Algorithms and computations in physics (Oxford Lectures 2024)

Werner Krauth*
Laboratoire de Physique, Ecole normale supérieure, Paris (France)
Rudolf Peierls Centre for Theoretical Physics & Wadham College
University of Oxford (UK)

Fourth lecture: 6 February 2024
(version: 04/02/2024)

This is the first of two lectures on classical many-particle systems. In this first lecture, we move from Newtonian mechanics to Boltzmann mechanics and from classical mechanics to statistical mechanics, in a way that we promise to be surprising and, as usual, entirely example-based.

Contents

4 Many-particle systems. From Newtonian mechanics to Boltzmann mechanics.

In the hard-sphere model, all configurations have the same potential energy and there is no energetic reason to prefer any configuration over any other. Only entropic effects come into play. In spite of this restriction, hard spheres and disks show a rich phenomenology and exhibit phase transitions from the liquid to the solid state. These "entropic transitions" were once quite unsuspected, and then hotly debated, before they ended up poorly understood, especially in

*werner.krauth@ens.fr, werner.krauth@physics.ox.ac.uk

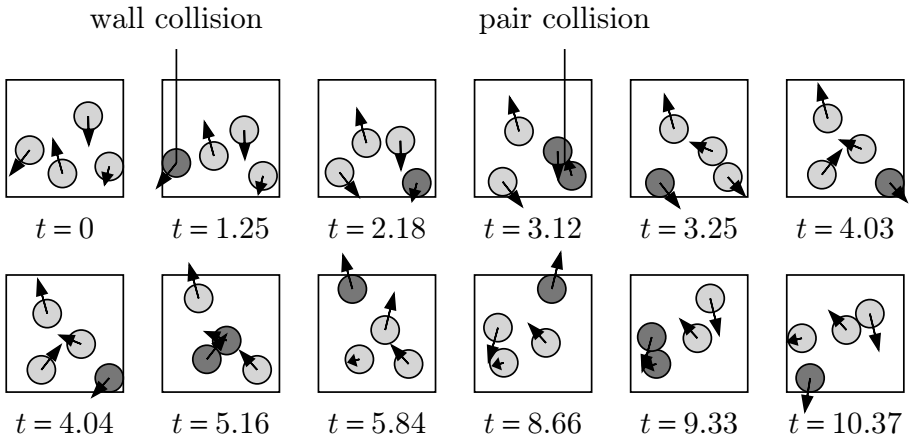


Figure 4.1: Newtonian evolution of four disks in a square box without periodic boundary conditions.

two dimensions. The physics of entropy will occupy us in Lecture 5. In the present lecture, our focus is on the emergence of statistical mechanics from classical mechanics.

4.1 Hard disks—Newton dynamics

We discuss Newtonian dynamics of hard disks, and we will later compare it to the point of view of statistical physics.

4.1.1 Event-driven molecular dynamics

Let us consider a model of hard disks in a box. Disks can undergo collisions with each other or with the walls. To get started with a naive [1] program, we realize that at any generic moment, there are $N(N-1)/2$ pairs of particles which could engage in pair collision, each indexed by a pair-collision time in the future and N individual wall collisions, also in the future. Up to the minimum of these times, the time evolution is straight, and at the next event, either a pair collision or a wall collision takes place (see Alg. 4.1 (`event-disks`)). Look here for a real-life

procedure `event-disks`
input $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}, t$
$\{t_{\rm pair}, k, l\} \leftarrow$ next pair collision
$\{t_{\rm wall}, j\} \leftarrow$ next wall collision
$t_{\rm next} \leftarrow \texttt{min}[t_{\rm wall}, t_{\rm pair}]$
for $m = 1, \ldots, N$**:**
  $\{\ \mathbf{x}_m \leftarrow \mathbf{x}_m + (t_{\rm next} - t)\mathbf{v}_m$
if $(t_{\rm wall} < t_{\rm pair})$ **then**
  $\{$ **call** `wall-collision` (j)
else:
  $\{$ **call** `pair-collision` (k, l)
output $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}, t_{\rm next}$

Algorithm 4.1: `event-disks`. Event-driven molecular dynamics algorithm for hard disks in a square box of sides 1.

