Analyzing and Solving the N-Body Problem

APPM 4610 Spring 2025 Section 001

Bryce Pfuetze, Edward Wawrzynek, Erick White

5 May 2025

1 Introductory Material

In this project, we analyze the N-body problem, a well-known open problem in a variety of physics and engineering fields. We discuss difficulties in approximating its solution and investigate current approaches to numerically solving this problem. We begin with a discussion of stiff differential equations, exploring why such equations present difficulties in numerical solutions and introducing dedicated Symplectic integration methods. We also cover basics of chaotic systems and their behavior. We then introduce the N-body problem, explaining how it arises in physical systems and where it is relevant; we follow this with a derivation of the N-body problem in the case of mutual gravitational attraction between N bodies. We then further analyze the Symplectic solvers introduced earlier, including analysis of their behavior for long-time integration and comparison to common explicit timestepping methods. We conclude with an application of the N-body problem in predicting the motion of objects in the Solar System. We focus on the case of N small, on the order of ones or tens of objects.

1.1 Stiff Differential Equations

One of the simplest ways to guarantee the existence and uniqueness of a solution involves proving the boundedness of derivatives. However, there are some differential equations with solutions that exist that have unbounded derivatives. These problems, known as *stiff differential equations*, are tricky to solve and require specific conditions for accuracy.

The stiff differential equation discussed in class is known as the test problem:

$$\begin{aligned} y' &= \lambda y \qquad \lambda < 0 \\ y(0) &= \alpha \neq 0 \\ \text{where } y(t) &= \alpha e^{\lambda t} \end{aligned} \tag{1}$$

The test problem allows for the calculation of acceptable h values for a particular time-stepping method. The application of the single or multi-step method of choice on the test problem leads to the associated characteristic polynomial $Q(\lambda h)$. This polynomial can then be plotted on the complex plane. All solutions that fall to the right of the imaginary axis are 'nonphysical' because λ is known to be a negative number, requiring the step size h to also be negative. This means the only acceptable solutions fall on the negative end of the real axis inside of the bounds created by $|Q(\lambda h)| < 1$; these combinations of λ and h are stable.

If the stable region of a method is the entire left half-plane (*e.g.*, $\Re(\lambda) < 0 \rightarrow |Q(\lambda h)| < 1$), it is known as *A-Stable*. All combinations of λ and h for A-Stable methods are stable. While this property is nice, these methods are typically computationally expensive or generally difficult to use. It can often be better to choose a less stable method with a valid (stable) combination of λ and h [3].

In class we applied Euler's Method and the Implicit Trapezoidal method to the test problem, resulting in the following associated characteristic polynomials plotted on the complex plane:

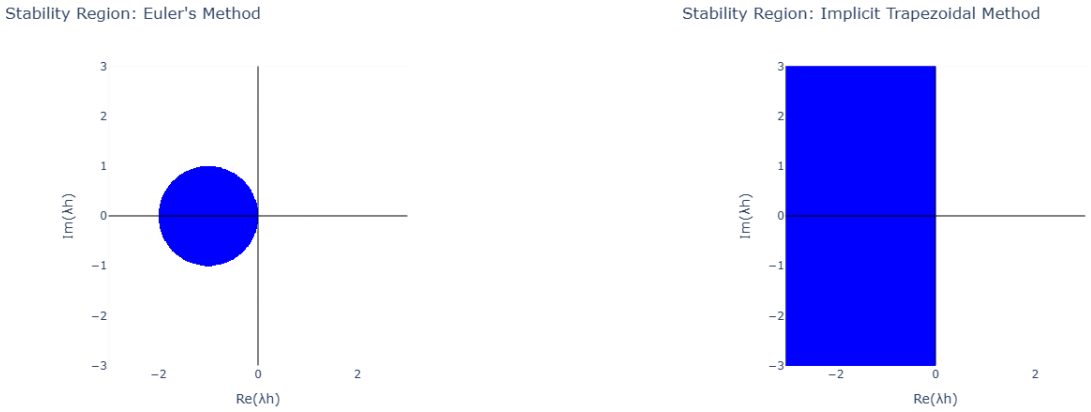

Figure 1: Regions of stability for Euler's method (left) and the implicit trapezoidal method (right) in the complex plane.

Method	Euler's Method	Implicit Trapezoidal Method
Definition	$w_{i+1} = w_i + hf(t_i, w_i)$	$w_{i+1} = w_i + \frac{h}{2}(f(t_{i+1}, w_{i+1}) + f(t_i, w_i))$
Characteristic Polynomial	$Q(\lambda h) = (1 + \lambda h)$	$Q(z, \lambda h) = (1 - \frac{\lambda h}{2})z - (1 + \frac{\lambda h}{2})$
Bounds	$\lvert 1 + \lambda h \rvert < 1$	$\lvert \frac{2+\lambda h}{2-\lambda h} \rvert < 1$

These regions of stability can be found in Figure 1. Euler's method defines a circle of radius 1 centered at $\lambda h = -1$. Any combination of λh that lies within this circle is a stable combination. Meanwhile the Implicit Trapezoidal Method is A-stable because the entire left half plane consists of valid combinations of λh.

1.2 Chaotic Systems

Often, the first words that come to mind associated with the term "chaotic" are "unpredictable" and "disorderly". Ideally, a chaotic system can be predicted, but this often falls apart as soon as any sort of numerical system with rounding errors is applied.

A chaotic system is mathematically defined as one that is sensitive to initial conditions, topologically transitive, and possesses a dense periodic orbit. [10] For the purposes of this paper, we will focus on the first property in the problem of approximating solutions to stiff differential equations.

A system is sensitive to initial conditions if, after a series of iterations, the output differs wildly from a small change in inputs. The most widely used example of a chaotic system is the double pendulum. Simple dynamics allow for the prediction of the position of single pendulum, like on a clock. If the starting position and velocity is known, then the position at any given time is only a few calculations away. However, if a second pendulum is attached to the end of the first, we have a much more complex system. A small change in the angle of the second pendulum relative to the first will result in an entirely different position after a sufficiently long time.

To illustrate this sensitivity to initial conditions we will use another simple example: logistics maps. These are based on the following function:

$$x_{n+1} = Ax_n(1 - x_n)$$

Here, A is the nonlinearity parameter and x_0 is the initial condition. For cases $A < 3$, the final state can be easily predicted by setting the left hand side equal to x_n so that $x_n = Ax_n(1 - x_n)$. The resulting quadratic is satisfied when $x = 0$ or when $x = 1 - \frac{1}{A}$, but only in cases where $A < 3$ and the value of x_n converges. End states of the logistic map fall into three categories: stable, oscillatory, and chaotic.

