Simulation of the Solar System

Massimo Giordano, Benjamin Haas

October 18, 2014

Link to the repository - code of the program
jasgnön

Abstract

The following article describes a way to simulate the solar system with consideration of the occuring many body problem. The physical theory is based on Newton's second law of motion, which gets discretized into certain time steps. A solution is then obtained by either making use of the Runge-Kutta-Four- or the Verlet-Algorithm. Furthermore, both algorithms are compared of stability.

Contents

1 Introduction

how important differential equations are for science. often sets of coupled differential equations, example solar system. too complex for analytic aproacccce. discreticing. first part focus on solvers for sun and earth with small earthmass, second part expanding problem to solar system

2 Theoretical Concept

As mentioned in the introduction, the first part of the work focuses on testing the Runge-Kutta-Four- and the Verlet-Algorithm, before the problem is expanded to the solar system in the second part.
To start therefore, we first consider a system of Sun and one planet, say Earth. The acting force between both bodys is given by the gravitational force F_G

$$F_G = \frac{G M_{\text{Sun}} M_{\text{Earth}}}{r^2}, \tag{1}$$

where M_{sun} is the mass of the Sun, M_{Earth} is the mass of Earth, G is the Gravitational Constant and r the distance between Sun and Earth. As we want to test different solvers for ordinary differential equations, we assume a much larger mass of the Sun than the mass of the Earth. Consequently, we can neglect the movement of the Sun. Furthermore, there exists no tangential force. For this reason, the movement takes place in a plane, namely the xy-plane. In this case, Newton's second law of motion reads

$$\frac{d^2x}{dt^2} = \frac{F_{G,x}}{M_{\text{Earth}}}, \tag{2a}$$

and

$$\frac{d^2y}{dt^2} = \frac{F_{G,y}}{M_{\text{Earth}}}, \tag{2b}$$

where $F_{G,x}$ and $F_{G,y}$ are the x and y components of the gravitational force.
These two second order differential equations can be rewritten as a set of four coupled first order differential equations. We define a velocity $v(t) = dx/dt$ and get

$$\frac{dx(t)}{dt} = v_x \tag{3}$$

$$\frac{v_x(t)}{dt} = \frac{F_{\text{G,x}}}{M_{\text{Earth}}} \tag{4}$$

$$\frac{dy(t)}{dt} = v_y \tag{5}$$

$$\frac{v_y(t)}{dt} = \frac{F_{\text{G,y}}}{M_{\text{Earth}}} \tag{6}$$

These first order differential equations are used in the Runge-Kutta-Algorithm, but not in the Verlet-Algorithm, as the latter just calculates with accelerations.

The x and y component of the gravitational force can easily be derived by the theorem of intersecting lines. It holds

$$\frac{r}{x} = \frac{F_G}{F_{G,x}} \tag{7a}$$

$$\frac{r}{y} = \frac{F_G}{F_{G,y}} \tag{7b}$$

and therefore

$$F_{G,x} = \frac{x}{r} F_G \tag{8a}$$

$$F_{G,y} = \frac{y}{r} F_G \tag{8b}$$

with F being the absolute value of the gravitational force.
We can also estimate the Gravitational Constant by the assumption of a cicular orbit of the Earth around the Sun. This estimate sounds quite reasonable as the acutal orbit's eccentricity is about 0.017, namely almost a cicular orbit. In this case, the gravitational force is balanced by the centripetal force. Therefore, the absolute values must be equal

$$F_{\mathrm{ZP}} = \frac{M_{\mathrm{Earth}} v^2}{r} = \frac{G M_{\mathrm{Sun}} M_{\mathrm{Earth}}}{r^2} = F_G, \tag{9}$$

with v being the orbital velocity of Earth.

2.1 Analytical Two Body Problem

In the previous considerations, we assumed a much larger mass of the Sun than the mass of the Earth. For that reason, we could neglect the motion of the Sun and easily derive relations. We now do consider Earth's mass and solve the occuring two body problem.
To start with, we let x_1, m_1 and x_2, m_2 be the position vectors and masses of body 1 and body 2. We can easily calculate the acting gravitational forces

$$\vec{F}_{2,1}(\vec{x}_1, \vec{x}_2) = -G m_1 m_2 \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3}. \tag{10a}$$

and

$$\vec{F}_{1,2}(\vec{x}_1, \vec{x}_2) = -G m_1 m_2 \frac{\vec{x}_2 - \vec{x}_1}{|\vec{x}_1 - \vec{x}_2|^3} \tag{10b}$$

what leads to

$$\frac{\partial^2 \vec{x}_1}{\partial t^2} = -G m_2 \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3} \tag{11a}$$

and

$$\frac{\partial^2 \vec{x}_2}{\partial t^2} = -G m_1 \frac{\vec{x}_2 - \vec{x}_1}{|\vec{x}_1 - \vec{x}_2|^3}. \tag{11b}$$

