Leap Frog Solar System

David-Alexander Robinson Sch.[*]
08332461

20th November 2011

Contents

Abstract

The *Leap Frog* integrator was used to solve the three body problem for a simple sun-earth-moon solar system as implemented with C++ programing code. A *Class* was written to read the planet data from an input file, as specified in the command line, and store it. A *Void* function was written to print the total energy of the system. The energy versus time was plotted for a range of values of step size h. It was found that the energy is conserved at -430±150 units.

Moreover, a void function was written to calculate the angular momentum. Again, the angular momentum versus the time was plotted for a range of values of h, and it was found that angular momentum was conserved at -4751±2 units.

Finally, the masses, initial positions and initial velocities for a stable three body system were found. The year-month ratio was found to be 1:50.6.

1 Introduction & Theory

1.1 The Leap Frog Integrator

To integrate equations of motion, such as those that define Hamiltonian dynamics, a symplectic integrator must be defined, as otherwise, for a simple Euler Integrator say, the amplitude of any oscillations will increase exponentially in time. The simplest such integrator is the Leap Frog, where we put

$$\begin{aligned} x_{k+1/2} &= x_k + 1/2hp_k \\ p_{k+1} &= p_k + hf(x_k + 1/2, t + 1/2h) \\ x_k + 1 &= x_{k+1/2} + 1/2hp_{k+1} \end{aligned} \tag{1}$$

where h is the step size, and f is the function.

1.2 Class

Like the concept of a data structure, a class is a method of holding data, which may hold both numbers and functions. By declaring a Class *Public* the members of the class may be accessed from anywhere else.

1.3 Void

A Void type function can be used to perform tasks, such as an iterative calculation or printing data, while not providing a result to the command that called it. This way they can be used to perform a simple closed task, for which the main function does not rely on the result of this task in any way.

2 Experimental Method

2.1 Conservation of Energy

The total energy of the system is given by

$$
\begin{aligned}
E &= T + U \\
&= \frac{1}{2}\sum_a m_a|\vec{v_a}|^2 - G\sum_{b,a<b}\frac{m_a m_b}{r_{ab}}
\end{aligned}
\tag{2}
$$

where a and b are the planet body indices, T and U are the kinetic energy and potential energy terms respectively, m_a is the mass of the a-th body, v_a is the velocity of the a-th body, r_{ab} is the distance between the a-th and b-th body, and G is Newton's Gravitational Constant.

Code was written to iterate over every planet body which calculates the norm of the displacement from planet body a to b, given that $a \neq b$, by iterating over the two components of displacement, and summing the squares. The gravitational potential energy for the interaction between planet bodies $a \neq b$ is then calculated using this value. The kinetic energy of each planet body can be calculated independently, and finally these six values are added together to give the total energy.

It is necessary to note that the three-arrays which store the kinetic and gravitational energy must be initialised to zero, to avoid the addition of unwanted double precision numbers.

2.2 Conservation of Angular Momentum

The total angular momentum for a three-body system which rotates in two dimensions only is given by

$$
\begin{aligned}
\vec{L}_{total} &= \sum_a \vec{r} \times \vec{p} \\
&= \sum_a [m((d_x \times v_y) - (d_y \times v_x)]_a
\end{aligned}
\tag{3}
$$

where again a is our planet body index, m_a is the mass of the a-th body, $[v_i]_a$ is the i-th component of the a-th body's velocity, and

$$[d_x]_a = [r_x]_a - [r_x]_{sun}$$

such that the angular momentum is calculated from the origin in the sun's rest frame.

This is implemented by again iterating over the planet body index, and computing the above. The values are stored as a three-array, over which a sum is performed to get the total angular momentum as a double precision number.

2.3 The Sun-Earth-Moon System

To find the number of years passed, code was written to store the coordinates of the earth-planet before performing the update step in a two-array. The old x-coordinate and the updated x-coordinate were then multiplied together, and if the resultant was less then zero, the earth-planet was known to have crossed the y-axis, and so a half of a year had passed. A each time this happens 0.5 is added to a newly defined double called *year*.

To calculate the number of months passed two double precision numbers were defined to store the value of the cross product of the sun-earth displacement with the sun-moon displacement, again, both before and after the update step

$$[r_{sun\vec{-}earth} \times r_{sun\vec{-}moon}]_{old} = [(r_{sun_x} - r_{earth_x})(r_{sun_y} - r_{moon_y}) - (r_{sun_y} - r_{earth_y})(r_{sun_x} - r_{moon_x})]_{old}$$

and similarly for the cross product after updating.

If the product of these two numbers was found to be less than zero, it is know that the sun-earth-moon system has crossed an alignment. A month here is defined as the time taken between subsequent alignments of sun-earth-moon system in that order. Thus every time this product is less than zero a half month has passed.