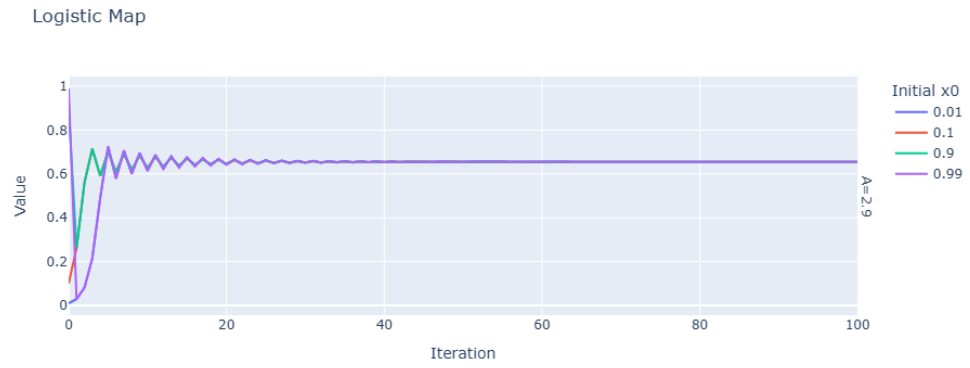

Figure 2: Logistic Map for A = 2.9

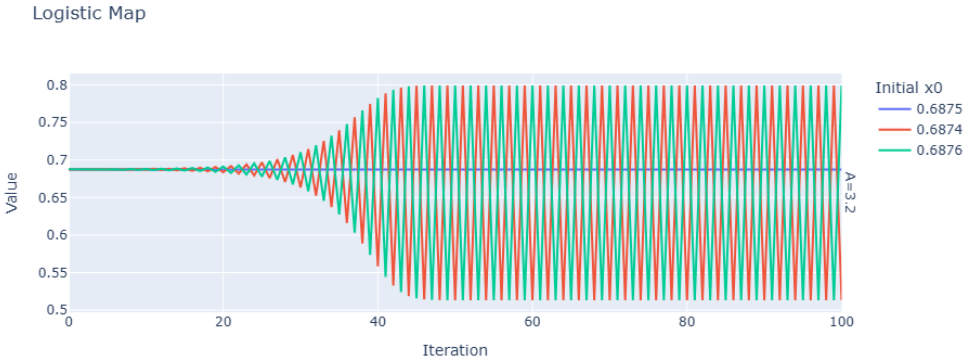

Figure 3: Logistic Map for A = 3.2

We begin with the case where $A = 2.9$ for a variety of x_0 values as seen in Figure 2. This is an example of a stable choice of conditions for the logistic map, with all starting values converging within 100 iterations.

When A is increased to 3.2 in Figure 3, we begin to see the sensitivity to initial conditions. A change as small as 10^{-4} is the difference between a stable and oscillatory solution.

Finally, $A = 3.9$ results in a chaotic solution that is highly dependent on x_0 (Figure 4). The solutions stay close for the first 10 iterations, after which they all diverge to completely different paths. Knowing the value for one would not allow you to easily predict the value for the other. If the initial conditions were readings made in a experiment, the sensitivity of the tools used becomes very important.

1.3 Symplectic Methods for Timestepping

Many dynamic systems depend on the assumption that energy is conserved. Unfortunately, the majority of numerical differential equation solvers are not capable of generating solutions that conserve energy. There are, however, some designed specifically for this purpose that are known as symplectic methods.

Orbits and the N-body problem are the perfect application for a symplectic integrator [2]. These systems, often involving planets or atomic particles, experience very little energy loss (this would not be the case in, *e.g.*, a system with a lot of friction). In order to apply a symplectic method, the equations of motion for the system must be written in the form of the Hamiltonian. This is a scalar function that takes the momentum and position of a particle as the inputs:

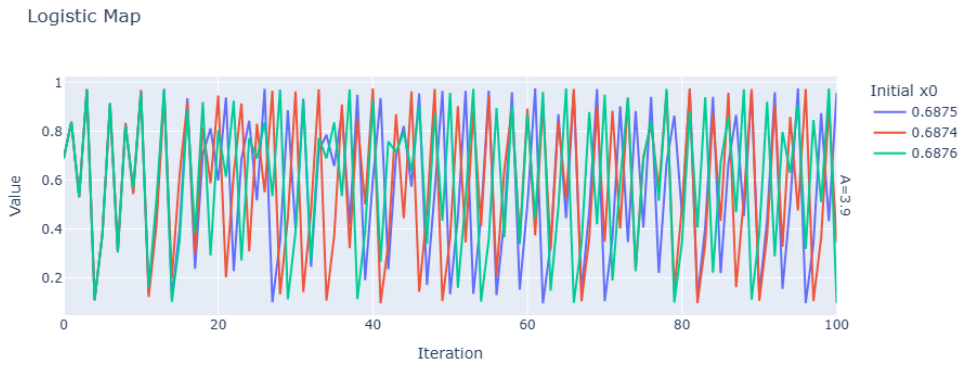

Figure 4: Logistic Map for A = 3.9

$$\mathcal{H}(p, x) = T + V$$

Here, T is the kinetic and V the potential energy. For instance, in the case of gravitation between bodies with mass m_i, positions x_i and momenta p_i, the Hamiltonian is given by

$$\mathcal{H}(p, x) = \sum_i \frac{|p_i|^2}{2m_i} - \sum_{i<j} \frac{Gm_i m_j}{|x_i - x_j|}.$$

such Hamiltonians can be derived for a variety of dynamical systems, although the T and V terms will naturally vary based on the underlying governing equations.

In solving the N-body problem, we will look at how symplectic methods out-performs other solvers (*e.g.*, explicit Euler and Runge-Kutta). We will also explore the use of variable timestepping in order to improve accuracy when these systems approach a singularity.

2 The N-body Problem

2.1 Introduction

The N-body problem arises in a number of physical situations involving mutual attraction between N distinct bodies [7]. Typically, such relations are represented using the *force* on a body with the goal of finding the *position* at a given time; two of the most common formulations of this are Coulomb's Law,

$$\mathbf{F}_{12} = \frac{kq_1q_2}{r_{12}^2}\hat{\mathbf{r}}_{12},$$

which governs electrostatic attraction and repulsion, and Newton's Law of Gravitation,

$$\mathbf{F}_{12} = -\frac{Gm_1m_2}{r_{12}^2}\hat{\mathbf{r}}_{12},$$

which governs gravitational attraction [11]. The important thing to note in each of these cases is the inverse square relation between the force acting on the two bodies and the distance between them:

$$F_{12} \propto \frac{1}{r_{12}^2}$$

(note that the unit vector $\mathbf{r}$, which points from one body to the other, is simply used to convert this scalar quantity to a force vector). This proportionality is a characteristic of the governing forces (*e.g.*, any force that diminishes as distance squared leads to an inverse square law which can then be modeled using N-body dynamics). This inverse square relationship stems from a force governed by a constant-flux source. Given a source of strength Γ in space that is taken to be the only source of a given force, the total flux of this force through a sphere of radius r centered on this source must also be Γ; however, this means that the flux per unit area of the sphere is now

$$\frac{\Gamma}{4\pi r^2},$$

hence the origin of the inverse square law [5].

