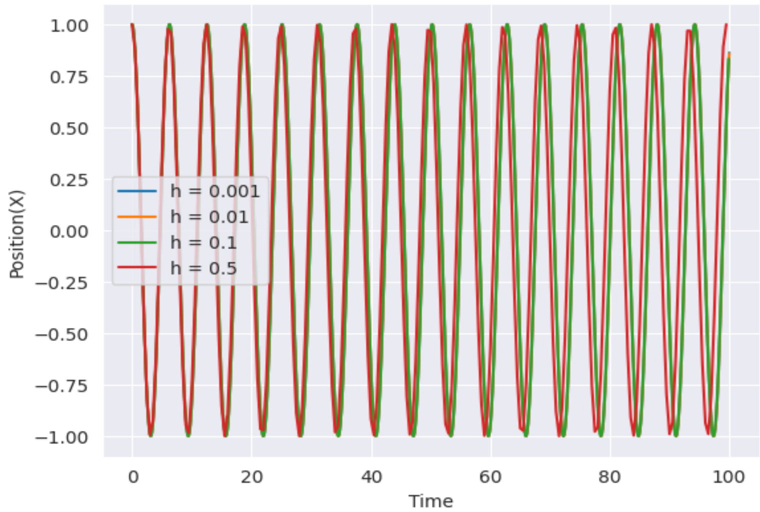

Figure 14: X vs Time with velocity verlet method

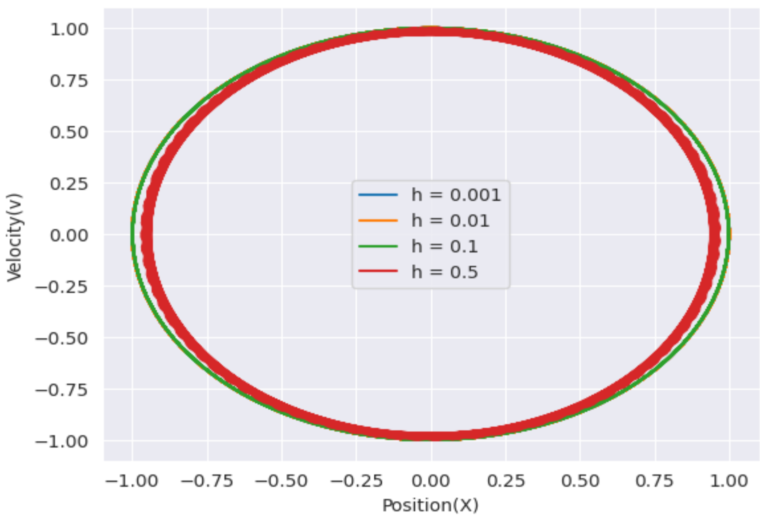

Figure 15: Velocity vs Time with velocity verlet method

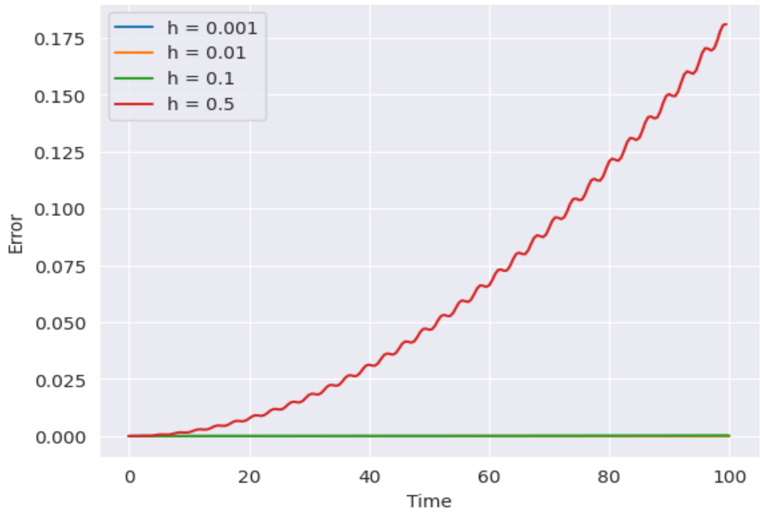

Figure 16: velocity verlet Method Error

at last Beeman methods results are as follows:

