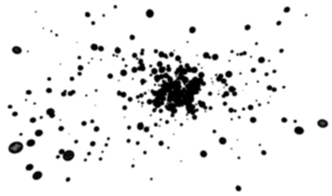

`MoveUrBody` code documentation

n-body problem - Gravity forces

1 Governing equations

The n-body problem is a classic one from the **Newtonian mechanics**: it consists in **resolving the gravity equations**. However, this problem can be generalized (from the algorithmic point of view) and we can often find it in numerical simulation: this is why it is a good real case study.

At the problem beginning (the time t), for each body i, its position $q_i(t)$, its mass m_i and its velocity $\vec{v_i}(t)$ are known. The applied force between two bodies i and j, at the t time, is defined by:

$$\vec{f_{ij}}(t) = G.\frac{m_i.m_j}{||\vec{r_{ij}}||^2}.\frac{\vec{r_{ij}}}{||\vec{r_{ij}}||}, \tag{1}$$

with G the gravitational constant ($G = 6,67384 \times 10^{-11} m^3.kg^{-1}.s^{-2}$) and $\vec{r_{ij}} = q_j(t) - q_i(t)$ the vector from body i to body j. The resulting force (alias the total force) for a given i body, at the t time, is defined by:

$$\vec{F_i}(t) = \sum_{j \neq i}^{n} \vec{f_{ij}}(t) = G.m_i.\sum_{j \neq i}^{n} \frac{m_j.\vec{r_{ij}}}{||\vec{r_{ij}}||^3}, \tag{2}$$

with n the number of bodies in space. For the time integration, the acceleration is required. For a given body i, at the t time, the acceleration characteristic is defined by:

$$\vec{a_i}(t) = \frac{\vec{F_i}(t)}{m_i} = G.\sum_{j \neq i}^{n} \frac{m_j.\vec{r_{ij}}}{||\vec{r_{ij}}||^3}. \tag{3}$$

1.1 An approximation of the total force

The total force $\vec{F_i}$ is given by Eq. 2:

$$\vec{F_i}(t) = G.m_i.\sum_{j \neq i}^{n} \frac{m_j.\vec{r_{ij}}}{||\vec{r_{ij}}||^3}.$$

As bodies approach each other, **the force between them grows without bound**, which is an undesirable situation for numerical integration. In astrophysical simulations, collisions between bodies are generally precluded; this is reasonable if the bodies represent galaxies that may pass right through each other. Therefore, **a softening factor** $\epsilon^2 > 0$ is added, and the denominator is rewritten as follows:

$$\vec{F_i}(t) \approx G.m_i.\sum_{j=1}^{n} \frac{m_j.\vec{r_{ij}}}{(||\vec{r_{ij}}||^2 + \epsilon^2)^{\frac{3}{2}}}. \tag{4}$$

Note the condition $j \neq i$ is no longer needed in the sum, because $\vec{f_{ii}} = 0$ when $\epsilon^2 > 0$. The softening factor models the interaction between two Plummer point masses: masses that behave as if they were spherical galaxies. In effect, **the softening factor limits the magnitude of the force between the bodies**, which is desirable for numerical integration of the system state.

As before, we need to compute the acceleration in order to perform the integration over the time:

$$\vec{a_i}(t) = \frac{\vec{F_i}(t)}{m_i} \approx G.\sum_{j=1}^{n} \frac{m_j.\vec{r_{ij}}}{(||\vec{r_{ij}}||^2 + \epsilon^2)^{\frac{3}{2}}}. \tag{5}$$

2 Collisions

In `MoveUrBody` code there are various implementations: some of them simulate colliding bodies. In this section we try to explain what kind of collisions are used if the implementation considers colliding bodies.

2.1 Collisions detection

We choose *a posteriori* method in order to detect collisions: this is a discrete and easy to implement method. We detect the collisions after they append. This is less expensive in term of computations than *a priori* method (continuous). In `MoveUrBody` we consider that **the shape of all the bodies is a sphere**, so two bodies i and j are colliding if:

$$||\vec{r_{ij}}|| - (r_i + r_j) \leq 0, \tag{6}$$

where r_i and r_j are respectively the radii of body i and body j.