Python program that we will motivate next. We now implement Alg. 4.1 (`event-disks`) without

[1] "naive" means "basically correct, but inefficient".

discretizing time. To do so, we consider an arbitrary pair of particles. They will collide when the norm of their spatial distance vector

$$\underbrace{\mathbf{x}_k(t) - \mathbf{x}_l(t)}_{\Delta\mathbf{x}(t)} = \underbrace{\Delta_\mathbf{x}}_{\mathbf{x}_k(t_0)-\mathbf{x}_l(t_0)} + \underbrace{\Delta_\mathbf{v}}_{\mathbf{v}_k-\mathbf{v}_l} \cdot (t - t_0) \tag{4.1}$$

equals twice the radius σ of the disks (see Fig. **??**). This can happen at two times t_1 and t_2, obtained by squaring eq. (4.1), setting $|\Delta\mathbf{x}| = 2\sigma$, and solving the quadratic equation

$$t_{1,2} = t_0 + \frac{-(\Delta_\mathbf{x} \cdot \Delta_\mathbf{v}) \pm \sqrt{(\Delta_\mathbf{x} \cdot \Delta_\mathbf{v})^2 - (\Delta_\mathbf{v})^2((\Delta_\mathbf{x})^2 - 4\sigma^2)}}{(\Delta_\mathbf{v})^2}. \tag{4.2}$$

The two disks will collide in the future only if the argument of the square root is positive and if they are approaching each other ($(\Delta_\mathbf{x} \cdot \Delta_\mathbf{v}) < 0$). The smallest of all the pair collision times obviously gives the next pair collision in the whole system (see Alg. 4.1 (`event-disks`)). Analogously, the parameters for the next wall collision follow from a straightforward time-of-flight analysis.

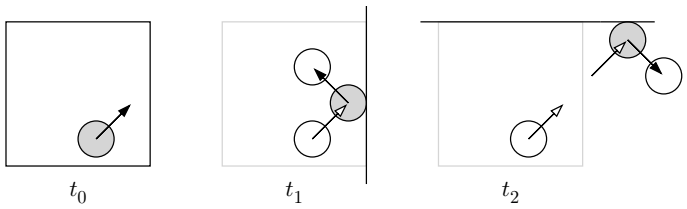


Figure 4.2: Wall collision. The time of a collision is easy to compute, and so is the new velocity

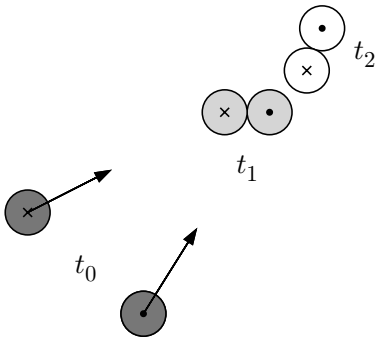
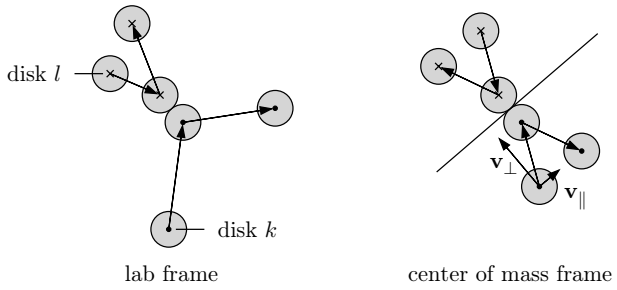


Figure 4.3: Approach of a pair of two disks, as programmed in eq. (4.2)

