FYS4150: Project 3

Mikael Toresen

October 28, 2013

1 Discretizing the problem

1.1 Newtons law of gravitation

Newtons law of gravity in two dimensions can be writen as:

$$F_x = G\frac{Mm}{r^3}x, \quad F_y = G\frac{Mm}{r^3}y$$

where $r = (x^2 + y^2)^{1/2}$ is the distance between two objects with mass M and m, and G is the gravitational constant. Combining these two expressions with Newtons second law we find an expression for the acceleration in each dimension. These again can be rewritten so that we have a set of coupled differential equations:

$$a_i = \frac{\mathrm{d}v_i}{\mathrm{d}t} = -G\frac{M}{r^3}x_i \tag{1}$$

$$v_i = \frac{\mathrm{d}x_i}{\mathrm{d}t} \tag{2}$$

where $i \in (x, y)$ and M is the mass of the other object. For a two body system with correct initial conditions we can get circular movement (in the reference frame of one of the bodies). This can be shown to conserve potential and kinetic energy. This is because there are no external forces, so $E = E_{kin} + E_{pot}$ is constant. The "correct" initial condition would be the one which keeps r constant and so preserves potential energy (which is spherically symmetric: $\Phi = -G(Mm)/r$ as can be shown by applying $F = -\nabla\Phi$). If potential energy is conserved and there are no external forces, kinetic energy has to be conseved. If these "perfect" inital conditions are not in place one would expect an elliptic movement where the other body is in one of the focal points of the ellipse, or a hyperbolic movement where there would be no orbit. This last case is equivalent to the total energy being greater than 0, or the kinetic energy is more positive than the potential is negative($|E_{kin}| > |E_{pot}|$, potenitial is zero at $r = \infty$). This is equivalent to having a velocity $v_{esc} = \sqrt{\frac{GM}{r}}$

For a general case with n bodies the acceleration can be written as:

$$a_j = \frac{1}{m_j}\sum_{i \neq j} m_i a_i$$

1.2 Units and values

In this project I use the units of AU, years and M_{Earth}. This means that the Sun gets $M_{Sun} \approx 333000 M_{Earth}$ etc. As we assume circular movement where the Earth orbits the Sun in 1 year, we get:

$$v_{Earth} = \frac{2\pi r}{T} = 2\pi \text{ AU/yr}$$

From the equation for centripetal acceleration of circular movement combined with the gravitational acceleration we get:

$$G\frac{M}{r^2} = \frac{v^2}{r} \quad \rightarrow \quad v = \sqrt{\frac{GM}{r}}, \quad \text{or} \quad G = \frac{v_{Earth}^2 r}{M_{Sun}} = \frac{4\pi^2}{333000}$$

The first expression derived from this gives us a simple expression to find an appropriate velocity for any given planet. The second expression gives us the value of the gravitaional constant in the units we use.

1.3 Runge-Kutta 4(RK4) algorithm

Runge-Kutta is based on the idea of sampling the slope at several points to get a weighted average which is used to get the next point. In particular RK4 works this way:

$$y_{i+1} = y_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}h$$

where y is a discretized quantity measured at points i with intervals h. Here k_1 is the standard Forward-Euler method slope. If one takes a half step length using this, then sample it again we get k_2. k_3 is the equivalent to k_2 using k_2 as instead of k_1 to measure the slope. k_4 is measured using k_3 a whole step length from i. Putting this into the gravitational coupled differential equations we get the following algorithm for RK4:

- compute $r(x, y, t_i)$
- compute $k_{1v} = a(x, y, t_i)$ and $k_{1p} = v(v, x, y, t_i)$ using (1) and (2) (p here is for position) at time t_i. k_{1v} has to be computed both for the x and y-direction.
- use this to measure slope at $t_h = t_i + h/2$ using $z_h = z_i + k_1 h/2$
- compute $r(x, y, t_i + h/2)$
- compute $k_{2v} = a(x, y, t_i + h/2)$ and $k_{2p} = v(v, x, y, t_i + h/2)$ using same equations at time $t_i + h/2$.
- use this to measure slope again at $t_h = t_i + h/2$ using $z_h = z_i + k_2 h/2$ ($z \in (x, y, v_x, v_y)$).
- compute $r(x, y, t_i + h/2)$ for the new values measured from the k_2 computations.
- compute $k_{3v} = a(x, y, t_i + h/2)$ and $k_{3p} = v(v, x, y, t_i + h/2)$ using same equations once again at time $t_i + h/2$.
- use this to measure slope again at $t_{i+1} = t_i + h$ using $z_{i+1} = z_i + k_3 h$.
- compute $r(x, y, t_{i+1})$ for the new values measured from the k_3 computations.
- compute $k_{4v} = a(x, y, t_{i+1})$ and $k_{3p} = v(v, x, y, t_{i+1})$.
- find the values to be used to propagate the quantity z in time: $z_{i+1} = z_i + h(k_1 + 2k_2 + 2k_3 + k_4)/6$ (using appropriate k's)

