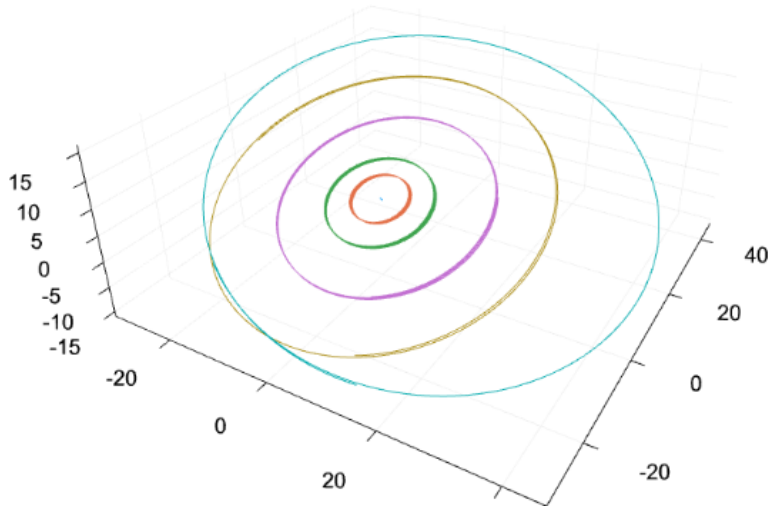
PAMS Project

The N-body problem and Hamiltonian Dynamics



Martin Alejandre Chaya

January 25, 2023

Contents

1 Introduction

This project is centered around solving the N body problem. Let there be N bodies free on the $\mathbb{R}^d$ space, each with a mass m_i, position q_i, and velocity v_i. This problem can be classified as a Hamiltonian Dynamic System, meaning it obeys the Hamiltonian Equations of motion, given by

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}, \qquad \frac{dp}{dt} = -\frac{\partial H}{\partial q}, \tag{1}$$

where q denotes the positions, p the momenta, and H is the Hamiltonian, i.e. the total energy of the system:

$$H(q,p) = K(p) + V(q), \tag{2}$$

with $V(q)$ the potential energy and $K(p)$ the kinetic energy. Hamiltonian Dynamics can be applied to a wide range of problems, but in our case, the formula for the Hamiltonian would be given by

$$K(p) = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 \qquad V(q) = -G\sum_{i=1}^{N}\sum_{j=i+1}^{N} \frac{m_i m_j}{r_{ij}} \tag{3}$$

where v_i denotes the velocity of the ith body, and r_{ij} the distance from body i to body j. Note that, for the N body problem, we write the equations in function of v instead of $p = mv$. According to the Hamiltonian Equations of motion (1), we can already write a system of ODEs that models our problem:

$$\frac{dq}{dt} = v \qquad \frac{dv_i}{dt} = -G\sum_{j=i, j\neq i}^{N} \frac{m_j}{r_{ij}2} u_{ij} \tag{4}$$

where u_{ij} denotes the unitary vector that points to body j from body i.

Now, we can not solve this ODE system analytically (it is still a open problem in mathematics), but along this project we will learn methods to approximate solutions.

2 Numerical Methods for solving ODEs

In this section we will see some methods for solving ODEs, and analyze their stability and error. For this purpose we will try to solve the following ODE:

$$\frac{dy}{dt} = ky. \tag{5}$$

We know that one solution to this ODE (5) is $y(t) = e^{kt}$. In particular this is the solution when $y(0) = 1$. This way we can compare the approximation given by our methods with the actual solution. We are interested in global solutions, so we will analyze the global error.

2.1 Euler Method

The Euler method is the simplest method to approximate the solution of a differential equation. It consists on setting the initial value z^0 and constructing z^{n+1} from z^n as

$$z^{n+1} = z^n + \Delta t f(z^n) \tag{6}$$

or in this case

$$z^{n+1} = z^n + \Delta t k z^n \tag{7}$$

Note that z^n denotes the approximation of $z(t_n) = z(n\Delta t)$.

In Figure 1 you can see the actual solution overlapped with the Euler approximation.

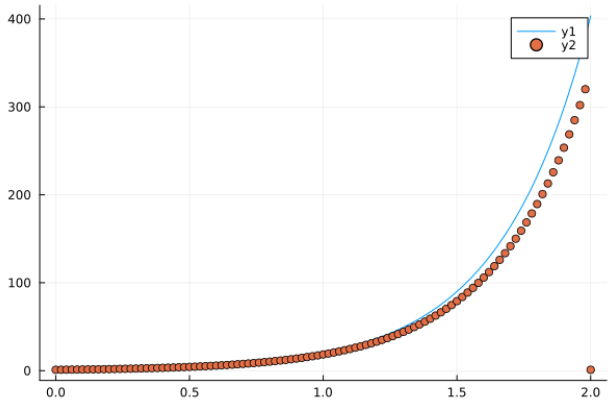


Figure 1: Euler approximation overlapped with actual solution

As you can see, the local error is small, but globally, the approximation diverges from the solution. There is actually a whole area of error analysis where bounds for local and global errors are studied. In the case of the Euler method, the bounds for the local and global errors are bounded by

$$LTE = \frac{1}{2}(\Delta t)^2 M + O(\Delta t^3) \tag{8}$$

For the Global error we assume that f is Lipschitz Continuous with L Lipschitz constant of f

$$GTE \leq \frac{2M}{L}\Delta t \tag{9}$$

Which means that the local error is proportional to the square of the time-step, meanwhile the global error is only bounded by the time-step. The bound of the error is also called the order of the method. Euler method is of order 1, but there are higher order methods that we will explore, like symplectic maps, or composition methods.

2.2 Range-Kutta Method

The Range-Kutta method is one of said higher order methods. It calculates the solution at step $n+1$ by applying this formula

$$z^{n+1} = z^n + \Delta t \sum_{i=1}^{s} b_i f(Z_i) \tag{10}$$

where Z_i is given by

$$\begin{aligned} Z_1 &= z^n \\ Z_2 &= z^n + c_2 \Delta t f(Z_1) \\ Z_3 &= z^n + c_3 \Delta t f(Z_2) \\ &\dots \end{aligned} \tag{11}$$

This is similar to the Euler method, but we subdivide the time step into s steps and we apply the weights b_i. If for example all weights where equal to $1/s$ and the points Z_i are equally spaced, with $c_i = 1/s$, then it would be the same as the Euler method but repeated s times for each time-step. We wouldn't gain anything. But if you space correctly the points Z_i and choose the weights b_i accordingly you can get better integrators. The sum of all the weights should add up to 1.

