Exploring Solar System Using a Personal Computer

Vikram Vyas
Physics Department, St. Stephen's College, Delhi University
Email:vvyas@stephens.du.ac.in

Abstract

This module introduces the idea of solving for the motion of more than two bodies under the influence of their mutual gravitational interaction using a computer. Through such techniques one can address questions like, is our solar system stable?

Contents

1 Introduction

> *"blind fate could never make all the Planets move one and the same way in Orbs concentric, some inconsiderable irregularities excepted, which could have arisen from the mutual Actions of Planets upon one another, and which will be apt to increase, until this System wants a reformation"* **Newton, Opticks, Book III**

One of the great triumphs of physics is the Newton's solution for the motion of two objects which interact with each other through their mutual gravitational force. This solution provides us, for example, with a qualitative and quantitative understanding of the motion of the earth around the sun. The solution also raises the question, how will the motion of earth get modified when we take into account the force on it due to other objects in our Solar System? Our first instinct would be that their effect should be very small and negligible. After all the orbit of earth calculated on the basis of the two-body solution matches very well with its real orbit, also the force due to even the most massive of these objects, Jupiter, is extremely small (see Exercise 2) Though these effects are small but are they negligible over the life span of the Solar System? In fact Newton himself was keenly aware of these problems as the quote in the beginning of the section shows.

Was Newton right? Do we need more than a blind fate for a stable Solar System? In fact, is Solar System stable? How do we answer these questions. One way would be to try and extend our two-body solution to a three-body case. Unfortunately there does not exist, as far as we know, analytical solution for a three-body problem. To appreciate the difficulty involved, consider the motion of the Sun, the Earth and the Jupiter under their mutual gravitational interaction. Now you can easily convince yourself that the force acting on the Earth is no longer a central force and the motion of earth cannot be reduced to an effectively one-dimensional problem as we could in the case of the motion of a particle in the central force field (see Exercise 1).

In the absence of an exact solution we can try to "correct" our two-body solution by considering a "small" disturbance or perturbance due to a third body. This approach too has its limitation, there is no proof that such corrections to the elliptical orbits do not accumulate over the time periods that we are interested in.

Fortunately there is in fact a universal tool that can help us answer these questions. The tool is the computer, and the idea is to use, as explained in Appendix (2) and (3), a computer to solve the Newton's equations of motion for the bodies making up the solar system. The numerical solution to these equations have lead to some surprising new insights into the question of the long term fate of our solar system. Therefore our immediate task is to understand how to solve equations of motion on a computer.

2 Solving Equations of Motion Using a Computer

> *"In thinking and trying out ideas about "what is a quantum field theory", I found it very helpful to demand that a correctly formulated field theory be soluble by computer, the same way an ordinary differential equation can be solved on a computer, namely with arbitrary accuracy in return for sufficient computing power"*
>
> **Ken Wilson, Nobel Prize Acceptance Speech 1982.**

To formulate a problem in a manner which a computer can solve requires a clear, and often deep, understanding of the underlying laws. If we want to solve the equations of motion on a computer we have to have a clear understanding of Newton's second law of motion. Consider first the simple case of the motion in one dimension. Let us say that at some initial time, which we take as zero, we know the position of the particle $x(0)$ and its velocity $(\frac{dx}{dt})_{t=0} = v(0)$, and we would like to find it's position after a very short interval of time, Δt, that is we want to know what is $x(\Delta t)$? We can immediately give an approximate answer, using the definition of velocity,

$$x(\Delta t) \quad = \quad x(0) + v(0)\Delta t, \tag{1}$$

we expect our answer to me more and more accurate as we consider smaller and smaller time interval Δt. What is a short interval, more often than not in a problem of interest there is an intrinsic time scale τ which can be determined by simple dimension analysis, and what we require is $\Delta t \ll \tau$, this important point is further explored in Exercise- **??**. Now let us say we want to repeat this process and find the position of the particle at $t = 2\Delta$, we immediately face a problem, to find $x(2\Delta t)$ we need $v(\Delta t)$. How do we find that? Well, we can use the definition of the acceleration a to write

$$v(\Delta t) = v(0) + a(0)\Delta t, \tag{2}$$

but what is the value $a(0)$? This is as far as kinematic can take us, to make further progress we have to look at the environment of the particle to find out, what is causing the acceleration, what are the forces acting on the particle, this is the heart of the Newtonian program [R. P. Feynman]. It **gives** us acceleration

$$a(t) = \frac{1}{m} F(t), \tag{3}$$

where $F(t)$ is the total force acting on the particle, and m is its mass. Now we can find the $v(\Delta t)$

$$v(\Delta t) = v(0) + \frac{1}{m} F(0)\Delta t, \tag{4}$$

and this in turn allows us to find $x(2\Delta t)$ as

$$x(2\Delta t) = x(\Delta t) + v(\Delta t)\Delta t. \tag{5}$$

Now one can continue with this process and solve for the motion of the particle for any finite time interval, most importantly we can carry out this procedure for any force.