Pair collisions are best analyzed in the center-of-mass frame of the two disks, where $\mathbf{v}_k + \mathbf{v}_l = 0$ (see Fig. **??**). Let us write the velocities in terms of the perpendicular and parallel components $\mathbf{v}_\perp$ and $\mathbf{v}_\parallel$ with respect to the tangential line between the two particles when they are exactly in contact. This tangential line can be thought of as a virtual wall from which the particles rebound:

$$\underbrace{\begin{array}{rcl} \mathbf{v}_k &=& \mathbf{v}_\parallel + \mathbf{v}_\perp \\ \mathbf{v}_l &=& -\mathbf{v}_\parallel - \mathbf{v}_\perp \end{array}}_{\text{before collision}}, \quad \underbrace{\begin{array}{rcl} \mathbf{v}'_k &=& \mathbf{v}_\parallel - \mathbf{v}_\perp \\ \mathbf{v}'_l &=& -\mathbf{v}_\parallel + \mathbf{v}_\perp \end{array}}_{\text{after collision}}.$$

The changes in the velocities of particles k and l are $\mp 2\mathbf{v}_\perp$. Introducing the perpendicular unit vector $\hat{\mathbf{e}}_\perp = (\mathbf{x}_k - \mathbf{x}_l)/|\mathbf{x}_k - \mathbf{x}_l|$ allows us to write $\mathbf{v}_\perp = (\mathbf{v}_k \cdot \hat{\mathbf{e}}_\perp)\hat{\mathbf{e}}_\perp$ and $2\mathbf{v}_\perp = (\Delta_\mathbf{v} \cdot \hat{\mathbf{e}}_\perp)\hat{\mathbf{e}}_\perp$, where $2\mathbf{v}_\perp = \mathbf{v}'_k - \mathbf{v}_k$ gives the change in the velocity of particle k. The formulas coded into



lab frame center of mass frame

Figure 4.4: Computing the velocities after the pair collision.

procedure `pair-time`
input $\Delta_\mathbf{x}$ $(\equiv \mathbf{x}_k(t_0) - \mathbf{x}_l(t_0))$
input $\Delta_\mathbf{v}$ $(\equiv \mathbf{v}_k - \mathbf{v}_l \neq 0)$
$\Upsilon \leftarrow (\Delta_\mathbf{x} \cdot \Delta_\mathbf{v})^2 - |\Delta_\mathbf{v}|^2(|\Delta_\mathbf{x}|^2 - 4\sigma^2)$
if $(\Upsilon > 0$ and $(\Delta_\mathbf{x} \cdot \Delta_\mathbf{v}) < 0)$ **then**
$\quad\{\ t_{\rm pair} \leftarrow t_0 - \left[(\Delta_\mathbf{x} \cdot \Delta_\mathbf{v}) + \sqrt{\Upsilon}\right] / \Delta_\mathbf{v}^2$
else:
$\quad\{\ t_{\rm pair} \leftarrow \infty$
output $t_{\rm pair}$

Algorithm 4.2: `pair-time`. Pair collision time for two particles starting at time t_0 from positions $\mathbf{x}_k$ and $\mathbf{x}_l$, and with velocities $\mathbf{v}_k$ and $\mathbf{v}_l$.

Alg. 4.3 (`pair-collision`) follow. We note that $\hat{\mathbf{e}}_\perp$ and the changes in velocities $\mathbf{v}'_k - \mathbf{v}_k$ and $\mathbf{v}'_l - \mathbf{v}_l$ are relative vectors and are thus the same in all inertial reference frames. The program can hence be used directly with the lab-frame velocities.

procedure `pair-collision`
input $\{\mathbf{x}_k, \mathbf{x}_l\}$ (particles in contact: $|\mathbf{x}_k - \mathbf{x}_l| = 2\sigma$)
input $\{\mathbf{v}_k, \mathbf{v}_l\}$
$\Delta_\mathbf{x} \leftarrow \mathbf{x}_k - \mathbf{x}_l$
$\hat{\mathbf{e}}_\perp \leftarrow \Delta_\mathbf{x}/|\Delta_\mathbf{x}|$
$\Delta_\mathbf{v} \leftarrow \mathbf{v}_k - \mathbf{v}_l$
$\mathbf{v}'_k \leftarrow \mathbf{v}_k - \hat{\mathbf{e}}_\perp(\Delta_\mathbf{v} \cdot \hat{\mathbf{e}}_\perp)$
$\mathbf{v}'_l \leftarrow \mathbf{v}_l + \hat{\mathbf{e}}_\perp(\Delta_\mathbf{v} \cdot \hat{\mathbf{e}}_\perp)$
output $\{\mathbf{v}'_k, \mathbf{v}'_l\}$

Algorithm 4.3: `pair-collision`. Computing the velocities of disks (spheres) k and l after an elastic collision (for equal masses).

