

Computing the long term evolution of the solar system with geometric numerical integrators

Shaula Fiorelli Vilmart • Gilles Vilmart ^[1]

Simulating the dynamics of the Sun–Earth–Moon system with a standard algorithm yields a dramatically wrong solution, predicting that the Moon is ejected from its orbit. In contrast, a well chosen algorithm with the same initial data yields the correct behavior. We explain the main ideas of how the evolution of the solar system can be computed over long times by taking advantage of so-called geometric numerical methods. Short sample codes are provided for the Sun–Earth–Moon system.^[2]

1 Computing the trajectories

Let us step back in time and imagine we are on the first of January of the year 1600, when Johannes Kepler (1571–1630) has just moved to Prague to become the new assistant of the astronomer Tycho Brahe. Kepler had to escape from persecution in Graz, particularly caused by his adhesion to the controversial Copernican theory, boldly saying that the planets revolve around the Sun.

^[1] Partially supported by the Swiss National Science Foundation grants 200020_144313/1 and 200021_162404.

^[2] An earlier version in French of this article first appeared in [11].

How the strange motion of Mars inspired Kepler. Tycho Brahe is very interested in planetary motion and has already calculated very precisely the orbits of known planets. But Mars escapes comprehension: he cannot properly *predict its trajectory*. Without warning him of the difficulty, Brahe asks Kepler to calculate the precise orbit of Mars.

It will take about six years for Kepler to complete this work. Indeed, while Venus has a nearly circular orbit, the trajectory of Mars is more complex: it turns out to be an ellipse, whose flattening^[3] is, after Mercury's, the second largest of all planets' in the solar system.

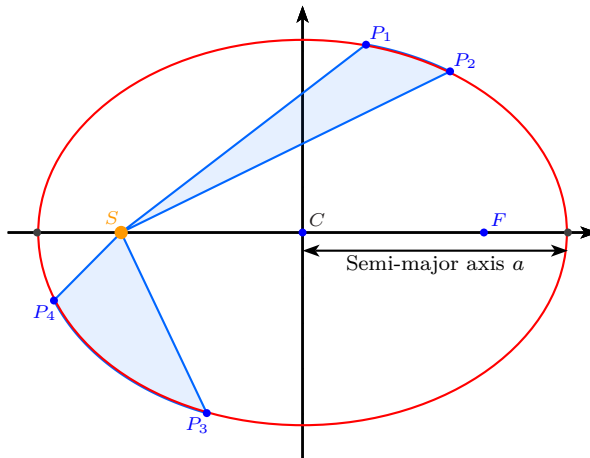


Figure 1: Illustration of Kepler's laws. The Sun S and the point F are the foci of the elliptic trajectory (in red) of the planet P . By the law of equal areas, when the time elapsed between positions P_1 and P_2 is equal to the time elapsed between positions P_3 and P_4 , then the domains SP_1P_2 and SP_3P_4 (in blue) have the same area.

Kepler's three laws. This takes Kepler to propose his three basic laws (compare Figure 1):

1. the planets describe elliptical orbits, with the Sun at a focus of each;
2. the segment connecting the Sun and a planet sweeps out equal areas during equal times; this so-called *invariant* of the problem is called the *law of equal areas*^[4];

^[3] The flattening may be measured by the *eccentricity* of the ellipse, which is the ratio of the distance between C and F to the semi-major axis a , compare Figure 1. The smaller its eccentricity, the more an ellipse is shaped like a circle.

^[4] That the derivative with respect to time of the area swept out is zero turns out to be connected to the conservation of the planet's angular momentum.

3. the square of the orbital period T of a planet is proportional to the cube of the semi-major axis a of the elliptical orbit: $T^2 = \text{Constant} \cdot a^3$.

Newton's laws of motion and the universal law of gravitation. In 1687, Isaac Newton (1642–1727) publishes his book *Philosophiæ Naturalis Principia Mathematica*. Inspired by Kepler's work, he proposes the three *laws of motion* and the universal *law of gravitation*, stating that all cosmic objects attract each other mutually with equal forces (but in opposite directions), proportional to the product of their masses and inversely proportional to the square of the distance between them.

Let m_S and m_P denote the masses of two bodies S and P , let D be the distance between S and P , and let \vec{u} be a vector with unit length in the direction from S to P . The gravitational force $\vec{F}_{S \rightarrow P}$ applied by S to P is then given by the formula

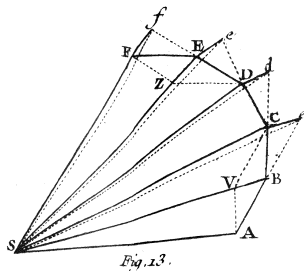
$$\vec{F}_{S \rightarrow P} = -\vec{F}_{P \rightarrow S} = -G \frac{m_S m_P}{D^2} \vec{u},$$

where the constant of proportionality G is called the *universal constant of gravitation*. It is this law that we will use later to calculate the motion of the planets.

