Building a model for the solar system using ordinary differential equations

Introduction.

The aim of this project is to develop a code for simulating the solar system using a widely popular algorithm for solving coupled ordinary differential equations, the so-called velocity Verlet algorithm. An important aspect of this project is to be able to object orient your code. There are several coupled ordinary differenatial equations where the basic equations, except for various physical constants and variables, are rather similar. Thus, write once and run many times, one of the central points of object orientation, is something which makes our program easier to extend and build upon when we add more planets or moons or other astronomical objects. The basic equations which govern the system are rather simple, a set of coupled equations that codify Newton's law of motion due the gravitational force.

In the first part however, we will limit ourselves (in order to test the algorithm) to a hypothetical solar system with the Earth only orbiting around the sun. The only force in the problem is gravity. Newton's law of gravitation is given by a force F_G

$$F_G = \frac{GM_\odot M_{\rm Earth}}{r^2},$$

where $M_\odot$ is the mass of the Sun and $M_{\rm Earth}$ is the mass of the Earth. The gravitational constant is G and r is the distance between the Earth and the Sun. We assume that the Sun has a mass which is much larger than that of the Earth. We can therefore safely neglect the motion of the Sun in this problem. In the first part of this project, your aim is to compute the motion of the the Earth using different methods for solving ordinary differential equations.

We assume that the orbit of the Earth around the Sun is co-planar, and we take this to be the xy-plane. Using Newton's second law of motion we get the following equations

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

and

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

where $F_{G,x}$ and $F_{G,y}$ are the x and y components of the gravitational force.

We will use so-called astronomical units when rewriting our equations. Using astronomical units (AU as abbreviation)it means that one astronomical unit of length, known as 1 AU, is the average distance between the Sun and Earth, that is 1 AU $= 1.5 \times 10^{11}$ m. It can also be convenient to use years instead of seconds since years match better the time evolution of the solar system. The mass of the Sun is $M_{\rm sun} = M_\odot = 2 \times 10^{30}$ kg. The masses of all relevant planets and their distances from the sun are listed in the table here in kg and AU.

Planet	Mass in kg	Distance to sun in AU
Earth	$M_{\mathrm{Earth}} = 6 \times 10^{24}$ kg	1AU
Jupiter	$M_{\mathrm{Jupiter}} = 1.9 \times 10^{27}$ kg	5.20 AU
Mars	$M_{\mathrm{Mars}} = 6.6 \times 10^{23}$ kg	1.52 AU
Venus	$M_{\mathrm{Venus}} = 4.9 \times 10^{24}$ kg	0.72 AU
Saturn	$M_{\mathrm{Saturn}} = 5.5 \times 10^{26}$ kg	9.54 AU
Mercury	$M_{\mathrm{Mercury}} = 3.3 \times 10^{23}$ kg	0.39 AU
Uranus	$M_{\mathrm{Uranus}} = 8.8 \times 10^{25}$ kg	19.19 AU
Neptun	$M_{\mathrm{Neptun}} = 1.03 \times 10^{26}$ kg	30.06 AU
Pluto	$M_{\mathrm{Pluto}} = 1.31 \times 10^{22}$ kg	39.53 AU

Pluto is no longer considered a planet, but we add it here for historical reasons. It is optional in this project to include Pluto and eventual moons.

In setting up the equations we can limit ourselves to a co-planar motion and use only the x and y coordinates. But you should feel free to extend your equations to three dimensions, it is not very difficult and the data from NASA are all in three dimensions.

http://www.nasa.gov/index.htmlNASA has an excellent site at http://ssd.jpl.nasa.gov/horizons.cgitoph From there you can extract initial conditions in order to start your differential equation solver. At the above website you need to change from **OBSERVER** to **VECTOR** and then write in the planet you are interested in. The generated data contain the x, y and z values as well as their corresponding velocities. The velocities are in units of AU per day. Alternatively they can be obtained in terms of km and km/s.

For the first system below involving only the Earth and the Sun, you could just initialize the position with say $x = 1$ AU and $y = 0$ AU.

a: The Earth-Sun system. We assume that mass units can be obtained by using the fact that Earth's orbit is almost circular around the Sun. For circular motion we know that the force must obey the following relation

$$F_G = \frac{M_{\mathrm{Earth}} v^2}{r} = \frac{G M_\odot M_{\mathrm{Earth}}}{r^2},$$

where v is the velocity of Earth. The latter equation can be used to show that

$$v^2 r = G M_\odot = 4\pi^2 \mathrm{AU}^3/\mathrm{yr}^2.$$

Discretize the above differential equations and set up an algorithm for solving these equations using Euler's forward algorithm and the so-called velocity Verlet method discussed in the lecture notes and lecture slides.

b: Writing an object oriented code for the Earth-Sun system. Write then a program which solves the above differential equations for the Earth-Sun system using Euler's method and the velocity Verlet method. Write these codes without object orientation first in order to make sure everything is running correctly. Thereafter you should start planning to object orient your code. Try to figure out which parts and operations could be written as classes and generalized. Your task here is to think of the program flow and figure out which parts can be abstracted and reused for many types of operations.

For those of you who will focus on the Molecular Dynamics version of Project 5, much of the structures developed here as well as the implementation of the Verlet algorithm, can be used in that project as well.

c: Test of the algorithms. Find out which initial value for the velocity that gives a circular orbit and test the stability of your algorithm as function of different time steps Δt. Make a plot of the results you obtain for the position of the Earth (plot the x and y values and/or if you prefer to use three dimensions the z-value as well) orbiting the Sun.