4.1.2 Chaos

Algorithm 4.1 (`event-disks`) is entirely deterministic, and we may think that it actually computes the positions and velocities of N hard disks at time t from the values at time $t = 0$. But this is not really the case. It suffices to run the program at different precision levels [2] in order

[2] this is easy to implement in the NumPy extension of Python.

to see that we can really *compute* positions and velocities up to a handful of collisions. Little errors in the numerical computations blow up inexorably, and change the sequence of collisions. This is manifestation of *chaos* that, in our case, is caused by the convex curvature of the disks.

4.1.3 Complexity of molecular dynamics, heaps

4.2 Hard disks—Boltzmann dynamics

We enter into the discussion of statistical mechanics proper, in the case of the hard-disk model, where things are easier than for general case, that we will sketch in Sec. **??**. The basic property that we can study is the equal-probability principle, that means that configurations with the same statistical weight have the same probability.

4.2.1 Equal-probability principle, direct-disk sampling

In the hard-disk case,

$$\pi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \begin{cases} 1 & \text{if configuration legal} \\ 0 & \text{otherwise} \end{cases}, \tag{4.3}$$

which, as in lecture 1, is to be understood with a Cartesian measure $\mathrm{d}\mathbf{x}_1, \ldots, \mathbf{x}_N$ on both sides. The sampling algorithm consists in the following Algorithm 4.4 (`direct-disks`) is one of a

procedure `direct-disks`
1 **for** $k = 1, \ldots, N$**:**
  $x_k \leftarrow \texttt{ran}(x_{\min}, x_{\max})$
  $y_k \leftarrow \texttt{ran}(y_{\min}, y_{\max})$
  for $l = 1, \ldots, k-1$**:**
    if $\texttt{dist}\,(\mathbf{x}_k, \mathbf{x}_l) < 2\sigma$**:** **goto** 1 (reject sample—tabula rasa)
output $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

Algorithm 4.4: `direct-disks`. Direct sampling for N disks of radius σ in a fixed box.

number of direct-sampling algorithms for this system, of which some are even fast, in the limit $N \to \infty$. The *tabula rasa* aspect of it can be understood easily.

4.2.2 Markov-disk sampling (reversible)

We now consider a reversible Markov-chain algorithm for four hard disks in a box.

procedure `markov-disks`
input $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ (configuration a)
$k \leftarrow \texttt{nran}\,(1, N)$
$\delta\mathbf{x}_k \leftarrow \{\texttt{ran}(-\delta, \delta)\,, \texttt{ran}(-\delta, \delta)\}$
if disk k can move to $\mathbf{x}_k + \delta\mathbf{x}_k$**:** $\mathbf{x}_k \leftarrow \mathbf{x}_k + \delta\mathbf{x}_k$
output $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ (configuration b)

Algorithm 4.5: `markov-disks`. Generating a hard-disk configuration b from configuration a using a Markov-chain algorithm (see Fig. **??**).

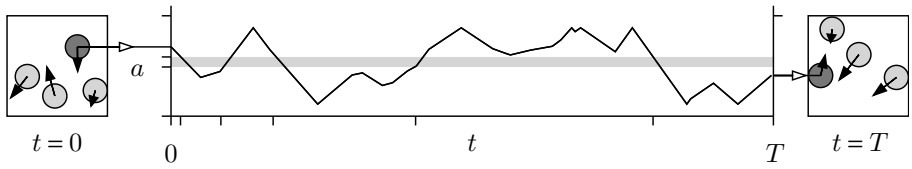


Figure 4.5: Density at position y computed the hard way, by following the entire trajectory

4.2.3 Observables

It can be computed exactly for given particle trajectories between times $t = 0$ and $t = T$:

$$\left\{ \begin{matrix} y\text{-density} \\ \text{at } y = a \end{matrix} \right\} = \eta_y(a) = \frac{1}{T} \sum_{\substack{\text{intersections } i \\ \text{with gray strip} \\ \text{in Fig. ??}}} \frac{1}{|v_y(i)|}. \tag{4.4}$$

In Fig. 4.5, there are five intersections (the other particles must also be considered). At each intersection, $1/|v_y|$ must be added, to take into account the fact that faster particles spend less time in the interval $[a, a + \mathrm{d}a]$, and thus contribute less to the density at a. A more leisurly approach consists in simpy analyzing stroboscopic pictures, that is, configurations at equal time intervals. This is the approach we also use for the Monte Carlo algorithm.

