When you start coding the simulation of the motion of the Argon atoms, you soon find that the normal SI units (kg, m, s, etc.) lead to numbers of quite different magnitude, e.g. the mass of the Argon atoms is $6.6 \times 10^{-26}\,\mathrm{kg}$ and typical distances are of order Ångstrom, $10^{-10}\,\mathrm{m}$. Working with these different orders of magnitude is cumbersome, and in the simulation prone to round-off errors. It is hence useful to identify some natural units in the simulation, so we can make the variables to have the same order of magnitude.

There are two main ways of doing this. The first one is to identify multiples (or here fractions) of the SI base units and run the simulation using this. In the case of the Argon atom simulation, we could, for instance, use a unit length U_{L} of $1\,\mathrm{nm}$ and a unit mass U_{M} of $10^{-27}\,\mathrm{kg}$. We would then have an atom mass of $m = 66\,U_{\mathrm{M}}$ and an interaction length of $\sigma = 0.3405\,U_{\mathrm{L}}$. *Question for you: What do we pick for our time unit U_{T} ?* We can then build all the units we need from this base. As an example, energies are expressed as multiple of a base energy $U_{\mathrm{E}} \equiv U_{\mathrm{M}} \cdot U_{\mathrm{L}}^2 \cdot U_{\mathrm{T}}^{-2}$.

This technique is widely applied and allows simulation practitioners to deal with more sensible numbers. Simulations of the solar system would, for instance, define U_{M} as the mass of the Earth or Jupiter and U_{L} as the distance between the Earth and the Sun. Simulations of the atmosphere will use kilometers as their base length unit and so on.

However, this is not what we are going to do. An other approach is to identify sensible dimensionless quantities. The extra advantage this as is that it allows us to make abstraction of the dimension aspects and focus on the physics of the model.

From the Lennard-Jones potential we can already find two natural units, namely σ for position and ϵ for energy. Let us thus define $\tilde{\mathbf{x}} = \mathbf{x}/\sigma$ (and hence also $\tilde{r} = r/\sigma$). We can then define a dimensionless Lennard-Jones potential

$$\tilde{U}\left(\tilde{r}\right) = U\left(r\right)/\epsilon = 4\left(\tilde{r}^{-12} - \tilde{r}^{-6}\right). \tag{1}$$

Next we derive Newton's equation for dimensionless units. We start from

$$\frac{d^2\tilde{\mathbf{x}}}{dt^2} = \sigma^{-1}\frac{d^2\mathbf{x}}{dt^2} = -\sigma^{-1}\frac{1}{m}\nabla U\left(r\right) = -\frac{\epsilon}{m\sigma}\nabla\tilde{U}\left(\tilde{r}\right) = -\frac{\epsilon}{m\sigma^2}\tilde{\nabla}\tilde{U}\left(\tilde{r}\right). \tag{2}$$

The only variable that still has a dimension is time t. If we define also a dimensionless time

$$\tilde{t} = t/\sqrt{\frac{m\sigma^2}{\epsilon}}, \tag{3}$$

then Newton's equation takes the following simple form:

$$\frac{d^2\tilde{\mathbf{x}}}{d\tilde{t}^2} = -\tilde{\nabla}\tilde{U}\left(\tilde{r}\right). \tag{4}$$

We can leave away the $\sim$'s and simply say that length is measured in units of σ, energy in units of ϵ and time in units of $\sqrt{m\sigma^2/\epsilon}$. *Question for you: What do the discretized equations for positions and velocities we wrote last time look like in this framework?*

The advantages of this approach are:

- Less cumbersome notation of formulas in the program (and thus less error prone!)
- Simpler equations that are easier to code.
- Insight into what are the expected length and time scales in our system.

Let's focus on this last item. If we put the numbers for Argon in the units of time, we arrive at

$$\sqrt{m\sigma^2/\epsilon} = 2.15 \times 10^{-12}\,\mathrm{s}. \tag{5}$$

To check if this is a typical time for our system, we double check with the expected velocities. From the equipartition theorem we have $v = \sqrt{3k_BT/m}$. The natural unit of velocity in our units is

$$\tilde{v} = \frac{\tilde{x}}{\tilde{t}} = \frac{x}{\sigma}\sqrt{\frac{m\sigma^2}{\epsilon}}\frac{1}{t} = v\sqrt{\frac{m}{\epsilon}} \quad \rightarrow \quad \tilde{v} = v\Big/\sqrt{\frac{\epsilon}{m}} \tag{6}$$

Question for you: Do the units check out?

Hence, in dimensionless units, and using the equipartition theorem, we get:

$$\tilde{v} = v/\sqrt{\epsilon/m} = \sqrt{3k_BT/\epsilon}. \tag{7}$$

As discussed before, for Argon one has $\epsilon/k_B \approx 100\,\mathrm{K}$. This means that the ratio of these two velocities is of order one if we simulate the system at temperatures of around 100 K. *At this system energy (temperature), the particles thus move a typical distance of order σ in time $\sqrt{m\sigma^2/\epsilon}$.*

As discussed previously, we need to approximate the continuous trajectories in the real system by sufficiently fine discretised trajectories of our MD simulation. This is especially the case for particles in close contact (at a typical distance σ) where we should make sure that e.g. the particles do not pass through each other. From this we can conclude that the time step h should be smaller than one (of order 10^{-3} to 10^{-2}) in our internal units. In other words, we pick a time-step length h such that particles move a fraction of their typical distance per step.

Second Milestone

Derive the expression of the kinetic energy in dimensionless units.

Update your molecular dynamics code to use dimensionless units and simulate *a few* atoms in three dimensions.

Plot the evolution of the kinetic, potential and total energy. Check what happens when you change the time-step length h (defined in the previous lecture) and especially what happens if you increase it beyond the numbers introduced just above. Try for instance particles on a collision course.

Hint: Write the energies in a separate file from the trajectories and plot the content of that file.