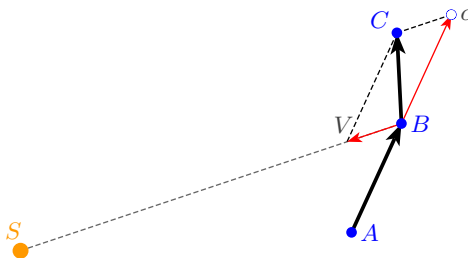
Newton's explanation of Kepler's law of equal areas. Newton also provides a justification for Kepler's laws: they represent the motion of a body subject to a single gravitational force, that of the Sun. To explain the law of equal areas, he uses a method that can be interpreted as the first *geometric numerical integrator*, as presented in [3, 9]. The idea is to apply the Sun's attractive gravitational force not continuously over time, but via a sequence of impulses: Suppose that S is the position of the Sun, and suppose that a planet is located initially at point A , see Figure 2.

Let us first assume that the Sun applies no gravitational force at all; in this case, the planet moves during a certain time from A to a point B along a straight line, with a constant velocity in the direction \vec{AB} (see Figure 2). Waiting for the same time again, the planet should continue on the same straight line until it reaches the point c , with $\vec{AB} = \vec{Bc}$.

However, let us apply an *impulse of force* from the Sun to the planet: this force adds a velocity component to the motion of the planet that Newton represents by the vector \vec{BV} along the segment SB . The planet's velocity is now the sum of two components: the vector \vec{Bc} and the vector \vec{BV} , and the resulting vector is \vec{BC} which defines the point C . The planet thus moves with this new constant velocity until it reaches C .



(a) Figure from Newton's *Principia Mathematica* (1687).



(b) The planet's motion subject to a single "force impulse".

Figure 2: Newton's proof of Kepler's second law.

Iterating this process, the planet follows the path A, B, C, D, E, F, \dots . Newton proves that all the triangles SAB, SBC, SCD, SDE, SEF have the same area by the following argument:

Newton's geometric proof. We first note that the triangles SAB and SBc have the same area, because they have the same basis $\overrightarrow{AB} = \overrightarrow{Bc}$ and the same height issued from S . Next, observing that $BcCV$ is a parallelogram, we deduce that \overrightarrow{Cc} is parallel to \overrightarrow{SB} . The triangles SBc and SBC thus have the same basis \overrightarrow{SB} and the same heights issued from c and C respectively; and hence they have the same area. Thus, the triangles SAB and SBC have the same area.

Newton thus proves a *discrete* version of Kepler's second law, meaning a motion with successive jolts. The process which permits to get from A to B , then B to C , and so forth, corresponds in fact to a geometric numerical scheme, known today as the *symplectic Euler method*, which we will present later on. We will also see that as the time interval between two force impulses tends to zero, the obtained approximation of the trajectory converges towards the exact solution of the problem.

2 What is a differential equation, and how can we solve it?

Many physical phenomena can be modeled by *differential equations*, that is to say, equations in which the unknown is not a number but a function, and involving one or more derivatives of this function.^[5]

^[5] Note to advanced readers: in this snapshot, we are only concerned with the case where these functions depend on one variable only, so-called *ordinary* differential equations.

A differential equation example: de Beaune's problem. As an example, let us consider a problem formulated by Florimond de Beaune in 1638:^[6]

Find a curve \mathcal{C} in the plane, given by a function $y(t)$, such that the tangent to \mathcal{C} in any point M with abscissa t intersects the horizontal axis at the point $u = t - 1$ (see Figure 3).

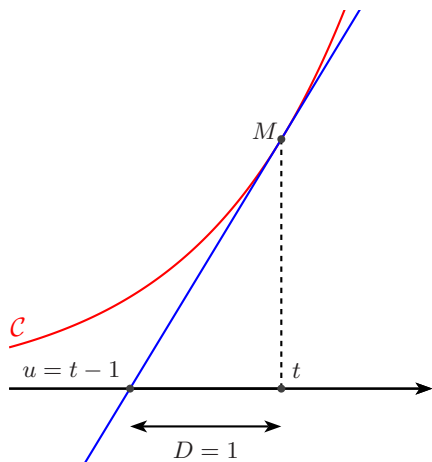


Figure 3: The problem of de Beaune (1638).

