# Kepler Project

Thomas Czuba*

(Dated: August 24, 2023)

The equations of motion for the $N$-body Keplerian problem are written. The objective of these notes is to define a proper framework to begin implementing an oriented-object Python code.

## CONTENTS



## I. REMINDERS

Units used in this work are scaled to the Keplerian problem : time is in years (yr), masses are in solar masses ($M_\odot$) distances are in astronomical units (AU) and angles are given in radians rad. Their values in SI units are recalled in Table I.

| Units | Values in SI units |
|---|---|
| 1 AU | $1.496 \times 10^{11}$ m |
| 1 $M_\odot$ | $1.989 \times 10^{30}$ kg |
| 1 yr | $3.156 \times 10^{7}$ s |

TABLE I: Units used in this work and how they translate to the International System of Units.

### A. The two-body problem

Let us have two point-like objects of respective masses $M_1$ and $M_2$ with initial positions and velocities $\mathbf{r}_1 = (r_1, \theta_1)$, $\mathbf{v}_1 = (\dot{r}_1, \dot{\theta}_1)$, and $\mathbf{r}_2 = (r_2, \theta_2)$, $\mathbf{v}_2 = (\dot{r}_2, \dot{\theta}_2)$ in the central Keplerian potential:

$$V(\mathbf{r}_1, \mathbf{r}_2) = -G\frac{M_1 M_2}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{1}$$

with $G = 6.674310^{-11}$ m$^3$ kg$^{-1}$ s$^2 = 4\pi^2$ AU$^3$ $M_\odot^{-1}$ yr$^2$ the universal gravitational constant.

The equations of motion then writes:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_1 = -G\frac{M_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3}(\mathbf{r}_1 - \mathbf{r}_2), \tag{2}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_2 = -G\frac{M_1}{|\mathbf{r}_2 - \mathbf{r}_1|^3}(\mathbf{r}_2 - \mathbf{r}_1). \tag{3}$$

It is always possible to rewrite the problem in terms of the motion of the center of mass $X$ and the relative position vectors $\mathbf{x}$, defined by:

$$\mathbf{X} = \frac{M_1\mathbf{r}_1 + M_2\mathbf{r}_2}{M_1 + M_2}, \tag{4}$$

$$\mathbf{x} = \mathbf{r}_1 - \mathbf{r}_2. \tag{5}$$

This leads to the new set of equations of motion:

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{X} &= M_1\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_1 + M_2\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_2 \\ &= (M_1 + M_2)\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{X} \\ &= 0,\end{aligned} \tag{6}$$

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{x} &= \frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_1 - \frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{r}_2 \\ &= \mu\mathbf{F}_{12},\end{aligned} \tag{7}$$

with:

$$\mu = \frac{M_1 M_2}{M_1 + M_2}. \tag{8}$$

There are three conservations laws to keep in mind when working on the two-body problem:

- Conservation of total energy $E$:

$$\dot{E} = 0, \tag{9}$$

$$E = \frac{1}{2}M_1\mathbf{v}_1^2 + \frac{1}{2}M_2\mathbf{v}_2^2 - G\frac{M_1 M_2}{|\mathbf{r}_1 - \mathbf{r}_2|}, \tag{10}$$

- Conservation of total momentum $\mathbf{P}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P} = 0, \tag{11}$$

$$\mathbf{P} = \mathbf{p}_1 + \mathbf{p}_2 = M_1\mathbf{v}_1 + M_2\mathbf{v}_2, \tag{12}$$

- Conservation of total angular moment $\mathbf{L}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L} = 0, \tag{13}$$

$$\mathbf{L} = \mathbf{L}_1 + \mathbf{L}_2 = \mathbf{r}_1 \times \mathbf{p}_1 + \mathbf{r}_2 \times \mathbf{p}_2, \tag{14}$$

* thomas.czuba1@gmail.com

- Conservation of individual angular momentum $\mathbf{L}_i$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_i \;=\; = 0. \tag{15}$$

Starting from here we have everything we need to solve the equations of motion. The question arise as to what algorithm will be used for what equations. It is possible to work directly using (2), (3), as it allows for easier generalization when working on the python code.

