# A645/A445: Exercise #4

# Numerical computation of orbits

Due: 2019 Oct 24

1 **A useful fact about turning points in spherical systems.**
As we know, the apocenter $r_a$ and pericenter $r_p$ of an orbit in a spherical stellar system is given by the roots of

$$2\left(E - V(r)\right) - \frac{J^2}{r^2} = 0. \tag{1}$$

Prove that this equation has either zero or two roots assuming that the orbit is bound ($E < 0$) and the potential $V(r)$ is physically realizable (that is: it corresponds to a mass distribution with finite density). [Hints: (i) Change variables to $u = 1/r$; (ii) use the Poisson equation after taking the second derivative of equation (1).]

2 **Symplectic integration.**
Generalize your ODE program for integrating and plotting the equations of motion using the symplectic leapfrog integrator. I did not have time to discuss this algorithm in class today, but I'm attaching a brief introduction as an appendix to get you started. I will discuss this at the beginning of class next Thursday.

(a) As a test, use this integrate the pendulum equations of motion in all regimes:
   i. rotation,
   ii. libration,
   iii. near the unstable equilibrium

(b) Demonstrate that the leapfrog integrator is second-order accurate in the sense that errors in $\mathbf{q}$ and $\mathbf{p}$ after a timestep $h$ are $O(h^3)$.

(c) Compute the energy conservation for the trajectories in each regime over about 10,000 orbital periods. Compare the energy conservation for RK4 and leapfrog.

## 3 Orbits in an axisymmetric potential

Consider an axisymmetric potential used as a simple model for a galaxian disk first proposed by Toomre (1964, ApJ):

$$U(r) = -(1+r^2)^{-1/2}.$$

(a) Pick some initial conditions and integrate the orbits using your orbit integration code. Verify that orbits do not close but precess. These are sometimes called *rosette* orbits because of their shape or *tube* orbits because the have an inner and outer boundary.

(b) Compute the energy and angular momentum of your trial orbits (e.g. using the initial conditions). Compute the inner and outer radii of the tube (the turning points) using the conserved quantities and check these values against your direct orbit integration.

## 4 Non-axisymmetric orbits

Finally, a more complicated non-axisymmetric case—two fixed point masses—which has the potential:

$$U(x,y) = -[(x-a)^2+y^2]^{-1/2} - [(x+a)^2+y^2]^{-1/2}.$$

Although far from obvious, this potential can be separated in confocal elliptical coordinates and put in a Staeckel form (as described in B&T and therefore has regular orbits. If the two point masses were in orbit around each other, the orbits would not be regular. In other words, the real three body problem has chaotic trajectories!

(a) Again, pick some initial conditions and integrate the orbits. Just to be definite $a = 1/2$ (although the problem scales with $a$ so you can choose value you wanted and get the same results).

(b) There are two different kinds of orbits in this potential: *tube* orbits and *box* orbits. Box orbits come arbitrarily close to the center of force filling a closed area (similar to Lissajous figures). In fact, there are two cases of box orbits here: those which come close to one and those which come close to both centers of force. Find an example of each case.

(c) **Extra credit** Investigate the relationship between conserved quantities and the envelope of the orbits. Report your findings.

Note: because the force is generated by two point masses, box orbits will come arbitrarily close to one or both of the force centers. This makes the solution numerically tricky. Consider using small error tolerances and/or small time steps ...

## 5 Irregular orbits: Part 1

Most often, one does not know the fraction of irregular orbits for particular density model. A very useful technique for both determining whether an orbit is regular or irregular and classifying the type of orbit is the "surface of section" method. This is described in detail in Binney and Tremaine, §3.2.2.

(a) Adapt your ODE program to find the $x$—$\dot{x}$ surface of section; that is, have your program plot with points the the values of $x$ and $\dot{x}$ when $y = 0$ and $\dot{y} > 0$.

(b) Use your ODE program to find a tube orbit and a box orbit in the potential given by B&T equation 3-103 with $R_c = 0.14$, $q = 0.9$ and $v_c = 1$. (See Figure 3-8 for inspiration).

(c) Now, plot the surface of section for these orbits. Note/discuss/explain the differences. [We will explore the surface of sections for irregular orbits in a later PS.]

## 6 The Jacobi Integral

Recall the Jacobi integral discussed in class.

(a) Show that the Jacobi integral is a constant of the motion, i.e.

$$\frac{dE_J}{dt} = 0.$$

(b) Show that the Jacobi integral for an orbit may be written as:

$$E_J = E - \mathbf{\Omega}_p \cdot \mathbf{L}$$

where $\mathbf{\Omega}_p$ is the constant rotation speed of the system whose direction is along the axis of rotation and $\mathbf{L}$ is the angular momentum of the orbit.

# Very brief description of the leapfrog integrator

Martin D. Weinberg
Department of Astronomy
University of Massachusetts, Amherst, MA, USA 01003;
*weinberg@astro.umass.edu*

October 10, 2019

Numerical solution of orbits in a gravitational potential is at the heart of N-body simulation. There are good ways and bad ways of solving the ODEs as we saw in the Jupyter notebook that I posted on the course web site. I will assume that you are familar with that notebook. Here, I will briefly introduce on the most frequently used technique used for orbit integration: the *leapfrog algorithm*.

