**Molecular dynamics simulation algorithm:**

The idea:

The energy function allows us to calculate the force experienced by any atom given the positions of the other atoms.

Newton’s laws tell us how those forces will affect the motions of the atoms.

Basic algorithm:

Divide time into discrete time steps, no more than a few femtoseconds (10–15 s) each

• At each time step:

– Compute the forces acting on each atom, using a molecular mechanics force field

– Move the atoms a little bit: update position and velocity of each atom using Newton’s laws of motion

More details:

First, we need to extract the first location (x, y and z coordinates) of the atoms from the pdb file, we get from our client.

We can assume all atoms start with zero velocity.

Second, divide time into discrete time steps, no more than a few femtoseconds (10–15 s) each.

Later, we need to calculate the total force in each axis (x, y and z).

As we all remember from physics 1m, we can get the force with the well-known formula:



where x represents coordinates of all atoms, and U is the potential energy function

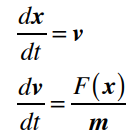
(for now I couldn’t find the right formula for the potentially energy, maybe the client knows)

Newton’s second law: F = ma

– where F is force on an atom, m is mass of the atom, and a is the atom’s acceleration

Velocity is the derivative of position, and acceleration is the derivative of velocity.

We can thus write the equations of motion as:

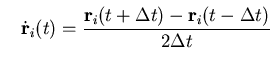


**Very important to remember - this is a system of ordinary differential equations – For n atoms, we have 3n position coordinates and 3n velocity coordinates!**

• “Analytical” (algebraic) solution is impossible

We obtain the location and velocity:

  
The error in the atomic positions is of the order of $\Delta t^4$.



with an error of the order of $\Delta t^2$.

Because the time step is very short, we can assume that in this period the velocity is constant.

**I used the leapfrog algorithm.**

**The leapfrog algorithm is computationally less expensive than the Predictor-Corrector approach for example and requires less storage.**

**This could be an important advantage in the case of large-scale calculations.**

**Moreover, the conservation of energy is respected, even at large time steps.**

**Therefore, the computation time could be greatly decreased when this algorithm is used.**