And as you can see in Figure 2, its results are much more exact, even when we go to higher steps simulations

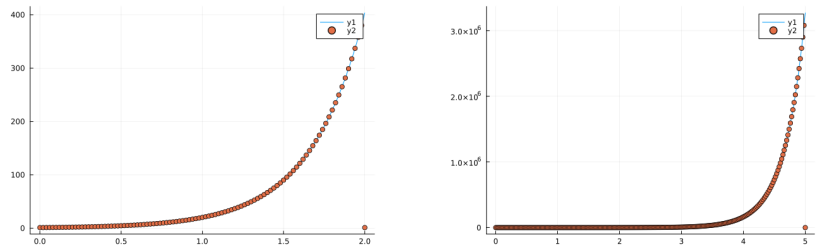


Figure 2: Actual solution overlapped with Range-Kutta approximation

The Range-Kutta method is of higher order, in particular in this case it is of order 2. The global error is bounded by the squared time step: $GTE \leq M(\Delta t)^2 + O((\Delta t)^3)$. In this case we added two points, one at the middle and another one at $\frac{3}{4}$ of the way. And we used the weights $\frac{1}{6}$, $\frac{2}{6}$, $\frac{2}{6}$, $\frac{1}{6}$.

Here in figure 3 you can see a comparison of the Euler and Range Kutta global errors, plotted in a log-graph.

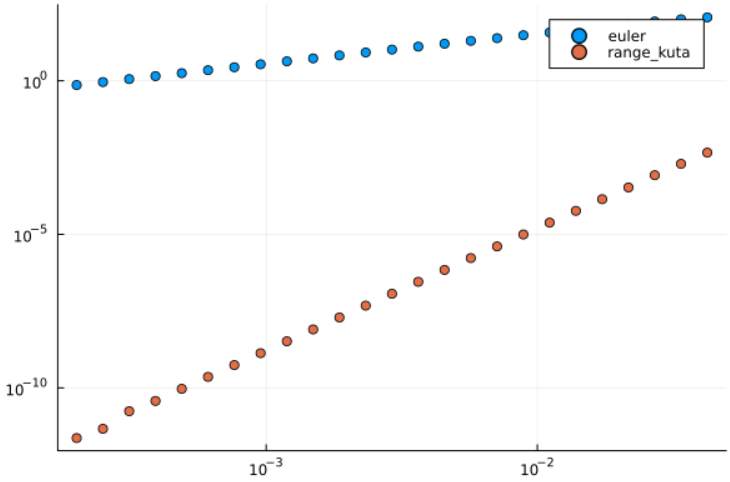


Figure 3: Euler vs Range-Kutta global error in a log-graph (x axis is the time step)

3 Hamiltonian Dynamics

Conservative mechanical systems, i.e. Hamiltonian Dynamical Systems, present certain global qualitative or geometric properties on their flow maps. Those properties can be exploited to find better approximations of solutions.

Let $z : \mathbb{R} \to \mathbb{R}^d$ represent a solution to the ODE

$$\begin{cases} \frac{dz}{dt} = f(t) \\ z(0) = z_0 \end{cases} \tag{12}$$

The flow map is defined as $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$ such that

$$\phi_t(z) = z(z_o; t). \tag{13}$$

Conceptually, if z represented the position of a body, ϕ_t maps every point p to the position after t time of the body whose initial condition was $z(0) = p$.

3.1 Symplectic Maps and Hamiltonian Flow Maps

A map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ is symplectic respect to the structure matrix J (constant and invertible) if its Jacobian $\phi_z(z)$ satisfies

$$[\phi_z(z)]^T J^{-1} \phi_z(z) = J^{-1} \quad \forall z \in \mathbb{R}^d \tag{14}$$

The flow map ϕ_t of a Hamiltonian System as in (1) is symplectic for the matrix $\begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$. This implies certain global conservation laws:

- For $d = 1$, preservation of area
- For $d > 1$, preservation of volume

3.2 Geometric Integrators for solving ODEs

A geometric integrator is a method for solving ODEs that conserves the wedge product. This means that its flow map of its solutions is symplectic. This kind of integrators are better estimators because they take into account the conservation laws, hence, giving more stable results.

3.2.1 Euler-B Method

The normal Euler method given in (6) is not a geometric integrator. If we try to simulate the solar system we get this result:

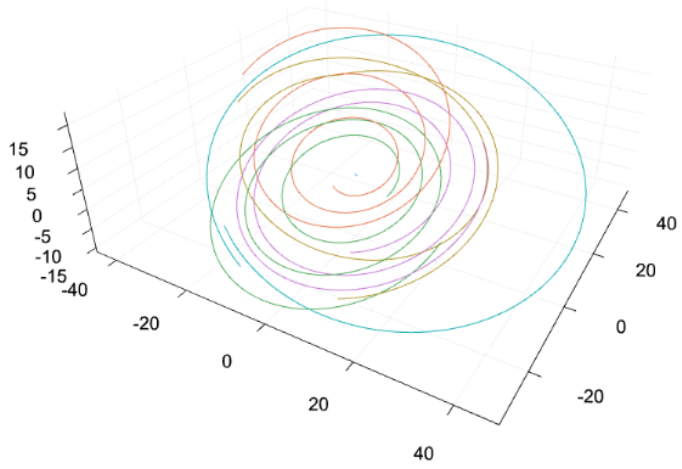


Figure 4: Euler method Solar System simulation

The Euler-B method given by

$$q^{n+1} = q^n + \Delta t \nabla_p H(q^n, p^{n+1}) \qquad p^{n+1} = p^n - \Delta t \nabla_q H(q^n, p^{n+1}) \tag{15}$$

preserves the wedge product, hence its a geometric integrator. Notice that its very similar to the Euler Method, but the velocity component is implicit. This isn't a difficulty in the N-body problem, because its a separable Hamiltonian System, hence $\nabla_q H(q^n, p^{n+1}) = g(q)$ some function of only q. So this method reduces to finding first the new velocity, and use it to calculate the new position. If we try to simulate the solar system with this method we get this much more stable result:

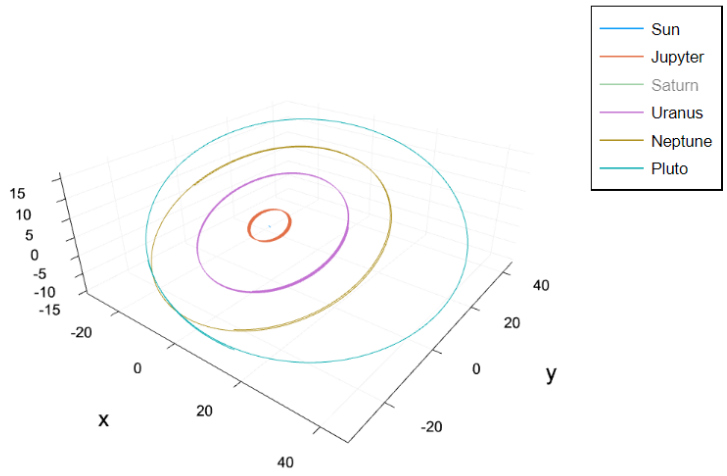


Figure 5: Solar System simulation with Euler-B Method

3.2.2 Störmer-Verlet Method

The Störmer-Verlet method introduces an extra middle step in the computations of the velocity, hence making the method more accurate, and more costly.

