Contents

1 Langevin Equation

The Langevin Equation for the i-th particle is:

$$m_i\ddot{\vec{x}}_i = \vec{F}(\vec{x}_i(t)) - \gamma_i m_i \dot{\vec{x}}_i + \vec{R}_i \tag{1}$$

where $\vec{F}(\vec{x}_i(t)) = \vec{F}_i$ is the interaction force, γ_i is the collision frequency and $\vec{R}_i$ represents a random force which is related to the mass m_i, the Temperature T and the collision frequency γ_i by:

$$\langle \vec{R}_i(t)\vec{R}_j(t')\rangle = 2m_l k_B T \gamma_l \delta_{i,j}\delta(t-t') \tag{2}$$

and

$$\langle \vec{R}_i(t)\rangle = 0 \tag{3}$$

This force can be calculated using a Gaussian distribution

$$\rho(\vec{R}_i) = \frac{1}{\sqrt{4\pi\gamma_i m_i k_B T}} \cdot e^{\frac{-\vec{R}_i^2}{4\gamma_i m_i k_B T}} \tag{4}$$

or using

$$\vec{R}_i(t) = \sqrt{2k_B T \gamma_i m_i}\eta(t) \tag{5}$$

where $\eta(t) = \dot{W}(t)$ is a white-noise ($W(t)$ being a Wiener process). In order to solve the differential equation one needs to integrate the equation of motion. A second-order integrator for Langevin equations was found by Eric Vanden-Eijnden and Giovanni Ciccotti [1] and is a generalizaton of the BBK integrator. The algorithm reduces to the velocity-Verlet algorithm when $\gamma_i = 0$.

2 Integrator

The Integrator found is

$$\begin{cases} \vec{v}^{(n+1/2)} = \vec{v}^{(n)} + \frac{1}{2}\Delta t\vec{f}(\vec{x}^{(n)}) - \frac{1}{2}\Delta t\gamma\vec{v}^{(n)} + \frac{1}{2}\sqrt{\Delta t}\sigma\vec{\xi}^{(n)} \\ \qquad\qquad - \frac{1}{8}\Delta t^2\gamma(\vec{f}(\vec{x}^{(n)}) - \gamma\vec{v}^{(n)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma\left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right) \\ \vec{x}^{(n)} = \vec{x}^{(n)} + \Delta t\vec{v}^{(n+1/2)} + \Delta t^{3/2}\sigma\frac{1}{\sqrt{12}}\vec{\eta}^{(n)} \\ \vec{v}^{(n+1)} = \vec{v}^{(n+1/2)} + \frac{1}{2}\Delta t\vec{f}(\vec{x}^{(n+1)}) - \frac{1}{2}\Delta t\gamma\vec{v}^{(n+1/2)} + \frac{1}{2}\sqrt{\Delta t}\sigma\vec{\xi}^{(n)} \\ \qquad\qquad - \frac{1}{8}\Delta t^2\gamma(\vec{f}(\vec{x}^{(n+1)}) - \gamma\vec{v}^{(n+1/2)}) - \frac{1}{4}\Delta t^{3/2}\gamma\sigma\left(\frac{1}{2}\vec{\xi}^{(n)} + \frac{1}{\sqrt{3}}\vec{\eta}^{(n)}\right) \end{cases} \tag{6}$$

where $(\vec{\xi}^{(n)}, \vec{\eta}^{(n)})$ are independent Gaussian variables with mean zero and covariance

$$\langle \xi_i^{(n)}\xi_j^{(n)}\rangle = \langle \eta_i^{(n)}\eta_j^{(n)}\rangle = \delta_{i,j} \qquad \langle \xi_i^{(n)}\eta_j^{(n)}\rangle = 0 \tag{7}$$

3 Reduced Units

As molecular dynamics actions take place on a small time- and spacescale it's convenient to change the unit system to the system of reduced units (MD units). Following changes are made

$$\tilde{r} \to r\sigma \tag{8}$$
$$\tilde{E} \to E\epsilon \tag{9}$$
$$\tilde{m} \to m m_a \tag{10}$$

where σ and ϵ are paramter of the Lennard-Jones potential (ϵ governs the strength of the interaction and σ defines a length scale, both just numbers) and m_a is the atomic mass. In order to derive the factor for the conversion factor of time one can use the fact that for example the formula for the kinetic energy shouldn't change under unit system transformations. This leeds to

$$\tilde{t} \to t\sqrt{\frac{m_a\sigma^2}{\epsilon}} \tag{11}$$

By setting $k_B = 1$ the MD unit of temperature is now also defined. The following table contains the most used quantities, as well as values for argon:

