

## 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.

#### Solution:

(a) To find out whether the method is time-reversible, one calculates backwards. Starting from the calculated positions and velocities at time  $(t + \Delta t)$ , one calculates those at time  $t$  by using  $\tilde{\Delta t} = -\Delta t$  from the previous formula as a time step. The resulting values at time  $(t + \Delta t + \tilde{\Delta t}) = t$  are denoted as  $\tilde{r}$  and  $\tilde{v}$ , since we don't know yet whether they are really equal to  $r$  and  $v$ . By replacing  $\Delta t$  by  $\tilde{\Delta t}$  and  $t$  by  $t + \Delta t$  in eq. (1) and (2), we obtain:

$$\tilde{r}(t + \Delta t + \tilde{\Delta t}) = r(t + \Delta t) + \tilde{\Delta t} \cdot v(t + \Delta t) + \frac{\tilde{\Delta t}^2}{2} \cdot a(t + \Delta t) \quad (5)$$

$$\tilde{v}(t + \Delta t + \tilde{\Delta t}) = v(t + \Delta t) + \frac{\tilde{\Delta t}}{2} \cdot (a(t + \Delta t) + \tilde{a}(t + \Delta t + \tilde{\Delta t})). \quad (6)$$

Replacing  $\tilde{\Delta t}$  by  $-\Delta t$  in Eq. (5) yields:

$$\tilde{r}(t) = r(t + \Delta t) - \Delta t \cdot v(t + \Delta t) + \frac{\Delta t^2}{2} \cdot a(t + \Delta t). \quad (7)$$

Substituting Eq. (1) and Eq. (2) into Eq. (7) results in:

$$\tilde{r}(t) = (r(t) + \Delta t \cdot v(t) + \frac{\Delta t^2}{2} \cdot a(t)) \quad (8)$$

$$- \Delta t \cdot (v(t) + \frac{\Delta t}{2} \cdot (a(t) + a(t + \Delta t))) + \frac{\Delta t^2}{2} \cdot a(t + \Delta t) \quad (9)$$

$$= r(t). \quad (10)$$

We thus have shown that concerning the calculation of the positions, the Velocity Störmer Verlet method is time-reversible. This also means that  $\tilde{a}(t)$  equals  $a(t)$  since the acceleration only depends on the current position of the molecules. Now we will have a look at the velocities. Replacing  $\tilde{\Delta t}$  by  $-\Delta t$  in Eq. (6) yields:

$$\tilde{v}(t) = v(t + \Delta t) - \frac{\Delta t}{2} \cdot (a(t + \Delta t) + \tilde{a}(t)) \quad (11)$$

$$= v(t + \Delta t) - \frac{\Delta t}{2} \cdot (a(t + \Delta t) + a(t)). \quad (12)$$

Substituting Eq. (2) into Eq. (12) results in:

$$\tilde{v}(t) = v(t) + \frac{\Delta t}{2} \cdot (a(t) + a(t + \Delta t)) - \frac{\Delta t}{2} \cdot (a(t + \Delta t) + a(t)) \quad (13)$$

$$= v(t). \quad (14)$$

Thus, the velocity calculation is also time-reversible.

- (b) It is enough to consider the position update. We carry out the same analysis as for the Velocity Störmer Verlet method:

$$\begin{aligned} \tilde{r}(t + \Delta t + \tilde{\Delta t}) &= r(t + \Delta t) + \tilde{\Delta t} v(t + \Delta t) \\ &\stackrel{\text{Eq.(3),Eq.(4)}}{=} r(t) + \Delta t v(t) + \tilde{\Delta t}(v(t) + \Delta t a(t)) \\ &\stackrel{\tilde{\Delta t} = -\Delta t}{=} r(t) - \Delta t^2 a(t) \\ &\neq r(t). \end{aligned} \quad (15)$$

## 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

$$a_1(t) := F(r_1(t) - r_2(t)) = 24 \left( \frac{1}{(r_1(t) - r_2(t))^7} - \frac{2}{(r_1(t) - r_2(t))^{13}} \right) \quad (16)$$

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.