2.1 Simple Harmonic Oscillator on a Computer

Let us use our understanding of Newton's Laws to solve equation of motion on a computer. To illustrate this, we will consider the familiar problem of the motion a particle in one dimension under the influence of a linear restoring force. The equation of motion is

$$m\frac{d^2x}{dt^2} = -kx, \tag{6}$$

where m is the mass of the particle and k is a constant ("spring" constant). We would like to find the function $x(t)$ which solves Eq.(6) *and* satisfies the initial conditions

$$\begin{aligned} x(0) &= x_0 \\ \left(\frac{dx}{dt}\right)_{t=0} &= v_0. \end{aligned} \tag{7}$$

Let us rewrite the equation of motion as an expression for acceleration

$$\frac{d^2x}{dt^2} = -\omega_0^2 x, \tag{8}$$

where we have defined

$$\omega_0^2 = \frac{k}{m}. \tag{9}$$

Dimension analysis tells us that ω_0 has the dimension of inverse time, thus there is an *intrinsic* time scale in the problem given by

$$T = \omega_0^{-1} = \sqrt{\frac{m}{k}}. \tag{10}$$

(What is the physical significance of T?) With this in mind let us define a new time variable

$$\tau = \omega_0 t,$$

note by construction τ is dimensionless. Let us express velocity and acceleration in terms of τ

$$\frac{dx}{d\tau} = \frac{dx}{dt}\frac{dt}{d\tau} = \frac{1}{\omega_0}\frac{dx}{dt},$$

similarly

$$\frac{d^2x}{d\tau^2} = \frac{1}{\omega_0^2}\frac{d^2x}{dt^2}.$$

Using these results we can write the equation of motion as

$$\frac{d^2x}{d\tau^2} = -x(\tau). \tag{11}$$

This is the equation we would like to solve on the computer, with the corresponding initial conditions as

$$x(\tau = 0) = x(t = 0) = x_0, \tag{12}$$

and

$$\left(\frac{dx}{d\tau}\right)_{\tau=0} = \frac{1}{\omega_0}\left(\frac{dx}{dt}\right)_{t=0} = \frac{v_0}{\omega_0} \equiv u_0. \tag{13}$$

What does it mean to solve this equation on a computer? Given the initial position x_0 and velocity u_0 at time $\tau = 0$, we would like to know what is the value of x at any other arbitrary value of $\tau = \bar{T}$.

2.2 Discretizing the time

From our discussion in the beginning of this section we know that to obtain the value of $x(\tau = \bar{T})$ we have to successively obtain the value of $x(\tau)$, starting from its value at $\tau = 0$, for small time intervals $\Delta\tau$ till we reach $\tau = \bar{T}$. Its important to note that because τ is a dimensionless variable, small $\Delta\tau$ means that $\Delta\tau \ll 1$. Thus, one reasonable choice is

$$\Delta\tau = 0.01.$$

Let us call $x_0 = x(0)$, $x_1 = x(\Delta\tau)$, $\ldots$, $x_N = x(N\Delta\tau) = x(\bar{T})$ where

$$N = \frac{\bar{T}}{\Delta\tau}.$$

2.3 Leap Frog Method

We are now in a position to calculate x_1 and $u(\Delta\tau)$

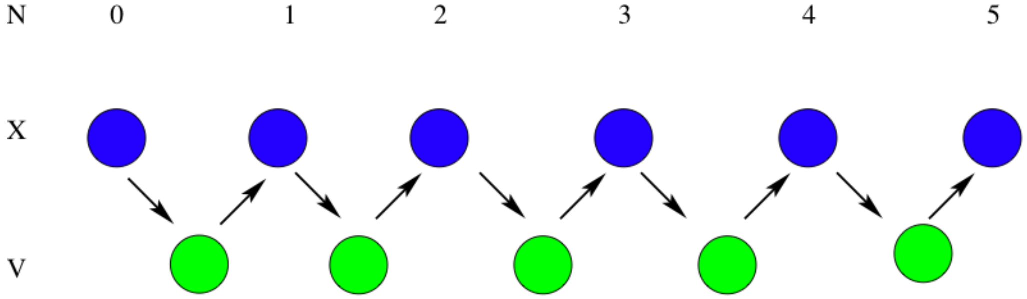
$$\begin{aligned} x_1 &= x_0 + u(0)\Delta\tau \\ a(0) &= -x_0 \\ u(\Delta\tau) &= u_0 + a_0\Delta\tau, \end{aligned}$$

where a refers to the acceleration. Repeating this procedure we can calculate x_2 and $u(2\Delta\tau)$

$$\begin{aligned} x_2 &= x_1 + u(\Delta\tau)\Delta\tau \\ a(\Delta\tau) &= -x_1 \\ u(2\Delta\tau) &= u(\Delta\tau) + a(\Delta\tau)\Delta\tau. \end{aligned}$$

Notice, to calculate $x_2 = x(2\Delta\tau)$ we used the velocity $u(\Delta\tau)$, but we know the velocity is changing so it might be a better idea to use the velocity not at the beginning of the interval but in the middle of the interval at $\tau = \frac{\Delta\tau}{2}$. Let us define $u_1 = u(\frac{\Delta\tau}{2})$ and then $u_2 = (\frac{\Delta\tau}{2} + \Delta\tau)$. Once we do that then we observe that we can improve out calculation of acceleration too, in calculating u_2 from u_1 we can use the value of acceleration evaluated at the middle of the interval $(\frac{\Delta\tau}{2}, \frac{3}{2}\Delta\tau)$ which is just $\Delta\tau$! Thus, with a simple reorganising of our calculation we can improve our accuracy, effectively halving our time interval $\Delta\tau$, *without* increasing number of step required to reach x_N. This reorganizing of our calculation is referred as the leap-frog method. To start

Figure 1: Leap Frog Method



the leap-frog process we just have to do one additional calculation, which is

$$\begin{aligned} a_0 &= -x_0, \\ u_1 &= u_0 + a_0 \frac{\Delta\tau}{2}. \end{aligned}$$

The above procedure can be easily converted into a computer algorithm, as you will do in project-1.

2.4 Round off Errors

The above example shows that solving of equation of motion can be reduced to a repetitive arithmetical operations which can be carried out on a computer. But there is a catch, on a computer we cannot represent a real number like π or $\frac{1}{3}$ exactly, as a result real numbers have to be approximately represented, or rounded off. This approximation results in the error in our arithmetic operations. For example, a typical personal computer will not be able to distinguish between 1 and $1+\epsilon$ where $\epsilon \sim 10^{-16}$. Another source of error, which you may have noticed is due to the small but finite value of the time interval $\Delta\tau$. This error can of course be reduced by reducing the size of $\Delta\tau$, but that would increase the number of intermediate steps required to reach the value $x(\bar{T})$, that means more arithmetical operations than before and a corresponding increases in the round off error. The art of numerical solution is to find a way to minimize the error due to discritizing, without overly increasing the roundoff errors. But, how can we quantify the error in our solution, particularly in the cases where there are no known analytical solutions? Here two important properties of Newton's law help us.