Since gravitational (in the Newtonian sense) and electrostatic attraction due to point sources can both be modeled as some source of constant flux, they are both subject to this inverse square law (as shown above in Coulomb's Law and Newton's Law of Gravitation). It is worth noting, however, that not all physical relationships follow this law; for instance, the electric field strength due to an electric dipole and the magnetic field strength due to a magnetic dipole are both governed by inverse *cube* laws [11]:

$$F \propto \frac{1}{r^3}.$$

The N-body problem is also heavily dependent on the number N of bodies involved. For small N, it is typically reasonable to consider each body individually and compute the forces exerted on each body by each other $\left(\mathcal{O}\left(N^2\right)\right)$; however, this becomes infeasible for large N, where alternate methods (*e.g.*, using techniques such as particle-mesh grids) become necessary [1]. We focus here on the "small"-N case, where N is on the order of ones or tens of objects.

The applications of the N-body problem come largely through modeling complex dynamical systems that are too costly or difficult (or, oftentimes, impossible) to model directly. A primary example of this is in the field of astrodynamics, where the N-body problem is crucial when planning missions belong the sphere of influence of a single body (*e.g.*, interplanetary missions); this is an example of a small-N problem, as such studies typically involve only the most significant gravitational forces in the solar system (*e.g.*, the Sun, the major planets, and nearby moons) [9]. In a similar vein, the N-body problem for large N can be used to model galactic dynamics (*i.e.*, the evolution of galaxies), where the number of bodies far exceeds what can be computed directly [6].

2.2 Derivation

We proceed with the derivation of the N-body equation for a gravitationally-attractive system. Assuming that an appropriate inertial reference frame is chosen, we can invoke Newton's Second Law, which states that the force on an object is equal to its mass times its acceleration [11]:

$$\mathbf{F} = m\mathbf{a}.$$

However, nothing that the acceleration of an object is nothing more than the second derivative of its position, we can rewrite this as

$$\mathbf{F} = m\ddot{\mathbf{r}}$$

(note that this positional quantity is relative to the origin of some inertial coordinate system, where *inertial* denotes a system which is not accelerating or rotating [9]).

Recall that the gravitational force between two bodies is given by

$$\mathbf{F}_{12} = -\frac{Gm_1m_2}{r_{12}^2}\hat{\mathbf{r}}_{12},$$

where r_{12} is the distance between the bodies, or, equivalently, by

$$\mathbf{F}_{12} = -\frac{Gm_1m_2}{r_{12}^3}\mathbf{r}_{12},$$

where $\mathbf{F}_{12}$ is the gravitational force on second body from the first body, G is the gravitational constant, $\mathbf{r}_{12}$ is the vector from the location of the first body to the second in inertial space, and m_1 and m_2 are the masses of the two objects [9].

By Newton's Second Law,

$$\mathbf{F}_{12} = m_2\mathbf{a}$$

in inertial space and

$$\mathbf{a} = \ddot{\mathbf{r}}$$

in an inertial frame; therefore,

$$m_2\ddot{\mathbf{r}}_2 = -\frac{Gm_1m_2}{r_{12}^3}\mathbf{r}_{12},$$

where $\ddot{\mathbf{r}}_1$ is the acceleration of the first body in inertial space. Dividing on both sides by m_2 yields

$$\ddot{\mathbf{r}}_2 = -\frac{Gm_1}{r_{12}^3}\mathbf{r}_{12}.$$

Note that

$$\mathbf{r}_{12} = \mathbf{r}_2 - \mathbf{r}_1;$$

therefore, the above equation can be rewritten as

$$\ddot{\mathbf{r}}_2 = -\frac{Gm_1}{\|\mathbf{r}_2 - \mathbf{r}_1\|^3}(\mathbf{r}_2 - \mathbf{r}_1).$$

This can be extended to N bodies by superposition; that is, for bodies $2, 3, \ldots, N$ acting on body one, the total acceleration is given by [9]:

$$\boxed{\ddot{\mathbf{r}}_1 = G\sum_{j=2}^{N}\frac{m_j}{\|\mathbf{r}_j - \mathbf{r}_1\|^3}(\mathbf{r}_j - \mathbf{r}_1)}.^{1}$$

Note that the vector $\mathbf{r}_n - \mathbf{r}_1$ involved still points *from* body one *to* body j; however, we are now considering the force *on body one* (hence the sign change before the force).

[1] Because this model is based on Newtonian mechanics, it is only valid for velocities much less than the speed of light ($v << c$) [4]. This also assumes that the bodies in question can be modeled as spherically symmetric point masses of constant mass [11].

Implementing this system for N bodies results in a highly-coupled system of nonlinear second-order vector differential equations; each of these must be solved independently but simultaneously for the time-evolution of the bodies. Note that, due to the fact that each body directly impacts the motion of each other body, it is not sufficient to solve this system for each body separately over a given interval of time.

The simplified two-body problem consists of a system with 12 degrees of freedom (three in position and three in velocity for two bodies); solving this system analytically requires finding 12 *integrals of motion*, which are physical constants related to the system. In this case, they are constants provided by physical parameters such as conservation of linear momentum (six constants), conservation of energy (three constants), and conservation of angular momentum (three constants); therefore, the two-body problem has a closed-form solution. Unfortunately, for $N \geq 3$, this is not the case; it has been proven that there do not exist closed-form solutions for these problems (there are not enough integrals of motion to solve such problems); therefore, numerical approaches are the only ones possible [9].

2.3 Verlet Integration

We consider a second order initial value problem with a differential equation of the form

$$\frac{\mathrm{d}^2\mathbf{q}}{\mathrm{d}t^2} = \mathbf{f}(\mathbf{q}). \tag{2}$$

We showed in the previous section that the N-body problem takes this form. Many other problems in mechanics also take this form. To reduce (2) to a first order system, we introduce the velocity $v = \frac{\mathrm{d}q}{\mathrm{d}t}$ and have

$$\begin{aligned} \frac{\mathrm{d}q}{\mathrm{d}t} &= v, \\ \frac{\mathrm{d}v}{\mathrm{d}t} &= f(q). \end{aligned} \tag{3}$$

One approach to solving (3) is the use of one-step methods such as the Euler or Runge-Kutta methods. For the n-body problem, we will later show that these two methods introduce energy error—they produce solutions with unbounded energy. Instead, we want a method which is guaranteed to bound energy error. One such method is the Stormer-Verlet method, described below.