We note that the slope of the tangent at $M = (t, y(t))$ to the curve \mathcal{C} equals the quotient of the height $y(t)$ over the width $D = 1$, which means the slope is equal to $y(t)$. In addition, we recall that by definition the slope is the derivative of y at t . Together, this gives the differential equation

$$y'(t) = y(t).$$

The general solution of this equation is $y(t) = C \cdot e^t$, where C is a constant that can be determined by adding an *initial condition* $y(0) = y_0$. The differential equation with a given initial value then possesses a unique solution $y(t) = y_0 \cdot e^t$.

Numerical methods. For a general differential equation, however, it is in practice often difficult or even impossible to find a formula for the exact solution. Therefore a *numerical integrator*, which is an algorithmic method for calculating an approximate solution, must be used.

[6] De Beaune (1601–1652) is famous mostly for the problem presented here. It is one of four problems that he submitted to René Descartes (1596–1650), who had just published *La Géométrie* in 1637.

There exists a wide range of algorithms for solving a differential equation of the form

$$y'(t) = f(y(t)), \quad y(0) = y_0,$$

where y_0 is a given initial condition, f is a given function and y is an unknown function depending on t . Note that for the problem of de Beaune, the function f reduces to the identity, $f(y) = y$. We now present a few simple examples of such methods, chosen for their importance.

Euler methods: approximations of the continuous model. When proving the law of equal areas, Newton used gravitational force impulses, applied to the planet at regular intervals. The numerical methods rely on the same idea.

We first choose a stepsize h and we compute an approximation $y_n \simeq y(t_n)$ of the continuous solution at times $t_n = nh$, where $n = 0, 1, 2, 3, \dots$. The first quantity y_0 is known, this is the initial condition. Next, we compute y_1 , then y_2 , then y_3 , and so on. The common approach of all so-called *Euler methods* is to approximate the derivative by a difference quotient:

$$y'(t_n) = \lim_{\varepsilon \rightarrow 0} \frac{y(t_n + \varepsilon) - y(t_n)}{\varepsilon} \simeq \frac{y_{n+1} - y_n}{h}.$$

Explicit Euler method. This is the simplest numerical method. It is due to Leonhard Euler (1707–1783).^[7] By approximating $f(y(t_n))$ with $f(y_n)$ in the differential equation, we obtain the *explicit Euler method*, $f(y_n) = (y_{n+1} - y_n)/h$, which can be written as

$$y_{n+1} = y_n + h f(y_n).$$

Convergence rate of a numerical integrator. Successively calculating y_1 , y_2 , y_3 , and so on, we obtain a polygonal line passing through the points (t_n, y_n) , see Figure 4. One can show that the polygonal line converges to the exact solution as h goes to zero. Additionally, we observe in Figure 4 that when dividing the stepsize h by a factor of two, the numerical solution (blue polygonal line) gets closer to the exact solution (red curve) with an error divided by the same factor, two to the power of one. The *convergence rate* is thus of *order 1*.

There exist many variants of the Euler method, in particular the so-called Runge–Kutta methods, which can be more accurate, with distance proportional to h^2 , h^3 , and so on, corresponding to a rate of convergence of order 2, 3, or higher.

[7] This was a great contribution of Euler among numerous ones in impressively many areas of science. He presented it in the first volume of his *Institutiones calculi integralis*. Having left Berlin in 1766, where he had written the book, Euler published it in 1768 in Saint Petersburg. Thus, Euler’s method can already celebrate its 250 year anniversary.

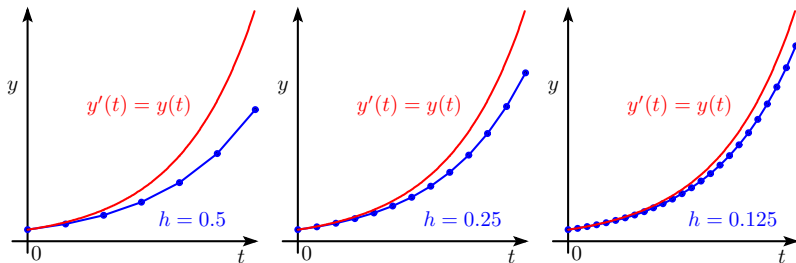


Figure 4: The explicit Euler method for solving the problem of de Beane $y'(t) = y(t)$, with $y(0) = 1$ (in blue), and the exact solution (in red). The stepsizes are $h = 0.5, 0.25, 0.125$, respectively.

The implicit Euler method. In contrast to the explicit Euler method, by instead choosing the approximation $f(y(t_n)) \simeq f(y_{n+1})$, we get the *implicit Euler method*

$$y_{n+1} = y_n + h f(y_{n+1}).$$

This method is called *implicit* because the computation of y_{n+1} in general requires solving a non-linear system. Indeed, one has to compute y_{n+1} while the value $f(y_{n+1})$ is not a priori known. There exist specific methods for solving such problems.