$$\begin{aligned} p_{n+\frac{1}{2}} &= p_n + \frac{\Delta t}{2} f(q_n) \\ q_{n+1} &= q_n + \Delta t p_{n+\frac{1}{2}} \\ p_{n+1} &= p_{n+\frac{1}{2}} + \frac{\Delta t}{2} f(q_{n+1}) \end{aligned} \tag{16}$$

This method is a very stable integrator, as we will see in Section 5.2.2

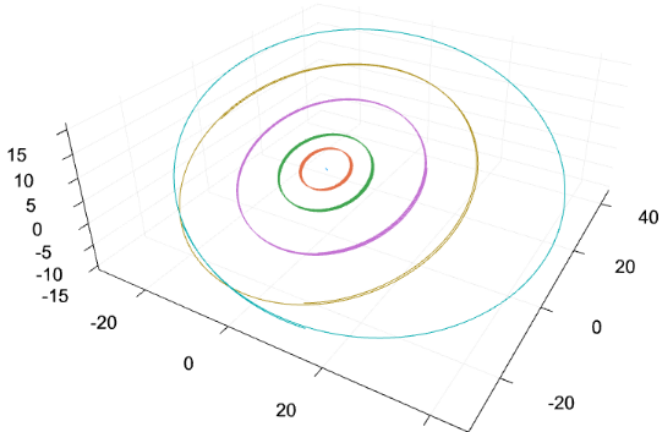


Figure 6: Solar System simulation with Störmer-Verlet Method

To compare the Störmer-Verlet and the Euler-B method, here you can see the same three-body simulation with very long integration time. As you can see, from far away it looks like the trajectories exactly overlap (left), but if you zoom in you can see there is a little difference between them (right).

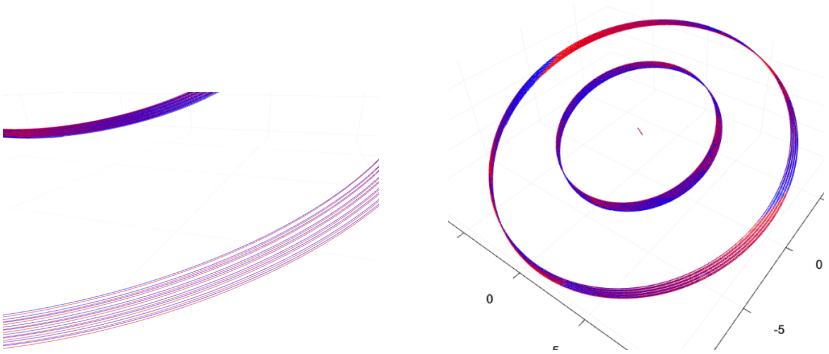


Figure 7: Euler-B and Störmer-Verlet simulations overlapped.

4 Simulations

Now that we have a integrator that behaves well, we can try to run some simulations. For the next simulations the Euler-B method was used, since it behaves well because of symplecticness. Störmer-Verlet method could have also been used and similar results would have followed.

4.1 Moons

We will begin by checking if the integrator is good enough to simulate moons of planets. But first, we can't eyeball the mass, initial position and initial velocity of the moon of a planet. Thankfully, using very basic physics we can find out the orbital velocity of a planet (without taking into account any of the other planets).

$$\|v\| = \sqrt{\frac{GM}{R}} \tag{17}$$

Note that, to the orbital velocity, we must add the planet velocity, since its calculated respect to the planet. For direction of the orbital velocity, it must be perpendicular to the planet velocity, and the vector that joins the planet and its moon.

Now we can run some simulations, lets start with one planet having one moon.

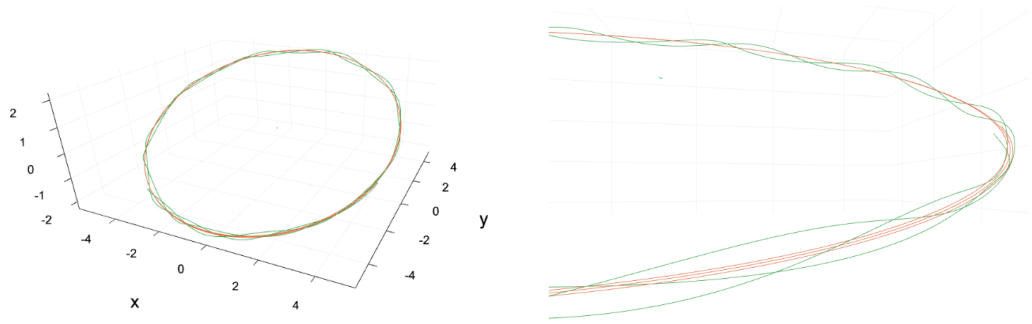


Figure 8: One Moon

Even though when we run the simulation the integrator is taking into account the forces between the sun and the moon, for our calculations for the orbital velocity, we don't factor them in. This simplifies the process, and gives good results because relatively speaking, the moon is much closer to the planet than to the sun, so the rest of the forces are negligible. See this simulation with other planets, multiple moon's orbiting a planet, and moons orbiting other moons.