We wish to approximate (3) with finite differences. We pick the uniform grid $t_n = t_0 + nh$, where h is the step size. We can then approximate v as the centered difference

$$v_n = \frac{q_{n+1} - q_{n-1}}{2h}. \tag{4}$$

We introduce also the staggered grid $t_{n+\frac{1}{2}} = t_0 + \left(n + \frac{1}{2}\right) h$, shown in Figure 5. On this grid, we approximate q as the average of its neighbors

$$q_{n-\frac{1}{2}} = \frac{q_n + q_{n-1}}{2} \tag{5}$$

and v as the centered difference

$$v_{n-\frac{1}{2}} = \frac{q_n - q_{n-1}}{h}. \tag{6}$$

To combine these, we discretize the second order formulation (2) with the centered second difference,

$$\frac{q_{n+1} - 2q_n + q_{n-1}}{h^2} = f(q_n). \tag{7}$$

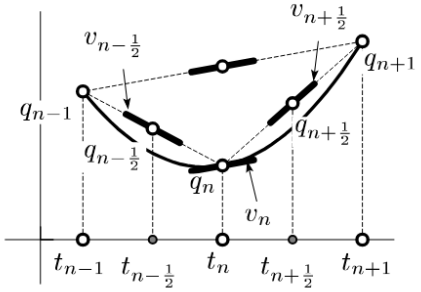

Figure 5: Staggered grid and location of evaluation of finite differences for Verlet integration.

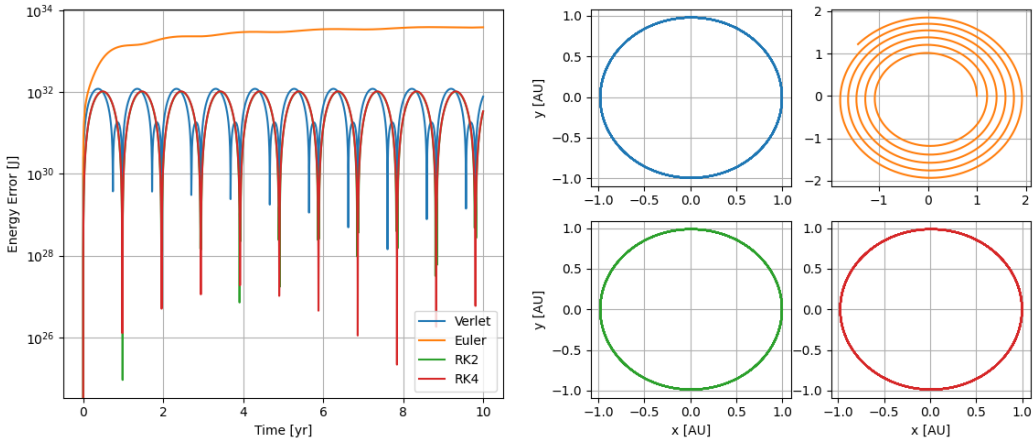

Figure 6: Integration of the Earth-Sun system over 10 years with $h = 1$ day.

Substitution of (4), (5), and (6) into (7) yields the steps

$$\begin{aligned} v_{n+\frac{1}{2}} &= v_n + \frac{h}{2} f(q_n), \\ q_{n+1} &= q_n + h v_{n+\frac{1}{2}}, \\ v_{n+1} &= v_{n+\frac{1}{2}} + \frac{h}{2} f(q_{n+1}). \end{aligned} \tag{8}$$

In (8), the last line of the n-th step takes the same form as the first line of the $(n+1)$-th step, so we can write the method as

$$\begin{aligned} v_{n+\frac{1}{2}} &= v_{n-\frac{1}{2}} + h f(q_n), \\ q_{n+1} &= q_n + h v_{n+\frac{1}{2}}. \end{aligned} \tag{9}$$

This symplectic stepping scheme is called the Stormer-Verlet method. It can be interpreted as a one-step method, were we've evaluated the positions q on the original grid t_n and the velocities on the staggered grid $t_{n+\frac{1}{2}}$.

Numerical results for integration of the Earth-Sun system are shown in Figures 6 - 8. For short integration periods, such as that shown in Figure 6, notice that explicit Euler loses energy as the Earth moves out of its orbit, but that the other methods produced well behaved results. For longer integration periods, such as those shown in Figure 7, both the Verlet method and RK4 produce orbits with similar spatial extent, but that the Verlet method's energy error is bounded while RK4's grows through the simulation. For extremely long integrations, such as the 10,000 year integration shown in Figure 8, the Verlet method performs favorably in both spatial stability of the orbit and energy error.

2.4 Application

We now approach the challenge of solving the N-body problem. In solving this problem numerically using the Runge-Kutta method, we employ linear algebra to increase the simplicity and computational efficiency of the equations of motion. Recall that the acceleration of each individual element of an N-body system is given by

$$\ddot{\mathbf{r}}_1 = G \sum_{j=2}^{N} \frac{m_j}{\|\mathbf{r}_j - \mathbf{r}_1\|^3} (\mathbf{r}_j - \mathbf{r}_1).$$

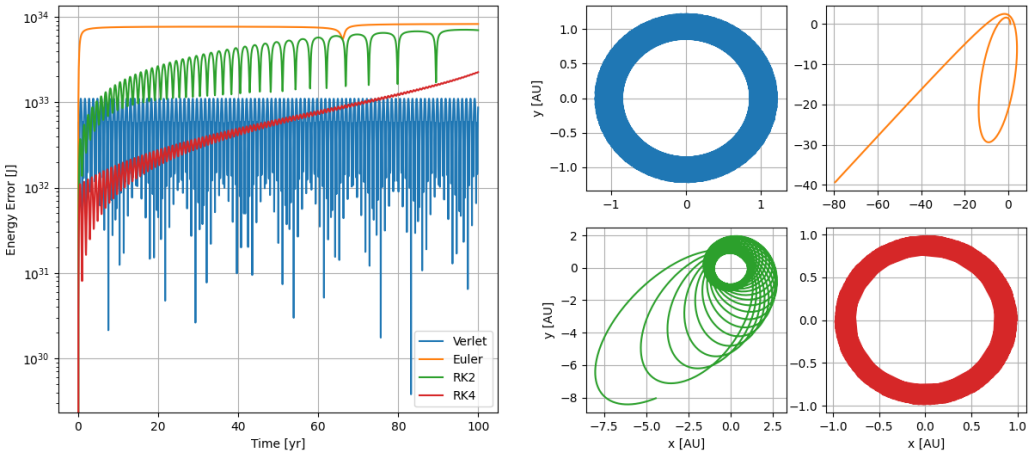

Figure 7: Integration of the Earth-Sun system over 100 years with $h = 20$ days.