Many other schemes could be considered. However, in some situations, the rate of convergence is not the only important aspect to be taken into consideration. Having a correct qualitative behavior for long times can be very important, in particular regarding invariants, that is, conserved quantities such as the energy of the Sun–Earth–Moon system. We will address this issue in the next section.

3 How to conserve the energy of a physical system?

Before focussing on the planetary problem, let us consider first another, simpler example of a physical system: a mass oscillating on a spring close to its rest position (see Figure 5). We denote by $q(t)$ the elongation of the spring at time t , by $p(t)$ the *momentum*, and by m the mass of the body. In the absence of friction or damping forces, we can model the motion by the following system of differential equations:

$$q'(t) = \frac{1}{m} p(t), \quad p'(t) = -kq(t). \quad (*)$$

The first equation above is just the definition of momentum. The second equation is a consequence of the following two facts: First, if the spring is

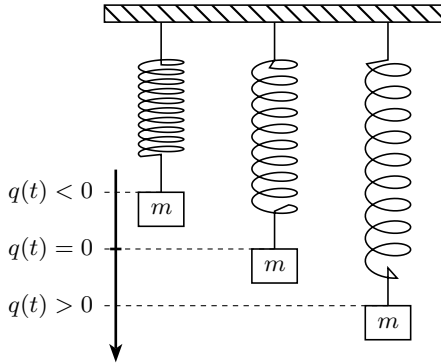


Figure 5: A mass oscillating on a spring, where $q(t)$ is the elongation at time t . The spring is compressed on the left, at its rest position in the middle, and stretched on the right.

stretched or compressed with $q(t)$ units with respect to its rest position, then a restoring force $-kq(t)$ proportional to $q(t)$ but with opposite direction is applied. This is called Hooke's law, where k is a positive constant corresponding to the stiffness of the spring.

Second, Newton's second law of motion states that the sum of forces equals the product of mass and acceleration: $\vec{F}(t) = m\vec{a}(t)$, where $\vec{a}(t)$ has coordinates $q''(t)$. We deduce $-kq(t) = mq''(t) = p'(t)$, yielding the second differential equation in (*).

Energy conservation. The total energy of this system is given by

$$E(p, q) = \text{kinetic energy} + \text{potential energy} = \frac{1}{2m}p^2 + \frac{k}{2}q^2. \quad \text{[8]}$$

It is *conserved* in time by the exact solution, which means

$$E(p(t), q(t)) = E(p(0), q(0)) \text{ for all } t.$$

Indeed, the derivative of the energy with respect to time is zero:

$$\frac{d}{dt}(E(p(t), q(t))) = \frac{1}{m}p'(t)p(t) + kq'(t)q(t) = \frac{1}{m}(-kq(t))p(t) + \frac{k}{m}p(t)q(t) = 0.$$

In comparison, let us now examine how the numerical solutions behave in this aspect.

[8] The equations of motion have a peculiar structure when expressed in terms of q , p , and E , namely $q' = \frac{\partial E}{\partial p}(p, q)$, $p' = -\frac{\partial E}{\partial q}(p, q)$. This observation is at the foundation of the so-called *Hamiltonian formulation* of classical mechanics, going back, alongside many other innovations in mathematics and physics, to William Rowan Hamilton (1805–1865).

Behavior of the energy under explicit and implicit Euler methods. We fix at a chosen instant $t = 0$ the initial conditions $p(0) = p_0$ and $q(0) = q_0$. For simplicity, we also fix the mass as $m = 1$. Applying the explicit Euler method, we obtain the recurrence relation

$$q_{n+1} = q_n + hp_n, \quad p_{n+1} = p_n - hkq_n.$$

A calculation yields

$$\begin{aligned} E(p_{n+1}, q_{n+1}) &= \frac{1}{2}p_{n+1}^2 + \frac{k}{2}q_{n+1}^2 = (1 + kh^2)\left(\frac{1}{2}p_n^2 + \frac{k}{2}q_n^2\right) \\ &= (1 + kh^2)E(p_n, q_n). \end{aligned}$$

We see that at each step of the scheme, the energy is *amplified* by the factor $1 + kh^2$, which is strictly larger than 1. Analogously, considering the implicit Euler method,

$$q_{n+1} = q_n + hp_{n+1}, \quad p_{n+1} = p_n - hkq_{n+1},$$

the same calculation yields

$$E(p_{n+1}, q_{n+1}) = \frac{1}{1 + kh^2}E(p_n, q_n).$$

This time the energy is not amplified, but *damped* at each step of the method.

