Problem Set 4

Exercises for the lecture Fundamentals of Simulation Methods, WS 2020

Prof. Dr. Frauke Gräter, Prof. Dr. Friedrich Röpke
Tutors: Benedikt Rennekamp (benedikt.rennekamp@h-its.org), Fabian Kutzki (fabian.kutzki@h-its.org), Giovanni Leidi (giovanni.leidi@h-its.org), Siddhant Deshmukh (sdeshmukh@lsw.uni-heidelberg.de)
Offices: Heidelberg Institute for Theoretical Studies, Schloss-Wolfsbrunnenweg 35

Hand in until Wednesday, 02.12., 23:59
Tutorials on Friday, 04.11.
(Group 1, Giovanni Leidi 09:15)
(Group 2, Benedikt Rennekamp 09:15)
(Group 3, Siddhant Deshmukh 11:15)
(Group 4, Fabian Kutzki 14:15)

1. The 2-body problem: Orbit of planet around the Sun [8pt]

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 AU=1.496×10^{11} m. The planet's velocity is $(0, 0.5, 0)$ in units of the Kepler velocity at that location, which is $v_K(1\mathrm{AU}) = 2.98 \times 10^4$ 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 and RK4 algorithms and see how the system behaves. Discuss the difference to the leapfrog algorithm. For this, also plot the time evolution of the relative error of the total energy and the time evolution of the total kinetic energy of the system.

2. N-body problem [12pt]

Let us now consider a N-body problem, with $N > 2$ gravitationally interacting objects. We will solve this system with the *direct summation method*. First, however, let us cast the problem in dimensionless units. The N-body equations read

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i \tag{1}$$

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

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:

$$\frac{d\tilde{\vec{x}}_i}{d\tilde{t}} = \tilde{\vec{v}}_i \tag{3}$$

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

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. One way to prevent numerical divergences in case of very close encounters is to use the *softening length parameter* (ϵ) and modify the gravitational force among massive particles:

$$m_i\frac{d\vec{v}_i}{dt} = Gm_i\sum_{k\neq i}m_k\frac{\vec{x}_k - \vec{x}_i}{|\vec{x}_k - \vec{x}_i|^3 + \epsilon^3} \tag{5}$$

However, ϵ should be small enough to keep the simulation physically consistent. Try to set:

$$\epsilon = LN^{-\frac{1}{3}}\frac{1}{10000} \tag{6}$$

where L is the typical size of the system (in dimensionless units!) and N is the number of particles (note that $LN^{-\frac{1}{3}}$ is the average distance between 2 particles). If the evolution is unstable or you get odd features in the trajectories, try to change the value of ϵ.

2. 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). Plot also the time evolution of the relative error of the total energy of the system.

3. Now add a third star with mass 0.1 and initial position $\vec{x}_3 = (1, 6, 2)$ and intial 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). Again, plot the time evolution of the relative error of the total energy of the system.

4. 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 changing the time step can also be stronger than in this case, to see this also 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. Evolve the system over an appropriate time scale (is there a characteristic time scale for a gravitationally interacting cluster?). For these simulations you will have to use an adaptive time step: at the beginning of each iteration, you can estimate the value of the time step as:

$$\Delta t = C \frac{d_{min}}{|\vec{v}_{max}|} \tag{7}$$

   where C=0.01, d_{min} is the minimum distance between 2 particles and $|\vec{v}_{max}|$ is the velocity of the fastest particle in the system. If the evolution is unstable or if the relative error of the total energy gets higher than 0.05, decrease the value of C. Note that in this case you will need more iteration steps to evolve the system over the same time scale.

2. Plot the time evolution of the relative error of the total energy.

3. Plot the evolution in 3-D and note any interesting patterns that emerge from the evolution. What are these? Why do they appear?

4. Now redo the simulation with $N = 60$ and masses 0.5. Run this and the previous simulation for 100 iterations each. Compare both the results and run times between the two.