

Adaptive time stepping for n-body simulations

Dan Čermák

November 18, 2014

1 Motivation

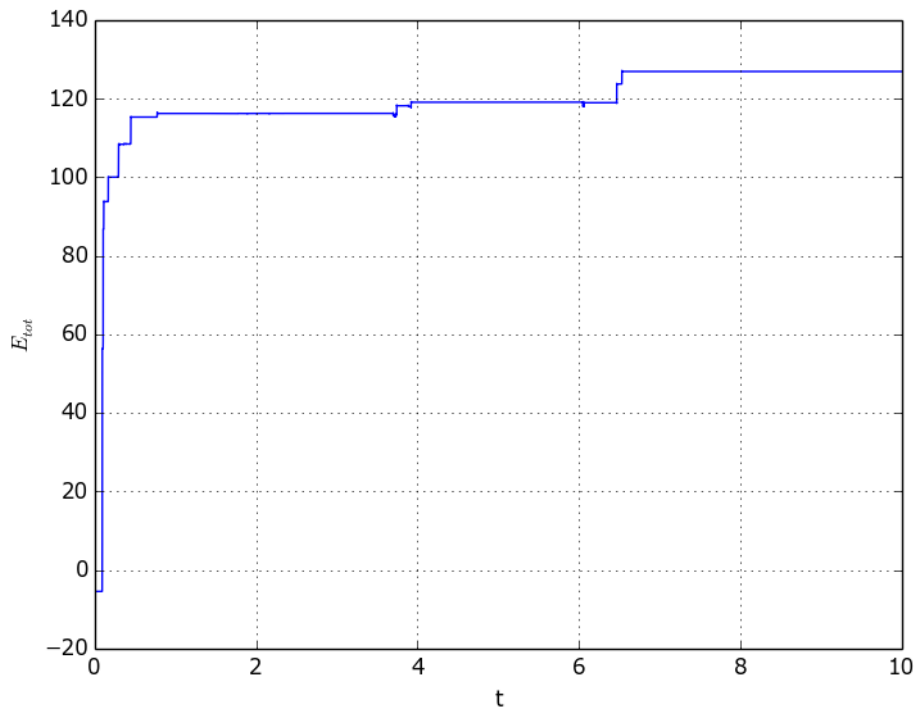


Figure 1: Typical result of close encounters with too large time steps on the total energy.

We require an adaptive time stepping routine to adjust Δt during close encounters of two bodies. If we do not adjust it, the integrator is usually not able to correctly calculate the outcome. The typical result of too large time steps is the sudden ejection of one object or pairs of objects (depending on the mass difference). This increases the total energy of the system and thus creates wrong results. You can find an example in figure 1.

2 Implementation

There is no correct way how to implement adaptive time steps.

The most robust way is to estimate the local truncation error and adjust the time step accordingly. This however requires usually additional evaluations of the differential equation and if done incorrectly (which is by the way very easy) causes more problems then it solves.

A very common way is to estimate typical time scales of the physical system and to adjust the time step according to some empiric relation. This usually works pretty well if one is able to include all time scales in the calculation. On the downside this is usually quite tricky and requires a lot of testing.