Check also for the case of a circular orbit that both the kinetic and the potential energies are conserved. Check also if the angular momentum is conserved. Explain why these quantities should be conserved.

Discuss eventual differences between the Verlet algorithm and the Euler algorithm. Consider also the number of FLOPs involved and perform a timing of the two algorithms for equal final times.

We will use the velocity Verlet algorithm in the remaining part of the project.

d: Escape velocity. Consider then a planet which begins at a distance of 1 AU from the sun. Find out by trial and error what the initial velocity must be in order for the planet to escape from the sun. Can you find an exact answer? How does that match your numerical results?

Try also to change the gravitional force, by replacing

$$F_G = \frac{GM_\odot M_{\mathrm{Earth}}}{r^2},$$

with

$$F_G = \frac{GM_\odot M_{\mathrm{Earth}}}{r^\beta},$$

where you let $\beta \in [2,3]$. What happens to the earth-sun system when β creeps towards 3? Comment your results.

e: The three-body problem. We will now study the three-body problem, still with the Sun kept fixed as the center of mass of the system but including Jupiter (the most massive planet in the solar system, having a mass that is approximately 1000 times smaller than that of the Sun) together with the Earth. This leads to a three-body problem. Without Jupiter, the Earth's motion is stable and unchanging with time. The aim here is to find out how much Jupiter alters the Earth's motion.

The program you have developed can easily be modified by simply adding the magnitude of the force betweem the Earth and Jupiter.

This force is given again by

$$F_{\text{Earth}-\text{Jupiter}} = \frac{GM_{\text{Jupiter}}M_{\text{Earth}}}{r^2_{\text{Earth}-\text{Jupiter}}},$$

where M_{Jupiter} is the mass of the sun and M_{Earth} is the mass of Earth. The gravitational constant is G and $r_{\text{Earth}-\text{Jupiter}}$ is the distance between Earth and Jupiter.

We assume again that the orbits of the two planets are co-planar, and we take this to be the xy-plane (you can easily extend the equations to three dimensions). Modify your first-order differential equations in order to accomodate both the motion of the Earth and Jupiter by taking into account the distance in x and y between the Earth and Jupiter. Set up the algorithm and plot the positions of the Earth and Jupiter using the velocity Verlet algorithm. Discuss the stability of the solutions using your Verlet solver.

Repeat the calculations by increasing the mass of Jupiter by a factor of 10 and 1000 and plot the position of the Earth. Study again the stability of the Verlet solver.

f: Final model for all planets of the solar system. Finally, using our Verlet solver, we carry out a real three-body calculation where all three systems, the Earth, Jupiter and the Sun are in motion. To do this, choose the center-of-mass position of the three-body system as the origin rather than the position of the sun. Give the Sun an initial velocity which makes the total momentum of the system exactly zero (the center-of-mass will remain fixed). Compare these results with those from the previous exercise and comment your results. Extend your program to include all planets in the solar system (if you have time, you can also include the various moons, but it is not required) and discuss your results. Use the above NASA link to set up the initial positions and velocities for all planets.

g: The perihelion precession of Mercury. An important test of the general theory of relativity was to compare its prediction for the perihelion precession of Mercury to the observed value. The observed value of the perihelion precession, when all classical effects (such as the perturbation of the orbit due to gravitational attraction from the other planets) are subtracted, is $43''$ (43 arc seconds) per century.

Closed elliptical orbits are a special feature of the Newtonian $1/r^2$ force. In general, any correction to the pure $1/r^2$ behaviour will lead to an orbit which is not closed, i.e. after one complete orbit around the Sun, the planet will not be at exactly the same position as it started. If the correction is small, then each orbit around the Sun will be almost the same as the classical ellipse, and the orbit can be thought of as an ellipse whose orientation in space slowly rotates. In other words, the perihelion of the ellipse slowly precesses around the Sun.

You will now study the orbit of Mercury around the Sun, adding a general relativistic correction to the Newtonian gravitational force, so that the force becomes

$$F_G = \frac{GM_{\text{SunM}_{\text{Mercury}}}}{r^2}\left[1 + \frac{3l^2}{r^2c^2}\right]$$

where M_{Mercury} is the mass of Mercury, r is the distance between Mercury and the Sun, $l = |\vec{r} \times \vec{v}|$ is the magnitude of Mercury's orbital angular momentum per unit mass, and c is the speed of light in vacuum. Run a simulation over one century of Mercury's orbit around the Sun with no other planets present, starting with Mercury at perihelion on the x axis. Check then the value of the perihelion angle θ_{p}, using

$$\tan\theta_{\mathrm{p}=\frac{\mathrm{y_p}}{\mathrm{x_p where}}}$$

$\mathrm{x_p}$ (y_{p}) is the x (y) position of Mercury at perihelion, i.e. at the point where Mercury is at its closest to the Sun. You may use that the speed of Mercury at perihelion is 12.44 AU/yr, and that the distance to the Sun at perihelion is 0.3075 AU. You need to make sure that the time resolution used in your simulation is sufficient, for example by checking that the perihelion precession you get with a pure Newtonian force is at least a few orders of magnitude smaller than the observed perihelion precession of Mercury. Can the observed perihelion precession of Mercury be explained by the general theory of relativity?