

Problem Set 3

Exercises for the Lecture Fundamentals of Simulation Methods

Prof. Dr. Ralf Klessen (Lecture Tuesday 9h - 11h and Thursday 9h - 11h)

Loke Lönblad Ohlin (Tutor Group Thursday 11h - 13h, e-mail: loke.ohlin@uni-heidelberg.de)

Toni Peter (Tutor Group Thursday 14h - 16h, e-mail: toni.peter@uni-heidelberg.de)

Marcelo Barraza (Tutor Group Friday 11h - 13h, e-mail: barraza@mpia-hd.mpg.de)

Submit the solution to your tutor in electronic form by **Wednesday November 6, 2019**.

1. The 2-body problem: Orbit of planet around the Sun

(5 points)

Consider the problem of a star of mass $M_* = M_\odot$ (where $M_\odot = 1.99 \times 10^{30}$ kg is one solar mass) surrounded by a planet of $M_p = 10^{-3} M_\odot$. At time $t = 0$ the planet is located at coordinates $(1, 0, 0)$ in units of $\text{AU} = 1.496 \times 10^{11}$ m. The planet's velocity is $(0, 0.5, 0)$ in units of the Kepler velocity at that location, which is $v_K(1\text{AU}) = 2.98 \times 10^4$ m/s.

1. Solve the Kepler orbit of this planet using numerical integration with the leapfrog algorithm. Find an appropriate time step. Plot the result for the first few orbits.
2. Now integrate for 100 orbits and see how the orbit behaves.
3. Repeat this with the RK2 algorithm and see how the system behaves. Discuss the difference to the leapfrog algorithm.

2. A simple N-body system

(20 points)

Let us now consider a true N -body problem, with $N > 2$ gravitating objects. First, however, let us cast the problem in dimensionless units. The N -body equations read

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i \quad (1)$$

$$m_i \frac{d\vec{v}_i}{dt} = -Gm_i \sum_{k \neq i} m_k \frac{\vec{x}_k - \vec{x}_i}{|\vec{x}_k - \vec{x}_i|^3} \quad (2)$$

1. Show that with an appropriate scaling $\vec{x} = \xi \tilde{\vec{x}}$, $\vec{v} = \phi \tilde{\vec{v}}$, $t = \tau \tilde{t}$ and $m = \mu \tilde{m}$ (with ξ , ϕ , τ and μ constants) the equations for N -body dynamics can be brought into dimensionless form:

$$\frac{d\tilde{\vec{x}}_i}{d\tilde{t}} = \tilde{\vec{v}}_i \quad (3)$$

$$\tilde{m}_i \frac{d\tilde{\vec{v}}_i}{d\tilde{t}} = -\tilde{m}_i \sum_{k \neq i} \tilde{m}_k \frac{\tilde{\vec{x}}_k - \tilde{\vec{x}}_i}{|\tilde{\vec{x}}_k - \tilde{\vec{x}}_i|^3} \quad (4)$$

where the gravitational constant G is absorbed into the variables. Which conditions must ξ , ϕ , τ and μ obey and how many of them can be freely chosen?

From now on we will do everything in dimensionless units, *and we will omit the tilde in spite of being in dimensionless units.*

1. Write an N -body code for arbitrary N with the leapfrog integration and a constant time step. As a simple test problem solve the following simple binary star problem: two stars of mass 1, at initial locations $\vec{x}_1 = (-0.5, 0, 0)$, $\vec{x}_2 = (+0.5, 0, 0)$, and initial velocities $\vec{v}_1 = (0, -0.5, 0)$, $\vec{v}_2 = (0, +0.5, 0)$. Choose an appropriate time step! Plot the resulting trajectories of both stars in the (x, y) -plane (projection).
2. Now add a third star with mass 0.1 and initial position $\vec{x}_3 = (1, 6, 2)$ and initial velocity $\vec{v}_3 = (0, 0, 0)$. Show how this third star “falls into” the binary, interacts with it, and gets eventually ejected. Plot the trajectories of all three stars in the (x, y) -plane (projection).
3. Play with the time step by varying it at least a factor of 10 (but keep the final time of the integration fixed) and describe whether the results change or not. The effect of the time step can even be stronger than for this case: try $\vec{x}_3 = (1, 6, 3)$.

Now let us try a true N -body problem ($N \gg 3$).

1. Set up a spherical cloud of 30 randomly positioned stars of mass 1. The cloud radius is 1. Use a uniform random number generator for this. An easy way is to choose randomly (x, y, z) between -1 and +1, and reject (and redo) stars that have $\sqrt{x^2 + y^2 + z^2} > 1$. Give each particle a random velocity (uniform) with a maximum of 0.1. Use the same rejection trick as for the positions. Make a number of simulations for appropriate simulation times and reasonable time steps but different random seeds.
2. Now redo the simulation with $N = 300$ and masses 0.1. How much slower is the simulation and why?
3. Go back to $N = 30$, but now implement a “close encounter checker” which, if triggered, will reduce the time step of the entire simulation (not just of the two close particles) to ensure a better integration of the close passage of the two stars. Use time-substepping. Convince yourself that you catch these close encounters. Is the simulation much slower?