Trajectories in the (p, q) -plane. The state of the spring at a given time t is described by the knowledge of both its position q and momentum p . For the exact solution, the law of conservation of energy implies that they satisfy the identity

$$p^2 + kq^2 = 2E(p, q) = \text{Constant}.$$

Setting $k = 1$, the exact solution therefore corresponds to a circle in the (p, q) -plane, passing through the given initial condition – see the red curve in Figure 6. Here, at initial time the spring is stretched, $q_0 = 1$, and without velocity, $p_0 = 0$.

We can also view the numerical solutions obtained for the spring problem with stepsize $h = 1/4$ in Figure 6. We observe that the solutions obtained with the explicit and implicit Euler methods are spiraling towards the exterior or interior, respectively. This is due to the amplification or damping factor described previously.

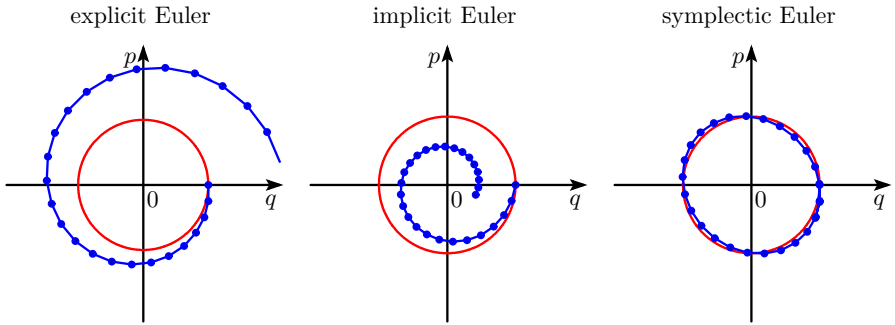


Figure 6: Comparison of the explicit, implicit, and symplectic Euler methods (in blue) for the *harmonic oscillator* (a mass $m = 1$ on a spring with stiffness parameter $k = 1$). The trajectory of the exact solution is a circle (in red).

The symplectic method, an integrator that preserves the energy well over long times. To get a better preservation of the energy, the idea is to combine the explicit and implicit Euler methods. For the spring problem, the *symplectic Euler method* is given by

$$q_{n+1} = q_n + hp_n, \quad p_{n+1} = p_n - hkq_{n+1}.$$

Note that this method alternates between updating the position and the momentum, just as in Newton's proof of the law of equal areas presented earlier.

We now consider the *modified numerical energy* \tilde{E}_h , defined as $\tilde{E}_h(p, q) = E(p, q) + hkpq$. A calculation yields

$$\tilde{E}_h(p_{n+1}, q_{n+1}) = \tilde{E}_h(p_n, q_n).$$

This means that the modified energy is exactly conserved by the numerical scheme, without amplification and attenuation factor. In the (p, q) -plane, the curve $\tilde{E}_h(p, q) = \tilde{E}_h(p_0, q_0)$ is in fact an ellipse close to the circle of the exact solution when h is small. This is because the numerical energy \tilde{E}_h is a small *perturbation* of the exact energy E of size proportional to h . This shows that the numerical error in the energy always remains small with size h . Indeed, we observe in Figure 6 (right picture) that the numerical trajectory (in blue) remains close to the exact one (in red).

Even though not exactly, the symplectic Euler method conserves the energy of the harmonic oscillator well. Compared to the explicit or implicit Euler method, it is therefore better adapted to the problem. Numerical integrators with the property that they preserve the invariants and symmetries of the exact solution of the system well are called *geometric*.

4 The Sun–Earth–Moon system

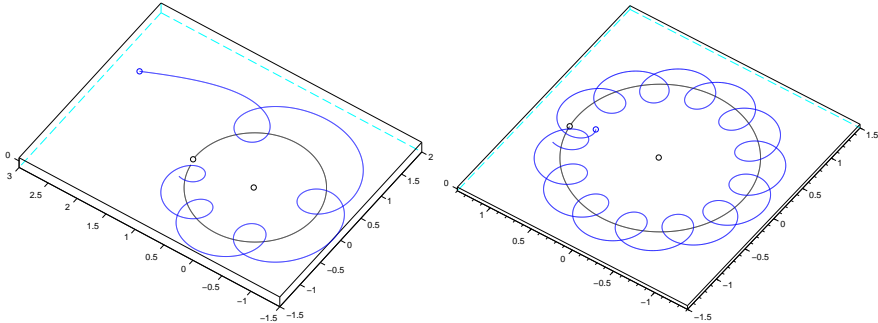


Figure 7: Comparison of the explicit Euler method (left) and the symplectic Euler method (right) for the Sun–Earth–Moon system simulated over one year. The distance between the Moon (blue trajectory) and the Earth (black trajectory) is scaled by a factor of 100 in the plots, to better distinguish the Earth and the Moon.