**Solution:** First, we can compute the new position of the molecule 1:

$$\tilde{r}_1(t + \Delta t) = r_1(t) + \Delta t v_1(t) + \frac{\Delta t^2}{2} (a_1(t) + \epsilon) = r_1(t + \Delta t) + \frac{\Delta t^2}{2} \epsilon. \quad (17)$$

In order to determine the new velocity, we need to obtain an approximate expression for the new (disturbed) force at  $t + \Delta t$ . For this purpose, we approximate the force via (a very crude!) Taylor expansion,

$$F(d + h) \approx F(d) + F'(d)h, \quad (18)$$

where the first derivative of the force  $F'$  is given by

$$F'(d) = 24 \left( \frac{-7}{d^8} + \frac{26}{d^{14}} \right). \quad (19)$$

The (disturbed) acceleration/force at  $t + \Delta t$  thus evolves at (we use  $d := r_1(t + \Delta t) - r_2(t + \Delta t)$  for the sake of compact writing):

$$\tilde{a}_1(t + \Delta t) = F(\tilde{r}_1(t + \Delta t) - r_2(t + \Delta t)) \approx F(d) + F'(d) \frac{\Delta t^2}{2} \epsilon = a_1(t + \Delta t) + F'(d) \frac{\Delta t^2}{2} \epsilon. \quad (20)$$

The new velocity can thus be approximated by:

$$\begin{aligned} \tilde{v}_1(t + \Delta t) &= v_1(t) + \frac{\Delta t}{2} \cdot (\tilde{a}_1(t) + \tilde{a}_1(t + \Delta t)) \\ &= v_1(t) + \frac{\Delta t}{2} \cdot \left( a_1(t) + \epsilon + a_1(t + \Delta t) + F'(d) \frac{\Delta t^2}{2} \epsilon \right) \\ &= v_1(t + \Delta t) + \frac{\Delta t \epsilon}{2} \left( 1 + F'(d) \frac{\Delta t^2}{2} \right). \end{aligned} \quad (21)$$

Though we linearised the problem and thus obtain an error term of order  $O(\epsilon)$ , it is interesting to consider the factor

$$\left( 1 + F'(d) \frac{\Delta t^2}{2} \right) = 1 + 12\Delta t^2 \left( \frac{-7}{(r_1(t + \Delta t) - r_2(t + \Delta t))^8} + \frac{26}{(r_1(t + \Delta t) - r_2(t + \Delta t))^{14}} \right)$$

in the latter equation. Due to the power-14 term, the error will inherently increase for small distances  $r_1(t + \Delta t) - r_2(t + \Delta t)$ ! Strong deviations in the particle trajectories or even instabilities are thus expected in this case.

In most relevant applications, however, this does not pose a severe problem: typically, a random movement of the molecules is required where this additional “numerical” random noise does often not yield any significant impact. Still, very small time steps are required to keep the simulation stable.

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

$$F_{ij} = \frac{m_i m_j}{r_{ij}^3} \vec{r}_{ij} \quad (22)$$

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<sup>1</sup> in Tab. 1.

- Write a matlab method `initialise()` which returns the initial positions, velocities and forces as well as the masses of the planets. Test the method.
- 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.
- 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.
- 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.
- Write a matlab method `explEuler(positions, velocities, dt, masses)` which carries out explicit Euler time stepping (instead of the Verlet scheme). Re-run your simulation with this time stepping method. What do you observe?

### Solution:

- See `initialise.m`.
- See `computeForces.m`.
- See `velocityStoermerVerlet.m`.

<sup>1</sup>See also M. Griebel, S. Knapek, G. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer, 2007.

(d) See `planets.m`.

(e) See `planets.m`. We can observe that the planets start drifting away from their orbit when using the explicit Euler scheme.

This behaviour does not occur for the velocity Störmer Verlet scheme.