2.5 Conservation of Energy

If the forces acting on the particle are conservative then we know the total energy must be conserved. From the initial conditions we can calculate the initial value of energy which must remain constant as we evolve our system. The deviation from the initial value is than one measure of the error. In project 1 you will explore this for a non-linear oscillator.

2.6 Time reversal

Another important property of the equation of motion for conservative forces is the time reversal property. To understand what this property means, consider a particle that starts with an initial position $x_i = x(0)$ and an initial velocity $u_i = u(0)$ and let its position and velocity at latter time $\bar{T}$ be $x_f = x(\bar{T})$ and $u_f = u(\bar{T})$, now if we *reverse* its velocity, so that its velocity is $-u_f$, and follow the motion, then after a time interval $\bar{T}$ it will come back to the starting position x_i and will have velocity $-u_i$. In other words by reversing the velocity the particle retraces its trajectory, you will prove this important property in Exercise - 4. Our numerical solution should also satisfy this property, this we can check even if we do not know the exact analytical solution of the problem.

3 Solving N-body Problem on a Computer

> *"An intelligence knowing, at a given instant of time, all forces acting in nature, as well as the momentary positions of all things*

> *of which the universe consists, would be able to comprehend the motions of the largest bodies of the world and those of the smallest atoms in one single formula, provided it were sufficiently powerful to subject all data to analysis. To it, nothing would be uncertain; both future and past would be present before its eyes.”*
>
> **Laplace**

Having understood that Newton’s equation of motion are amenable to a numerical solution, we are in a position to make a more realistic model of our solar system which takes into account not only the interaction between a planet and the sun but also includes the mutual gravitational interaction with the rest of the objects that make up the solar system, which are the planets, their moons, and the asteroids. The equations of motion that describe the solar system are

$$\mathbf{a}_i(t) = \frac{d^2\mathbf{x}_i}{dt^2} = -G \sum_{j=1, j\neq i}^{N} \frac{m_j}{|\mathbf{x}_i - \mathbf{x}_j|^3}(\mathbf{x}_i - \mathbf{x}_j), \tag{14}$$

where $\mathbf{x}_i$ is the position vector of the i^{th} body with respect to suitably chosen origin, m_i is the mass of the body and G is the Gravitational constant. The sum on the left hand side is over all the bodies except the i^{th} body whose acceleration we are evaluating. The label i takes value from 1 to N, where N is the total number of bodies that form our solar system. Therefore we have to solve $3N$ coupled differential equations to obtain the future or the past of our solar system.

We can use *exactly* the same procedure for solving these equations as we used for solving the equation of motion for a simple harmonic oscillator. The key point of Newtonian dynamics is that it provides us with the acceleration, knowing the initial conditions we can use this acceleration to find the position of the various particles at latter time Δt. From the point of view of solving the equations, the only difference between the simple harmonic oscillator and the solar system is in the expression for the acceleration. In fact we can immediately set up an iterative procedure for solving Eq.(14)

$$\begin{aligned} \mathbf{x}_i(t+\Delta t) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+\frac{\Delta t}{2})\Delta t, \\ \mathbf{v}_i(t+\frac{\Delta t}{2}) &= \mathbf{v}_i(t-\frac{\Delta t}{2}) + \mathbf{a}_i(t)\Delta t. \end{aligned} \tag{15}$$

The all important $\mathbf{a}_i(t)$ is given by (14.)You will notice that this is nothing but the leap-frog method (See Fig. 1). To start our iterative process we of course need the initial conditions, the initial positions and the initial velocities of all the bodies that make up our solar system: $\mathbf{x}_{i0} = \mathbf{x}_i(0)$ and $\mathbf{v}_{i0} = \mathbf{v}_i(0)$, and for implementing the leap-frog method we need one additional calculation

$$\mathbf{v}_i(\frac{\Delta t}{2}) = \mathbf{v}_{i0} + \mathbf{a}_i(0)\frac{\Delta t}{2}.$$

To find the position of all the bodies that make up the solar system at some latter time T, which might be million years from now, we have to just iterate Eq. (15) for $\frac{T}{\Delta t}$ number of times. Similarly, to find the position of the bodies that make our present solar system million years in the *past* we just have to *reverse* the initial velocities and then iterate Eq. (15). So indeed we seem to have realised Laplace's dream, with apparently both the past and future of solar system can be determined by simple repetitive arithmetic operations.

4 Stability of a Dynamical System

We have an intuitive notion of stability of a static object. For example, we consider a pencil lying horizontally on a table to be in a stable position, a gentle tapping on the table does not flip the horizontal lying pencil into a vertical position. We consider the same pencil to be unstable when it is balanced vertically on its sharp tip, a slightest of disturbance will flip it to a horizontal position. To answer the question raised in the introduction, that will the Earth continue to move in a closed orbit even after, say a half-billion year, requires that we extend the notion of stability to the motion of interacting bodies, like the sun and the planets that make up the solar system.