Physical quantity	Unit	Value for Ar
Length	σ	$3.4 \cdot 10^{-10}\ m$
Energy	ϵ	$1.65 \cdot 10^{-25}\ J$
Mass	m	$6.69 \cdot 10^{-26}\ kg$
Time	$\sigma(m/\epsilon)^{1/2}$	$2.17 \cdot 10^{-12}\ s$
Velocity	$(\epsilon/m)^{1/2}$	$1.57 \cdot 10^{2}\ m/s$
Force	ϵ/σ	$4.85 \cdot 10^{-12}\ N$
Pressure	ϵ/σ^3	$4.20 \cdot 10^{7}\ Nm^{-2}$
Temperature	ϵ/k_B	$120\ K$

Table 1: Conversation factors for MD units and specific values for argon

4 Lennard-Jones Potential

The Lennard-Jones Potential approximates the interaction between neutral atoms or molecules. A common expression is

$$U = 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) \tag{12}$$

where $r_{ij} = |\vec{r}_{ij}|$ is the distance between the particle i and j, σ is the finite distance at which the inter-particle potential is zero and ϵ is the depth of the potential well.
For simulations it is necessary to cut off the potential at a given radius r_c in order to reduce the simulationtime, and the potential becomes

$$U = \begin{cases} 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{13}$$

The force is

$$-\vec{\nabla}U = \vec{F}_{ij} = \begin{cases} \frac{48\epsilon}{\sigma^2}\left(\left(\frac{\sigma}{r_{ij}}\right)^{14} - \frac{1}{2}\left(\frac{\sigma}{r_{ij}}\right)^{8}\right)\vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{14}$$

When reduced units are used, the force reduces to the following form

$$-\vec{\nabla}U = \vec{F}_{ij} = \begin{cases} 48\left(r_{ij}^{-14} - \frac{1}{2}r_{ij}^{-8}\right)\vec{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases} \tag{15}$$

5 Implementation details

5.1 General algorithm

In the case where the force depends on all other particles (e.g. Lennard-Jones-Potential) one is forced to split the algorithm in two different parts. One for the calculation of the half-step for the velocity and the new position. And the other part for the calculation of the new velocity. A pseudocode for the implementation could be like Algorithm 1 and a expiration chart of this algorithm may look like Figure 1.

Data: Initial positions and velocities of m particles $(x_1^1, x_2^1, ..., x_m^1)$, $(v_1^1, v_2^1, ..., v_m^1)$
Result: Final positions $(x_1^n, x_2^n, ..., x_m^n)$ and velocities $(v_1^n, v_2^n, ..., v_m^n)$
initialization;
for $k \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** m **do**
    calculate force for all m particles;
    update positions of all particles and calculate velocity half step;
    calculate force for new positions;
    update velocity;
  end
end

Algorithm 1: Main algorithm

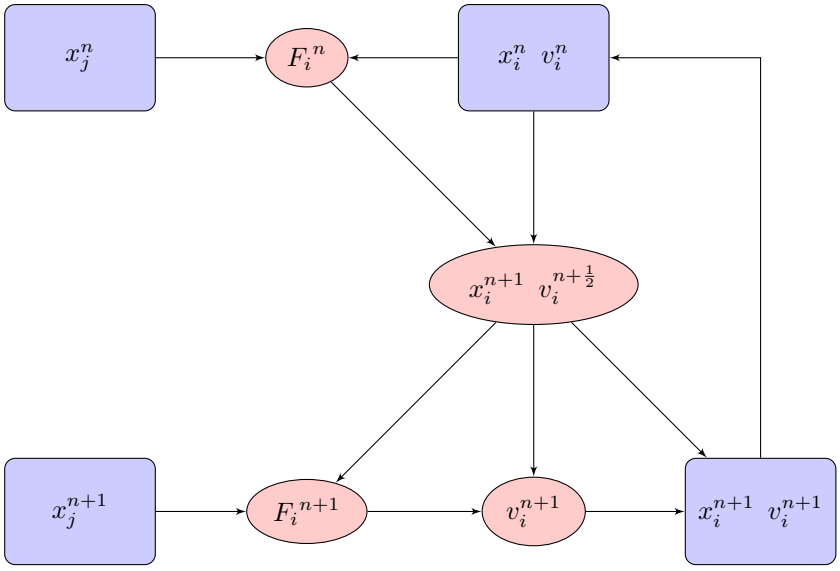

Figure 1: Expiration chart for the implementation. Red: calculation, Blue: storage