We finally consider the Sun–Earth–Moon system, where for simplicity we neglect the other bodies and influences in the solar system. We represent the positions of these three bodies at time t as $\vec{q}_i(t) \in \mathbb{R}^3$, $i = 0, 1, 2$, where the index $i = 0$ corresponds to the Sun, $i = 1$ to the Earth, and $i = 2$ to the Moon. The masses of the three bodies are denoted by m_i , $i = 0, 1, 2$. We also consider the momenta

$$\vec{p}_i(t) = m_i \vec{q}_i'(t). \quad (**)$$

Newton’s second law of dynamics then reads

$$\begin{aligned} \vec{p}_0' &= \vec{F}_{E \rightarrow S} + \vec{F}_{M \rightarrow S}, \\ \vec{p}_1' &= \vec{F}_{S \rightarrow E} + \vec{F}_{M \rightarrow E}, \\ \vec{p}_2' &= \vec{F}_{S \rightarrow M} + \vec{F}_{E \rightarrow M}. \end{aligned} \quad (***)$$

Taking (**) and (***) together, and expressing the right sides in (***) through Newton’s law of gravitation, we get a system of six differential equations for the functions \vec{q}_i and \vec{p}_i , to which we apply our numerical schemes.

In Table 1 we provide masses, positions, and initial velocities of the Sun, the Earth, and the Moon at a given date (here 1st of January 2016), and a value for the gravitational constant G ^[9].

^[9] The gravitational constant G is among the most difficult physical constants to measure. In precise celestial computations, one uses rather the product of G times the mass of the body under consideration as input data (*standard gravitational parameter*).

body	mass (relative to M_{\odot})	position (ua)	velocity (ua/d)
Sun	$m_0 = 1$	0 0 0	0 0 0
Earth	$m_1 = 3.00348959632 \cdot 10^{-6}$	-0.1667743823220 0.9690675883429 -0.0000342671456	-0.0172346557280 -0.0029762680930 -0.0000004154391
Moon	$m_2 = 1.23000383 \cdot 10^{-2} m_1$	-0.1694619061456 0.9692330175719 -0.0000266725711	-0.0172817331582 -0.0035325102831 0.0000491191454
Gravitational constant $G = 2.95912208286 \cdot 10^{-4} \text{ua}^3/\text{d}^2/M_{\odot}$			

Table 1: Initial data from [8] for Sun, Earth, Moon on 01/01/2016 at 0h00. Masses are given relative to the Sun mass M_{\odot} .

The Sun is chosen as reference and located at the origin (0, 0, 0). Distances are expressed in astronomical units ua, a quantity based on the Earth–Sun distance (1 ua is about 150 million kilometers), and the time is in Earth days.

Try yourself, with the free open source software Scilab. We give below a short sample code for the free open source software Scilab [10] for computing the evolution of the system with either the explicit or the symplectic Euler method. Results are presented in Figure 7. Notice that the trajectories are almost in a plane, but actually evolve in 3D. The Sun itself is slightly moving as well – this effect is, by the way, used as a common method to detect exoplanets – but the software represents the trajectories relative to the Sun.

To get started with running your own simulations, put the following code into a file `sunearthmoon.sce`. You can also download the file from this snapshot’s website.^[10]

```
//Newton's law of gravitation
function f=fun_v(q)
  def(' [v]=vecf(v0)', 'v=v0/norm(v0).^3');
  sun=1:3; earth=4:6; moon=7:9;
  f(sun) = -G*m0*m1*vecf(q(sun) -q(earth))..
           -G*m0*m2*vecf(q(sun) -q(moon));
  f(earth)=-G*m1*m0*vecf(q(earth)-q(sun))..
           -G*m1*m2*vecf(q(earth)-q(moon));
  f(moon) = -G*m2*m0*vecf(q(moon)-q(sun))..
           -G*m2*m1*vecf(q(moon)-q(earth));
endfunction
```

