Department of Mathematics

250031-1 – Modelling Interacting Particle Systems in Science

The Three-Body Problem

Author:
Verena Alton

Summer Semester 2021

Contents

List of Figures

1 Introduction

Archie E. Roy states the many-body problem in his book *Orbital Motion* [Roy, 1988] as follows: "Given at any time the positions and velocities of three or more massive particles moving under their mutual gravitational forces, the masses also being known, calculate their positions and velocities for any other time."

As one of the oldest issues in classical mechanics, the many-body problem is of great importance for astronomical modelling – to understand the interactions and influences between celestial bodies. [Krishnaswami and Senapati, 2019] Already Newton was challenged by the problem, but especially in the near history, with the uprising of satellites and space flights, it gained significance. Furthermore, since only the special case of the two-body problem can be solved analytically, the invention of high-performance computers paved the way for a new access to the problem: numerical approximations. [Mikkola, 2020]

In this work, a system of three particles is considered and solved by four different numerical methods. To see the code, visit https://github.com/Entonia314/ModIntPartSys with ThreeBodyProblem_Version2.py as the file that is used for the latest dashboard. The dashboard is running on alton.pythonanywhere.com.

2 Theory

The first part of the work is an theoretical approach to the three-body problem, to analyse the properties and the behaviour of the model.

2.1 The Model

For the model of the three-body problem in two dimensions, the given values for the bodies b_1, b_2 and b_3 are respectively the masses m_1, m_2 and m_3, the initial positions $r_1(0) = (x_1, y_1)$, $r_2(0) = (x_2, y_2)$ and $r_3(0) = (x_3, y_3)$ as well as the initial velocities $r_1'(0) = v_1(0) = (u_1, w_1)$, $r_2'(0) = v_2(0) = (u_2, w_2)$ and $r_3'(0) = v_3(0) = (u_3, w_3)$. The notation $r_i'(t)$ means in this context the derivative to the time t (i.e. $\frac{dr_i(t)}{dt}$).

With these values, the model for the three-body problem appears as

$$m_i r_i''(t) = m_i \frac{d^2 r_i(t)}{dt^2} = F_i \tag{1}$$

for $i \in \{1, 2, 3\}$, with

$$F_1 = -Gm_1m_2 \frac{r_1 - r_2}{|r_1 - r_2|^3} - Gm_1m_3 \frac{r_1 - r_3}{|r_1 - r_3|^3} \tag{2}$$

$$F_2 = -Gm_2m_1 \frac{r_2 - r_1}{|r_2 - r_1|^3} - Gm_2m_3 \frac{r_2 - r_3}{|r_2 - r_2|^3} \tag{3}$$

$$F_3 = -Gm_3m_1 \frac{r_3 - r_1}{|r_3 - r_1|^3} - Gm_3m_2 \frac{r_3 - r_2}{|r_3 - r_2|^3}. \tag{4}$$

The vector $F_i \in \mathbb{R}^2$ describes the force that affects the body b_i, $i \in \{1, 2, 3\}$. [Aarseth, 2003] Thus, six ordinary differential equations of second order have to be solved, which can be transformed to twelve differential equations of first order.

Particularly, (1) is a second order Newtonian equation: $F = mv'(t) = ma(t)$, with m the mass, v the velocity and a the acceleration. In this case, the force F is given as it is deviated from the masses and positions in (2)-(4). So, the model is of the type of conservative Newton dynamics, like the model of the harmonic oscillator that was discussed in the course.

2.2 Conservation of Energy

The forces in the model are conservative, hence there is a potential ψ with

$$F(r) = - \bigtriangledown \psi(r), \qquad r = (r_1, r_2, r_3) \tag{5}$$

As shown in the course, the mechanical energy of the system appears as

$$E(r) = \underbrace{\frac{1}{2}\sum_{i=1}^{3} m_i |v_i|^2}_{\text{kinetic energy}} + \underbrace{\psi(r)}_{\text{potential energy}} .$$