5.2 Boundary Conditions

5.2.1 Periodic boundary conditions

When simulating liquids the problem is the huge amount of particles. To reduce this problem one can introduce periodic boundary conditions, where the simulation takes place in a container of some kind and considering an infinte, space-filling array of idential copies of the simulation region. This means, the particles near the boundaries effect the particles on the other side of the simulation space. This leeds to a wraparound effect. If a particle moves outside the region, it has to be set on the other side. The following pseudocode takes care of this behavoir

Data: New position x_i of particle, Simulationspace size L
Result: If particle leaves simulation region, update position based on periodic boundary conditions
initialization;
if $x_i \geq \frac{L}{2}$ **then**
  $x_i = x_i - L$;
end
if $x_i \leq -\frac{L}{2}$ **then**
  $x_i = x_i + L$;
end

Algorithm 2: Particle wraparound

5.3 Force calculation

For a system of N particles, most of the simulation time is spend in calculating the force between different particles. A way to reduce this calculation time is to cut off after a specific radius r_c and only include particle within this radius. For the Lennard-Jones potential typical values are $r_c = 2.5\sigma - 3.5\sigma$. A way to improve this technique was introduced by Loup Verlet [3], the so-called Neighbor list or Verlet list. The idea is to use a second radius $r_v > r_c$ and calculate the force for all particles within this radius, but only at every n-th step. In order to estimate r_v one can use

$$r_v = r_c + \Delta r \tag{16}$$

with

$$\Delta r \leq n\tilde{v}\Delta t \tag{17}$$

where Δt is the time step, $\tilde{v}$ the root-mean-square. See 5.3 for a graphic representation of this method. This means that within the following $n-1$ steps no particles, other the ones in the list, will move into the cut-off radius. A typical number for n is 10 - 20. The computation time is reduced by a factor of 10.

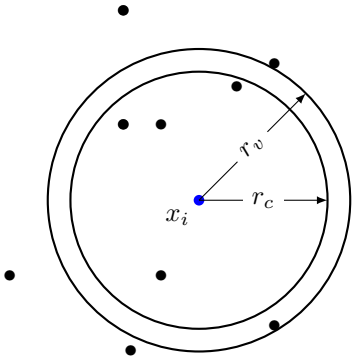

Figure 2: Neighbor list

For periodic boundary conditions the distance between to particles depends on the simulation region, as two particles near the boundaries can effect each other. A pseudocode for the calculation of the force F_i for the i-th particle is listed below (see algorithm 3) where a boolean *periodic* distinguishes between periodic and non-periodic boundary conditions.

Data: Position of all N particles at timestep m $(x_1^m, x_2^m, ..., x_N^m)$, boolean periodic
Result: Force F_i^m for i-th particle
initialization;
if $m \mod n == 0$ **then**
  for $j \leftarrow 1$ **to** N **do**
    calculate distance d from particle i to other particles;
    if *periodic* **then**
      if $d > \frac{L}{2}$ **then**
        $d = d - \frac{L}{2}$;
      end
    end
    if $d \leq r_v$ **then**
      add particle position to neighbor list;
    end
  end
end
calculate force on i-th particle based on neighbor list;

Algorithm 3: Neighbor list with boundary conditions

5.4 OpenMP

OpenMP [2] is application programming interface (API) which supports parallel programming in C/C++ and Fortran.

5.5 C functions

To calculate the distance between two particles the following C-function is used:

```
double distance(double xi[], double xj[], int l){
// Returns the (euclidian) distance between two points
  double d=0;
  int i;
  for (i = 0; i<l; i++){
    d += pow(xi[i]-xj[i],2);
  }
  d = sqrt(d);
```

```
    return d;
}
```

Listing 1: Returns the euclidian distance between two points

6 Check implementation for a simple potential

In order to check the implementation one can use a potential of the form

$$V(x) = \frac{1}{2}kx^2 \tag{18}$$

If $\gamma = 0$, the Langevin Equation reduces to a simple harmonic oscillator where the probability density for position can be derived in the following way:

6.1 Probability density for position for the harmonic oscillator

A particle moving with velocity $v(t)$ spends the time dt in a small region of space dx via

$$dt = \frac{dx}{v(t)} \tag{19}$$

For a periodic motion with period τ, the probability of finding the particle in this region is the ration of the time spent to the total time for one traversal $\tau/2$ is

$$P(x)dx = \text{Probability}[(x, x + dx)] = \frac{dt}{\tau/2} = \frac{2}{\tau}\frac{dx}{v(t)} \tag{20}$$

Since the kinetic energy is $T = \frac{mv^2}{2}$, we can rewrite the equation to

$$P(x) = \frac{2}{\tau}\sqrt{\frac{m}{2T}} = \frac{2}{\tau}\sqrt{\frac{m}{2(E - V(x))}} \tag{21}$$

For the harmonic oscillator we have $V(x) = \frac{1}{2}kx^2$ and $\tau = 2\pi\sqrt{\frac{m}{k}}$. The total energy is conserved and therefore one can use the initial conditions x_0 and v_0 to calculate the energy. This leads to

$$P(x) = \frac{1}{\pi}\sqrt{\frac{k}{2E - kx^2}} \tag{22}$$

After implementation one can check if the total energy is conserved. In figure 3 it can be seen that this is the case.

