Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Dipl.-Tech. Math. P. Neumann
Dipl.-Inf. W. Eckhardt

SS 2013
Sheet 9
01.07.2013

Scientific Computing II

Molecular Dynamics

Exercise 11: Time-Reversibility of Time Integrators

An important property of time integrators is time-reversibility: when starting from a state $t + \Delta t$ and going back in time using a time step $\tilde{\Delta t} := -\Delta t$, a time-reversible time integrator delivers (in exact arithmetics) the original state t.

(a) Show that the Velocity Störmer Verlet method

$$r(t + \Delta t) = r(t) + \Delta t \cdot v(t) + \frac{\Delta t^2}{2} \cdot a(t) \quad (1)$$

$$v(t + \Delta t) = v(t) + \frac{\Delta t}{2} \cdot (a(t) + a(t + \Delta t)) \quad (2)$$

is time-reversible; r, v, a denote position, velocity and acceleration.

(b) Show that the explicit Euler method

$$r(t + \Delta t) = r(t) + \Delta t \cdot v(t) \quad (3)$$

$$v(t + \Delta t) = v(t) + \Delta t \cdot a(t) \quad (4)$$

is not time-reversible.

Exercise 12: Round-Offs and Stability

In the following, we want to investigate the error propagation if a small error occurs during the force/ acceleration evaluation. Therefore, consider a one-dimensional scenario with two particles. Both particles are updated according to the Velocity Störmer Verlet scheme, cf. Eqs. (1) and (2), and the acceleration evolves from a Lennard-Jones force

$$a_1(t) := F(r_1(t) - r_2(t)) = 24 \left(\frac{1}{(r_1(t) - r_2(t))^7} - \frac{2}{(r_1(t) - r_2(t))^{13}} \right) \quad (5)$$

where r_1, r_2 correspond to the positions of particles 1 and 2.
Assume an error enters the force computation at time step t, that is we obtain $\tilde{a}_1(t) = a_1(t) + \epsilon$ instead of $a_1(t)$. Investigate the impact of this error onto the movement of particle 1 at $t + \Delta t$. You may further assume that the error only occurs for particle 1 and that particle 2 is not affected by the error.

Planet	Mass	Init. position	Init. velocity
Sun	$m_0 = 1$	$\vec{x}_0 = (0,0)$	$\vec{v}_0 = (0,0)$
Earth	$m_1 = 3e-6$	$\vec{x}_1 = (0,1)$	$\vec{v}_1 = (-1,0)$
Jupiter	$m_2 = 9.55e-4$	$\vec{x}_2 = (0,5.36)$	$\vec{v}_2 = (-0.425,0)$
Halley's Comet	$m_3 = 1e-14$	$\vec{x}_3 = (34.75,0)$	$\vec{v}_3 = (0,0.0296)$

Table 1: Initial configuration of our solar system.

Programming Exercise 9: Moving Planets

In the following, we will consider the interaction of the planets in our solar system. The planets interact according to the *gravitational potential* which yields a force

$$F_{ij} = \frac{m_i m_j}{r_{ij}^3} \vec{r}_{ij} \tag{6}$$

which acts from planet j onto planet i. The parameters m_i, m_j are the masses of planets i and j, $\vec{r}_{ij} = \vec{x}_j - \vec{x}_i$ is the vectorial distance of the planet positions $\vec{x}_i$, $\vec{x}_j$, and $r_{ij} := \|\vec{r}_{ij}\|$.
The initial conditions for the following planets are given[1] in Tab. 1.

(a) Write a matlab method `initialise()` which returns the initial positions, velocities and forces as well as the masses of the planets. Test the method.

(b) Write a matlab method `computeForces(positions, masses)` which computes the gravitational forces for all pairs of planets at the given positions and respective masses. The method should return the total force that acts onto each planet. Test the method.

(c) Write a matlab method `velocityStoermerVerlet(positions, velocities, forces, forcesOld, dt, masses)` which for a given time step `dt` updates the positions and velocities of the planets using the velocity Störmer Verlet algorithm. The method makes use of the force computation `computeForces(positions, masses)`. The forces from the current time step should be stored in `forcesOld` before finishing the time step. Test the method.

(d) Integrate all methods into a simulation of the solar system. Use a time step $dt = 0.015$ and a number of time steps $N = 32\,000$. Plot the trajectories of all planets and the comet.

(e) Write a matlab method `explEuler(positions, velocities, dt, masses)` which carries out explict Euler time stepping (instead of the Verlet scheme). Re-run your simulation with this time stepping method. What do you observe?

[1]See also M. Griebel, S. Knapek, G. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer, 2007.