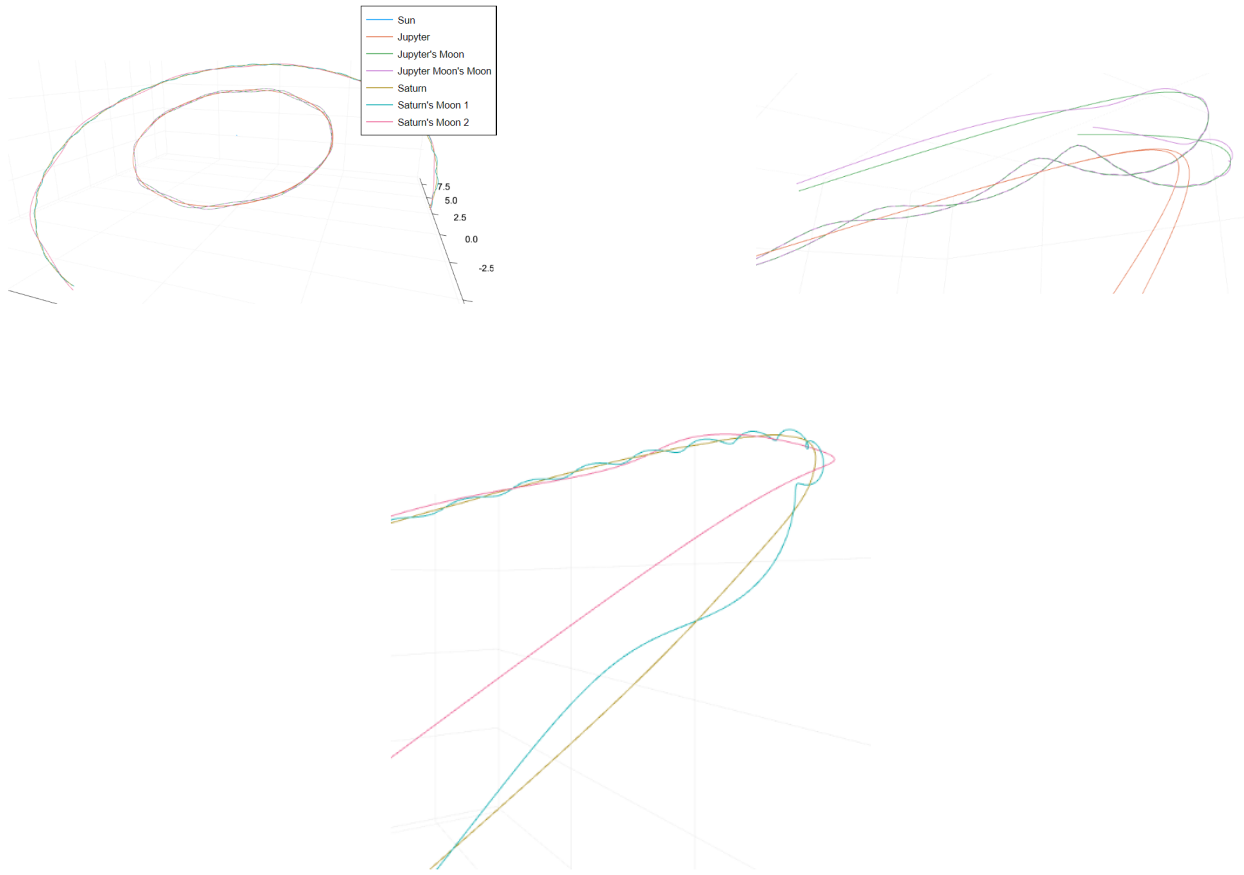


Figure 9: Moons

As you can see the integrator is capable of simulating the moons, but it has to be taken into account that time step ran for the first simulation [Figure 5] was reduced by a factor of 100 to run this last simulation.

5 Longtime Stability

Its important to analyze the consistency of our integrators, to determine if they are truly reliable (and how reliable). For this purpose we can observe how the methods conserve their first integrals.

5.1 First Integrals

In order to study the longtime stability of our methods, one good indicator is the conservation of first integrals. First integrals are values which are conserved through out the simulation. One could think of them like for example, the total mass of the system. If the total mass isn't preserved, the integrator may have some problems.

The Hamiltonian Hamiltonian dynamics are based on the fact that the Hamiltonian, or total energy of the system, should be preserved. Symplectic integrators as Euler-B or Störmer-Verlet, are stable in the sense that they preserve the Hamiltonian

$$H(p,q) = H(m,v,q) = \frac{1}{2}mv^2 - G\sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{m_i m_j}{r_{ij}} \tag{18}$$

where m is the vector of masses of the planets and v the vector of velocities.

Angular Momentum Total angular momentum is preserved in the N-body. In physics, total linear momentum is preserved when no external forces are applied. Total angular momentum is preserved when no torque forces are applied (as in the N-body problem).

$$L = \sum_{i=1}^{N} q_i \times p_i \tag{19}$$

5.2 Plots

Rather than studying the theoretical side of conservation of first integrals, we can see some direct results by integrating for long times, and tracking the evolution of the Hamiltonian and total angular momentum.

5.2.1 Hamiltonian and Euler-B

We will start by simply observing the behavior of the Hamiltonian with the Euler-B method. Then, we will compare the Euler-B and Störmer-Verlet method by running some simulations with diferent time steps and number of bodies, tracking always the conservation of Hamiltonian or total angular momentum.

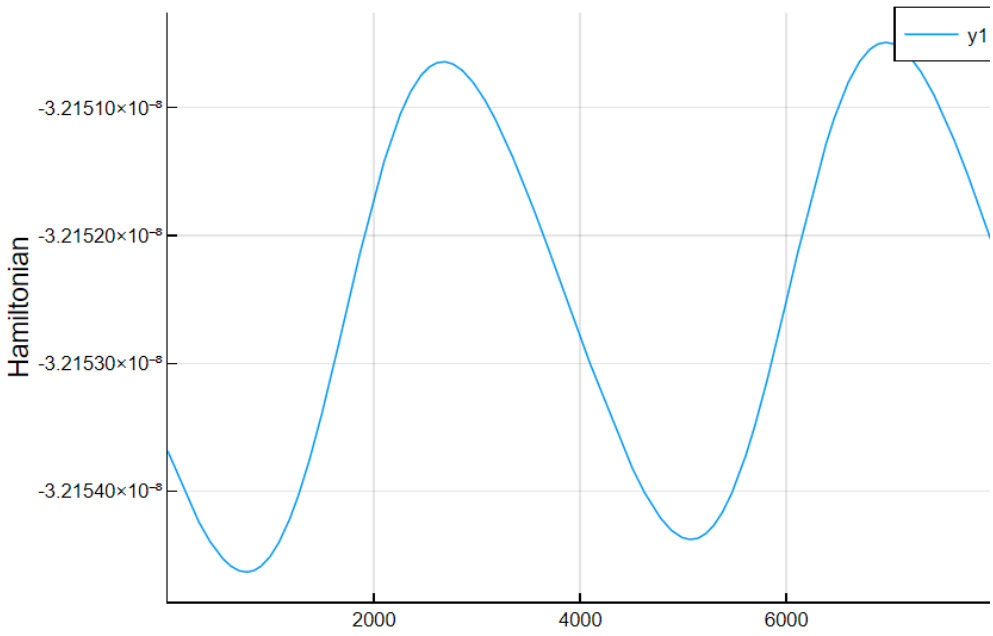


Figure 10: Hamiltonian through time in solar system simulation

We can see that the integrator behaves quite well with the Hamiltonian first integral (the error is of order 10^-8), but there is a strange oscillation. This oscillation is very difficult to get rid of. When a planet is going towards the sun the integrator calculates the speed on the current point, and then applies it to the body. Since the speed will grow as the planet goes towards the sun, energy is 'lost'. When the planet is moving away from the sun the opposite occurs, and the energy is 'recovered'.