With $\psi(r) = -G(\frac{m_1 m_2}{|r_1 - r_2|^3} + \frac{m_1 m_3}{|r_1 - r_3|^3} + \frac{m_2 m_3}{|r_2 - r_3|^3})$, the mechanical energy is

$$E = \frac{1}{2}\sum_{i=1}^{3} m_i |v_i|^2 - \sum_{i=1}^{3}\sum_{j>i}^{3} \frac{G m_i m_j}{|r_i - r_j|} = T + V, \tag{6}$$

where V is the potential energy and T the kinetic energy. Now, if the equations (1)-(4) are written as

$$\frac{d^2 r_1}{dt^2} = m_2 \frac{r_2 - r_1}{|r_2 - r_1|^3} + m_3 \frac{r_3 - r_1}{|r_3 - r_1|^3} \tag{7}$$

$$\frac{d^2 r_2}{dt^2} = m_1 \frac{r_1 - r_2}{|r_1 - r_2|^3} + m_3 \frac{r_3 - r_2}{|r_3 - r_2|^3} \tag{8}$$

$$\frac{d^2 r_3}{dt^2} = m_1 \frac{r_3 - r_1}{|r_2 - r_1|^3} + m_2 \frac{r_2 - r_3}{|r_2 - r_3|^3} \tag{9}$$

and the potential energy V is differentiated with respect to r_i for $i \in 1, 2, 3$, one concludes

$$m_1 \frac{d^2 r_1}{dt^2} = -\frac{dV}{dr_1} \tag{10}$$

$$m_2 \frac{d^2 r_2}{dt^2} = -\frac{dV}{dr_2} \tag{11}$$

$$m_3 \frac{d^2 r_3}{dt^2} = -\frac{dV}{dr_3}. \tag{12}$$

Multiplying (10)-(12) each with $v_i = \frac{dr_i}{dt}$ and summing the equations leads to

$$\sum_{i=1}^{3} m_i \frac{d^2 r_i}{dr^2} \frac{dr_i}{dt} = -\frac{dV}{dt} \tag{13}$$

Now, an integration with respect to t gives

$$\left[\text{Auxiliary calculation:} \int \frac{d^2 r_i}{dt^2}\frac{dr_i}{dt} dt = \frac{dr_i}{dt}\frac{dr_i}{dt} - \int \frac{d^2 r_i}{dt^2}\frac{dr_i}{dt} dt \iff 2\int \frac{d^2 r_i}{dt^2}\frac{dr_i}{dt} dt = \frac{dr_i}{dt}\frac{dr_i}{dt} = v_i^2\right]$$

$$\frac{1}{2}\sum_{i=1}^{3} m_i v_i^2 = -V + C \tag{14}$$

with a constant C. But the left side of (14) is exactly the kinetic energy T, hence

$$T = -V + C \iff T + V = C \iff E = C. \tag{15}$$

This shows that the energy of the system is a constant and therefore does not change over time, so it is conserved.

The behaviour of the particles in this system is similar to a harmonic oscillator. The solutions oscillate around the centre of mass of the three particles and do not reach an equilibrium point, because the energy is conserved. Depending on the starting values of the particles, the centre of mass is moving in a straight line or remains at rest. [Grützelius, 2004]

3 Implementation

In Python, four different numerical methods were implemented. The algorithms of the methods were mainly taken from the lecture notes of the course „Numerical methods for differential equations“ from Norbert Mauser and Lukas Exl [Mauser et al., 2021]. The functionalities of the methods are similar: with a given interval of the time $[t_0, t_1]$ and a step size h, the interval is divided into $n = \frac{t_1 - t_0}{h}$ steps. Now, with this discretised interval, for every step $k \in \{0, 1, ..., n\}$ a numerical integration is done, which uses the solution of the step before. A smaller step size leads to more accurate results, but needs more computing capacity. Therefore, a good medium should be used.

The derivation of the explicit Euler's method gives an example:
The definition of the derivative of $v(t)$ is

$$v'(t) = \lim_{h \to 0} \frac{v(t) - v(t+h)}{h}.$$

Now, the step size h is fixed and the formula can be written as

$$\begin{gathered} v'(t) = \frac{v(t) - v(t+h)}{h} + \theta(h^2) \\ \iff \\ v(t+h) = v(t) + h \cdot v'(t) + \theta(h^2). \end{gathered}$$

The derivatives $v_i'(t) = a_i(t)$ are known as the given functions of the force multiplied with the respective masses: $F_i m_i$, for $i \in \{1, 2, 3\}$. Hence, three functions $f_i(r_{k+1}) = m_i \cdot F_i(r_k)$ are defined and used for the calculation of each next step. The integration from v_k to r_k is done in the same way.