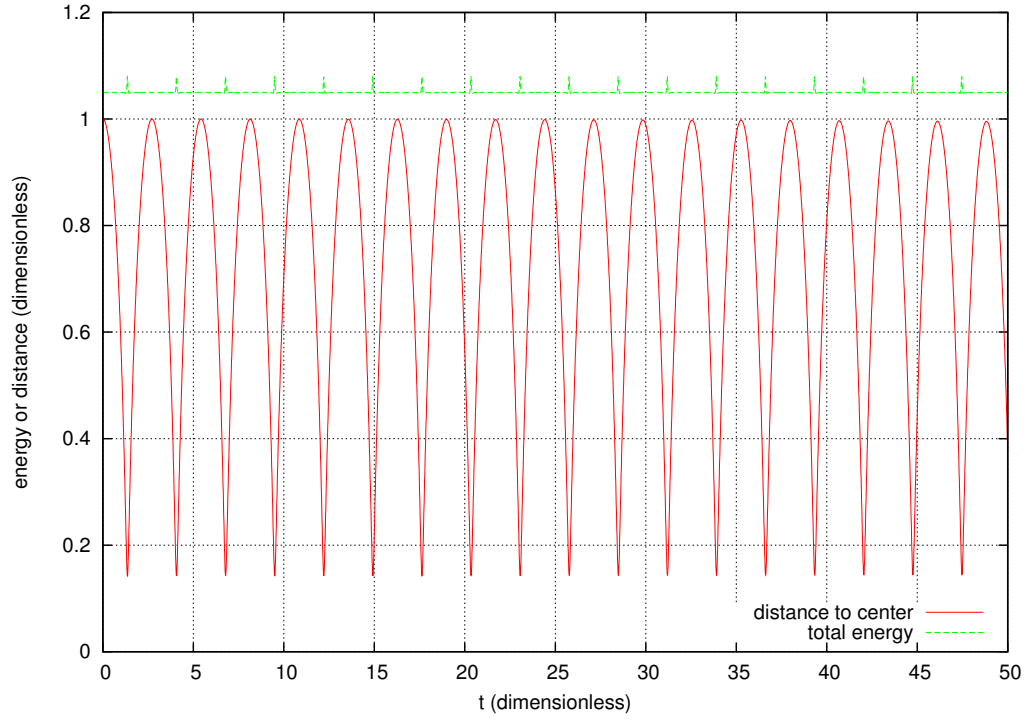


Figure 2: Simulation result with a fixed time step of the system from exercise sheet 3 task 1 a. All values are in dimensionless units as defined in task 2 a on the same exercise. The energy is multiplide by -1200 to fit the scale.

In my opinion the easiest solution is to use the total energy of the system, as we know that it should be conserved. When using the leapfrog we might ask ourselves whether this is really an option, as the total energy is conserved. Actually the total energy is not perfectly conserved, it is just bound in a small interval as long as there are no close encounters and the time step is constant. Please consider therefore figure 2. As you can clearly see, the energy has a spike¹ when the objects are close. We can therefore exploit the energy difference between two steps as an indicator for the distance of the objects, as we know that it will increase noticeably if the objects get close. With this we will be able to treat close encounters properly.

The actual implementation is very simple. First we have to calculate the normalized energy difference between two steps:

$$\Delta E_{\text{norm}} = \left| \frac{E_n - E_{n-1}}{E_{n-1}} \right| \quad (1)$$

If ΔE_{norm} is larger than a threshold then:

$$\Delta t_{\text{new}} = \frac{1}{2} \cdot \Delta t_{\text{old}} \quad (2)$$

Else:

$$\Delta t_{\text{new}} = 2 \cdot \Delta t_{\text{old}} \quad (3)$$

Furthermore we have to ensure that Δt_{new} does not become too small or too large (the latter happens nearly never, the former occurs quite frequently). This is done by setting:

$$\Delta t_{\text{new}} := \min(\max(\Delta t_{\text{new}}, \text{min value}), \text{max value}) \quad (4)$$

For me the following values worked:

$$\begin{aligned} \text{min value} &= 10^{-10} \\ \text{max value} &= 1 \\ \text{threshold} &= 10^{-10} \end{aligned} \quad (5)$$

You can see an example in figure 3 on the following page. As you can clearly see, the adaptive time stepping treats the close encounters properly and the total energy stays bound.