We can run a even longer time simulation using the relative error this time:

$$E = \frac{|H - H_0|}{|H_0|} \tag{20}$$

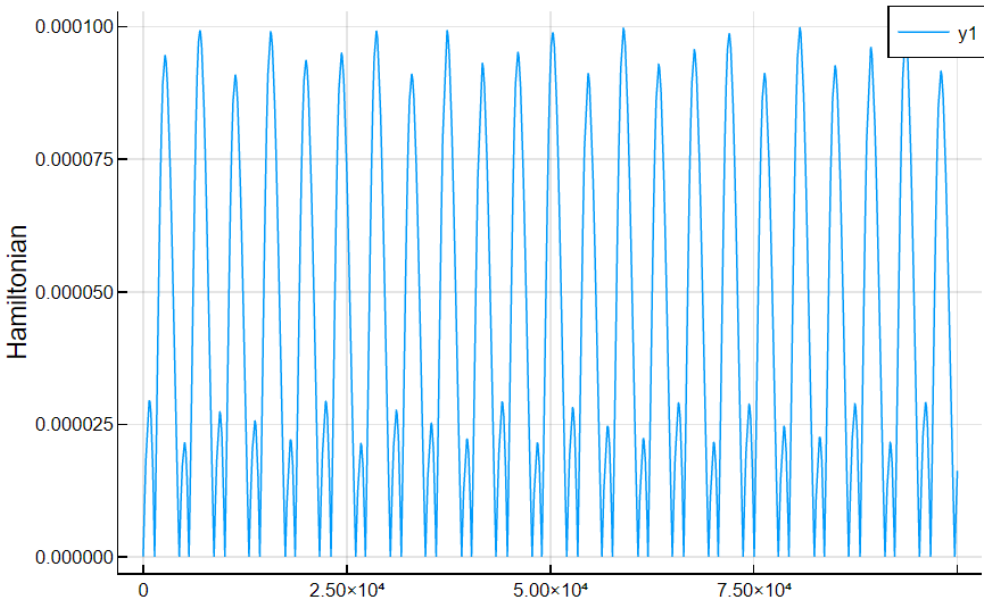


Figure 11: Hamiltonian relative error

Here we can see that the Euler-B method is stable, meaning the error oscillates but doesn't explode. This agrees with the theory of symplectic methods. To further illustrate this here you can see a plot of both Euler and Euler-B method Hamiltonian calculations through time, for two different time steps (the one on the left has $\Delta t = 1$, and the one on the right $\Delta t = 0.1$).

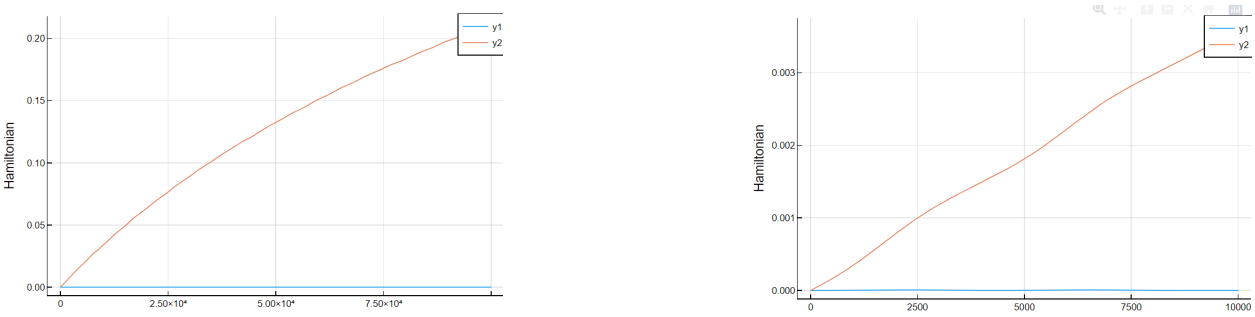


Figure 12: Hamiltonian relative error for Euler and Euler-B methods

Finally we can look at what happens when we take different time steps. Here you can see the same graph with time step $\Delta t = 1$, $\Delta t = 10$ and $\Delta t = 100$ respectively. Note that the for $\Delta t = 1$ the simulation runs for a total time of $T = N\Delta t = N$, while for $\Delta t = 100$ the simulation runs for a total time of $T = 100N$. This means that the error is not being plotted at same times, but the upper bounds still hold.

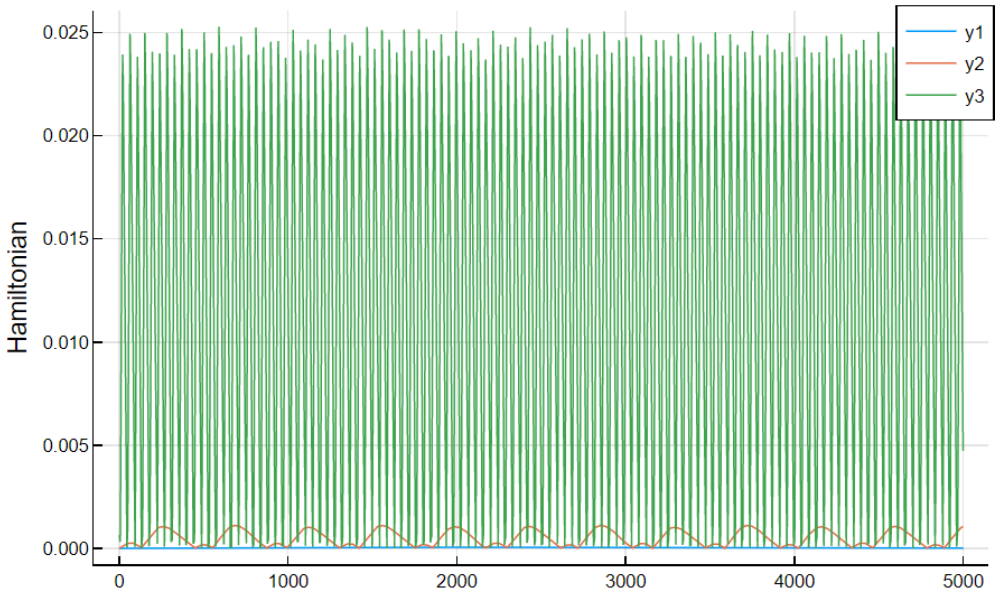


Figure 13: Hamiltonian relative error with different time steps

As a matter of fact we are more interested in the upper bound of the Hamiltonian error and to where the limit of Δt for stability is, than in the error through time. In the next plot you can see the upper bound for Hamiltonian for different time steps.