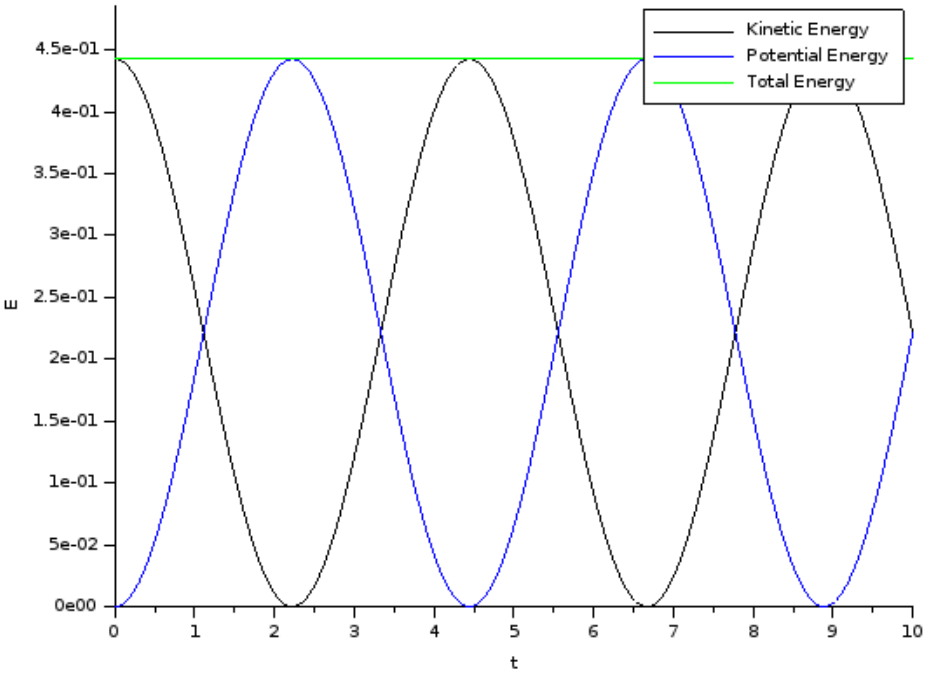

Figure 3: Kinetic-, Potential and total energy as a function of time

Another possibility to check the implementation is to plot $P(x)$.

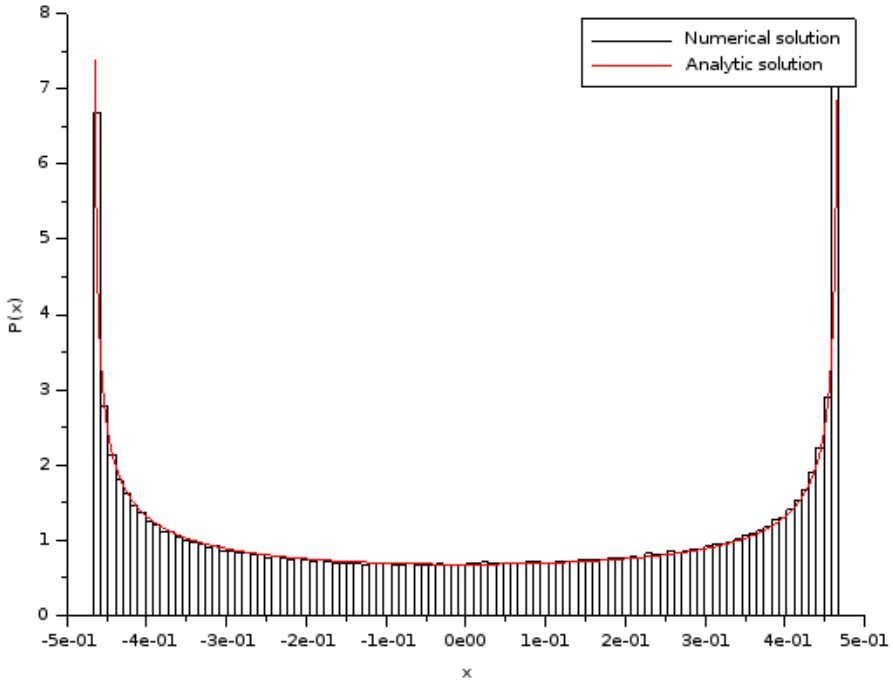

Figure 4: $P(x)$ for $g = 0$

In the cases where $\gamma \neq 0$ the probability density changes.