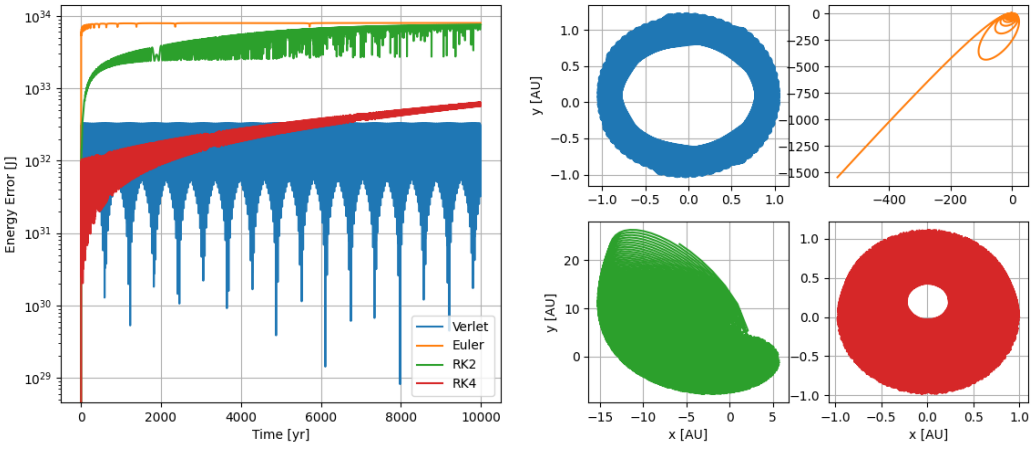

Figure 8: Integration of the Earth-Sun system over 10000 years with $h = 7$ days.

Focusing first on the mass-distance ratio, note that the denominator is the same for both the force acting *on* body i *from* body j and the force acting *from* body i *on* body j; therefore, only half the denominator terms need to be calculated. These calculations result in the following matrix:

$$R = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{r_{12}^3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{r_{13}^3} & \frac{1}{r_{23}^3} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{r_{1(N-2)}^3} & \frac{1}{r_{2(N-2)}^3} & \frac{1}{r_{3(N-2)}^3} & \cdots & 0 & 0 & 0 \\ \frac{1}{r_{1(N-1)}^3} & \frac{1}{r_{2(N-1)}^3} & \frac{1}{r_{3(N-1)}^3} & \cdots & \frac{1}{r_{(N-2)(N-1)}^3} & 0 & 0 \\ \frac{1}{r_{1N}^3} & \frac{1}{r_{2N}^3} & \frac{1}{r_{3N}^3} & \cdots & \frac{1}{r_{(N-2)N}^3} & \frac{1}{r_{(N-1)N}^3} & 0 \end{bmatrix}.$$

This matrix can then be transposed and added to itself (rather than re-calculating all distances) to yield

$$S = R + R^{\mathrm{T}} = \begin{bmatrix} 0 & \frac{1}{r_{12}^3} & \frac{1}{r_{13}^3} & \cdots & \frac{1}{r_{1(N-2)}^3} & \frac{1}{r_{1(N-1)}^3} & \frac{1}{r_{1N}^3} \\ \frac{1}{r_{12}^3} & 0 & \frac{1}{r_{23}^3} & \cdots & \frac{1}{r_{2(N-2)}^3} & \frac{1}{r_{2(N-1)}^3} & \frac{1}{r_{2N}^3} \\ \frac{1}{r_{13}^3} & \frac{1}{r_{23}^3} & 0 & \cdots & \frac{1}{r_{3(N-2)}^3} & \frac{1}{r_{3(N-1)}^3} & \frac{1}{r_{(N-1)N}^3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{1}{r_{1(N-2)}^3} & \frac{1}{r_{2(N-2)}^3} & \frac{1}{r_{3(N-2)}^3} & \cdots & 0 & \frac{1}{r_{(N-2)(N-1)}^3} & \frac{1}{r_{(N-2)N}^3} \\ \frac{1}{r_{1(N-1)}^3} & \frac{1}{r_{2(N-1)}^3} & \frac{1}{r_{3(N-1)}^3} & \cdots & \frac{1}{r_{(N-2)(N-1)}^3} & 0 & \frac{1}{r_{(N-1)N}^3} \\ \frac{1}{r_{1N}^3} & \frac{1}{r_{2N}^3} & \frac{1}{r_{3N}^3} & \cdots & \frac{1}{r_{(N-2)N}^3} & \frac{1}{r_{(N-1)N}^3} & 0 \end{bmatrix}.$$

Let the masses of the system be stored in a matrix

$$M = \begin{bmatrix} m_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & m_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & m_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & m_{N-2} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & m_{N-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & m_N \end{bmatrix};$$

taking the product of S and M then yields the matrix

$$P = SM = \begin{bmatrix} 0 & \frac{m_1}{r_{12}^3} & \frac{m_1}{r_{13}^3} & \cdots & \frac{m_1}{r_{1(N-2)}^3} & \frac{m_1}{r_{1(N-1)}^3} & \frac{m_1}{r_{1N}^3} \\ \frac{m_2}{r_{12}^3} & 0 & \frac{m_2}{r_{23}^3} & \cdots & \frac{m_2}{r_{2(N-2)}^3} & \frac{m_2}{r_{2(N-1)}^3} & \frac{m_2}{r_{2N}^3} \\ \frac{m_3}{r_{13}^3} & \frac{m_3}{r_{23}^3} & 0 & \cdots & \frac{m_3}{r_{3(N-2)}^3} & \frac{m_3}{r_{3(N-1)}^3} & \frac{m_3}{r_{(N-1)N}^3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{m_{N-2}}{r_{1(N-2)}^3} & \frac{m_{N-2}}{r_{2(N-2)}^3} & \frac{m_{N-2}}{r_{3(N-2)}^3} & \cdots & 0 & \frac{m_{N-2}}{r_{(N-2)(N-1)}^3} & \frac{m_{N-2}}{r_{(N-2)N}^3} \\ \frac{m_{N-1}}{r_{1(N-1)}^3} & \frac{m_{N-1}}{r_{2(N-1)}^3} & \frac{m_{N-1}}{r_{3(N-1)}^3} & \cdots & \frac{m_{N-1}}{r_{(N-2)(N-1)}^3} & 0 & \frac{m_{N-1}}{r_{(N-1)N}^3} \\ \frac{m_N}{r_{1N}^3} & \frac{m_N}{r_{2N}^3} & \frac{m_N}{r_{3N}^3} & \cdots & \frac{m_N}{r_{(N-2)N}^3} & \frac{m_N}{r_{(N-1)N}^3} & 0 \end{bmatrix}.$$

The diagonal terms of this matrix are all zero; the distance between an object and itself is zero, so these contributions must be ignored.