We introduce new vectors for the relative coordinate $\vec{r} := \vec{x}_1 - \vec{x}_2$ and the Center of Mass (COM) vector $\vec{R} := (m_1\vec{x}_1 + m_2\vec{x}_2)/(m_1 + m_2)$ and call $M = m_1 + m_2$. We plug equations 11a and 11b into the definition of $\vec{r}$ as well as into the definition of $\vec{R}$ and obtain

$$\frac{\partial^2 \vec{R}}{\partial t^2} = 0 \tag{12}$$

and

$$\frac{\partial^2 \vec{r}}{\partial t^2} = -GM \frac{\vec{r}}{r^3} \tag{13}$$

Equation 12 is used in section 2.3 to demonstrate energy conservation in the solar system.

2.2 Numerical n Body Problem

If we expand our system to the solar system with consideration of the Sun and all nine planets, we are dealing with a ten body problem. Note that we count Pluto as a planet for historical reasons. For such a "complex" system, the by now only well known way to get quantitative results is a numerical approach. To start with, we calculate for every timestep for every body all forces, which are acting on it. We get:

$$\frac{d\vec{v}_j(t)}{dt}m_j = \vec{F}_j(t) = Gm_j \sum_{i\neq j}\left[\frac{m_i}{|\vec{r}_i(t)-\vec{r}_j(t)|^2}\frac{\vec{r}_i-\vec{r}_j}{|\vec{r}_i-\vec{r}_j|}\right] \tag{14}$$

To solve equation 14, we integrate, discretize and finally obtain

$$\vec{v}_{n+1} = \vec{v}_n + G\int_{t_n}^{t_{n+1}}\sum_{i\neq j}\left[m_i\frac{\vec{r}_i-\vec{r}_j}{|\vec{r}_i(t)-\vec{r}_j(t)|^3}\right]dt. \tag{15}$$

Equation 15 will be solved in section **??**.

2.3 Conservation of Energy and Angular Momentum

The conservation of energy is tested in the simulation.

3 Method

To solve a generic problem of n couple of differential equation we can perfome a so-called $Rugge\ Kutta$ or $Verlet$Algorithm. These two algorithms use a different approach to solve the problem. We start to discuss the Verlet Algortihm that we use.

We can perfome a Taylor expantion of the unknown position at the time $t+\Delta t$.

$$p_n(t+\Delta t) = p_n(t) + v_n(t)\Delta t + \frac{1}{2}a_n(t)\Delta t^2 + O(\Delta t^3) \tag{16}$$

Since form the formulas that we use to describe the solar system we can calculate only the acceleration of the planets we can't to go further than the second degree in the Taylor expansion. Always with a Taylor expantion we can describe the velocity at $t+\frac{\Delta t}{2}$

$$v_n(t+\frac{\Delta t}{2}) = v_n(t) + t + \frac{a_n(t)\Delta t}{2} \tag{17}$$

and the acceleration at $t+\Delta t$:

$$a_n(t+\Delta t) = \sum_{1}^{all\ planets} -\frac{G\ M_i}{(r_i(t+\Delta t))^2} \tag{18}$$

Since the velocity follow this equation $v(t+\Delta t) = v(t) + a(t)\Delta t$ we can write that:

$$v_n(t+\Delta t) = v_n(t+\frac{\Delta t}{2}) + \frac{1}{2}a_n(t+\Delta t)\Delta t; \tag{19}$$

The Verlet algorithm provides a good preservation of the sympletic form on phase space also if it doesn't require a big computational cost. This means it preserves the kinetic energy for huge period of time.

The Rugge Kutta method use only the first derivates, is for that reason that we slip the second order differential equation in two first order differential equation.