6.2 Probability density for the Langevin Equation and the harmonic oscillator

The Langevin Equation can be rewriten as a Fokker-Planck equation. In this case the corresponding equation is known as the Klein-Kramers equation

$$\frac{\partial P}{\partial t} + v\frac{\partial P}{\partial x} = \frac{\partial}{\partial v}\left(\gamma v - f(x) + \sigma^2 \frac{\partial}{\partial v}\right) P \tag{23}$$

The steady state $\left(\frac{\partial P_s}{\partial t} = 0\right)$ is given by a Boltzmann distribution:

$$P_s(x, v) = ce^{\frac{-U(x)}{k_B T}} e^{\frac{-mv^2}{2k_B T}} \tag{24}$$

The normalization condition leeds to $c = \frac{\sqrt{km}}{2\pi k_B T}$ and therefore

$$P_s(x, v) = \frac{\sqrt{km}}{2\pi k_B T} e^{\frac{-U(x)}{k_B T}} e^{\frac{-mv^2}{2k_B T}} \tag{25}$$

In the case of $V(x) = \frac{1}{2}kx^2$ the expression becomes

$$P_s(x, v) = \frac{\sqrt{km}}{2\pi k_B T} e^{\frac{-kx^2}{2k_B T}} e^{\frac{-mv^2}{2k_B T}} \tag{26}$$

The results can be seen in figure 5. The potential was $V(x) = \frac{1}{2}kx^2$ and following parameters were used: $k = 1$, $\gamma = 0.3$, $T = 0.5$ and 200000 timesteps.

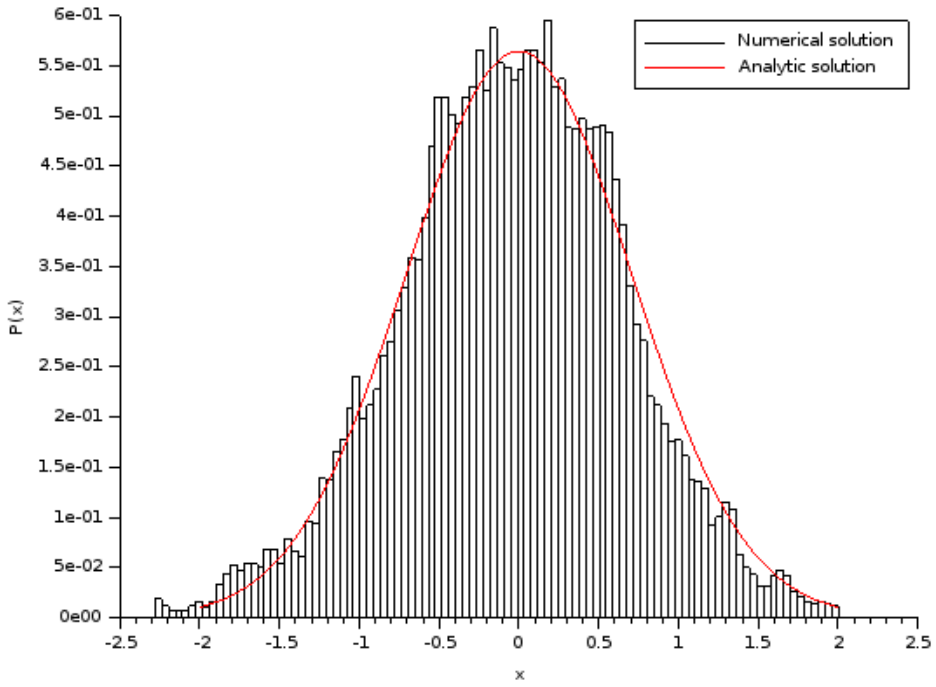

Figure 5: $P(x)$ for $\gamma = 0.3$, $T = 0.5$ and 200000 timesteps

References

[1] Giovanni Ciccotti Eric Vanden-Eijnden. Second-order integrators for langevin equations with holonomic constraints. *Chemical Physics Letters*, 429, 2006.

[2] OpenMP API for parallel programming, version 3.1.

[3] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159, 1967.