Since we work here in 2D space it can be interesting to write the equations of motion on $r_i$:

$$\ddot{r}_i \;=\; -G\frac{M_j}{|\mathbf{r}_i - \mathbf{r}_j|^2} + \frac{J^2}{|\mathbf{r}_i - \mathbf{r}_j|^3}, \tag{16}$$

$i \neq j$. The equation on $\theta_1$ can be easily found using conservation laws:

$$J_i \;=\; r_i^2\dot{\theta}_i = r_i^2(t=0)\dot{\theta}_i(t=0) = r_{i,0}^2\dot{\theta}_{i,0}, \tag{17}$$

which means that $\theta_i$ can be easily deduced numerically from $r_i$.

## B. Algorithms used to solve the problem

Two possibilities have been considered in this work: i) a Runge-Kutta 2 procedure because of its robustness and simplicity and ii) a Verlet integrator because of its symplectic nature making the conservation of important quantities such as total energy to be conserved.

### 1. *Runge-Kutta 2 procedure*

As a reminder, having an equation of motion of the general form:

$$\dot{y} \;=\; f\left(y, t\right), \tag{18}$$

with $t$ the argument of the function (here interpreted as time).

We discretize $t$ over its interval of definition into a collection of $n$ points $t_i$ with $1 \leq i \leq n$, and write $\Delta t = \frac{t_{i+1}-t_i}{n}$ the step between $t_i$ and $t_{i+1}$. The Runge-Kutta 2 algorithm is a two-steps procedure:

1. Estimate the function at the mid-point between $t_i$ and $t_{i+1}$:

$$k_2 \;=\; y(t_i) + \frac{\Delta t}{2} f\left(y, t_i + \Delta t/2\right), \tag{19}$$

2. Estimate the function at time $t_{i+1}$ using the mid-point:

$$y(t_{i+1}) \;=\; y(t_i) + \Delta t k_2. \tag{20}$$

It is usually quite robust for a $\Delta t$ little enough in comparison to the characteristic timescales of the system, while not being tremendously straining on computer resources since it doesn't need to hold many elements in memory.

The derivatives are not of first but second order in the Kepler problem. A common workaround is to rewrite the problem in terms of a set of two first order equations: one on the position, and one on the velocity. Assuming the problem is not ill-defined (*i.e.*, well-defined boundary conditions and closed form), the most general second order differential equation takes the form:

$$\ddot{y} \;=\; f\left(\dot{y}, y, t\right). \tag{21}$$

We then define the vector $\mathbf{A}$ as:

$$\mathbf{A} \;=\; \begin{pmatrix} y \\ \dot{y} \end{pmatrix}, \tag{22}$$

and the velocity $v = \dot{y}$. It is possible to rewrite Eq. (21) as an ordinary first order differential equation coupled with $y$:

$$\dot{v} \;=\; f\left(v, y, t\right). \tag{23}$$

The equation of motion of $\mathbf{A}$ appear to be a set of two coupled first order ordinary differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{A} = \begin{pmatrix} \dot{y} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \\ f\left(v, y, t\right) \end{pmatrix}, \tag{24}$$

that can be easily solved. Note that it is very akin in principle as to use the equations derived from the Hamiltonian formulation of Classical Mechanics.

Using Eqs. (2) and (3), the equations to be solved in the algorithm will take the form of the set of four vector equations of motion. The first two of the system are given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r}_1 \;=\; \mathbf{v}_1, \tag{25}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_1 \;=\; -G\frac{M_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3}\left(\mathbf{r}_1 - \mathbf{r}_2\right). \tag{26}$$

It is easy to find the two other since the labels 1 and 2 play symmetric roles. Since we work on a column vector, the midpoint will also be given by a vector $\mathbf{k}_2$.

### 2. *Verlet integrator*

Methods have been devised in the literature that conserve the symplectic nature of Hamiltonian mechanics, allowing for a much more robust conservation of important quantities such as total angular momentum and total mechanical energy. An example of such method is the Störmer-Velvet scheme, based around Taylor expansions. Denoting the right-hand side of Eq. (2) as $\mathbf{a}(t)$, this gives:

$$\mathbf{r}(t+\mathrm{d}t) \approx \mathbf{r}(t)+\mathbf{v}(t)\mathrm{d}t+\frac{1}{2}\mathbf{a}(t)\mathrm{d}t^2+\dots \quad (27)$$

$$\mathbf{r}(t-\mathrm{d}t) \approx \mathbf{r}(t)-\mathbf{v}(t)\mathrm{d}t+\frac{1}{2}\mathbf{a}(t)\mathrm{d}t^2+\dots \quad (28)$$