4.1 Regular and Chaotic Systems

The systems which can be described using Newton's equation of motion with conservative forces, falls naturally into two categories, the regular and the chaotic. To delineate these two kinds of system, consider a system consisting of just one particles. The initial conditions for such a system will be given by a pair of initial position and velocities, $(\mathbf{x}_0, \mathbf{v}_0)$. Similarly at a latter time the system can be described by giving its position and velocity $(\mathbf{x}(t), \mathbf{v}(t))$. Now we change the initial conditions by an infinitesimally small amount so that our new initial conditions are $(\mathbf{x}_0 + \delta\mathbf{x}_0, \mathbf{v}_0 + \delta\mathbf{v}_0)$, and correspondingly the system will be described at latter time by $(\mathbf{x}(t) + \delta\mathbf{x}(t), \mathbf{v}(t) + \delta\mathbf{v}(t))$. If the system is **regular** or integrable then we find

$$\begin{aligned} \delta\mathbf{x}(t) &\propto t, \\ \delta\mathbf{v}(t) &\propto t, \end{aligned} \tag{16}$$

which means that the new trajectory deviates from the original trajectory slowly, linearly with time. On the other hand, if the system is **chaotic** then

$$\begin{aligned} \delta\mathbf{x}(t) &\propto \exp(\frac{t}{t_l}), \\ \delta\mathbf{v}(t) &\propto \exp(\frac{t}{t_l}), \end{aligned}$$

where t_l is called the Lyapunov time. For a regular system small change in the initial condition leads to a "small" change in the final state, while for a chaotic system, on a time scale greater than t_l, a small change in the initial conditions leads to a "large" change in the final state. Chaotic systems are systems that are very sensitive to the initial conditions, but at the same time on the time scales less than t_l a chaotic system behaves like a regular system. This classification of dynamical system into regular and chaotic systems can be extended to systems with more than one particle. If the deviation from the initial condition grow only linearly we will call such a system regular, and if the deviation grows exponentially then the system is chaotic.

In predicting the future behaviour of a system, like the solar system, we need the initial conditions, the positions and the velocities of all the bodies making up the solar system, but our knowledge of these initial conditions is not prefect, there is some uncertainty in their values, and these small errors in our initial knowledge will grow either linearly or exponentially depending on whether the system is regular or chaotic. For chaotic system we will loose all predictability for a time scale much greater than the Lyapunov time, no matter how small our initial error be at $t \gg t_l$ they would have grown so much that the trajectory predicted based on our initial conditions and the actual trajectory could well be qualitatively different. Therefore in predicting the future fate of the solar system it is important to know that whether the solar system is a regular system or a chaotic system.

4.2 Central force: An example of a regular system

Perhaps the most famous and important example of a regular system is the system consisting of two bodies interacting through a central force. Qualitatively this can be seen from the fact the radial motion of the corresponding one body problem is completely described in terms of the effective potential and the initial total energy. A small change in the initial position and initial velocity would lead to a small change in the effective potential and a small change in the total energy. In fact a hallmark of regular or integrable system is the existence of sufficiently large number of conserved quantities, recall from your solution of the central force problem the role that conservation of energy and angular momentum played in integrating the equations of motions. In Exercise (1) you will see that how the situation changes when you have more than two interacting objects.

You can explore the idea of the stability of the orbits for central force further in Ex.(3), but more in the spirit of this module you can quantitatively explore the effect of small change in the initial conditions in a system of two bodies interacting through gravitational force using a computer in the project(3).

5 Is Solar System a Stable System?

> *"Speaking more precisely, given an arbitrary accuracy, no matter how precise, one can find a time long enough that we cannot make predictions valid for that long a time."*
>
> **Feynman, Feynman Lectures in Physics Vol-1**

We are in a position to try and answer the question that we posed in the introduction, can the small effect of the gravitational interaction with the rest of the bodies that make up the solar system modify the orbit of the earth in a manner that eventually it is no longer a closed elliptical orbit? We can make the question more sharper by writting the equation of motion for Earth as

$$\frac{d^2\mathbf{x}_E}{dt^2} = -G\frac{M_{sun}}{|\mathbf{x}_E - \mathbf{x}_{sun}|^3}(\mathbf{x}_E - \mathbf{x}_{sun}) + \delta\mathbf{a}_E, \tag{17}$$

where

$$\delta\mathbf{a}_E = -G\sum_{p=1}^{N}\frac{m_p}{|\mathbf{x}_E - \mathbf{x}_p|^3}(\mathbf{x}_E - \mathbf{x}_p) \tag{18}$$

and p labels all the other major bodies in the solar system. If we restrict ourselves to only major bodies than p would run over Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto and perhaps over other Pluto like objects. If we neglect $\delta\mathbf{a}_E$ in Eq.(17) then we know how to solve for the motion of Earth exactly. As you have learned in this course the motion is elliptical and in this approximation the earth would go on orbiting the sun till the sun exists in its present form. But what if we do not neglect $\delta \mathbf{a}_E$ then what will be the orbit of earth say five-million years from now? Will it still be an elliptical orbit? If the effect of $\delta \mathbf{a}_E$ is indeed negligible over the life time of the solar system than it would be reasonable to say that the effectively the orbit of earth is stable. We can ask similar question for all the bodies in the solar system that exhibit bounded motion, and if their orbits do not change qualitatively than again one can reasonably claim that the solar system is effectively stable.

In the recent times a great progress has been made in answering these questions. An essential component in answersing these question is the numerical solution of N gravitating bodies that we discussed in (3) and it has giving a surprisingly new insights [Renu Mahlotra, Carl Murry]. Perhaps, it is worth re-emphasizing the fact that from the point of solving Newton's equation of motion on a computer inclusion of $\delta \mathbf{a}_E$ introduced no new difficulty it just means that we have to do bit more airthematic to find the acceleration! In projects (2) and (4) you will explore these techniques on your own.

5.1 Future of the Solar System

Thanks to the heroic effort spent on solving the equations of motion that describe our solar system, Eq. (14), using computers with a tight control on round-off errors, we are beginning to find that the solar system is far from a perfect example of a clock work. Some of the salient features that emerge from these numerical solutions are

- The orbits of the planets that make up the solar system are chaotic with characteristic Lyapunov time of 5-10 million years.
- Although the numerical simulations all indicate chaos in planetary orbits, in a qualitative sense the planetary orbits are stable – because the

planets remain near their present orbits – over the lifetime of the sun.