3.1 Explicit Euler Method

The first method is the explicit Euler method, as shown above:

$$\begin{aligned} v_{k+1} &= v_k + hf(r_k) \\ r_{k+1} &= r_k + hv_k. \end{aligned}$$

To improve the method without significant increase of computational capacity, the second term of the Taylor series can be used, since the second derivative is already given.

$$\begin{aligned} v_{k+1} &= v_k + hf(r_k) \\ r_{k+1} &= r_k + hv_k + \frac{h^2}{2} f(r_k) \end{aligned}$$

3.2 Implicit Euler Method

The second method is the implicit Euler's method. *Implicit*, because the next step r_{k+1} is also in the formula of its calculation, hence an equation has to be solved in every time step to get r_{k+1}.

$$\begin{aligned} v_{k+1} &= v_k + hf(r_{k+1}) \\ r_{k+1} &= r_k + hv_k \end{aligned}$$

To solve a this equation numerically, a Newton-Raphson-Iteration was implemented. There, the iteration steps are given as $x_{l+1} = x_l + \frac{g(x_l)}{g'(x_l)}$. The derivative was calculated by hand, to spare computing capacity. Anyways, with suboptimal starting values or a too big step size it can happen, that the Newton-Raphson-Iteration does not converge and therefore the method fails.

Again, the second term of the Taylor series can be used to achieve better results.

$$\begin{aligned} v_{k+1} &= v_k + hf(r_{k+1}) \\ r_{k+1} &= r_k + hv_k + \frac{h^2}{2} f(r_k) \end{aligned}$$

3.3 Heun Method

The Heun Method uses two terms of the Taylor series to calculate a predictive $r_{k+1}^{(p)}$ and evaluates it to get the final guess of r_{k+1}. The approximation is not rectangular as in explicit Euler's method but triangular.

$$\begin{aligned} v_{k+1}^{(p)} &= v_k + h \cdot f(r_k) \\ r_{k+1}^{(p)} &= r_k + h \cdot v_k + \frac{h^2}{h} \cdot f(r_k) \\ v_{k+1} &= \frac{1}{2} v_k + \frac{1}{2}(v_{k+1}^{(p)} + h \cdot f(r_{k+1}^{(p)})) \\ r_{k+1} &= \frac{1}{2} r_k + \frac{1}{2}(r_{k+1}^{(p)} + h \cdot v_k + \frac{h^2}{h} \cdot f(r_k)) \end{aligned}$$

The original Heun Method uses explicit Euler for the predictive step. Here, the „improved Euler method" with the second derivative is used.

3.4 Runge-Kutta Method

The effort of the Runge-Kutta method is to be a one-step method with higher convergence order than the other ones. To achieve this, there are intermediate steps at every time step k, then the integral is replaced by a quadrature which is evaluated at these intermediate steps:

$$\begin{aligned} v_{k+1} &= v_k + h \sum_{j=1}^{\nu} b_j f(\xi_j) \\ r_{k+1} &= r_k + h \sum_{j=1}^{\nu} b_j f(\rho_j). \end{aligned}$$