For a given index i, the acceleration is simply the i^{th} row of the matrix S multiplied by the distances between i and each other object. Since the three dimensions of space the objects are simulated in are not coupled in the equations of motion beyond the distance term, each spatial dimension can be calculated separately; the x case will be handled here, but the y and z cases follow identically.

Let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-2} \\ x_{N-1} \\ x_N \end{bmatrix}$$

denote the x-positions of each object; furthermore, let

$$\mathbf{j} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Define

$$X = \mathbf{x}\mathbf{j}^{\mathrm{T}} = \begin{bmatrix} x_1 & x_1 & x_1 & \cdots & x_1 & x_1 & x_1 \\ x_2 & x_2 & x_2 & \cdots & x_2 & x_2 & x_2 \\ x_3 & x_3 & x_3 & \cdots & x_3 & x_3 & x)3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_{N-2} & x_{N-2} & x_{N-2} & \cdots & x_{N-2} & x_{N-2} & x_{N-2} \\ x_{N-1} & x_{N-1} & x_{N-1} & \cdots & x_{N-1} & x_{N-1} & x_{N-1} \\ x_N & x_N & x_N & \cdots & x_N & x_N & x_N \end{bmatrix},$$

then

$$D_x = X - X^{\mathrm{T}} = \begin{bmatrix} 0 & x_1 - x_2 & x_1 - x_3 & \cdots & x_1 - x_{N-2} & x_1 - x_{N-1} & x_1 - x_N \\ x_2 - x_1 & 0 & x_2 - x_3 & \cdots & x_2 - x_{N-2} & x_2 - x_{N-1} & x_2 - x_N \\ x_3 - x_1 & x_3 - x_2 & 0 & \cdots & x_3 - x_{N-2} & x_3 - x_{N-1} & x_3 - x_N \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_{N-2} - x_1 & x_{N-2} - x_2 & x_{N-2} - x_3 & \cdots & 0 & x_{N-2} - x_{N-1} & x_{N-2} - x_N \\ x_{N-1} - x_1 & x_{N-1} - x_2 & x_{N-1} - x_3 & \cdots & x_{N-1} - x_{N-2} & 0 & x_{N-1} - x_N \\ x_N - x_1 & x_N - x_2 & x_N - x_3 & \cdots & x_N - x_{N-2} & x_N - x_{N-1} & 0 \end{bmatrix}.$$

Finally, the x-acceleration of each object is given by the diagonal terms of the product

$$\begin{aligned} A_x &= SD_x \\ &= \left(R + R^{\mathrm{T}}\right) M \left(\mathbf{x}\mathbf{j}^{\mathrm{T}} - \mathbf{j}\mathbf{x}^{\mathrm{T}}\right). \end{aligned}$$

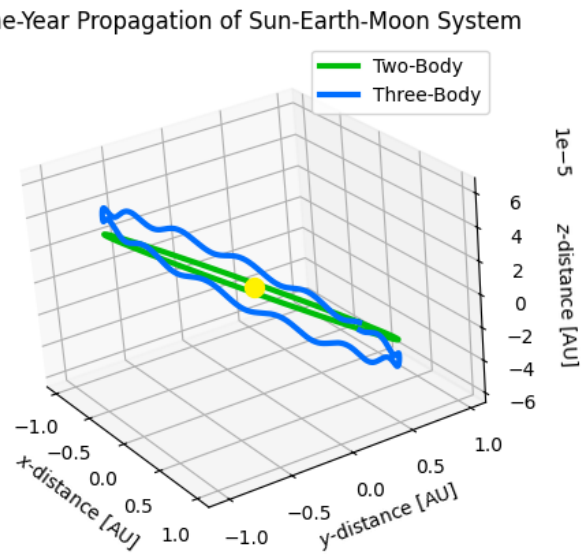

Figure 9: Comparison of two-body (Earth, Sun) and three-body (Earth, Sun, Moon) dynamics for a one-year Earth orbit around the Sun.

When calculating the accelerations of each object numerically, the off-diagonal terms do not need to be computed, saving significant computational time. This implementation reduces the need for nested loops or summations in the code while leveraging the vectorization afforded by modern numerical computing packages such as Python's NumPy.

An example of the fourth-order Runge-Kutta integration scheme is shown in Figure 9, where the motion of the Earth around the Sun (not to scale) was simulated both with and without three-body effects due to the Moon (not illustrated). The simulation was conducted over a period $T = 3.154 \times 10^7$ s (approximately equal to the Earth's orbital period) with timesteps of $h = 1 \times 10^4$ s.[2] A notable difference is visible between the two-body dynamics, which yield an ellipse with the Sun at one focus, and the three-body dynamics, which yield a much more complicated trajectory with no closed-form equation in any coordinate system (the two-body dynamics can be succinctly represented in terms of five fixed geometric parameters and an angle, known as orbital elements [9]). Note, however, the scale of the deviations; while the orbit is on the scale of 1 AU (where 1 AU is defined as the mean Earth-Sun distance), the perturbations are on the order of 1×10^{-5} AU.

2.5 Stability and Sensitivity Analysis

We focus now on the stability and sensitivity of our methods. For our truth data, we utilize the JPL Horizons database, a catalog of all the major planets and many small bodies propagated in the range of hundreds to thousands of years. We investigate most closely the fourth-order Runge-Kutta method and the Stormer-Verlet method, as these were shown in Figures 6, 7, and 8 to have the best spatial and energy performance. We also elect to use the Moon as the object of interest, since its low mass and proximity to the Earth make it particularly sensitive to gravitational perturbations.

In a direct comparison between the two methods (Figure 10), we see that the error between the two methods is originally similar in magnitude, with the Runge-Kutta method even beating out the Stormer-Verlet method between 20 and 40 years of propagation time; however, shortly after the 40-year mark, the Runge-Kutta

[2]See Section 4.2 for a discussion of the initial conditions of this and other figures.