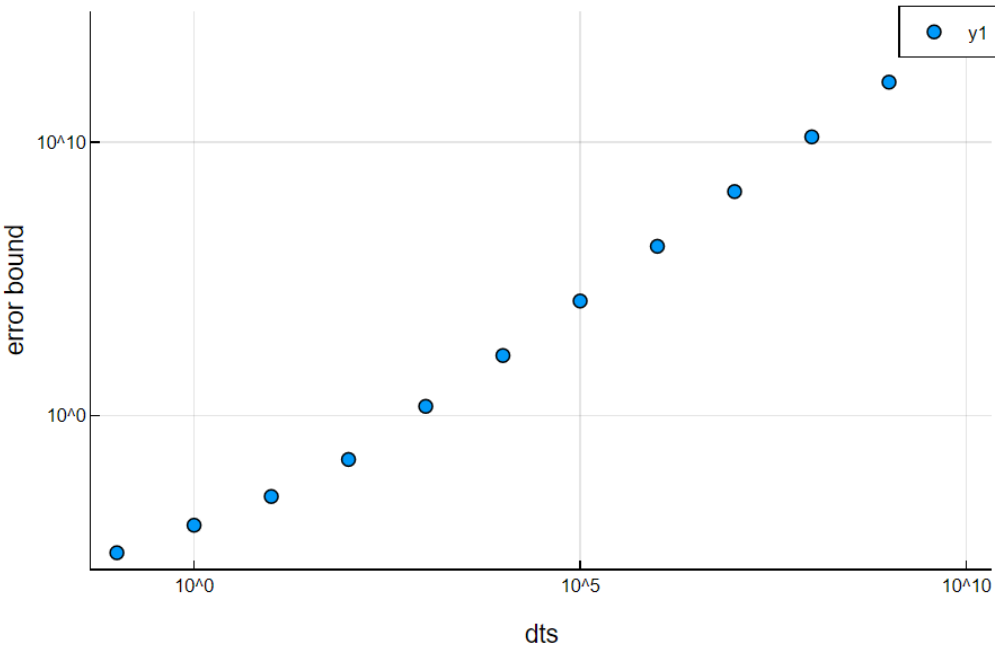


Figure 14: Hamiltonian relative error with different time steps Log-Log plot

As you can see the bound of the error of the Hamiltonian grows exponentially (log10 plot). Getting a relative error higher that 1 means that $|H - H_0| > |H0|$ so it is already considered too much. Hence the maximum Δt that the simulation can handle behaving well is around 100.

5.2.2 Comparison between Euler-B and Störmer-Verlet methods

We know that Euler-B method conserves the Hamiltonian, as opposed to the normal Euler method. And we know that Störmer-Verlet method also conserves the Hamiltonian because of symplecticness. Störmer-Verlet is more costly than Euler-B, hence we expect it to give better results, but how much better is Störmer-Verlet?

Firstly, we can plot the Hamiltonian and total angular momentum relative errors over time in a simulation. In this plot we can see how the Euler method doesn't preserve any of these two first integrals:

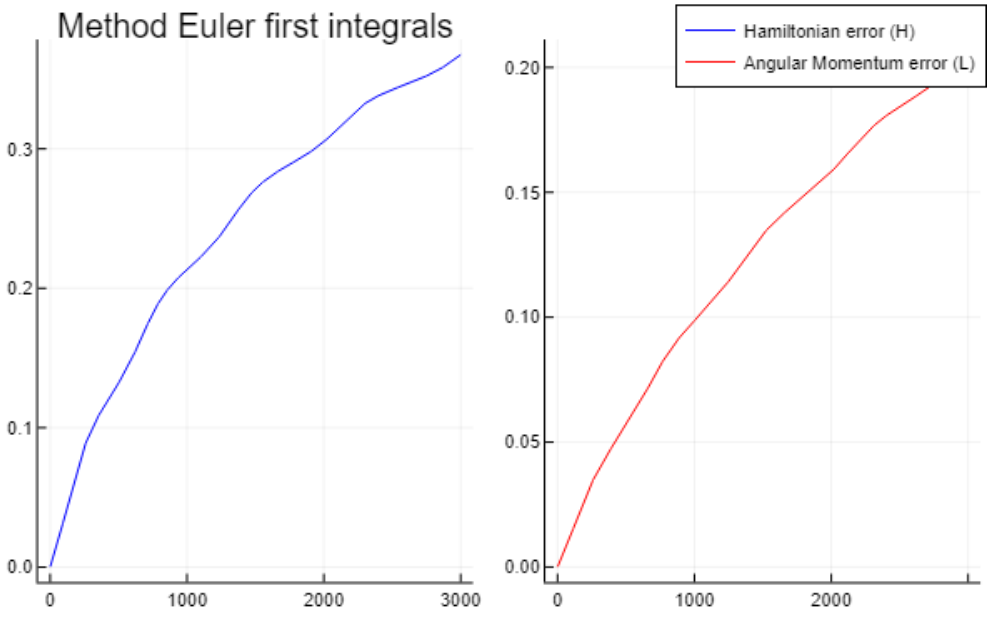


Figure 15: Hamiltonian relative error through time. Normal Euler integrator.

But, as expected, the Störmer-Verlet and Euler-B method both preserve the momentum with a 10^{-15} order of error, which can be attributed to the computer precision error. We can also note that the Störmer-Verlet method is better at preserving the Hamiltonian (error of order 10^{-6}).