The nodes are calculated as

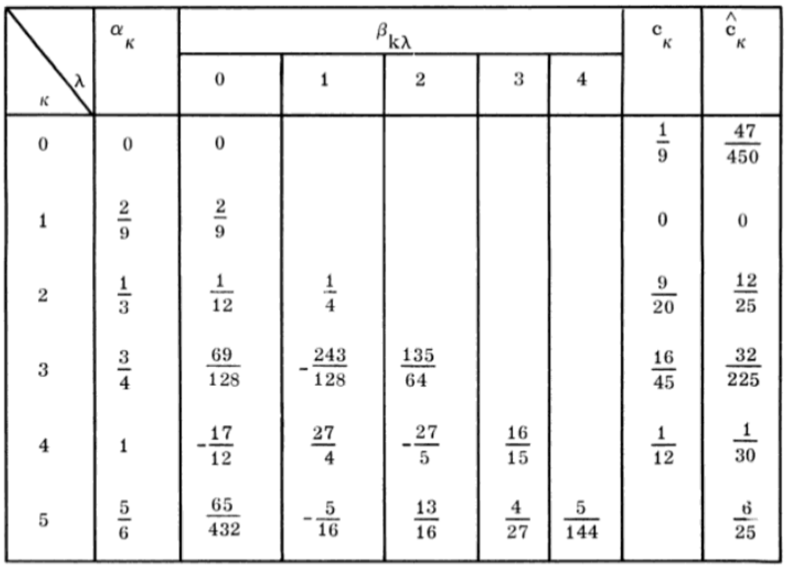
$$\begin{aligned}
\xi_1 &= r_k \\
\xi_2 &= r_k + a_{21} h f(\xi_1) \\
\xi_3 &= r_k + a_{31} h f(\xi_1) + a_{32} h f(\xi_2) \\
&\vdots \\
\xi_\nu &= r_k + h \sum_{j=1}^{\nu-1} a_{\nu j} f(\xi_j)
\end{aligned}$$

for v_k and as

$$\begin{aligned}
\rho_1 &= v_k \\
\rho_2 &= v_k + a_{21} h f(\rho_1) \\
\rho_3 &= v_k + a_{31} h f(\rho_1) + a_{32} h f(\rho_2) \\
&\vdots \\
\rho_\nu &= v_k + h \sum_{j=1}^{\nu-1} a_{\nu j} f(\rho_j)
\end{aligned}$$

for r_k.

For the weights b_i and a_{ij}, the following table from [Fehlberg, 1969] was chosen, where $b_i = \hat{c}_\kappa$ and $a_{ij} = \beta_{\kappa\lambda}$:



$\kappa \backslash \lambda$	α_κ	$\beta_{k\lambda}$					c_κ	$\hat{c}_\kappa$
		0	1	2	3	4		
0	0	0					$\frac{1}{9}$	$\frac{47}{450}$
1	$\frac{2}{9}$	$\frac{2}{9}$					0	0
2	$\frac{1}{3}$	$\frac{1}{12}$	$\frac{1}{4}$				$\frac{9}{20}$	$\frac{12}{25}$
3	$\frac{3}{4}$	$\frac{69}{128}$	$-\frac{243}{128}$	$\frac{135}{64}$			$\frac{16}{45}$	$\frac{32}{225}$
4	1	$-\frac{17}{12}$	$\frac{27}{4}$	$-\frac{27}{5}$	$\frac{16}{15}$		$\frac{1}{12}$	$\frac{1}{30}$
5	$\frac{5}{6}$	$\frac{65}{432}$	$-\frac{5}{16}$	$\frac{13}{16}$	$\frac{4}{27}$	$\frac{5}{144}$		$\frac{6}{25}$

Figure 1: Table of parameters („Butcher Tableau“) for the Runge-Kutta-Fehlberg Method from [Fehlberg, 1969, page 12]

The weights c_κ are used to calculate a second quadrature with the same nodes but these different weights, so the difference between both quadratures gives an error estimation. This is also implemented in the code of this project to introduce an adaptive step size. The method with adaptive step size is called *Runge-Kutta-Fehlberg* or *embedded Runge-Kutta*. More about the adaptive step size is depicted in the section „Adaptive Step Size“ below.

3.5 Dashboard

4 Simulations

Two different kinds of simulations were realised. One with simplified parameters and one with real values of astronomical objects.

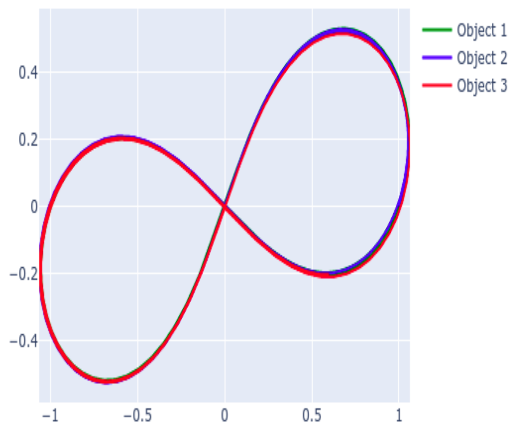
4.1 Simulations with simplified Parameters

To reduce the problem to the basics, the masses of the objects are all assumed as 1 as well as the gravitational constant.