- The presence of chaos implies that there is a finite limit to how accurately the positions of the planets can be predicted over long times.

These result are not only important for understanding the long term fate of our solar system, but they are also important as we search for other solar systems in our galaxy and for Earth like planets. Investigations like this may help us in answering the question, how unique are the Earth like planet in our Universe. You can explore some of these issues on your own in project (4).

6 References and Link

References

[R. P. Feynman] Feynman Lectures in Physics, Vol-1, Chp. 9

[Bikram Phukan] Bikram Phookun, *Planetary Orbits as Simple Harmonic Motion*, **Resonance** December 2003.

[Renu Mahlotra] Renu Mahlotra, *Chaos and stability of the solar system*, `http://www.pnas.org/cgi/reprint/98/22/12342.pdf`

[Carl Murry] Carl Murray, *Is the Solar System Stable?* `http://www.fortunecity.com/emachines/e11/86/solarsys.html`

[Visual Python] The Visual Python Web site:`http://www.vpython.org/index.html`

[Tutorial] Introduction to Visual Python through a tutorial.`http://www.vpython.org/VPython_Intro.pdf`

[Vectors] Implementation of vector algebra in Visual Python:`http://www.vpython.org/webdoc/index.html`

7 Exercises

1. Consider a simple model of Solar System that consists of the Sun, the Earth, and the Jupiter. Is the the total force acting on the Earth a central force? Is the angular momentum of the Earth conserved? Is the energy of the Earth conserved? Is the total angular momentum of the our model Solar system conserved?

2. Estimate the ratio of the force acting on the Earth due to Jupiter to the force acting on the Earth due to the Sun.

3. A planer moves around the Sun in a circular orbit of radius R_0. Consider the situation in which the planet at a given instant, which we can take as $t = 0$, gets a radial kick so that its radius changes from R_0 to $R_0 + \delta_0$. Assume that $\delta_0 \ll R_0$, and find the shape of the new orbit. Is the new orbit closed? For the pourpose of this exercise assume that the mass of the planet is much smaller than the mass of the Sun so that you can disregard the motion of the Sun.[Hint: The problem becomes simple once you realise that the circular orbits corresponds to the minimum of the effective potential and a small displacement about the minimum results in a simple harmonic motion. See [Bikram Phukan] for more interesting application of this approach.]

4. A particle moves in one dimension under the influence of a force which does not explicitly depend on time. Show that if we follow the trajectory of the particle, $x(t)$, from $t = 0$ to $t = T$ and then at time $t = T$ we reverse its velocity, then the particle will retrace its trajectory and will arrive back its original position at time $t = 2T$.

8 Projects

1. **Numerical solution of a non-linear oscillator.** In this project you will investigate the behaviour of a non-linear oscillator. It's equation of motion is
$$m\frac{d^2x}{dt^2} = -kx - \alpha x^3.$$
   (a) Define new unit of time and length so that the above equation of motion takes the following form
$$\frac{d^2y}{d\tau^2} = -y - y^3.$$
   (b) Solve this equation on a computer for various initial conditions.

   (c) Explore the cases when $y_0 \ll 1$ and when $y_0 \gg 1$, where y_0 is the initial position. To be definite take initial velocity as zero.

   (d) Write down an expression for the energy of the oscillator, and check to what accuracy is the energy conserved in your numerical solution for various choices of $\Delta\tau$.

2. **Building the solar system on your computer-I**: In this project you will use [Visual Python] to model a solar system made up of only two bodies, the sun and the earth. The way we will model will make it trivial to add more bodies. Before you start this project you may want to read the Appendix (A)go through a tutorial on visual python[Tutorial], and read the section on vectors [Vectors]

   - First write a Visual Python module that implements Eq. (14) check your answer against B.1.
   - Find the value of mass of the sun, mass of the earth, mean earth-sun distance from any physics text book.

- Now we are ready to build our solar system. Start a new Visual Python file. We will start by defining two spherical objects in visual python, they will graphically represent the sun and the earth. These objects have in addition three more properties, mass of the object, it's position, and it's velocity.

  ```
  from visual import *
  from nbodyAcc import *
  #Solar system on a computer
  #Constants that we will need
  #define the value of G
  G=6.673e-11
  sun_mass = 2e30
  earth_mass = 6e24
  #for initial conditions
  AU = 149.6e9       #mean earth sun orbital distance
  earth_vel = 2*math.pi * AU/(365.25 *24. *60.*60.)
  #setting for animations
  scene.background = color.white
  scene.autoscale = 0
  scene.range = 2*AU
  #objects making up our solar system
  sun = sphere(pos= (0,0,0), velocity = vector(0,0,0), mass=sun_mass,
                         radius = 0.1*AU, color =color.yellow)
  earth = sphere(pos= (AU, 0, 0), velocity = vector(0,earth_vel,0),
                             mass=earth_mass, radius=0.05*AU,
                                                 color =color.cyan)
  #note the radius of sun and earth are NOT their true radius
  #these are the radius of the spherical object
  #that will be drawn on the computer screen.
  ```