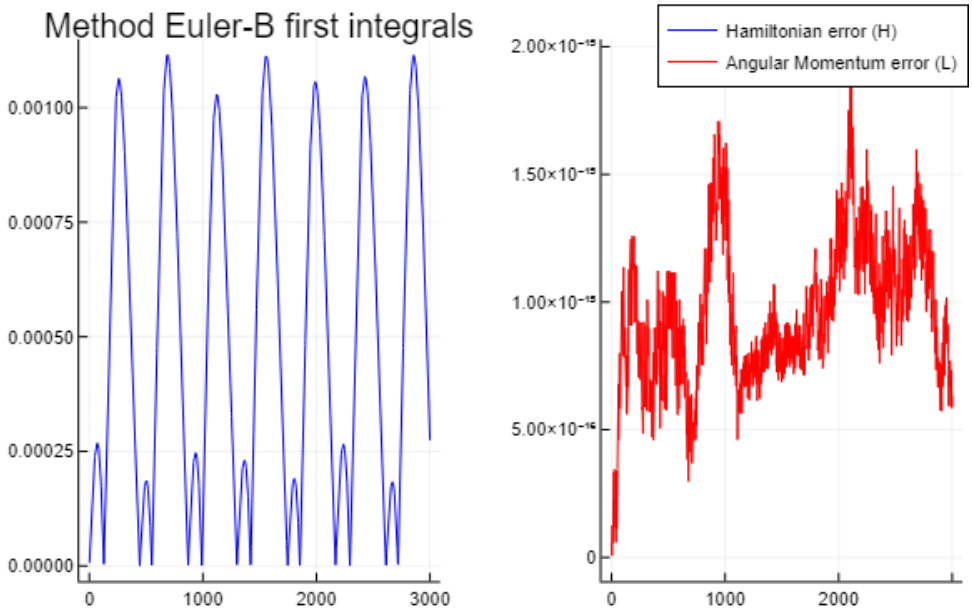


Figure 16: Hamiltonian relative error through time. Euler-B integrator

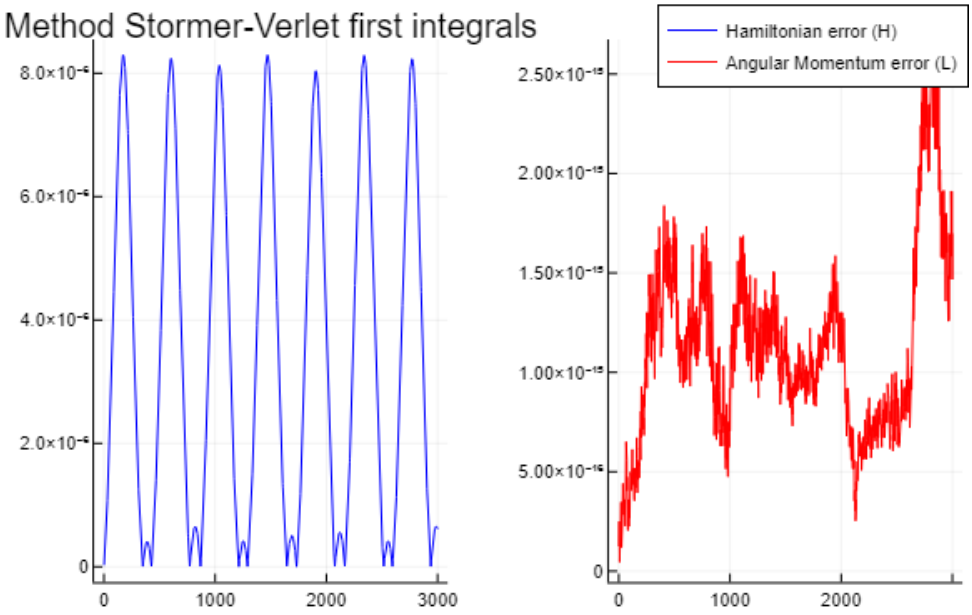


Figure 17: Hamiltonian relative error through time. Störmer-Verlet integrator

Again, here we are more interested in seeing the maximum Hamiltonian error, since as it oscillates in a stable way over time, this plots don't give us as much information. So we will plot the Hamiltonian maximum error for different time steps.

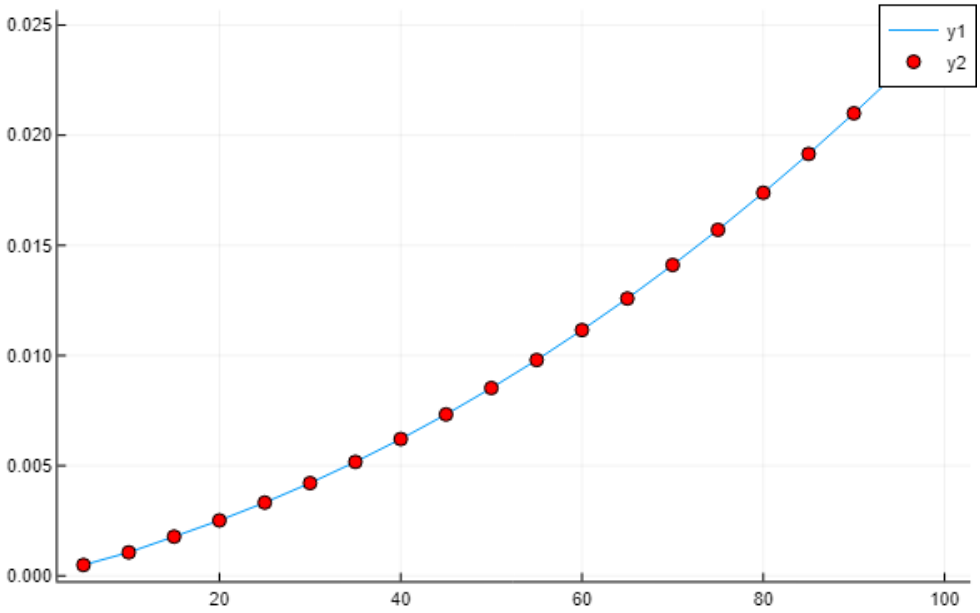


Figure 18: Hamiltonian maximum relative error for different time steps. Euler-B integrator

In the next plot you can see both methods overlapped. Finally we can clearly see how much better Störmer-Verlet method is at preserving the integral. For the cost of one more computation per time step, we get a much better result. There are other methods that partition this process even further (as seen in section 2.2).

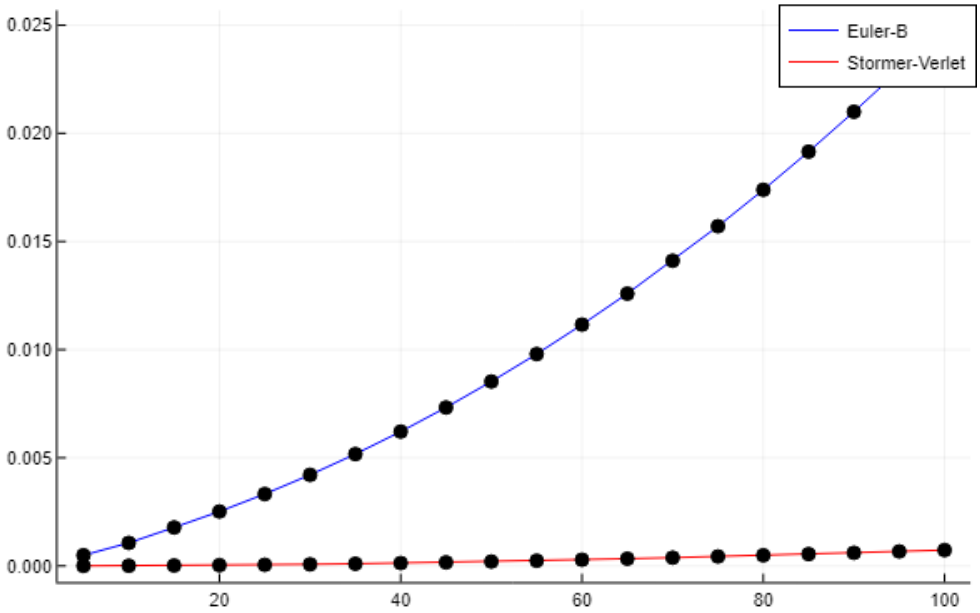


Figure 19: Hamiltonian maximum relative error for different time steps. Euler-B and Störmer-Verlet integrators

Now we can follow a similar process for the total momentum first integral. First we can plot $\frac{|L-L_0|}{|L_0|}$ over time. Note that the total angular momentum L is a vector in $\mathbb{R}^3$, so we always take it's norm.