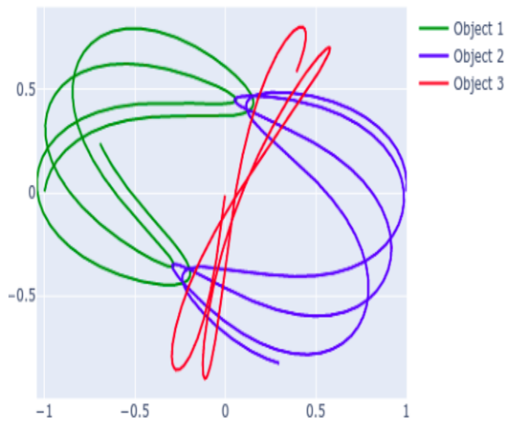
$$\begin{aligned} m_1 = m_2 = m_3 &= 1 \\ G &= 1 \end{aligned}$$

With these assumptions, starting values for relatively stable orbits can be found, for example in [Institute of Physics, 2021]. With a small enough step size, nearly all methods produce stable orbits for seven different starting values that were implemented. Of course, it is not very practical to use always a very small step size, because of limited computing capacity.

However, there is quite a difference between the stability of the scenarios. With a step size of 0.01, every method generates stable orbits of the „infinity symbol“. On the contrary, the „butterfly“ is chaotic for every method if the step size is bigger than 0.0001. Even with this really small step size, only Heun method is convergent.



(a) Runge-Kutta-Fehlberg method with starting values of „scenario 3: infinity symbol“



(b) Heun method with starting values of „scenario 7: butterfly“

Figure 2: Simulations with simplified parameters

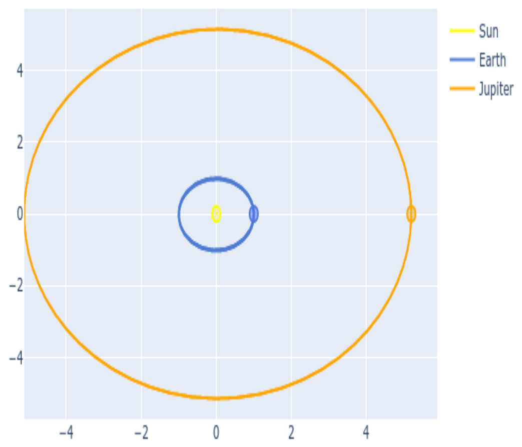
4.2 Simulation of the Sun System

Some of the astronomical bodies in the sun system were implemented, with their masses, their distance to the sun as starting position and average orbital speed as starting velocity.

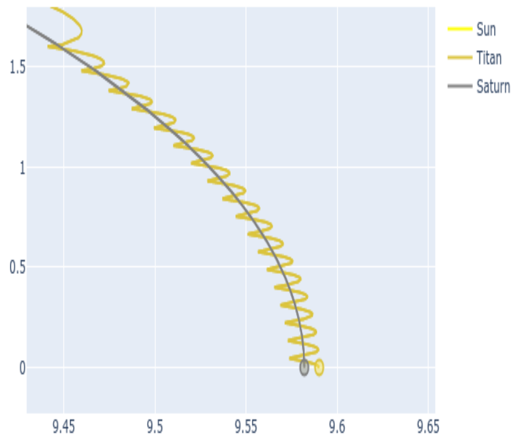
The teams with the sun and two planets are good examples for the restricted three-body problem, were two of the three bodies are nearly independent of each other, as their masses are too small or they are to far away from each other to affect the other one.

The systems of Sun, Earth and Moon or Sun, Saturn and Titan are not restricted, since the moons are influenced by the sun their planet. One can see in the simulation that the moons are actually orbiting their planet, only titan's orbit gets unstable with the time.