1.4 Implementation

In my project I chose to compute everything via centre of mass values. This would in theory, if one has many planets, save a lot of computation time. To do this we have to define the centre of mass quantities:

$$M_{com} = \sum_i m_i, \quad \alpha_{com} = \frac{1}{M_{com}} \sum_i m_i \alpha_i, \quad \alpha \in (x, y, r, v_x, v_y, a_x, a_y)$$

If one does this, one should not have to add up the forces from each of the other planets in the system to find the acceleration, but it should be sufficient to look at only the one planet and the centre of mass object. The force between the centre of mass without the object and the object should be the same(N3L). I realized a fatal flaw in my calculations which led to my mistakes. I assumed one simply had to add an adjustment factor(reduced mass divided by own mass to the power of 2 for the accelleration), but this is only correct for a two body case.

I used a class Celeb to describe the celestial bodies (ie. planets + sun). These had masses, positions and velocities in addition to having some useful functions for computing parameters between themselves. I used a class PlanetarySystem (ie. a "general" Solar system) to compute the change over time in a system of celestial bodies.

As I did not sucessfully achieve the task one was expected to, I made a simple program which illustrates the problem for a 3-body case which can be expanded to more(though it is not as scalable as a class implementation).

2 Results

Using my main program i did not get much results other than planets escaping the solar system. To show that I in theory could make the proper required program I made a short "quick 'n dirty" 3-body system with the Sun, Earth and Saturn. For a 3-body system with the Sun, Earth and

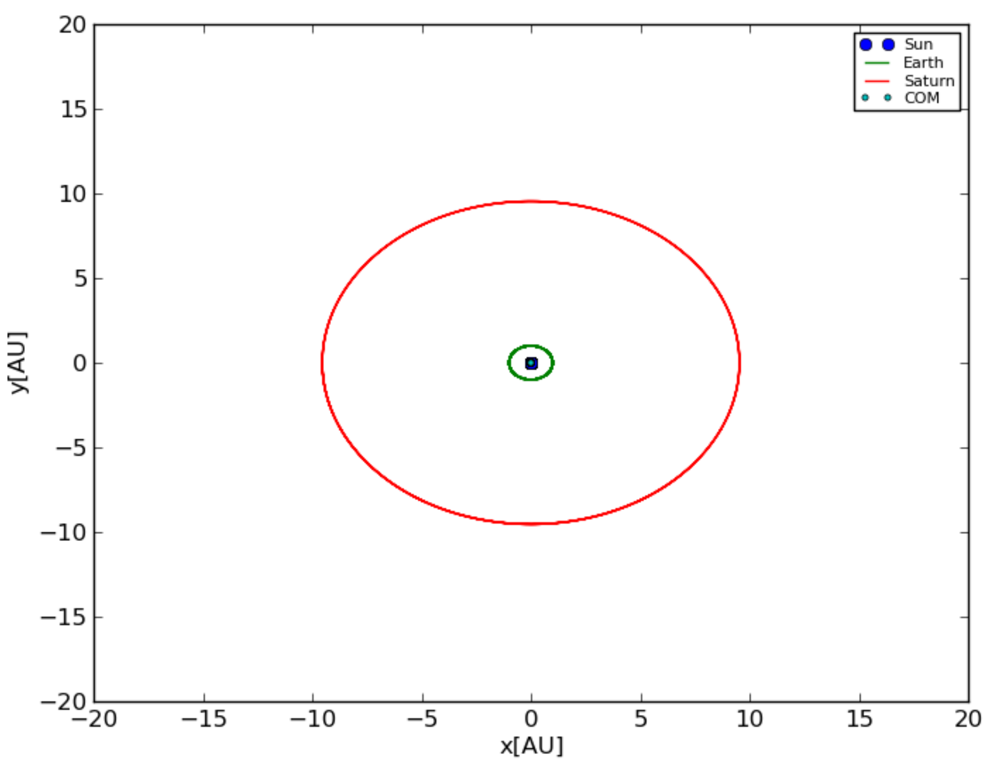

Figure 1: Sun in origin with Earth and Saturn in orbit.

Saturn, the potential and kinetic energy did not change significantly over a period of 10 years (changes were of the order 10^{-6}).

3 Remarks

The repository where you can find the programs is at `https://github.com/miktoki/FYS3150`.

Despite not being able to accomplish the specified task to a satisfactory degree, I did learn a lot about debugging, classes in C++ and was reminded of some good programming habits.