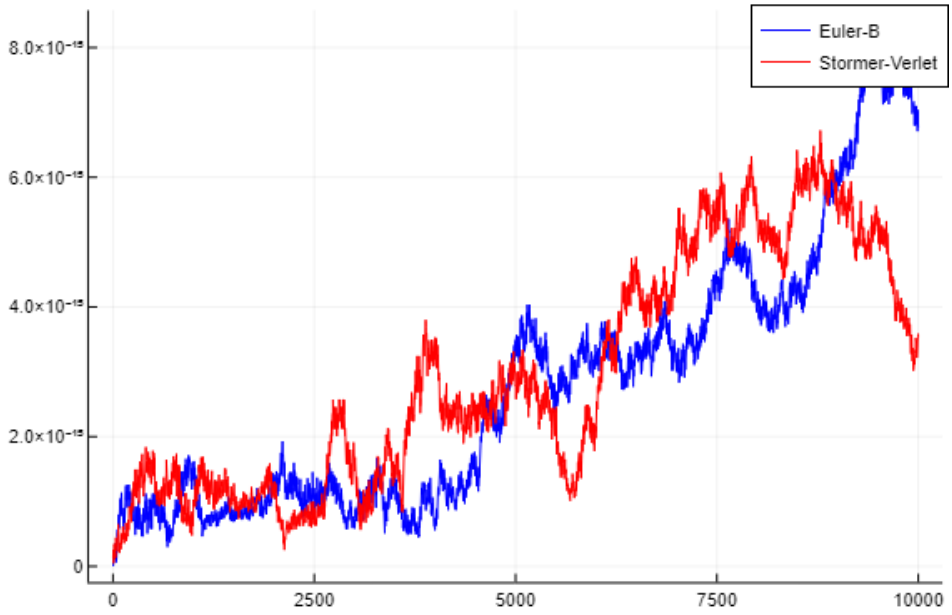


Figure 20: Total momentum relative error through time. Euler-B and Störmer-Verlet integrators

It seems as if L is almost perfectly preserved, and its oscillations feel somewhat random. In fact, as you can see in the next plot the total momentum error stays in the same 10^{-15} range for different time steps, concluding that it is perfectly preserved and the errors are due to the precision of the computer.

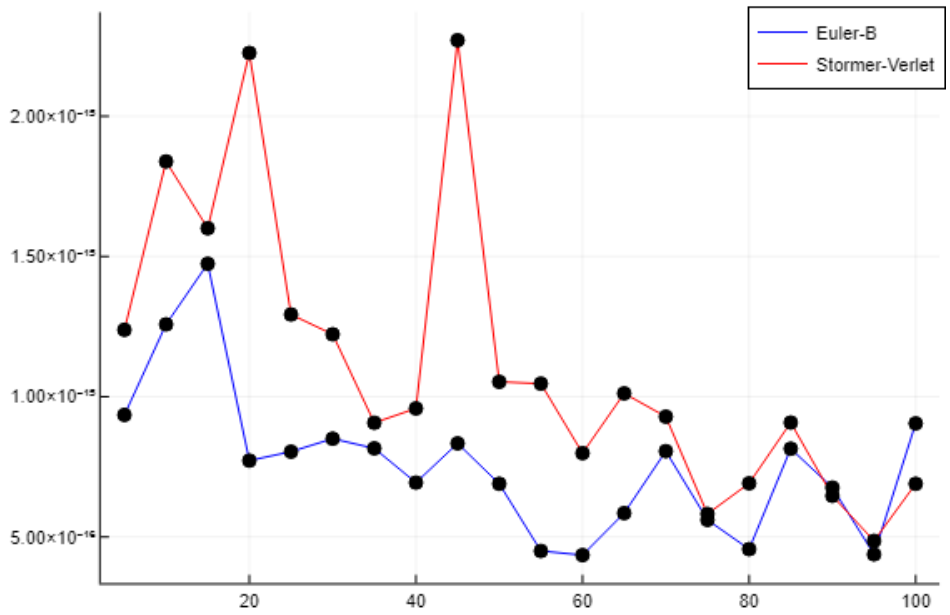


Figure 21: Maximum total momentum relative error for different time steps. Euler-B and Störmer-Verlet integrator

But, how big can we get Δt and still have the integrator preserve the angular momentum? In the next plot you can see that for Δt around 5000 we lose the preservation for the Euler-B method, and it seems as Störmer-Verlet holds for a bit longer until Δt around 8000.

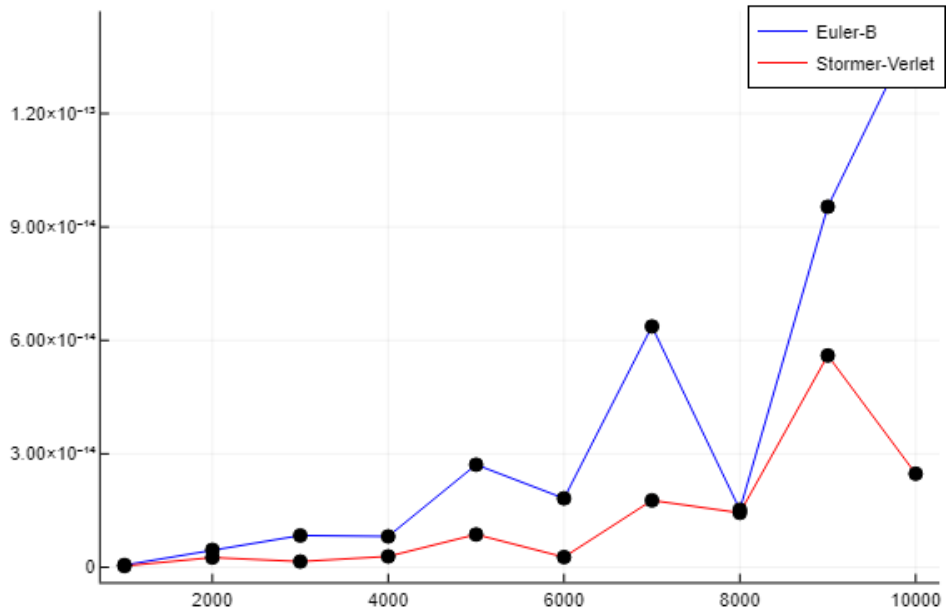


Figure 22: Maximum total momentum relative error for different time steps. Euler-B and Störmer-Verlet integrator

We can basically presume that the L momentum is perfectly preserved, since the other properties of the integrator are lost much sooner than such big time steps.

6 Conclusion

Hamiltonian Dynamics is a field that has been widely studied by mathematicians. Even though it is really hard (even impossible) to find the actual solutions to these EDP problems, there is still a lot of theory that can be made. Finding integrators that behave well, and actually proving their consistency and stability conditions, is really important and can help us estimate and study the behaviour of complex systems. This project illustrates the implications of stability and symplecticness, by implementing different methods on the Julia programming language, and running simulations.

During the development of the project, a lot of theory on Hamiltonian Dynamics was learned: Numerical methods for solving differential equations, error analysis, the meaning of symplecticness, what first integrals are, and the study of longtime stability. Finally, three of these methods were implemented.