Unfortunately, the international space station (ISS) does not work in the simulation, it leaves its orbit immediately. Since also a very small step size does not lead to better results, it can be assumed that the problem is with the starting values. In fact, average values were taken and not the exact positions and velocities at one exact date for all bodies.



(a) Sun, Earth and Jupiter (12 Years)



(b) Sun, Saturn and Titan, zoomed in to see Titan's orbit

Figure 3: Simulation of the Sun System

5 Conclusion

5.1 Strengths and Weaknesses of the Model

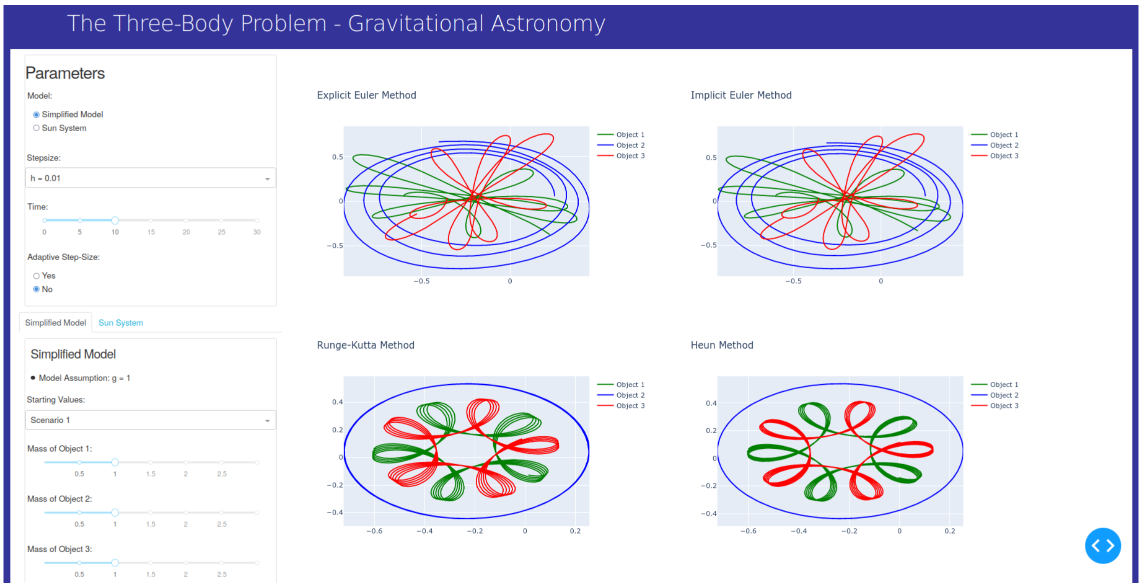
The fact that there is no friction or other kind of damping considered in the model makes sense, since it refers to astronomical objects in space, so the surrounding is void of air. Altogether, the model appears quite simple, without any random interferences of asteroids or other incidents imaginable in space. This appears reasonable, as it is not a very likely for an asteroid or satellite to have enough mass to influence the orbit of a planet or moon. Anyways, the system is chaotic and only small changes to the starting values of one of the objects can lead to crashes or leaving of orbits.

These unfortunate outcomes can also be observed when trying to approximate the three-body problem numerically for theoretical stable orbits. Since the time interval is discretisised, the gaps between the calculation steps can easily be too big, so that the results do not display the physical nature anymore. For example, if a body gets really fast, the time steps should be scaled down, otherwise the body could be very far away at the next step and fall out of orbit because the moment when its orbit would be corrected by the gravition of the other bodies was just overleaped. The same principle could lead to a crash of two bodies, namely when because of numerical errors a body does not shoot fast enough past another but falls out of orbit and onto the other body, what leads to a singularity as the positions of two bodies are the same.

This kind of false results could be prevented if the numerical error was measured at every step to scale down the step size if necessary and repeat the step until the error is small enough. Further considerations of this issue in the next section.

5.2 Error Analysis of the numerical Methods

A Dashboard was created with Plotly's Dash to show the outcomes of the methods simultaneously for comparison. Since there is no analytical solution, there is no way to determine the exact errors of the numerical methods.