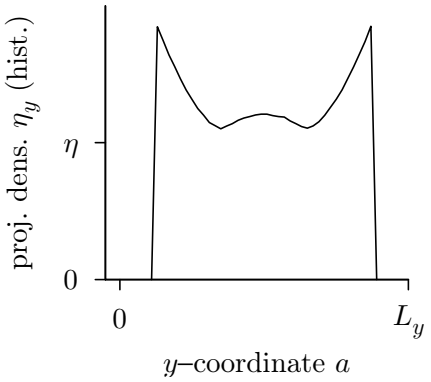


Figure 4.6: Projected density at position y, which is not constant.

4.3 Maxwell distribution, thermostats, Boltzmann distribution

4.3.1 Equal-probability principle for velocities

An obvious differences between molecular dynamics and Monte Carlo consists in that the former has positions and velocities whereas the latter only has positions. We just considered half of the problem, and it is not only the positions that satisfy an equal-probabability condition (under the given constraints) but also the velocities. For them, the condition is:

4.3.2 Thermostats and the Boltzmann distribution

In Sec. 4.2.3, we saw that molecular dynamics and Monte Carlo, in other words Newton and Boltzmann, gave absolutely the same results. We understand that, by construction, our Monte Carlo algorithm, by construction, outputs

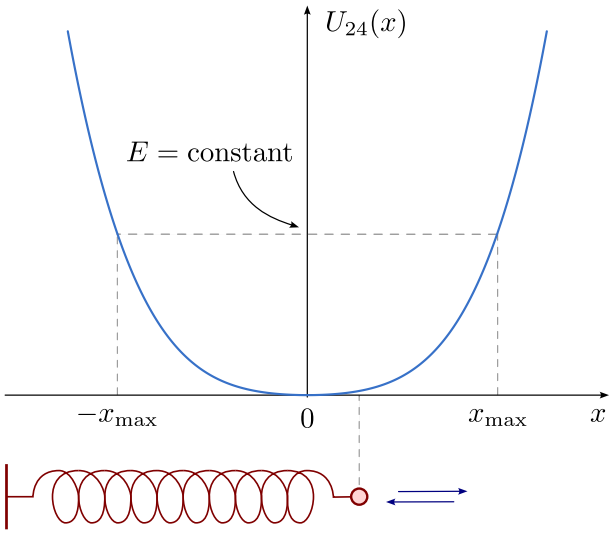


Figure 4.7: Isolated anharmonic oscillator

In between the turning points $-x_{\max}$ and $x_{\max}$ the kinetic energy (with the mass equal to unity) is $\frac{1}{2}(\mathrm{d}x/\mathrm{d}t)^2$, and conservation of energy can be written as

$$E = \frac{1}{2}\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + U_{24}(x) \Leftrightarrow \frac{\mathrm{d}x}{\mathrm{d}t} = \pm\sqrt{2\left[E - U_{24}(x)\right]}, \tag{4.5}$$

which gives

$$\mathrm{d}t = \pm\sqrt{\frac{1}{2\left[E - U_{24}(x)\right]}}\mathrm{d}x. \tag{4.6}$$

To simulate the isolated anharmonic oscillator, we could numerically integrate the first-order ordinary differential equation on the right of Eq. (4.5) over a quarter period and then piece together the entire trajectory of Fig. 4.8a. However, this method is specific to one-dimensional dynamical systems. To reflect the general case, we numerically integrate Newton's law for the force F:

$$F = m\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2}, \quad \text{with } F = -\frac{\mathrm{d}U_{24}}{\mathrm{d}x} = -x - x^3. \tag{4.7}$$