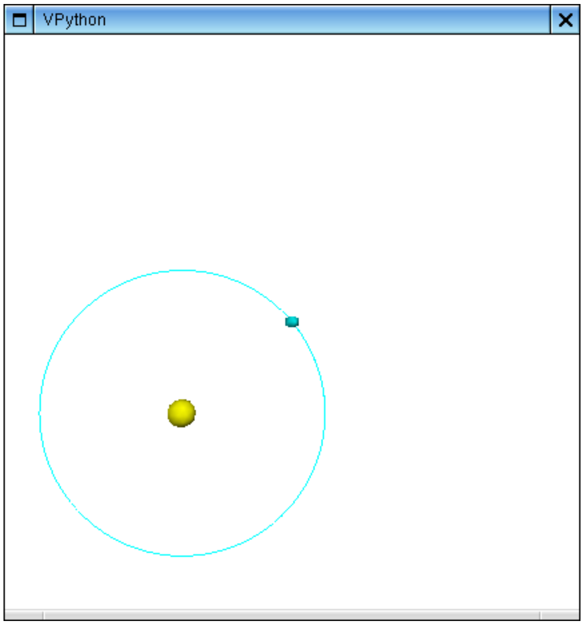
- Now make a list of these objects using the following code and add two new attributes

```
bodies = [sun, earth]
for b in bodies:
    b.acc = vector(0,0,0)
    b.track=curve (color = b.color)
#the last statement allows us to plot the orbits
```

- Now we are ready to implement leap-frog method

```
#Time interval for integration - let us take 30minutes
dt = 30.0*60.0
#Initialize leap-frog by finding the velocities at t=dt/2
for b in bodies:
    b.velocity = b.velocity + totalacc(b, bodies)*dt/2.0
#start leap-frog
while True:
    rate(100)  #not more than 100 time steps in a second
    for b in bodies:
        #update the positions
        b.pos = b.pos + b.velocity*dt
        b.track.append(pos=b.pos)
        #update the velocities
        b.velocity = b.velocity + totalacc(b, bodies)*dt
    scene.center = earth.pos #view centered on earth!
```

- If all goes well you should have

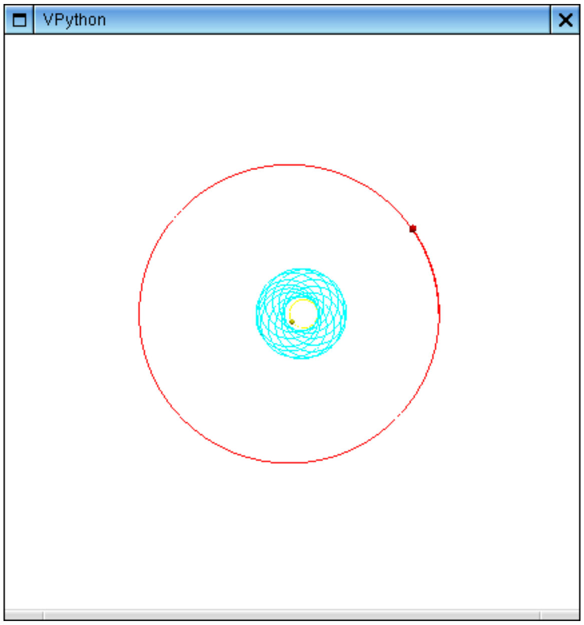
- 

3. **Stability of orbits in central force:** Modify the above program to simultaneously plot the orbit of the earth for two different initial conditions which differ by a very small amount. Let $\vec{r}_1$ represent the first orbit and $\vec{r}_2$ represent the second orbit, plot $|\vec{r}_1 - \vec{r}_2|$ as a function of time.

4. **Building the solar system on your computer-II:** Now that we have done the hard part, we can start playing with three body system.

   - Find the mass, average distance, and the time period for Jupiter.
   - Add Jupiter to the list of bodies in the previous program.
   - Set the total linear momentum of the solar system to zero using, so that we are in the centre of mass frame.

```
# set total momentum of system to zero
sum=vector(0,0,0)
for b in bodies:
    if (b!=sun):
        sum=sum+b.mass*b.velocity
sun.velocity=-sum/sun.mass
```

- Study the effect of Jupiter on Earth's orbit, to exaggerate this effect you can increase the mass of "Jupiter" by an arbitrary factor (this is the fun of using computer, we can do experiments!) Here is what you would see if the mass of Jupiter was increased by a factor of 100
- 

- You can also study the phenomenon of resonance in the solar system by introducing an hypothetical earth like planet, planetX whose orbit lies between earth and Jupiter (for more on resonance in solar system see [Carl Murry].) Find the distance from the sun for which the planet will, in the absence of Jupiter and Earth, would move in a circular orbit whose time period is in integral ratio with the time period of Jupiter's orbit. Now exaggerate the effect of resonance by increasing the mass of Jupiter. You should be able to see cases where the planetX is ejected from the solar system!

(The project (2) and this project are modification of Simon Catterall computational physics course Lab, phy307, Syracuse University `http://www.phy.syr.edu/courses/PHY307/LABS/`.)

A A Brief Introduction to Visual Python

What is Python?

It is a high level object oriented
Interpretive
programming language
which supports vector algebra.

Everything in Python is an object. Python objects can have attributes and methods, which are subsidiary variables and functions.

```
z = 2+8J
z.real # An attribute of complex numbers.
z.imag # Another attribute.
z.conjugate() # A method of complex numbers.
```

What is Visual Python?

Extension of python that allows for visual representation of three dimensional geometrical objects.

```
>>> from visual import *
>>> sphere() # draw a sphere
```

Use of Visual Python in Physics: SHO

```
from visual import *
ball = sphere (pos = (10,0,0), radius = 0.5, color = color.red)
t = 0
dt = 0.01
```

```
ball.velocity = vector(0,0,0)
tmax = 10
while t<tmax:

    rate(100)
    t = t + dt
    ball.pos = ball.pos + ball.velocity * dt
    accel = - ball.x - 0.01 * ball.velocity.x
    ball.velocity.x = ball.velocity.x + accel * dt
```