The Three-Body Problem - Gravitational Astronomy

Parameters

Model: [✓ Simplified Model] Sun System

Stepsize: h = 0.01

Time:

Adaptive Step-Size: Yes [✓ No]

Simplified Model | Sun System

Simplified Model

- Model Assumption: g = 1

Starting Values: Scenario 1

Mass of Object 1:

Mass of Object 2:

Mass of Object 3:

Figure 4: Dashboard

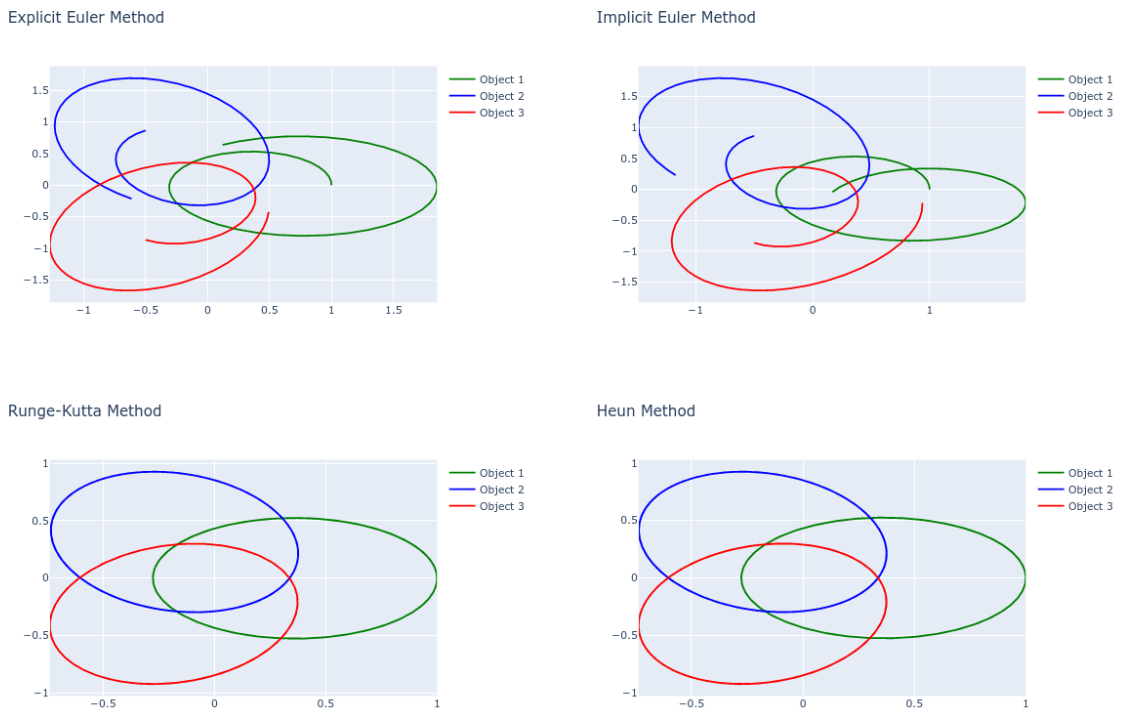


Figure 5: Comparison of methods (simplified Model)

From the visual comparison, the results from both Euler methods appear similarly accurate, and Runge-Kutta and Heun seem similar, whereat Runge-Kutta and Heun are significantly better than the other two. Heun Method achieves the best results, according to the visual appraisal.

5.2.1 Error Estimation

Multiplying (7)-(9) with the respective m_i and summing them all together gives

$$\sum_{i=1}^{3} m_i \frac{d^2 r_i}{dt^2} = 0.$$

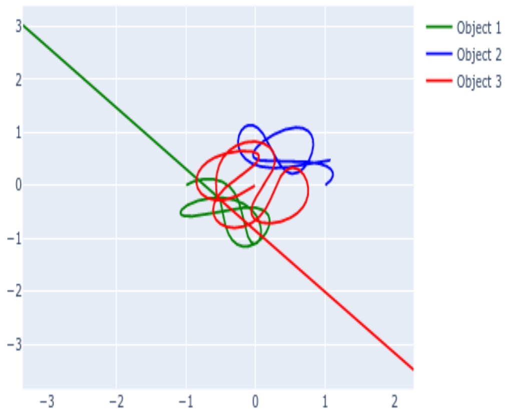
To estimate an error, the left term can be calculated at every step and checked how far away the result is from the expected zero. This is used for explicit and implicit Euler as well as for Heun method, while Runge-Kutta uses a second quadrature to estimate an error (see chapter „Runge-Kutta Method“ for details).