^[10] <http://dx.doi.org/10.14760/SNAP-2017-009-EN>

```

//Newton's second law of motion
function f=fun_u(p)
    f=[p(1:3)/m0;p(4:6)/m1;p(7:9)/m2];
endfunction

//Symplectic Euler method
function [vp,vq]=euler_symplectic(n,h,p,q)
vp=p;vq=q;
for i=1:n
    q=q+h*fun_u(p); p=p+h*fun_v(q);
    vq=[vq,q]; vp=[vp,q];
end
endfunction

//Explicit Euler method
function [vp,vq]=euler_explicit(n,h,p,q)
vp=p;vq=q;
for i=1:n
    tmp=q; q=q+h*fun_u(p); p=p+h*fun_v(tmp);
    vq=[vq,q]; vp=[vp,q];
end
endfunction

//Sun-Earth-Moon system integration
//
//body masses
m0=1;m1=3.00348959632E-6;m2=m1*1.23000383E-2;
//gravitational constant
G=2.95912208286e-4;
//Initial conditions
//Source: PORTAIL SYSTEME SOLAIRE
// OBSERVATOIRE VIRTUEL DE L'IMCCE
// Observatoire de Paris / CNRS
//http://vo.imcce.fr/webservices/miriade/?forms
//Target: p:Earth, s:Moon
//Epoch: 2016-01-01 00:00:00, 1, 1.0 - day, UTC
//Reference center:
//INPOP Ecliptic Rectangular AstrometricJ2000
//Ecliptic coordinates
//Initial positions and velocities
q0=[0;0;0;-0.1667743823220;0.9690675883429;-0.0000342671456;..
    -0.1694619061456;0.9692330175719;-0.0000266725711];
v0=[0;0;0;-0.0172346557280;-0.0029762680930;-0.0000004154391;..
    -0.0172817331582;-0.0035325102831;0.0000491191454];
//Initial momenta
p0=[v0(1:3)*m0;v0(4:6)*m1;v0(7:9)*m2];

//Time integration over 365 days -- choose one of the following:
[vp,vq]=euler_symplectic(365*10,0.1,p0,q0) //stepsize h=0.1
//[vp,vq]=euler_explicit(365*10,0.1,p0,q0) //stepsize h=0.1

//Trajectories with respect to the Sun placed at the origin
vq(4:6,:)=vq(4:6,:)-vq(1:3,:);
vq(7:9,:)=vq(7:9,:)-vq(1:3,:);
comet3d(0,0,0);

//Increase the Earth-Moon distance by a factor 100 for visualization.
vq(7:9,:)=vq(4:6,:)+100*(vq(7:9,:)-vq(4:6,:))
comet3d([vq(4,:)',vq(7,:)'],[vq(5,:)',vq(8,:)'],[vq(6,:)',vq(9,:)']);

```

Once you have Scilab installed, open and run `sunearthmoon.sce` from within Scilab using either `euler_symplectic` or `euler_explicit` as integrator. To change the integrator, you can just (un)comment the corresponding lines, starting with `[vp,vq]=euler_...` Do you see the difference?

Note that this sample code can be straightforwardly adapted to include additional planets of the solar system, using initial data from [8]. It could also be extended to predict solar or lunar eclipses when the Earth moves into the Moon’s shadow or the converse, taking into account the diameters of the bodies. Enjoy creating your own simulations!

5 Is the solar system stable?

A question closely related to the topic of this snapshot is the issue of the *stability of the solar system*. Soon after Newton proposed his universal law of gravitation, many researchers, including, amongst others, Pierre-Simon Laplace (1749–1827), Joseph-Louis Lagrange (1736–1813), and Siméon Denis Poisson (1781–1840), were studying the question whether the regular trajectories of the planets will continue nicely until the end of times, or if collisions or ejections will occur.

In 1885, King Oscar II of Sweden sponsored a competition about this question. The prize was awarded to Henri Poincaré (1854–1912), although he did not really solve the problem. His contribution, however, is at the origin of the theory of *dynamical systems*. It also led to important developments in “Hamiltonian perturbation theory” and gave rise to the so-called Kolmogorov–Arnold–Moser (KAM) theory, which deals with the persistence of quasi-periodic motions under small perturbations, see the survey [7]. Unfortunately, this beautiful theory does not apply to realistic solar system models.

The initial question “Is the solar system stable?” then remained open until the last decades, where the final negative answer, revealing that the solar system is *chaotic*, was given by the mathematician and astronomer Jacques Laskar and his collaborators. The argument is based on analytic means, but also uses numerical methods including geometric integrators.^[11] In addition, some of their recent computations show that collisions or ejections could even occur in the next five billion years, that is, before the end of the life of the Sun (see the survey [4]).

Notice that the past evolution of the solar system over long times has a surprising application: it serves as a measurement scale for geological dating.

^[11] The chaotic behavior of the solar system was shown by Laskar in 1989: a small error of a few meters in the initial position of the Earth is amplified by a factor 10 every 10 million years, leading to a huge error of dozens of millions of kilometers after 100 million years. This makes precise numerical predictions based on planetary trajectories in the solar system become infeasible beyond this time horizon of 100 million years.