By substituting the time differential $\mathrm{d}t$ by a very small finite interval Δt, appropriate for stepping from t to $t + \Delta t$, and to $t + 2\Delta t$, and so on, we obtain

$$x(t + \Delta t) = x(t) + v(t)\Delta t, \tag{4.8}$$

$$v(t + \Delta t) = v(t) - (x + x^3)\Delta t. \tag{4.9}$$

4.3.3 Molecular dynamics with a thermostat

For hard spheres, we celebrated brillant agreement between molecular dynamics and Monte Carlo, and it was exactly valid at finite N. Agreement between Newton and Boltzmann can usually only be obtained by a detour. To see this, we return to the Newtonian dynamics of the anharmonic oscillator [3], but we take it out of isolation and have it interact with an infinite *bath* of hard spheres *via* thermostat (see Fig. 4.9.)

Statistical mechanics teaches us that, although all the particles in the heat bath are Maxwell-distributed, the thermostat behaves differently. In particular, because the latter lies at a fixed

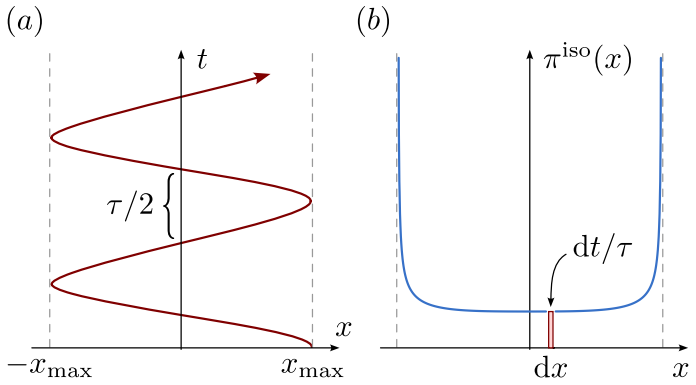


Figure 4.8: Trajectory of the isolated anharmonic oscillator

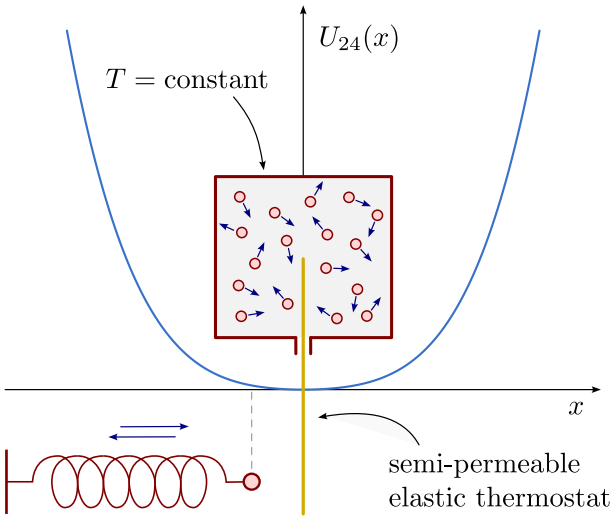


Figure 4.9: Anharmonic oscillator coupled to a heat bath

procedure `isolated-dynamics`
input x, v, t
$t \leftarrow t + \Delta t$
$x' \leftarrow x + v\Delta t$
$v \leftarrow v - \left(x + x^3\right) \Delta t$
$x \leftarrow x'$
output x, v, t

Algorithm 4.6: `isolated-dynamics`. Naive integration of Newton's equations for the isolated anharmonic oscillator (see Fig. 4.8).

position (up to an infinitesimal interval), its velocity follows the distribution

$$\pi(v)\mathrm{d}v = \beta|v|\mathrm{e}^{-\beta v^2/2}\mathrm{d}v, \tag{4.10}$$

often called the Maxwell boundary condition. It differs by the prefactor $\beta|v|$ from the Maxwell distribution of one velocity component. The velocity distribution of the thermostat in Eq. (4.10) can be sampled as

$$v = \pm\sqrt{\frac{-2\log \texttt{ran}(0,1)}{\beta}}, \tag{4.11}$$

(it is an exponential distribution of the random variable v^2).