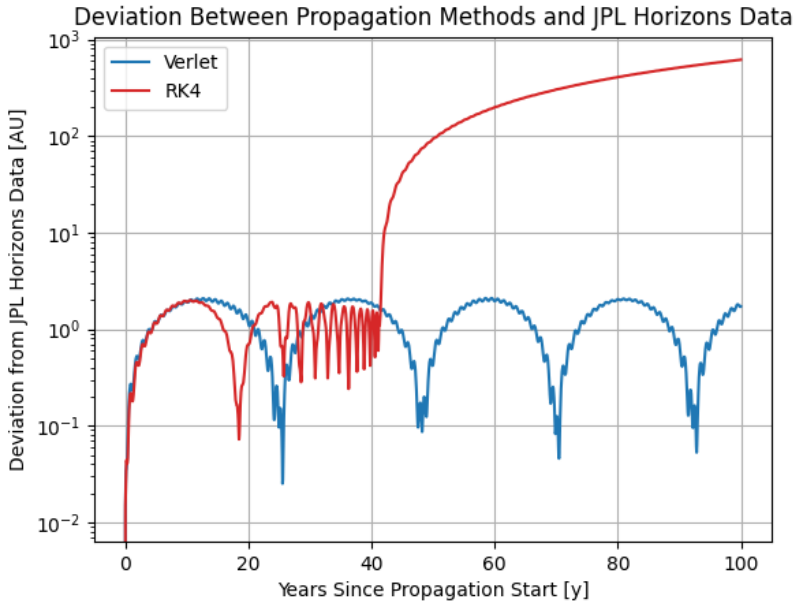

Figure 10: Deviation between fourth-order Runge-Kutta and Stormer-Verlet methods with JPL Horizons data using a timespan of 100 years with a timestep of 20 days.

method diverges rapidly from the truth data, while the error in the Stormer-Verlet method remains bounded (with a maximum of approximately 2 AU).

A similar trend is seen in a comparison between a perturbed and unperturbed system. In the case visualized in Figure 11, the Sun was perturbed by just one kilometer along every axis at $t = 0$; while such a perturbation had a near-negligible effect on the position of the Moon after 100 years using the Stormer-Verlet method (approximately 1×10^{-4} AU, or slightly less than 4% of the Earth-Moon distance); the error in the Runge-Kutta method was nearly eight orders of magnitude greater [9]!

These tests illustrate the importance of utilizing an energy-conserving method in long-time propagations of the N-body problem. Since, in the ideal case (*e.g.*, Newtonian mechanics, point masses, no outside forces), the N-body problem conserves energy, utilizing a timestepping method that also conserves energy serves to much better model the behavior of the system.

2.6 Periodic Planar Orbits

Now that we have studied, built, and tested these tools, we can have some fun! Xiaoming Li and Shijun Liao have documented the initial conditions required for 152 types of periodic planar orbits [8]. We can use these conditions to test our methods as we attempt recreate these orbits. The conditions specified by Li and Liao are as follows:

$$\begin{aligned} \text{All bodies have a mass of } m &= 1 \\ \text{Gravitational constant } G &= 1 \\ \mathbf{r}_1(t=0) &= [-1, 0] = -\mathbf{r}_2(t=0) \\ \mathbf{r}_3(t=0) &= [0, 0], \\ \dot{\mathbf{r}}_1(t=0) &= [v_1, v_2] = \dot{\mathbf{r}}_2(t=0) \\ \dot{\mathbf{r}}_3(t=0) &= [-2v_1, -2v_2] \end{aligned}$$

From their list we chose conditions **II.B-1** with the following values:

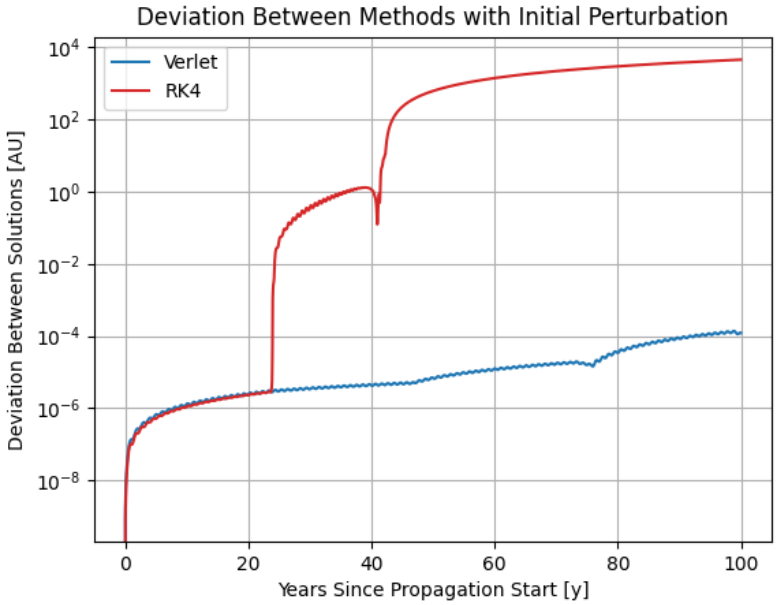

Figure 11: Deviation between fourth-order Runge-Kutta and Stormer-Verlet methods with and without a solar perturbation using a timespan of 100 years with a timestep of 20 days.

$$v_1 = 0.3962186234$$
$$v_2 = 0.5086826315$$
$$\text{Period } T = 96.4358796119$$

As we have shown above, our implementation of the Stormer-Verlet method significantly outperforms fourth-order Runge-Kutta in these problems, so it is what we will apply to this problem. The results can be found in Figure 12 with an animated version here.

This orbit is stable over the course of our simulation. However, the paths do not overlap perfectly, so we expect the paths to deviate and drift apart as the simulation time is increased.

We previously looked at sensitivity to perturbations; what would happen if these initial conditions were perturbed a small amount? We randomly added perturbations with a maximum value of 5% (using Numpy Random Uniform and Random Seed 40) to the initial positions and velocities. The results are plotted in Figure 13 with another animation here.

This makes it clear that even a small change in these initial conditions has a dramatic change on the positions after a short period (as also seen in Figure 11). This only reinforces the chaotic nature discussed in Section 1.2.

Finally, this perturbation experiment exposes one of the weaknesses of our implementation of the Verlet method (as well as our Euler and Runge-Kutta implementations): it takes fixed steps through time, causing instabilities as the bodies get closer to each other. This caused many of the other periodic orbits given by Li and Liao where the bodies were closer to fail. Future work for this project should include the implementation of variable time stepping to our method to prevent this behavior.