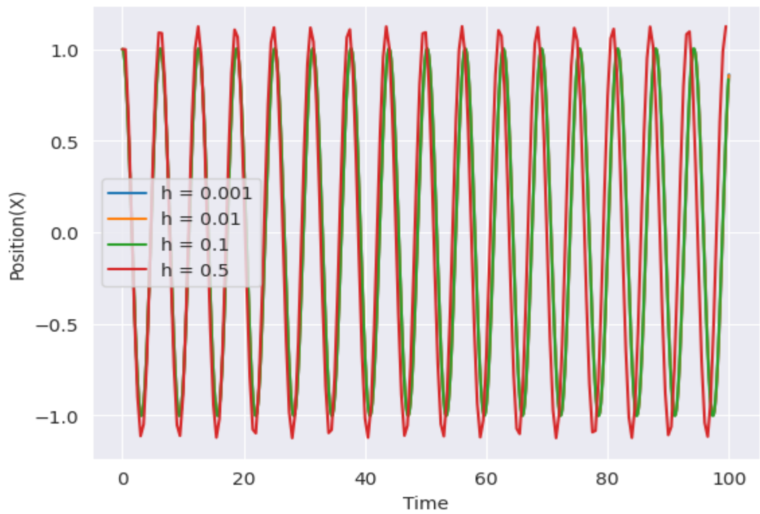

Figure 17: X vs Time with Beeman method

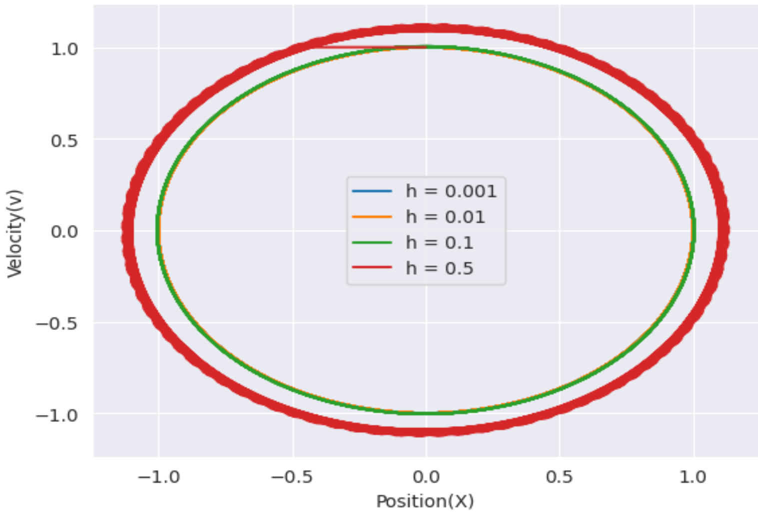

Figure 18: Velocity vs Time with Beeman method

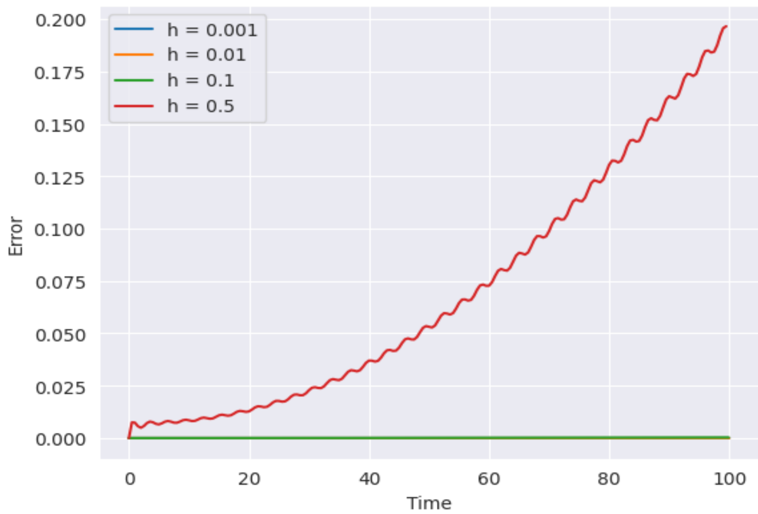

Figure 19: Beeman Method Error

As you can see, the errors of these last three algorithms are in the same range. The key difference lies in energy conservation:(for calculating energyconversion we can see which algorithms are more stable over long time period!or have less errors)

- **Euler's method** performs very poorly, with energy rapidly increasing or decreasing over time.

- **Verlet** and **velocity Verlet** show better conservation, maintaining stable fluctuations over longer periods.
- **Leapfrog** outperforms both Verlet methods.
- **Beeman algorithm** achieves the best results at the cost of higher computational expense (longer runtime).

Logistic Map

in this question we have to analyze the logistic map formula:

$$X_{n+1} = 4rX_n(1 - X_n)$$

For analyzing this problem, I used 1000 different r values. For each r, I generated the X series 50,000 times using the given formula. Then, I used only the last 100 values of each series to plot the bifurcation diagram and calculate the bifurcation constant. The bifurcation constant was computed using the following relation:

$$B = \lim_{n\to\infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n}$$

the Bifurcation Diagram is as follows:

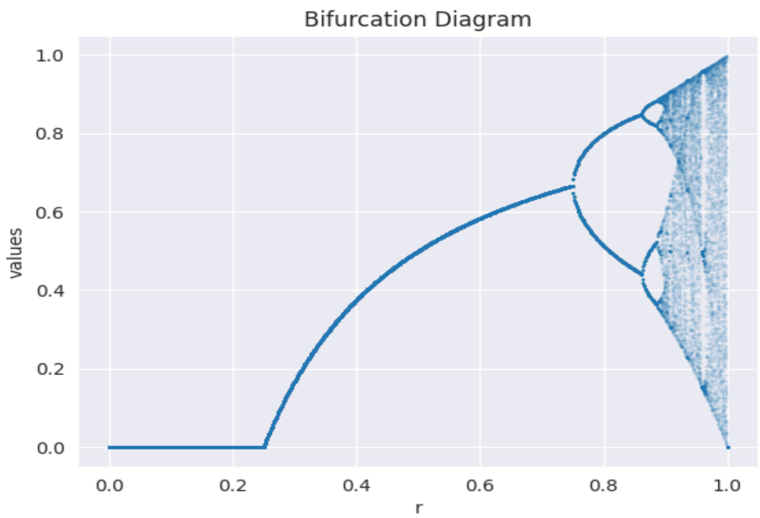

Figure 20: Euler–CromerMethod Error

From this simulation, I calculated the bifurcation constant to be 4.61, which is very close to the expected value of 4.666—only a 1.2% error.