procedure `thermostat-dynamics`
input x, v, t
$x' \leftarrow x + v\Delta t$
$t \leftarrow t + \Delta t$
$\Upsilon \leftarrow \texttt{ran}(0,1)$
if $x \cdot x' < 0$ and $\Upsilon < 1/2$**:**
$\left\{ \; v \leftarrow -\texttt{sign}(v)\sqrt{-2\beta^{-1}\log \texttt{ran}(0,1)} \right.$ (see Eq. (4.11))
else:
$\left\{ \begin{array}{l} v \leftarrow v - (x + x^3)\Delta t \\ x \leftarrow x' \end{array} \right.$
output x, v, t

Algorithm 4.7: `thermostat-dynamics`. Naive solution of Newton's equations for the anharmonic oscillator with the semi-permeable thermostat at $x = 0$ (see Fig. 4.10).

Output of Alg. 4.7 (`thermostat-dynamics`) can be histogrammed to see that the distribution of positions is exactly (up to discretization errors in Δt) the Boltzmann distribution of the anharmonic oscillator, proving (experimentally but by our own means) that the Boltzmann distribution describes a subsystem interacting with a heat bath. But what about the distribution of velocities? As we only give our particle a kick at $x = 0$, we'd surely suppose that it runs out of steam as it climbs up the potential. But this is not the case, and Fig. 4.11 illustrates it by histogramming probability distributions of the velocities at different values up the hill. As statistical mechanics dictates, we have independence of distributions of positions and velocities.

References

[1] L. Wasserman, *All of Statistics*. New York: Springer, 2004.

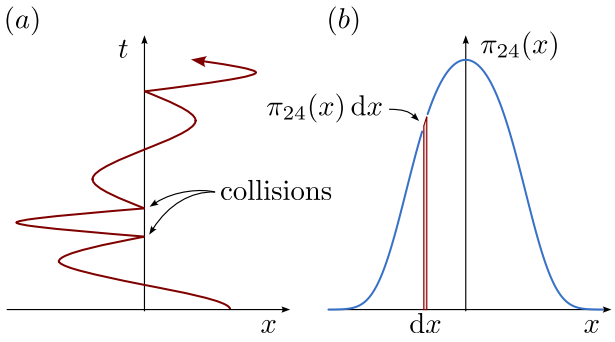


Figure 4.10: Trajectory of the anharmonic oscillator coupled to a heat bath.

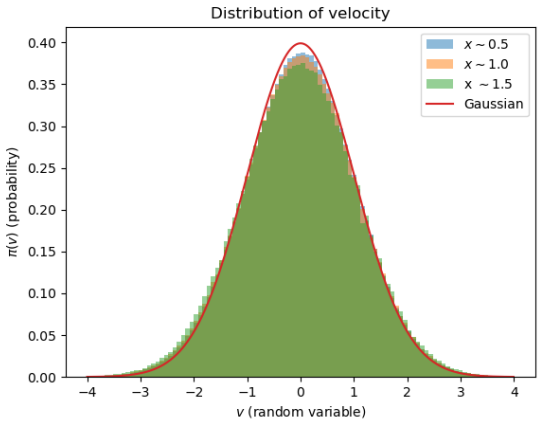


Figure 4.11: Velocity distributions of the anharmonic oscillator (Algorithm 4.7) at different approximate positions of x. Up to discretization effects, they all agree with the Maxwell distribution at inverse temperature $\beta = 1$.

[2] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[3] G. Tartero and W. Krauth, "Concepts in Monte Carlo sampling," *American Journal of Physics*, vol. 92, no. 1, p. 65–77, 2024.

[4] W. Krauth, *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, 2006.

[5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[6] W. Krauth, "Event-Chain Monte Carlo: Foundations, Applications, and Prospects," *Front. Phys.*, vol. 9, p. 229, 2021.

[7] E. A. J. F. Peters and G. de With, "Rejection-free Monte Carlo sampling for general potentials," *Phys. Rev. E*, vol. 85, p. 026703, 2012.