Adding these two expansions,

$$\mathbf{r}(t+\mathrm{d}t)+\mathbf{r}(t-\mathrm{d}t) \approx 2\mathbf{r}(t)+\mathbf{a}(t)\mathrm{d}t^2+\dots, \quad (29)$$

which can be rewritten:

$$\mathbf{r}(t_{i+1}) = 2\mathbf{r}(t_i)-\mathbf{r}(t_{i-1})+\mathbf{a}(\mathbf{r}(t_i))\mathrm{d}t^2+\dots \quad (30)$$

It is possible, from this equation, to find $\mathbf{r}(t_{i+1})$ from $\mathbf{r}(t_i)$, $\mathbf{r}(t_{i-1})$ and the expression of the acceleration of the studied object at time $t_i$. It is possible to link the velocity $v(t_{i-1})$ to $\mathbf{r}(t_i)$, $\mathbf{r}(t_{i-1})$ using Taylor expansions

$$\mathbf{r}(t_i) = \mathbf{r}(t_{i-1})+\mathbf{v}(t_{i-1})u+\frac{\mathrm{d}t^2}{2}\mathbf{a}(\mathbf{r}(t_{i-1})), \quad (31)$$

and link it to the velocity at mid-point in time:

$$\mathbf{v}\left(t_i-\frac{\mathrm{d}t}{2}\right) = \mathbf{v}\left(t_{i-1}\right)+\frac{\mathrm{d}t}{2}\mathbf{a}(\mathbf{r}(t_{i-1})), \quad (32)$$

yielding:

$$\mathbf{v}(t_i) = \mathbf{v}\left(t_i-\frac{\mathrm{d}t}{2}\right)+\frac{\mathrm{d}t}{2}\mathbf{a}(\mathbf{r}(t_i)). \quad (33)$$

Solving this new system of equations should yield robust conservation of quantities of interest and allow for a more stable numerical scheme.

## C. Results in an Earth orbiting the Sun scenario

### D. Chosen parameters

In this scenario, the equations of motion are solved using a time step $\mathrm{d}t = 10^{-2}$ days = 14 min and $\mathrm{N}_{\mathrm{time}} = 365 * 500$ with parameters given by Table II, resulting in a total time propagation of 5 yr.

$$v_{\theta,T} = \sqrt{\frac{G(M_1+M_2)}{r_1^3}} \approx 2\pi \text{ rad/yr}. \quad (34)$$

Note that the angular velocity of Earth corresponds to the value given for a circular orbit, ensuring that the total energy of the system is equal to zero.

#### 1. Results

Results are pretty much the same between the two procedures with notable oscillations of the distance between the Earth and the Sun that causes the visible non-conservation of energy (see pics created by the code).

The cause of these unphysical oscillations can be multiple:

| Parameters | Values |
|---|---|
| $m_1$ | $1\ M_\odot$ |
| $m_2$ | $3.00273 \times 10^{-6}\ M_\odot$ |
| $\mathbf{r}_1(t=0)$ | $(10^{-6}, 0)$ |
| $\mathbf{r}_2(t=0)$ | $(1, 0)$ |
| $\mathbf{v}_1(t=0)$ | $(0, 0)$ |
| $\mathbf{v}_2(t=0)$ | $(0, 2\pi)$ |

TABLE II: Initial conditions taken for the scenario. Positions are in AU, angles in rad, radial velocities are in AU/yr and angular velocities in rad/yr. Note that the initial position of the Sun $\mathbf{r}_1(t=0)$ is not equal to zero in an attempt to soften numerical instabilities.

1. Bad choice of time step.

2. Inaccurate integrators.

3. Numerical instabilities caused by little numbers in the position of the Sun and the mass of the Earth.

Changing the time step and going as far as propagate the system every second for 5 years do not change the outcome, but results and energy conservation become noticeably more robust when approximating the mass of the Earth to zero when writing the right-hand side of the Sun dynamical equation. It is therefore probable that numerical inaccuracies and ratios of small numbers are the main factor of the results inaccuracies in such short time periods.

Constructing a method that is both accurate and conserve the symplectic flow of the equation of motion is a well-known problem in the literature and seems out of my scope as a simple project. Note that inaccuracies become more and more problematic as the number of planets in the system is increased, since the N-body problem is well-known to be chaotic in nature.