3 Results & Analysis

3.1 Conservation of Energy

The energy of the system was found to conserve within a range of -430±150 units. The following graphs were plotted for a range of step sizes h

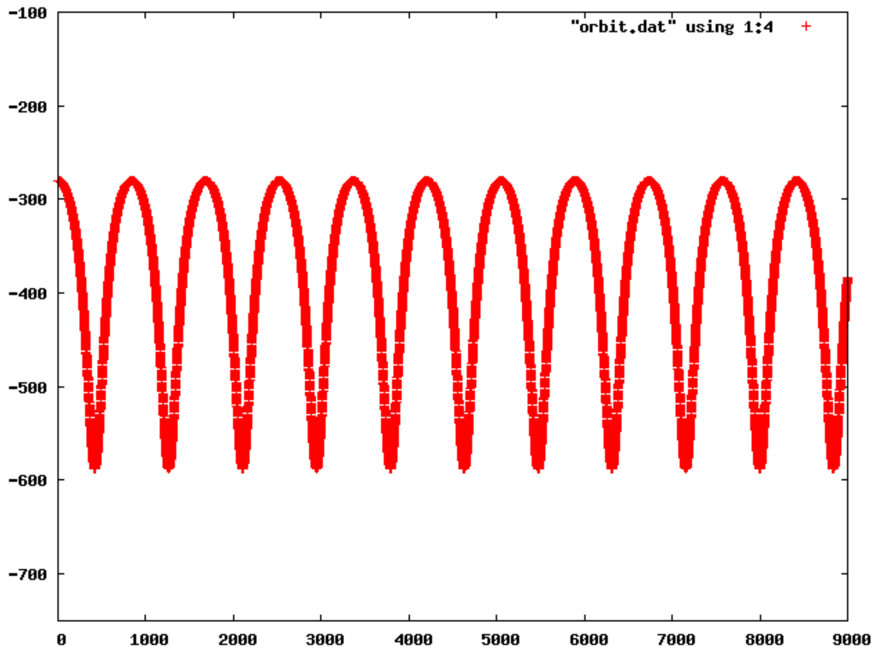

Figure 1: Total Energy vs. Time for $h = 0.1$

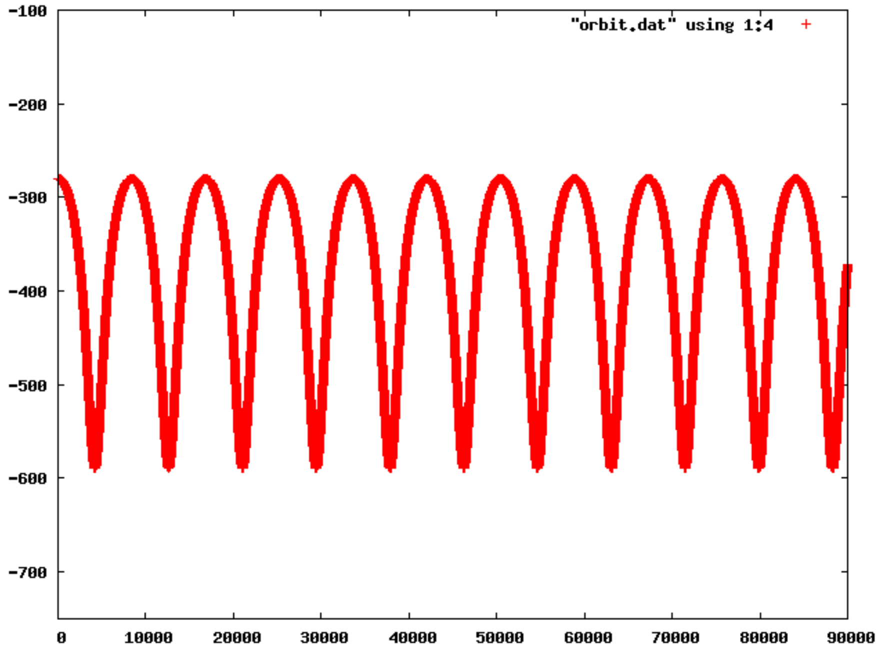

Figure 2: Total Energy vs. Time for $h = 0.01$

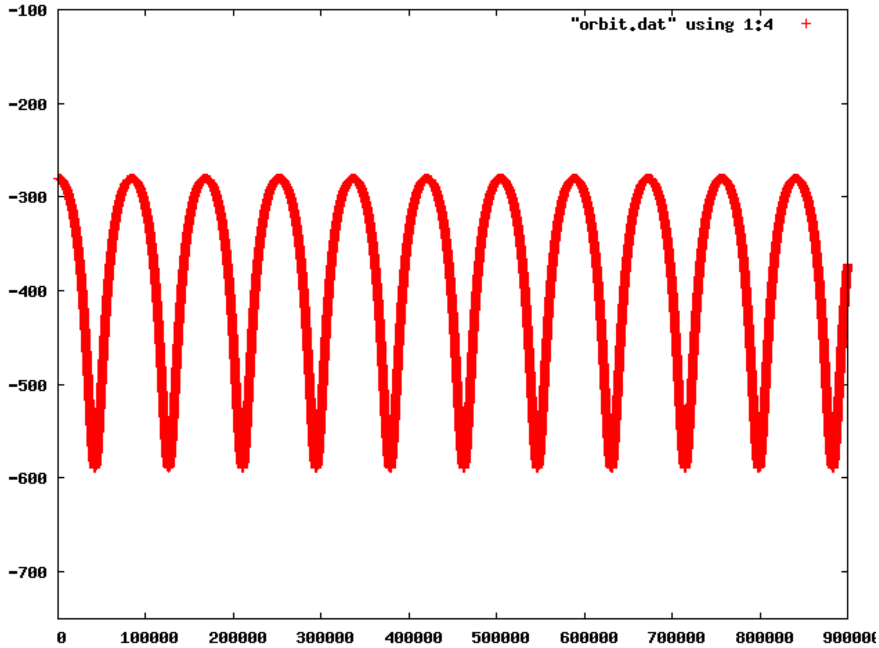

Figure 3: Total Energy vs. Time for $h = 0.001$

3.2 Conservation of Angular Momentum

The angular momentum was found to be conserved within a very narrow range of -4751±2 units. Again, the following graphs were plotted for a range of values of the step size h

