

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 occurring 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	2
2	Theoretical Concept	2
3	Method	3
4	Tuned Program	4
5	Energy states and probability functions	4
6	Analytic Solution	5
7	Conclusions	6
8	References	6

1 Introduction

how important differential equations are for science. often sets of coupled differential equations, example solar system. too complex for analytic aproaccce. 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

$$F_G = \frac{GM_{\text{Sun}}M_{\text{Earth}}}{r^2}, \quad (1)$$

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

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

and

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

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

$$\frac{dx(t)}{dt} = v_x \quad (3) \qquad \frac{v_x(t)}{dt} = \frac{F_{G,x}}{M_{\text{Earth}}} \quad (4)$$

$$\frac{dy(t)}{dt} = v_y \quad (5) \qquad \frac{v_y(t)}{dt} = \frac{F_{G,y}}{M_{\text{Earth}}} \quad (6)$$

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

$$\frac{r}{x} = \frac{F_G}{F_{G,x}} \quad (7a)$$

$$\frac{r}{y} = \frac{F_G}{F_{G,y}} \quad (7b)$$

and therefore

$$F_{G,x} = \frac{x}{r} F_G \quad (8a)$$

$$F_{G,y} = \frac{y}{r} F_G \quad (8b)$$

with F being the absolute value of the gravitational force.

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

$$F_{ZP} = \frac{M_{\text{Earth}} v^2}{r} = \frac{G M_{\text{Sun}} M_{\text{Earth}}}{r^2} = F_G, \quad (9)$$

with v being the orbital velocity of Earth.

3 Method

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

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

$$p_n(t + \Delta t) = p_n(t) + v_n(t)\Delta t + \frac{1}{2}a_n(t)\Delta t^2 + O(\Delta t^3) \quad (10)$$

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

$$v_n(t + \frac{\Delta t}{2}) = v_n(t) + t + \frac{a_n(t)\Delta t}{2} \quad (11)$$

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

$$a_n(t + \Delta t) = \sum_{i=1}^{\text{all planets}} -\frac{G M_i}{(r_i(t + \Delta t))^2} \quad (12)$$

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

$$v_n(t + \Delta t) = v_n(t + \frac{\Delta t}{2}) + \frac{1}{2}a_n(t + \Delta t)\Delta t; \quad (13)$$

The Verlet algorithm provides a good preservation of the symplectic 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 Runge Kutta method uses only the first derivatives, is for that reason that we split the second order differential equation in two first order differential equations.

Before starting to explain this algorithm we make some observations:

$$\frac{dp(t)}{dt} = v(t) \quad \text{and} \quad \frac{dv(t)}{dt} = a(t) \quad (14)$$

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

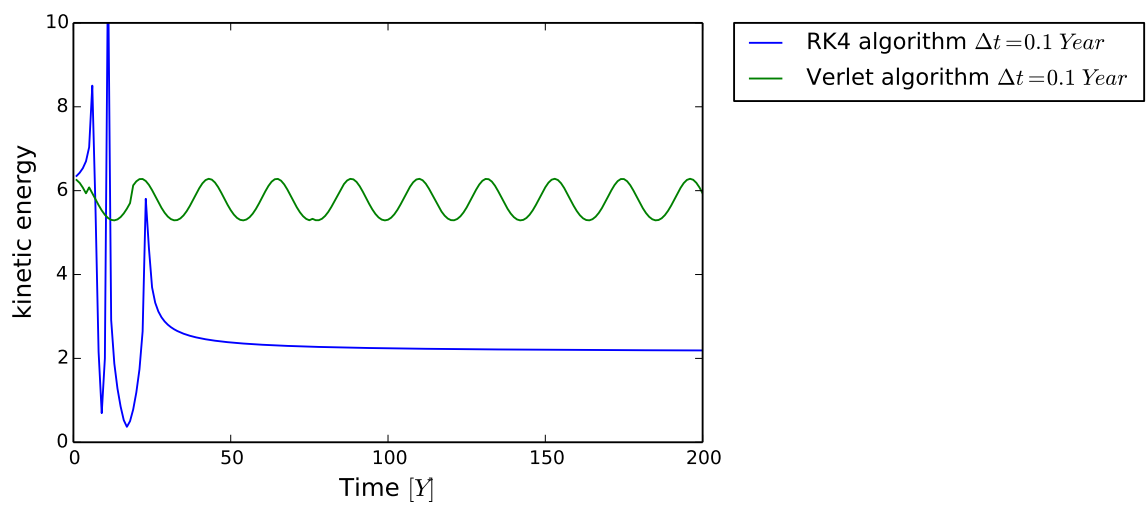
$$y(t) = \int a(t)dt \quad \text{and} \quad y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} a(t)dt \quad (15)$$

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

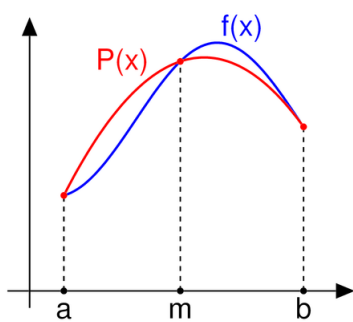
$$\int_{t_i}^{t_{i+1}} a(t)dt \simeq \Delta t a(t_{i+\frac{1}{2}}) \quad (16)$$

Using the Simpson's formula we can approximate an integral of a finite polynomial in this way:

$$55 \quad (17)$$



graf



Aproximation of Simposon's formula

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 Concussions

8 References

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