¹this is actually a dip, as the energy is multiplied by a negative number

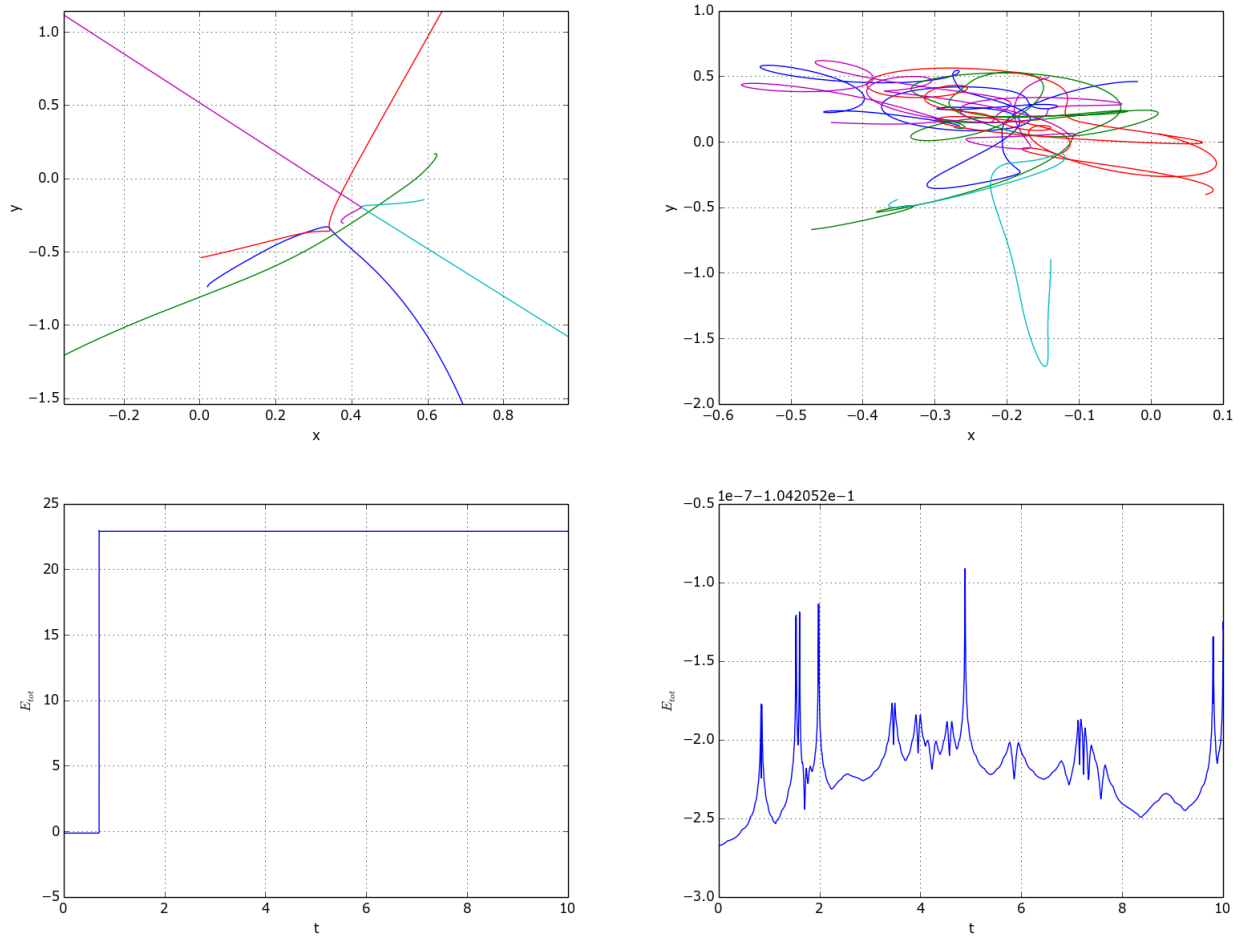


Figure 3: Simulation of 5 objects at random initial positions (like in exercise 2 e, sheet 3) simulated with adaptive and constant time stepping. The upper left image shows the positions in the x-y plane of the objects with a constant time step, whereas the image beside it shows the same but with adaptive time steps (the initial conditions are different, but only the qualitative result is interesting). The lower row shows the energy as a function of time for the same runs as in the images above.

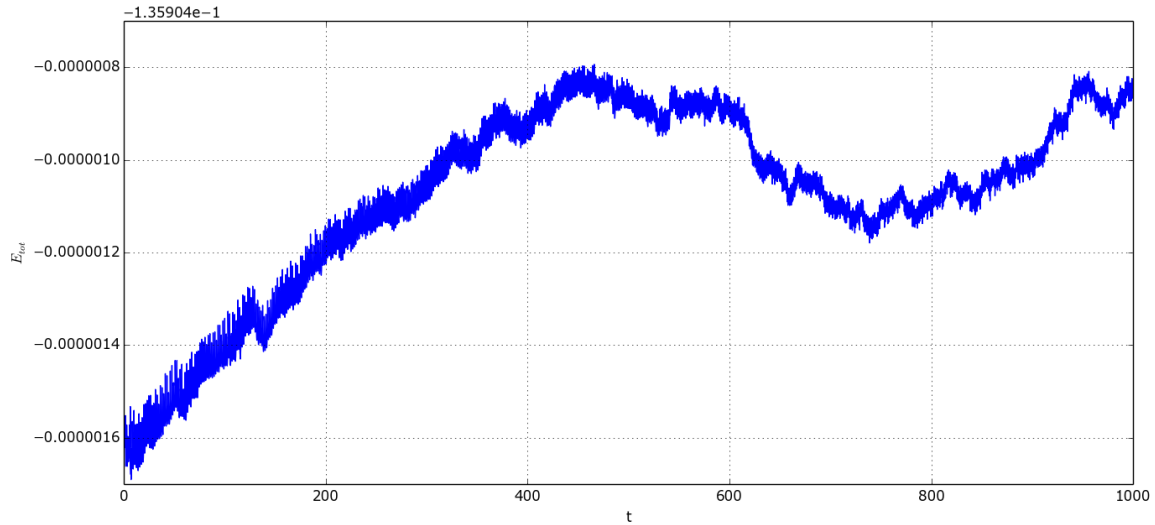


Figure 4: Simulation of 5 objects at random initial positions (like in exercise 2 e, sheet 3) simulated with adaptive time stepping. The plot shows the total energy as a function of time. As you can see there is an overall energy drift even though the leapfrog algorithm was used.

3 Additional considerations

3.1 Energy error

The calculation of the energy error in equation (1) on page 2 has the problem, that it will create a NaN if E_{n-1} is exactly zero. As that is unlikely, no special treatment has been done.

One could however check whether E_{n-1} is zero and in that case choose a reasonable value for the normalized difference (for instance 1 if one replaces the denominator of equation (1) on page 2 by E_n).

3.2 Additional threshold

It might not be very smart to refine the time step every time (as it is done currently). One could for instance introduce an interval in which the time step remains unchanged and only refine it if it falls out of this interval. For simplicity this was not implemented and thus not tested, therefore no guarantee that it works.

3.3 Long term evolution

When one changes the time step, the symplectic behavior of the leapfrog is partially lost. Thus an overall energy drift can happen. This is the price one has to pay for treating close encounters (now you can also imagine why big n-body simulations have high precision solvers for the close encounters but use low order solvers for the remaining time). You can see this in figure 4.