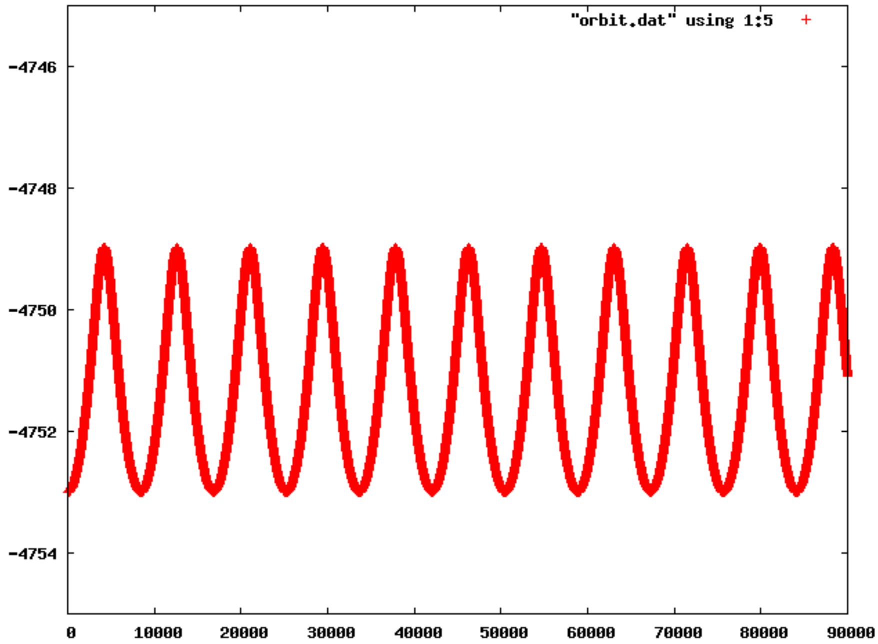

Figure 4: Total Angular Momentum vs. Time for $h = 0.1$

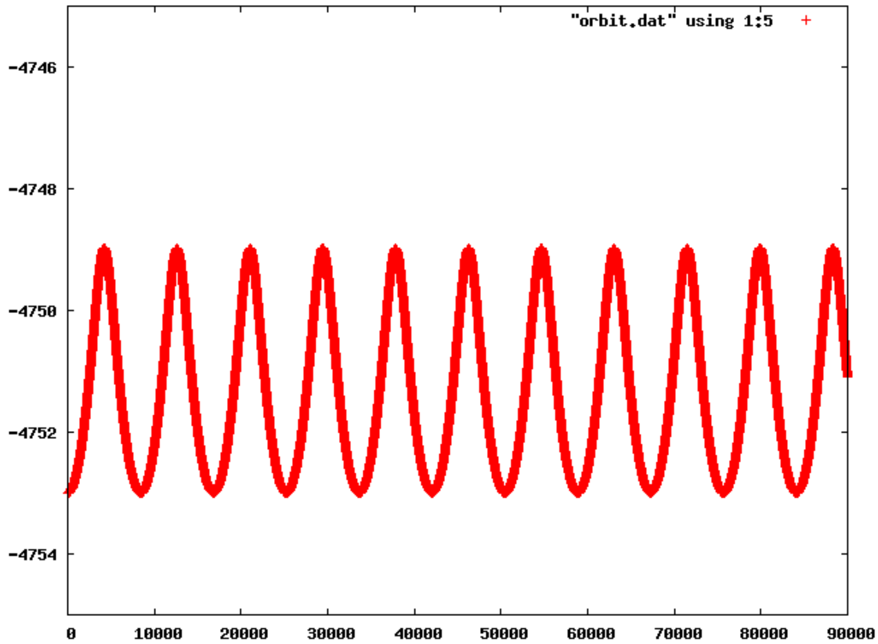

Figure 5: Total Angular Momentum vs. Time for $h = 0.10$

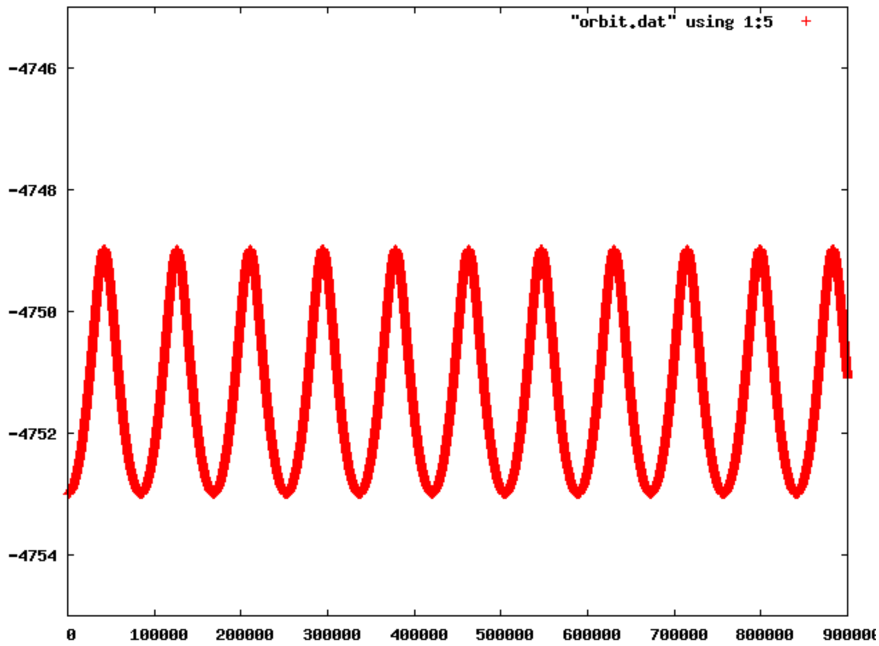

Figure 6: Total Angular Momentum vs. Time for $h = 0.001$

3.3 The Sun-Earth-Moon System

The following initial data was found to produce a stable sun-earth-moon system

Table 1: Initial Sun-Earth-Moon Data

Planet	Mass	Initial Position	Initial Velocity
Sun	3500	$0.0\ \hat{i} + 0.0\ \hat{j}$	$0.0\ \hat{i} + 0.0\ \hat{j}$
Earth	2	$0.0\ \hat{i} + 250.0\ \hat{j}$	$9.5\ \hat{i}$ - $4.0\ \hat{j}$
Moon	0.001	$0.0\ \hat{i} + 251.0\ \hat{j}$	$11.915\ \hat{i}$ -$4.35\ \hat{j}$

It was found that for smaller initial distance between the Sun and Earth, larger initial velocity was required. Moreover, for a smaller Sun-Earth initial distance, an initial Earth-Moon distance of two orders of magnitude smaller was needed. In addition, it was found that the ratio of the mass of the sun to that of the earth to that of the moon must be of the order of 1000 : 1 : 0.001.

For this system the ratio of a year to a month was found to vary somewhat with the step size h. The following ratios were found

Table 2: Year to Month ratio for varying step size h

Step Size h	Year to Month ratio
0.01	1:41
0.001	1:56
0.0001	1:259

Moreover, averaging over a ten year period, and using a step size of 0.01 gave an average year to month ratio of 1 : 50.6.

4 Conclusions

Using the Leap Frog integrator and the C++ programing language it was found that the three-body problem can be effectively solved, and a range of dynamical variables calculated. The method can be generalised such that a single program can solve any number of planets in a gravitating system by defining a Class to read data from an input file, which is specified in the command line. Moreover, C++ allows the construction of void functions to compute and print these dynamical variables, such as energy and angular momentum, without their use in the main function of the program.

For the case of a Sun-Earth-Moon system it was found that both Energy and Angular Momentum were conserved, with values of -430±150 and -4751±2 units respectively. This is in agreement with the laws of Classical Physics.

Finally, it was found that the system is only stable for certain initial values of the planet body masses and initial positions and velocities. For such a stable system code may be written to calculate the months and years passing in the system, and for the system considered, the ratio of years to months was found to be 1 : 50.6.