5.2.2 Adaptive Step Size

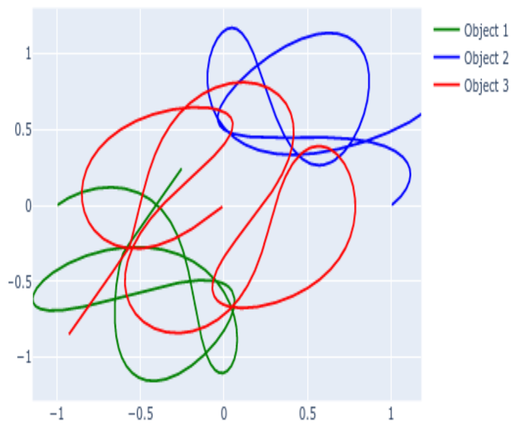
With an error estimation, not only a reference value to compare the accuracy of the methods is given, but also a possibility to check at every step, if the numerical results agree with the theoretical ones. If the error is too big, the step size can be scaled down and the step repeated, until the error is small enough. On the other hand, if the estimated error is already very small, the step size can be scaled up to spare computing capacity. This algorithm for an adaptive step size was implemented for all of the methods. To determine a realistic error tolerance, the maximum error of each method was considered.

Furthermore, the step size is only allowed in a certain interval, to prevent it from getting too small and therefore needing too much computing capacity. In this case, it was decided that $h \in [\frac{h_{start}}{256}, h_{start}]$. While the step size is just halved or doubled in the Euler methods and Heun method, Runge-Kutta-Fehlberg embeds the error estimator as well as the error tolerance into the new step size if it is scaled down: $h_{new} = 0.9 \cdot h \cdot |\frac{\epsilon}{TE}|^{\frac{1}{5}}$. To scale it up, the old step size is just multiplied by 1.1 since the additional term leads in this case to an unbalanced result.

Not only crashes and leaving of orbits can be prevented with adaptive step size, but it also leads to a way better convergence rate of the Newton-Raphson-Iteration in the implicit Euler method, because the initial guess for the iteration is more accurate.



(a) Without adaptive step size: „Crash“



(b) With adaptive step size

Figure 6: Adaptive Step Size – Heun Method

5.3 Further Ideas

The numerical error could be estimated in other ways, for example by calculating the energy of the system with the starting values and then calculating it at every step and looking at the difference.

It was shown above that the total energy of the system is conserved, so there should be no difference in theory and therefore, it would be an estimator for the numerical error. This was also tried in the implementation of the adaptive step size, but the errors were unrealistically big and unbalanced, hence the other estimator worked better.

To model the sun system realistically also for small bodies like the ISS, the starting values would have to be researched very detailedly, because the littlest changes lead to chaotic behaviour.

References

Aarseth, Sverre J (2003). *Gravitational N-body simulations: tools and algorithms*. Cambridge University Press.

Fehlberg, Erwin (1969). *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*. Vol. 315. National aeronautics and space administration.

Grützelius, Joakim H. (2004). *The three body problem*. Karlstad University.

Institute of Physics, Belgrade (2021). *Three-body Gallery*. URL: http://three-body.ipb.ac.rs/ (visited on 2nd Aug. 2021).

Krishnaswami, Govind S and Himalaya Senapati (2019). 'An introduction to the classical three-body problem'. In: *Resonance* 24.1, pp. 87–114.

Mauser, Norbert J., Lukas Exl and Peter Stimming (2021). *Numerische Methoden für Differentialgleichungen*. University of Vienna.

Mikkola, Seppo (2020). *Gravitational Few-Body Dynamics: A Numerical Approach*. Cambridge University Press.

Roy, Archie E (1988). 'Orbital motion'. In: *Bristol, England, Adam Hilger, 1988, 544*.