## 1 Example: Euler

To review, we want to solve Newton's equations which have the form

$$\frac{d^2\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x})$$

We know that can write this as coupled first-order equations as follows:

$$\frac{dx}{dt} = \mathbf{v}(\mathbf{x}) \tag{1}$$

$$\frac{d\mathbf{v}}{dt} = \mathbf{g}(\mathbf{x}) \tag{2}$$

The simplest algorithm is the **forward Euler method**:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}(\mathbf{x}) \tag{3}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h\mathbf{g}(\mathbf{x}) \tag{4}$$

where $h$ is the step size.

Its main disadvantage is that it is first order and unstable.

## 2 Second-order Leapfrog

Newton's equations are a very special 2nd-order differential equation. After reducing the equations to coupled first-order ODEs, the first order differential equations for the positions take a special form. This condition, the right-hand-side of the position equation only

depending on the velocity and the right-hand-side of the velocity equation only depending on position is called a *Verlet* condition. There are several specialized algorithms for equations of motion that obey the *Verlet* condition. The Verlet condition allows the ODE algorithm to be *split* into two parts!

In particular, we can construct a second order integrator called *leapfrog*. It is often written

$$\begin{aligned} \mathbf{v}_{n+1/2} &= \mathbf{v}_{n-1/2} + h\mathbf{g}(\mathbf{x}_n) && (5) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + h\mathbf{v}_{n+1/2}. && (6) \end{aligned}$$

We can think of this set as a centered difference algorithm in both space and velocity! In this form, this algorithm is second order but not self starting. The position update equivalent to evolving under the Hamiltonian

$$H_T = \frac{1}{2}\mathbf{p}\cdot\mathbf{p} = \frac{1}{2}p^2$$

We will call the operator that evolves the system under $H_T$ the "drift" operator, $D$. The velocity update equivalent to evolving under the Hamiltonian

$$H_V = V(\mathbf{q})$$

We will call this operator the "kick" operator, $K$. Now, leapfrog can be written

$$K(h)D(h)$$

repeated with $\mathbf{v}$ and $\mathbf{z}$ defined half steps apart.

One can also construct second order integrator

$$D(h/2)K(h)D(h/2)$$

or

$$\begin{aligned} \mathbf{x}_{n+\frac{1}{2}} &= \mathbf{x}_n + \frac{h}{2}\mathbf{v}_n && (7) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + h\mathbf{g}(\mathbf{x}_{n+\frac{1}{2}}) && (8) \\ \mathbf{x}_{n+1} &= \mathbf{x}_{n+\frac{1}{2}} + \frac{h}{2}\mathbf{v}_{n+1} && (9) \end{aligned}$$

This is equivalent to leapfrog but is self starting and therefore this form is computationally convenient. One can construct fourth order method similarly but going to higher order is usually not worth the expense.

As I emphasized in the Python notebook, it is important to monitor the error in the solution, especially for long time integrations. The error can be estimated by half-stepping: simultaneously solving the ODEs for stepsize $h$ and $h/2$. Finally, it is possible to extend the *leapfrog* algorithm using Richarson Extrapolation while maintaining the Hamiltonian character. The extrapolator also can give error estimate. One may use the error estimate to adjust step size to keep number of extrapolations reasonable.

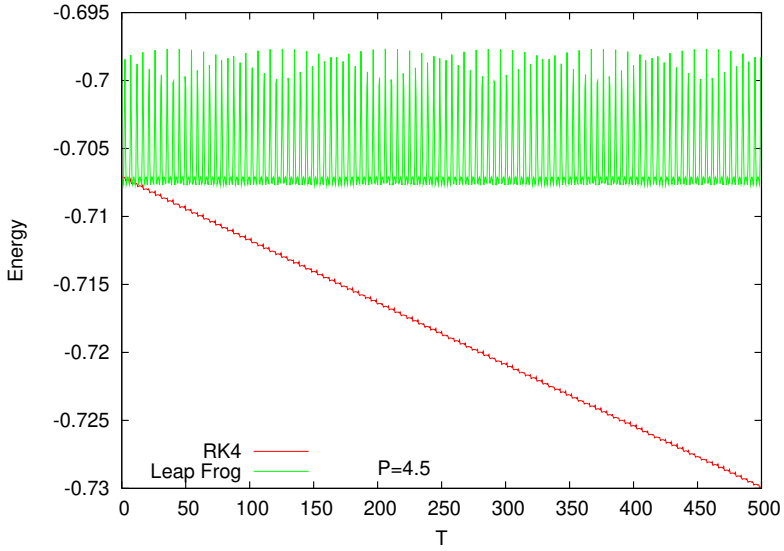


Figure 1: Comparison of Newton's equations solution for RK4 and Leapfrog in the potential $U(x,y) = -1/\sqrt{(a^2+x^2+y^2)}$ with initial condition $(x,y,u,v) = (1,0,0,0)$ and $h = 0.5$. The radial period is approximately 4.5 for this initial condition.

# 3 Example: comparison of RK4 and Leapfrog

In this example, we compare the energy conservation of an orbit in a softened point-mass potential

$$U(x,y) = -\frac{1}{\sqrt{(a^2+x^2+y^2)}}.$$

We choose $a = 1$, $h = 0.5$ and $(x,y,u,v) = (1,0,0,0)$. We compare the solutions using the 4th-order Runga Kutta integrator with the 2nd-order Leapfrog integrator. The period is approximately 4.5 time units and therefore Figure 1 shows about 100 radial periods.

Even though Runga Kutta is 4th-order, the solution shows a slow downward drift in energy. The symplectic Leapfrog, only second order, shows more step-to-step variation in energy but no net drift in energy over the same time range.