Indeed, the position of the planets and their orbital parameters – such as the inclination – influence how sediments are deposited on the surface of the Earth, which allows for geological dating by observing these sediment deposits. Such geological calculations are known again through the work of Laskar and collaborators. The calculation named La2010 [5], which uses geometric numerical methods with high order of accuracy, shows that the extinction of dinosaurs (about 65 million years ago) may have been slightly earlier than previously estimated.

6 Conclusion and outlook

Based on the examples of an oscillating spring and the three-body problem Sun–Earth–Moon, we have shown that in order to get a good qualitative behavior of the numerical solution of a problem with certain invariants, it is essential to use geometric numerical methods, which preserve these invariants.

We have seen that the energy, a key invariant of all mechanical systems, is well preserved by the symplectic Euler method. In contrast, the explicit Euler method, and more generally any standard explicit Runge–Kutta methods, do not preserve it and are thus not suitable for integration over long time intervals. A mathematical theory called “backward error analysis” permits to demonstrate that symplectic integrators have a good energy conservation for such mechanical systems.

The theory of geometric numerical integration [6, 3, 2, 1] turns out to be a powerful tool for the study and design of integrators in many areas of physics (here for example celestial mechanics), chemistry (molecular dynamics), and biology. In addition to that, it has connections with algebraic tools from other fields of mathematics and physics, such as the so-called renormalization in quantum field theory, see [3, Section III.1.5].

Acknowledgements

We thank Pierre-Alain Cherix, Martin J. Gander, Ernst Hairer, Christian Lubich, and Gerhard Wanner for their helpful comments.

Image credits

Fig. 2a Illustration of Theorem I, Book I, from I. Newton, *Philosophiæ Naturalis Principia Mathematica*, Londini anno MDCLXXXVII, 1687.

References

- [1] S. Blanes and F. Casas, *A concise introduction to geometric numerical integration*, Monographs and Research Notes in Mathematics, Chapman and Hall/CRC Press, 2016.
- [2] J. C. Butcher, *Numerical methods for ordinary differential equations*, 3rd ed., John Wiley & Sons, Ltd, 2016.
- [3] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations*, 2nd ed., Springer Series in Computational Mathematics 31, Springer-Verlag, 2006.
- [4] J. Laskar, *Is the solar system stable?*, Chaos, Progress in Mathematical Physics, vol. 66, Birkhäuser/Springer, 2013, pp. 239–270.
- [5] J. Laskar, A. Fienga, M. Gastineau, and H. Manche, *La2010: a new orbital solution for the long-term motion of the Earth*, *Astronomy&Astrophysics* **532** (2011), A89.
- [6] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics*, Cambridge Monographs on Applied and Computational Mathematics 14, Cambridge University Press, 2004.
- [7] J. Moser, *Is the solar system stable?*, *Mathematical Intelligencer* **1** (1978), 65–71.
- [8] Observatoire virtuel de l’IMCCE, Portail système solaire, *Miriade ephemeris generator*, Observatoire de Paris & CNRS (2016), <http://vo.imcce.fr/webservices/miriade>.
- [9] A. Ostermann and G. Wanner, *Geometry by its History*, Undergraduate Texts in Mathematics, Springer-Verlag, 2012.
- [10] Scilab Enterprises, *Scilab: Free and open source software for numerical computation*, Scilab Enterprises, 2012, <http://www.scilab.org>.
- [11] S. Fiorelli Vilmart and G. Vilmart, *Les planètes tournent-elles rond?*, Interstices (INRIA) (2013), <http://interstices.info/planetes>.

Shaula Fiorelli Vilmart is a research associate at the University of Geneva, Section of Mathematics, Sciencescope and NCCR SwissMap.

Gilles Vilmart is a senior lecturer at the University of Geneva, Section of Mathematics.

Mathematical subjects
Numerics and Scientific Computing

Connections to other fields
Engineering and Technology

License
Creative Commons BY-SA 4.0

DOI
10.14760/SNAP-2017-009-EN

Snapshots of modern mathematics from Oberwolfach provide exciting insights into current mathematical research. They are written by participants in the scientific program of the Mathematisches Forschungsinstitut Oberwolfach (MFO). The snapshot project is designed to promote the understanding and appreciation of modern mathematics and mathematical research in the interested public worldwide. All snapshots are published in cooperation with the IMAGINARY platform and can be found on www.imaginary.org/snapshots and on www.mfo.de/snapshots.

Junior Editors
Lara Skuppin and Sophia Jahns
junior-editors@mfo.de

Senior Editor
Carla Cederbaum
senior-editor@mfo.de

Mathematisches Forschungsinstitut
Oberwolfach gGmbH
Schwarzwaldstr. 9–11
77709 Oberwolfach
Germany

Director
Gerhard Huisken



Mathematisches
Forschungsinstitut
Oberwolfach



IMAGINARY
open mathematics