2.2 Elastic collisions

Notation: Throughout this section, m is the mass and v is the velocity. Be aware that $v = ||\vec{v}||$. Subscripts i and j distinguish between the two colliding bodies. An apostrophe after a variable means that the value is taken after the collision (called prime; i.e., v' is "v prime").

In `MoveUrBody` we consider that **all collisions are perfectly elastic**. An elastic collision is a collision in which kinetic energy is conserved. That means no energy is lost as heat or sound during the collision. In the real world, there are no perfectly elastic collisions on an everyday scale of size. But you can get the sense of an elastic collision by imagining a perfect pool ball which doesn't waste any energy when it collides. In an elastic collision, both kinetic energy and momentum are conserved (the total before and after the collision remains the same). Momentum is the product of mass and velocity:

$$\vec{p} = m.\vec{v}. \tag{7}$$

The kinetic energy of an object is one-half times its mass times the square of its velocity:

$$E_k = \frac{1}{2}.m.v^2. \tag{8}$$

Now it is easy to write the conservation of momentum and kinetic energy as two equations:

1. Conservation of momentum

$$m_i.\vec{v_i} + m_j.\vec{v_j} = m_i.\vec{v_i'} + m_j.\vec{v_j'} \tag{9}$$

2. Conservation of kinetic energy

$$\frac{1}{2}.m_i.(v_i)^2 + \frac{1}{2}.m_j.(v_j)^2 = \frac{1}{2}.m_i.(v_i')^2 + \frac{1}{2}.m_j.(v_j')^2 \tag{10}$$

2.2.1 1-Dimensional elastic collisions

Combining the two previous equations (Eq. 9 and 10) and doing a lot of algebra gives the final (after collision) velocities of body i and j:

$$v_i' = \frac{v_i(m_i - m_j) + 2.m_j.v_j}{m_i + m_j}, \quad v_j' = \frac{v_j(m_j - m_i) + 2.m_i.v_i}{m_j + m_i}. \tag{11}$$

This result allows us to find the velocity of two objects after undergoing a one-dimensional elastic collision. We will use this result later in the 3-dimensional case.

2.2.2 3-Dimensional elastic collisions

In previous sub-section (1D elastic collisions) vectors representation was not very important. Now in 3D, we will use the component representation of a vector: $\vec{v} = \{v_x, v_y, v_z\}$.

We will follow a 7-step process to find the new velocities of two objects after a collision. The basic goal of the process is to project the velocity vectors of the two objects onto the vectors which are normal (perpendicular) and tangent to the plan of the collision. This gives us a normal component and two tangential components (defining a plan) for each velocity. The tangential components of the velocities are not changed by the collision because there is no force along the tangent plan to the collision surface. The normal components of the velocities undergo a one-dimensional collision, which can be computed using the one-dimensional collision formulas presented above. Next the unit normal vector is multiplied by the scalar (plain number, not a vector) normal velocity after the collision to get a vector which has a direction normal to the collision surface and a magnitude which is the normal component of the velocity after the collision. The same is done with the unit tangent vectors and the tangential velocity components. Finally the new velocity vectors are found by adding the normal velocity and the two tangential velocity vectors for each object.

Step 1
Find the unit normal and the two unit tangent vectors. The unit normal vector is a vector which has a magnitude of 1 and a direction that is normal (perpendicular) to the surfaces of the objects at the point of collision. The two unit tangent vectors are vectors with a magnitude of 1 which are forming a tangent plan to the circle's surfaces at the point of collision.

First find a normal vector. This is done by taking a vector whose components are the difference between the coordinates of the centers of the spheres. Let $q_i = \{q_{ix}, q_{iy}, q_{iz}\}$, $q_j = \{q_{jx}, q_{jy}, q_{jz}\}$ coordinates of the centers of the spheres (it does not matter which spheres is labelled i or j; the end result will be the same). Then the normal vector $\vec{n}$ is:

$$\vec{n} = \{q_{jx} - q_{ix},\ q_{jy} - q_{iy},\ q_{jz} - q_{iz}\}.$$

Next, we have to find the unit vector of $\vec{n}$, which we will call $\vec{un}$. This is done by dividing by the magnitude of $\vec{n}$:

$$\vec{un} = \frac{\vec{n}}{||\vec{n}||} = \frac{\vec{n}}{\sqrt{n_x^2 + n_y^2 + n_z^2}}.$$