Visual Python in Physics: A Tutorial

Setting up the charges on the screen

```
    from visual import *
    #set the screens
    scene.width=1200
    scene.height= 1000
    scene.title=’Motion of an electron ...’
    #define electric charge
    Q = 1.6e-19
    #define two static charges
    q1= sphere(pos=(0.0,0.0,0.0), radius = 0.1e-10,
            color = color.red, charge = Q)
    q2= sphere(pos=(1.0e-10,0.0,0.0), radius = 0.1e-10,
            color = color.red, charge = Q)
    #define electron
    electron = sphere(pos=(0.5e-10,1.0e-10, 0.0), radius =
0.05e-10,
           color= color.blue, charge = -Q, mass = 9.0e-31,
           velocity = vector(1.8e6,0.0,0.0),
trail=curve(color=color.yellow))
    #define array of two charges
    charges=[q1, q2]
    #$k = \frac{1}{4\pi\epsilon_0}$
    k = 9.0e9
```

Showing the motion of the charges on the screen

```
    #time interval for integrating
    dt = 0.5e-18
    #start an infinite loop
    while 1:
          rate(50)
          #calculate the new position of the electron
          electron.pos = electron.pos + electron.velocity*dt
          #create a trail
          electron.trail.append(pos=electron.pos)
          #define the electric field vector
          E=vector(0.0,0.0,0.0)
          #calculate the electric field
          for ch in charges:
               #position vector of the electron
               r = electron.pos - ch.pos
               E = E + k*ch.charge/mag(r)**2 * norm(r)
      #calculate the force on the electron
          F = E*electron.charge
          #calculate the accelaration
```

```
a = F/electron.mass
#calculate the new velocity
electron.velocity = electron.velocity + a*dt
```

B Building a Solar System on Your Computer:

B.1 Calculating acceleration in N-body problem

This module implements Eq. (14) in visual python

```
##============================================================
##nbodyAcc
##============================================================
from visual import *
#define the value of G
G=6.673e-11
## Acceleration of object a due to object b
def acc(a, b):
    rel_pos = b.pos - a.pos
    return G*b.mass * norm(rel_pos)/rel_pos.mag2
## Acceleration of a due to all the
##objects b interacting with it
def totalacc (a, objlist):
    sum_acc = vector (0,0,0)
    for b in objlist:
        if (a!=b):
            sum_acc = sum_acc + acc(a, b)
    return sum_acc
```

B.2 N-body Problem in Visual Python

```
from visual import *
from nbodyAcc import *
#Solar system on a computer
#Constants that we will need
#define the value of G
G=6.673e-11
sun_mass = 2e30
earth_mass = 6e24
boost=1.0 #allow for boosting Jupiter's mass
jupiter_mass=boost*1.9e27
#for initial conditions
AU = 149.6e9       #mean earth sun orbital distance
earth_vel = 2*math.pi * AU/(365.25 *24. *60.*60.)
jupiter_vel=2*math.pi*AU*5.2/(11.86*365.25*24.*60.*60)
#setting for animations
scene.background = color.white
scene.autoscale = 0
scene.range = 10*AU
#objects making up our solar system
sun = sphere(pos= (0,0,0), velocity = vector(0,0,0),
             mass=sun_mass, radius = 0.1*AU, color =color.yellow)
earth = sphere(pos= (AU, 0, 0), velocity = vector(0,earth_vel,0),
               mass=earth_mass, radius=0.05*AU, color =color.cyan)
jupiter=sphere(pos=(5.2*AU,0,0),velocity=vector(0,jupiter_vel,0),
             mass=jupiter_mass, radius=0.15*AU, color=color.red)
#note the radius of sun, jupiter  and earth are NOT
# their true radius these are the radius of the spherical object
#that will be draw on the computer screen.
#Create a list of objects making up our solar system
```

```
#and add attributes for their acceleration and orbits
bodies = [sun, earth, jupiter]
for b in bodies:
    b.acc = vector(0,0,0)
    b.track=curve (color = b.color)

# set total momentum of system to zero
sum=vector(0,0,0)
for b in bodies:
    if (b!=sun):
        sum=sum+b.mass*b.velocity
sun.velocity=-sum/sun.mass
# dt corresponds to 3000 mins here
dt=30.*60.*100
#Initialize leap-frog by finding the velocities at t=dt/2
for b in bodies:
    b.velocity = b.velocity + totalacc(b, bodies)*dt/2.0
#start leap-frog
while True:
    rate(100)  #not more than 100 time steps in a second
    for b in bodies:
        #update the positions
        b.pos = b.pos + b.velocity*dt
        b.track.append(pos=b.pos)
        #update the velocities
        b.velocity = b.velocity + totalacc(b, bodies)*dt
    scene.center = sun.pos #view centered on sun
```

C Appendix: Solutions of the Selected Exercises

- Ex-2: To estimate this ratio, we will assume that both the Earth and the Jupiter are moving in a circular orbit, that is we are disregarding the small eccentricities of their orbits. Then the ratio is

  $$\frac{F_{EJ}}{F_{ES}} \approx \frac{M_J}{M_S}\frac{r_{ES}^2}{r_{EJ}^2},$$

  where, M_J and M_S are the masses of the Jupiter and the Sun respectively, while r_{ES} is the average distance between the Earth and the Sun, similarly r_{EJ} is the average distance between the Earth and the Jupiter. Using the following values: $M_J = 1.9 \times 10^{27} kg$, $M_S = 2.0 \times 10^{30} kg$, $r_{ES} = 1AU$ and $r_{EJ} = 4.2AU$, gives us

  $$\frac{F_{EJ}}{F_{ES}} \approx \frac{1}{10^3}\frac{1}{4.2^2} \approx 6 \times 10^{-5}.$$