Before starting to explain this algorithm we make some observations:

$$\frac{dp(t)}{dt} = v(t)\ \ and\ \ \frac{dv(t)}{t} = a(t) \tag{20}$$

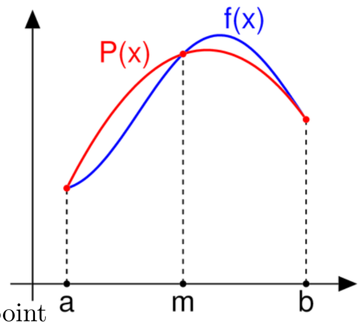
We discuss the case for solve the velocity knowing that it's the same for the position.

$$y(t) = \int a(t)dt\ \ and\ \ y_{i+1} = y_i + \int_{t_i} t_{i+1}a(t)dt \tag{21}$$

If we interpret the integral as the area under the curve we have that:

$$\int_{t_i} t_{i+1}a(t)dt \simeq \Delta t\ a(t_{i+\frac{1}{2}}) \tag{22}$$

Using the Simpson's formula we can amproximate an integral of a finit polinomius in this way:



$$\int_{t_i} t_{i+1} a(t)dt \simeq \frac{\Delta t}{6}\left[a(t_i) + 4a(t_{i+\frac{1}{2}}) + a(t_{i+1})\right] \tag{23}$$

To have a higher precision we can extimate more time the middle poit in this way:

$$k1 = a(t_i) \quad \rightarrow \quad y_{i+\frac{1}{2}} = y_i + \frac{\Delta t}{2}\, k1 \ \ \text{first aproximetion of middle point} \tag{24}$$

$$k2 = a(t_{i+\frac{1}{2}}) \quad \rightarrow \quad y_{i+\frac{1}{2}} = y_i + \frac{\Delta t}{2}\, k2 \ \ \text{second aproximetion of middle point} \tag{25}$$

$$k3 = a(t_{i+\frac{1}{2}}) \quad \rightarrow \quad y_{i+1} = y_i + \Delta t\ k3 \ \ \text{first aproximetion of point} \ \ i+1 \tag{26}$$

$$k3 = a(t_{i+1}) \quad \text{Note that it can be solve knowing the value of the position computed at the previus step} \tag{27}$$

Try overwrite toghedera
A grapghic interpreation of the algorithm is very usefull to understan better it:
prova ciao !!ee 33 32

4 Tuned Program

5 Energy states and probability functions

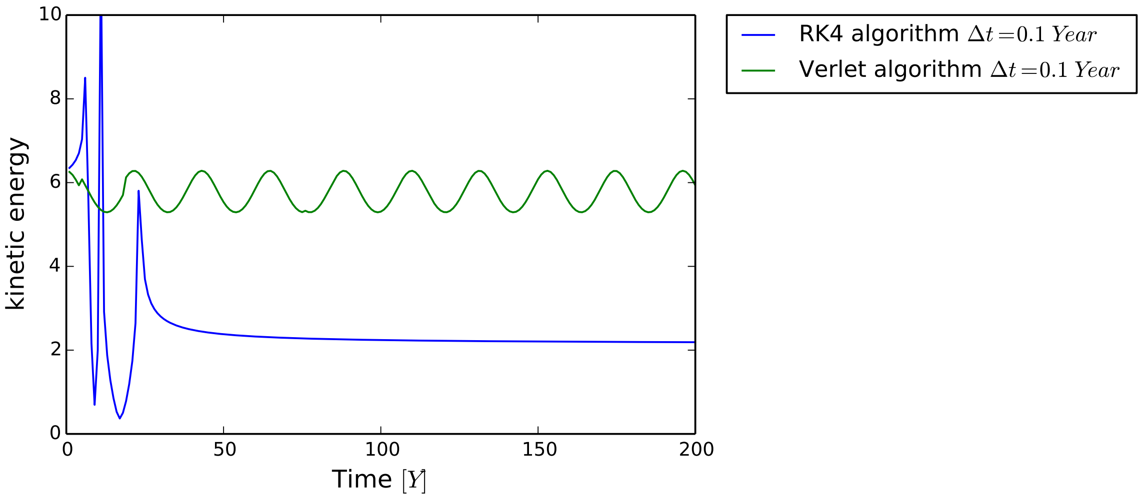
6 Analytic Solution

Prova hello!

7 Conculsions

8 References

- M. Hjorth-Jensen, Computational Physics, University of Oslo (2013).
- M. Hjorth-Jensen, Project3, University of Oslo (2013).
- www.cplusplus.com



10
8
6
4
2
0
0
50
100
150
200
kinetic energy
Time $[Y]$
RK4 algorithm $\Delta t = 0.1\ Year$
Verlet algorithm $\Delta t = 0.1\ Year$

graf