Once it's done we need to find the two unit tangent vectors which are forming the collision tangent plan. Those two unit tangent vectors ($\vec{ut}_1$ and $\vec{ut}_2$) are perpendicular to the unit normal vector $\vec{un}$ and there are also perpendicular between themselves (in order to form a Cartesian coordinate system). Before trying to determine two unit tangent vectors $\vec{ut}_1$ and $\vec{ut}_2$

we will try to find a tangent vector $\vec{t_1}$. If $\vec{t_1}$ is perpendicular to $\vec{un}$ then the scalar product (alias dot product) of the two should be null:

$$\vec{t_1}.\vec{un} = 0 \Leftrightarrow (t_{1x}.un_x) + (t_{1y}.un_y) + (t_{1z}.un_z) = 0.$$

The first and easy solution to this previous equation is the null vector $\vec{0}$ but we obviously want to avoid it. An idea is to fix one of the three components to 0 and an other to 1 in order to determine a perpendicular vector $\vec{t_1}$ (be aware that there is an infinity of perpendicular vectors to $\vec{un}$ and we just need to find one of them).
If $un_x \neq 0$:

$$\vec{t_1} = \left\{-\frac{un_y}{un_x}, 1, 0\right\}.$$

If $un_y \neq 0$:

$$\vec{t_1} = \left\{1, -\frac{un_x}{un_y}, 0\right\}.$$

If $un_z \neq 0$:

$$\vec{t_1} = \left\{1, 0, -\frac{un_x}{un_z}\right\}.$$

We have to choose one of the three previous propositions and be sure to avoid to divide by 0. Now we have to normalize $\vec{t_1}$ in order to find the unitary $\vec{ut_1}$ vector:

$$\vec{ut_1} = \frac{\vec{t_1}}{||\vec{t_1}||} = \frac{\vec{t_1}}{\sqrt{t_{1x}^2 + t_{1y}^2 + t_{1z}^2}}.$$

It remains to determine the $\vec{ut_2}$ vector, this can be done by computing the cross product (alias vector product) between $\vec{un}$ and $\vec{ut_1}$:

$$\vec{ut_2} = \vec{un} \wedge \vec{ut_1} = \{(un_y.ut_{1z} - un_z.ut_{1y}), (un_z.ut_{1x} - un_x.ut_{1z}), (un_x.ut_{1y} - un_y.ut_{1x})\}.$$

Step 2

Create the initial (before the collision) velocity vectors, $\vec{v_i}$ and $\vec{v_j}$. These are just the x, y and z components of the velocities put into vectors: $\vec{v_i} = \{v_{ix}, v_{iy}\}$ (and similarly for $\vec{v_j}$). Note that this step really isn't necessary if the velocities are already represented as vectors. This step is needed only if the velocities are initially represented as separate x, y and z values.

Step 3

Keep in mind that after the collision the tangential components of the velocities are unchanged and the normal component of the velocities can be found using the one-dimensional collision formulas presented earlier. So we need to resolve the velocity vectors, $\vec{v_i}$ and $\vec{v_j}$, into normal and tangential components. To do this, project the velocity vectors onto the unit normal and unit tangent vectors by computing the dot product. Let v_{in} be the scalar (plain number, not a vector) velocity of body i in the normal direction. Let v_{it1} and v_{it2} be the scalar velocity of body i in the tangential directions. Similarly, let v_{jn}, v_{jt1} and v_{jt2} be for body j. These values are found by projecting the velocity vectors onto the unit normal and unit tangent vectors, which is done by taking the dot (alias scalar) product:

$$v_{in} = \vec{v_i}.\vec{un}, \quad v_{it1} = \vec{v_i}.\vec{ut_1}, \quad v_{it2} = \vec{v_i}.\vec{ut_2},$$

$$v_{jn} = \vec{v_j}.\vec{un}, \quad v_{jt1} = \vec{v_j}.\vec{ut_1}, \quad v_{jt2} = \vec{v_j}.\vec{ut_2}.$$