- Ex3: The radial motion is determined by the effective potential

  $$U_{eff}(r) = \frac{-GM}{r} + \frac{l^2}{2r^2}, \tag{19}$$

  where M is the mass of the sun and

  $$l = \frac{L}{m}, \tag{20}$$

  where L is the conserved angular momentum of the planet and m is its mass. The radial equation of motion is

  $$\ddot{r} = -\frac{dU_{eff}}{dr}. \tag{21}$$

The angular motion is of-course given by

$$\dot{\phi} = \frac{l}{r^2}. \tag{22}$$

The circular motion corresponds to the minimum of the the effective potential

$$\left(\frac{dU_{eff}}{dr}\right)_{R_0} = 0 = \frac{GM}{R_0^2} - \frac{l^2}{R_0^3},$$

which givers the radius of the circular orbit as

$$R_0 = \frac{l^2}{GM}. \tag{23}$$

To find the radial motion of the perturbed orbit, $r(t) = R_0 + \delta(t)$, we need $U_{eff}(R_0 + \delta)$, since $\delta \ll R_0$, we do a Taylor expansion about $R_0$to obtain

$$U_{eff}(R_0 + \delta) = U_{eff}(R_0) + \frac{1}{2}\left(\frac{d^2U_{eff}}{dr^2}\right)_{R_0} \delta^2 + \cdots,$$

where we have used the fact that $U_{eff}(R_0)$ is the minimum and therefore the first derivative of the effective potential vanishes at R_0. Define

$$\omega^2 = \left(\frac{d^2U_{eff}}{dr^2}\right)_{R_0} = \frac{3l^2}{R_0^4} - \frac{2GM}{R_0^3},$$

using Eq.(23) we get

$$\omega^2 = \frac{GM}{R_0^3}, \tag{24}$$

and the equation for the radial motion is

$$\ddot{\delta} = -\frac{dU_{eff}}{d\delta} = -\omega^2\delta.$$

This is nothing but the equation for simple harmonic motion and the solution for our case is

$$\delta(t) = \delta_0 \cos \omega t$$

and the new radial motion is

$$r(t) = R_0 + \delta_0 \cos \omega t.$$

Note that the ω is same as the angular frequency $\dot{\phi}$ of the unperturbed circular motion, so after one complete radial oscillation the particle is at $\phi = \phi + 2\pi$ and therefore the perturbed orbit is closed.

- Ex 4: Consider the trajectory

  $$\bar{x}(t) = x(2T - t) = x(\tau),$$

  where $\tau = 2T - t$, for the time interval $[T, 2T]$. One can immediately verify that if $x(t)$ is the solution of the equation of motion then so is $x(\tau)$, that is,

  $$m\frac{d^2x(\tau)}{d\tau^2} = F.$$

  $x(\tau)$ is nothing but the original trajectory which has been reversed, at $t = 2T$, $\tau = 0$ and the particle returns to its starting point, but the velocity is reversed

  $$\left(\frac{d\bar{x}}{dt}\right)_{t=2T} = \left(\frac{dx}{d\tau}\right)_{\tau=0} \frac{d\tau}{dt} = -\left(\frac{dx}{dt}\right)_{t=0}.$$

D Movies

D.1 Sun-Earth

The movie (Fig.2) was made using the numerical solution of the earth sun system as described in the project 2.

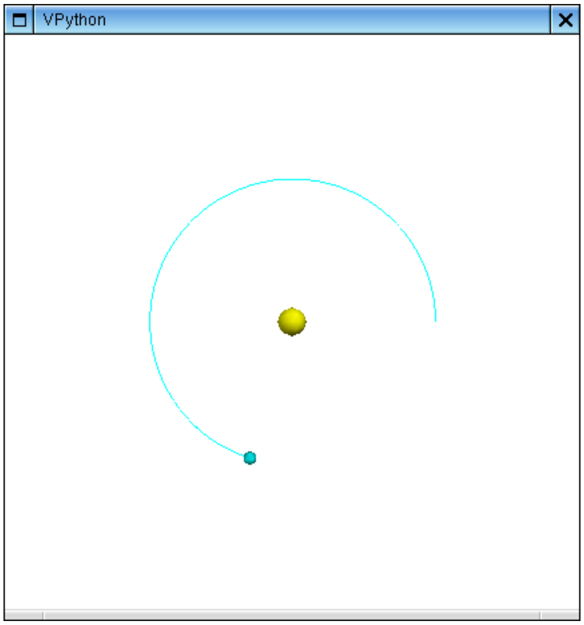


Figure 2: Movie of Earth Sun System

D.2 Sun-Earth and a Heavy Jupiter

The movie (Fig.3) is an example of a three body system and was made using the program developed in the project 4. Here we consider the motion of the Sun, the Earth and an imaginary planet which is at same distance as the Jupiter from the Sun but is hundred time massive than the Jupiter.

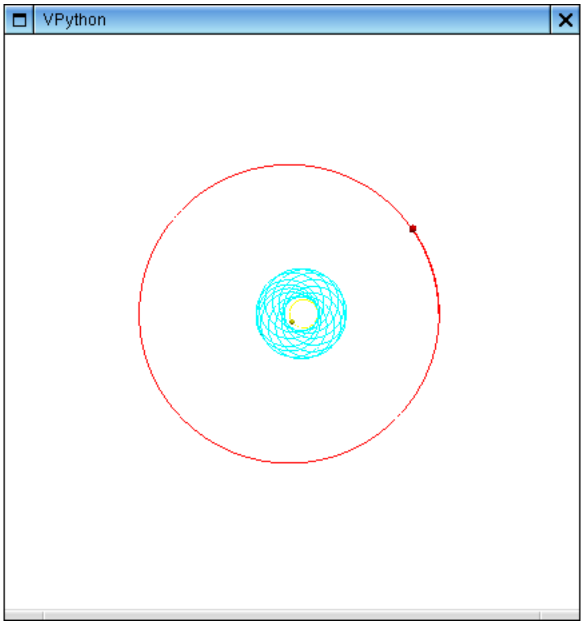


Figure 3: Movie of the Sun, Earth and a planet hundred times massive than the Jupiter