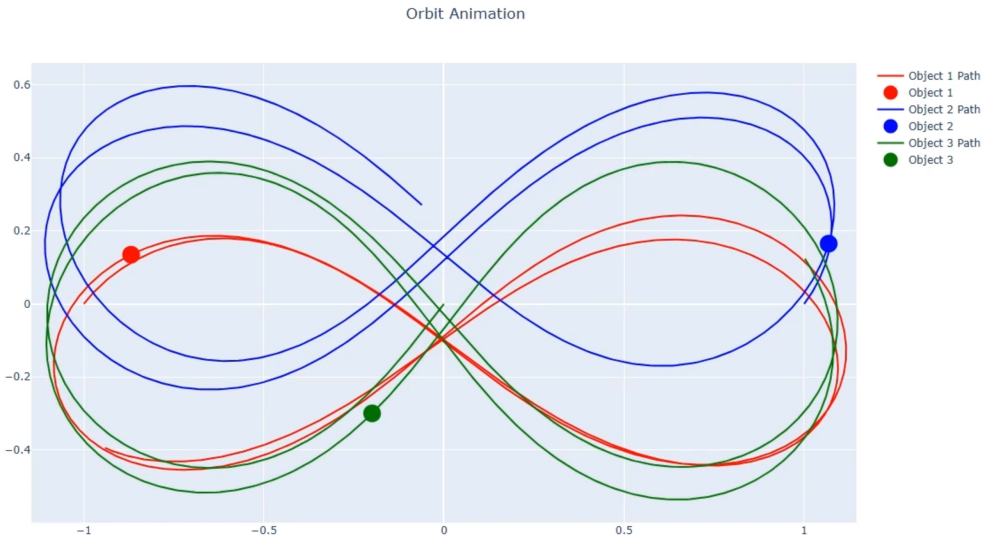

Figure 12: Periodic Planar Orbit **II.B-1**

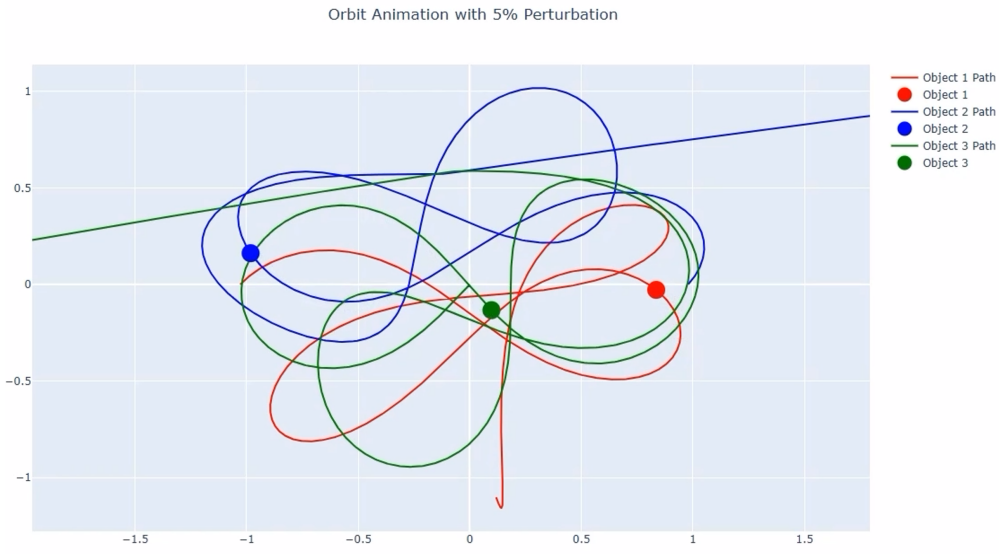

Figure 13: Perturbed Periodic Planar Orbit **II.B-1**

3 Conclusion

In this project, we study numerical methods for time stepping of the N-body problem and compare general-purpose timestepping methods to symplectic methods. The N-body problem is of significant practical importance, especially for orbital problems. We cover the basics of stiff and chaotic systems and show that they present challenges for timestepping. We then derive the N-body problem and provide an implementation as a linear system for computational efficiency. We also develop the Verlet integration scheme and show that it has a bounded energy error, while the Euler and Runge-Kutta methods accumulate unbounded energy error. Further, we show that the Verlet scheme performs better on the JPL Horizons data and also shows reduced sensitivity to perturbations in initial conditions.

References

[1] Sverre J. Aarseth, Christopheer A. Tout, and Rosemary A. Mardling. *The Cambridge N-Body Lectures*. Springer-Verland Berlin Heidelberg, 2008.

[2] Michael Brorson. *Symplectic Integrators*. From LibreTexts Differential Equations. 2022. URL: https://math.libretexts.org/Bookshelves/Differential_Equations/Numerically_Solving_Ordinary_Differential_Equations_(Brorson)/01%3A_Chapters/1.07%3A_Symplectic_integrators (visited on 04/14/2025).

[3] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. Cengage Learning, 2010.

[4] Steven Carlip. *General Relativity: A Concise Introduction*. Oxford University Press, 2019.

[5] Suresh Chandra, Mohit Kumar Sharma, and Monika Sharma. *Fundamentals of Mechanics*. Cambridge University Press India, 2014.

[6] Rafael Cubarsi. *Kinematics and Dynamics of Galactic Stellar Populations*. Cambridge Scholars Publishing, 2018.

[7] Donald Greenspan. *N-Body Problems and Models*. World Scientific Publishing, 2004.

[8] Xiaoming Li and Shijun Liao. *One hundred and fifty-two new families of Newtonian periodic planar three-body orbits*. Updated October 24, 2021. 2017. arXiv: 1705.00527v2. URL: https://arxiv.org/abs/1705.00527v2.

[9] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. 4th ed. Microcosm Press, 2013.

[10] Eric W. Weisstein. *Chaos*. From MathWorld–A Wolfram Web Resource. 2024. URL: https://mathworld.wolfram.com/Chaos.html (visited on 04/14/2025).

[11] Richard Wolfson. *Essential University Physics*. 2nd ed. Pearson Education, 2012.

4 Appendix A: Code and Data

4.1 Code

All figures in this report were created using code available at https://github.com/brycepfuetze/APPM-4610-stiff-ODEs. All code in this repository makes use of Python 3 and free and open-source packages; all data used is publicly available (see Section 4.2 and the `README.md` file in the repository for more details). The code can be run as-is (assuming the proper packages are installed and the data is downloaded).

4.2 Data

The data for this report were provided by the Jet Propulsion Laboratory (Figures 9, 10, and 11). All data used is publicly available through the JPL Horizons website at https://ssd.jpl.nasa.gov/horizons/.

Figures 9, 10, and 11 all make use of the Sun (10), Earth (399), and Moon (301) centers of mass referenced to the center of the Sun (@10). Initial conditions are all fetched at 2025-01-01, 00:00:00.00; for Figure 10, ephemerides are fetched with a frequency of $h = 20$ days through 2125-01-01, 00:00:00.00.