Step 4
Find the new tangential velocities (after the collision). This is the simplest step of all. The tangential components of the velocity do not change after the collision because there is no force between the spheres in the tangential direction during the collision. So, the new tangential velocities are simply equal to the old ones:

$$v'_{it1} = v_{it1}, \quad v'_{it2} = v_{it2},$$

$$v'_{jt1} = v_{jt1}, \quad v'_{jt2} = v_{jt2}.$$

Step 5
Find the new normal velocities. This is where we use the one-dimensional collision formulas (Eq. 11). The velocities of the two spheres along the normal direction are perpendicular to the surfaces of the spheres at the point of collision, so this really is a one-dimensional collision:

$$v'_{in} = \frac{v_{in}(m_i - m_j) + 2.m_j.v_{jn}}{m_i + m_j},$$

$$v'_{jn} = \frac{v_{jn}(m_j - m_i) + 2.m_i.v_{in}}{m_j + m_i}.$$

Step 6
Convert the scalar normal and tangential velocities into vectors. This is easy just multiply the unit normal vector by the scalar normal velocity and you get a vector which has a direction that is normal to the surfaces at the point of collision and which has a magnitude equal to the normal component of the velocity. It is similar for the tangential components:

$$\vec{v'_{in}} = v'_{in}.\vec{un}, \quad \vec{v'_{it1}} = v'_{it1}.\vec{ut}_1, \quad \vec{v'_{it2}} = v'_{it2}.\vec{ut}_2,$$

$$\vec{v'_{jn}} = v'_{jn}.\vec{un}, \quad \vec{v'_{jt1}} = v'_{jt1}.\vec{ut}_1, \quad \vec{v'_{jt2}} = v'_{jt2}.\vec{ut}_2.$$

Step 7
Find the final velocity vectors by adding the normal and tangential components for each body:

$$\vec{v'_i} = \vec{v'_{in}} + \vec{v'_{it1}} + \vec{v'_{it2}},$$

$$\vec{v'_j} = \vec{v'_{jn}} + \vec{v'_{jt1}} + \vec{v'_{jt2}}.$$

Now we have the final (after collision) velocity of each body as a vector.

3 Time

3.1 Time step selection

The easier way to determine Δt is to select it as a constant for all the simulation iterations. For visualization concerns we will sometimes use a constant Δt but, for simulation precision interests, it is better to compute a new time step for each iterations depending on the distance between the nearest bodies. The following equation describes the variable Δt calculation:

$$\|\vec{v_i}(t).\Delta t + \frac{\vec{a_i}(t)}{2}.\Delta t^2\| \leq 0.1 \times ||\vec{r_{ij}}||, \tag{12}$$

with j the nearest body to the body i. For each body i, a time step is calculated and the smallest one is chosen. Eq. 12 traduces that the distance between $i(t)$ and $i(t + \Delta t)$ must be below 10% of the $||\vec{r_{ij}}||$ distance. This equation assures that two masses cannot be closest than 20% between t time and $t + \Delta t$ time. However, Eq. 12 is not directly usable: this is a 4^{th} degree polynomial equation in Δt. It's why we will use the triangle inequality witch allows us to determine a new condition:

$$\|\vec{v_i}(t)\|.\Delta t + \frac{\|\vec{a_i}(t)\|}{2}.\Delta t^2 \leq 0.1 \times ||\vec{r_{ij}}||. \tag{13}$$

Eq. 13 is a 2^{nd} degree equation: this is more reasonable in term of computational time.

3.2 Time integration

~~The integrator used to update the positions and velocities is a leapfrog-Verlet integrator (Verlet 1967) because it is applicable to this problem and is computationally efficient (it has a high ratio of accuracy to computational cost).~~

Body i velocity characteristic at the $t+\Delta t$ time depends on the velocity and the acceleration at the t time:

$$\vec{v_i}(t + \Delta t) = \vec{v_i}(t) + \vec{a_i}(t).\Delta t. \tag{14}$$

At the end, body i position q_i at the $t + \Delta t$ time depends on the position, the velocity and the acceleration at the t time:

$$q_i(t + \Delta t) = q_i(t) + \vec{v_i}(t).\Delta t + \frac{\vec{a_i}(t).\Delta t^2}{2}. \tag{15}$$

Thanks to Eq. 3, 14 and 15, it is now possible to compute the new position and the new velocity for all bodies at the $t + \Delta t$ time.