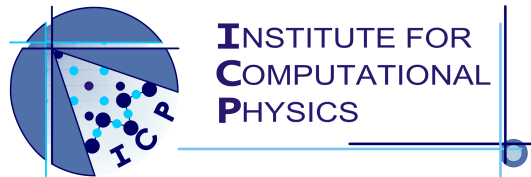


...script in development...



Simulation Methods in Physics 1

Prof. Dr. Christian Holm

Institute for Computational Physics
University of Stuttgart

WS 2012/2013

Contents

1	Introduction	3
1.1	Computer Science	3
1.1.1	Historical developments	3
1.1.2	TOP 500 Supercomputer list	4
1.1.3	Moore's law (1970)	5
1.2	Computer Simulations in Physics	6
1.2.1	The New Trinity of Physics	6
1.2.2	Optimization of simulation performance	8
1.2.3	Optimization of simulation algorithms	8
1.2.4	Computational approaches for different length scales	9
2	Molecular Dynamics integrators	11
2.1	Principles of Molecular Dynamics	11
2.1.1	Ergodicity	11
2.2	Integration of equations of motion	12
2.2.1	Algorithms	12
2.2.2	Estimation of time step	13
2.2.3	Verlet Algorithm	13
2.2.4	Velocity Verlet Algorithm	14
2.2.5	Leapfrog method	15
2.2.6	Runge-Kutta method	16
2.2.7	Predictor-Corrector methods	17
2.2.8	Ljapunov-Characteristics	17
3	Statistical mechanics - Quick & Dirty	20
3.1	Microstates	20
3.1.1	Distribution functions	20
3.1.2	Average values and thermodynamic limit	22
3.2	Definition of Entropy and Temperature	23
3.3	Boltzmann distribution	24
3.3.1	Total energy and free energy	25
3.4	Quantum mechanics and classical partition functions	26
3.5	Ensemble averages	26

1 Introduction

1.1 Computer Science

1.1.1 Historical developments

- The first calculating tool for performing arithmetic processes, the *Abacus* is used ~ 1000 b.C. primarily in parts of Asia.
- W. Schickhard (1592-1635): The *Calculating Clock* (1623) could add and subtract six-digit numbers.
- B. Pascal (1623-1662): The *Pascaline* invented in 1645 already could add and subtract 8 digits.
- G.W. Leibniz (1646-1716): First machine that could perform all four algebraic operations.
- G. Boole (1815-1864): Development of binary logic.
- Ch. Babbage (1792-1871): Development of the *Analytical engine*, which was designed to tabulate logarithms and trigonometric functions by evaluating approximate polynomials; it was the first attempt to design a mechanical general-purpose computer.
- H. Hollerith (1860-1929): Idea of using *Punchcards* for information technology.
- IBM 601 (1933): first electric multiplier.
- K. Zuse (1910-1995): Z1 (1938) was a binary electrically driven mechanical calculator with limited programmability; Z2 (1940) was purely electromagnetic (using telephone relay). Z3 (1940) was the first computer in the modern sense, being a *Universal Turing Machine*.
- Harvard Mark I (1943) was the first large-scale automatic digital computer in the USA (weight of 35 tons!).
- The *Eniac* was a Turing-complete computer using tubes with a von-Neumann CPU architecture.
- The Z4 (1949) at ETH Zürich was the first commercially available computer system.
- The *Los Alamos MANIAC* (1952) as another milestone in computer improvement.
- Around 1958: first computers with transistors and Integrated Circuits were built.
- 1960: DEC-PDP 1 was the worlds first mini-computer

- ~ 1970 : Intel IC 4004 a 4-bit processor with 2300 transistors
- ~ 1980 : First home computers were available

1.1.2 TOP 500 Supercomputer list

Today's ranking of supercomputers (high performance computing) is highly competitive and prestigious, the improvement of computational hardware due to a better performance finally results in cheaper components for home computers.

Here is the TOP 5 list (November 2012):

1. *Titan* - Oak Ridge National Laboratory, USA
2. *Sequoia* - Lawrence Livermore National Laboratory, USA
3. *K computer* - RIKEN, Japan
4. *Mira* - Argonne National Laboratory, USA
5. *JUQUEEN* - Forschungszentrum Jülich, Germany

...

27. *HERMIT* - HLR Stuttgart, Germany

As it can be seen, the HLRS is internationally well positioned.



Figure 1.1: *Top left*: Charles Babbage (1792-1871). *Top middle*: MANIAC computer (1952). *Top right*: Konrad Zuse (1910-1995). *Bottom*: Titan - Supercomputer, Oak Ridge National Laboratory, USA.

1.1.3 Moore's law (1970)

The prophecy of Moore allows a prediction of computational performance in the future. A popular formulation of Moore's law is given by:

Every two years the number of transistors on a chip doubles. Up to 2007 this rate has been maintained very well (Fig. 1.2). Nowadays the miniaturization is so far advanced that instead one improves the instruction sets (actually we use 64-bit computers) and tries to add more and more cores to a single CPU (for example Intel's Core2 processors, 2008).

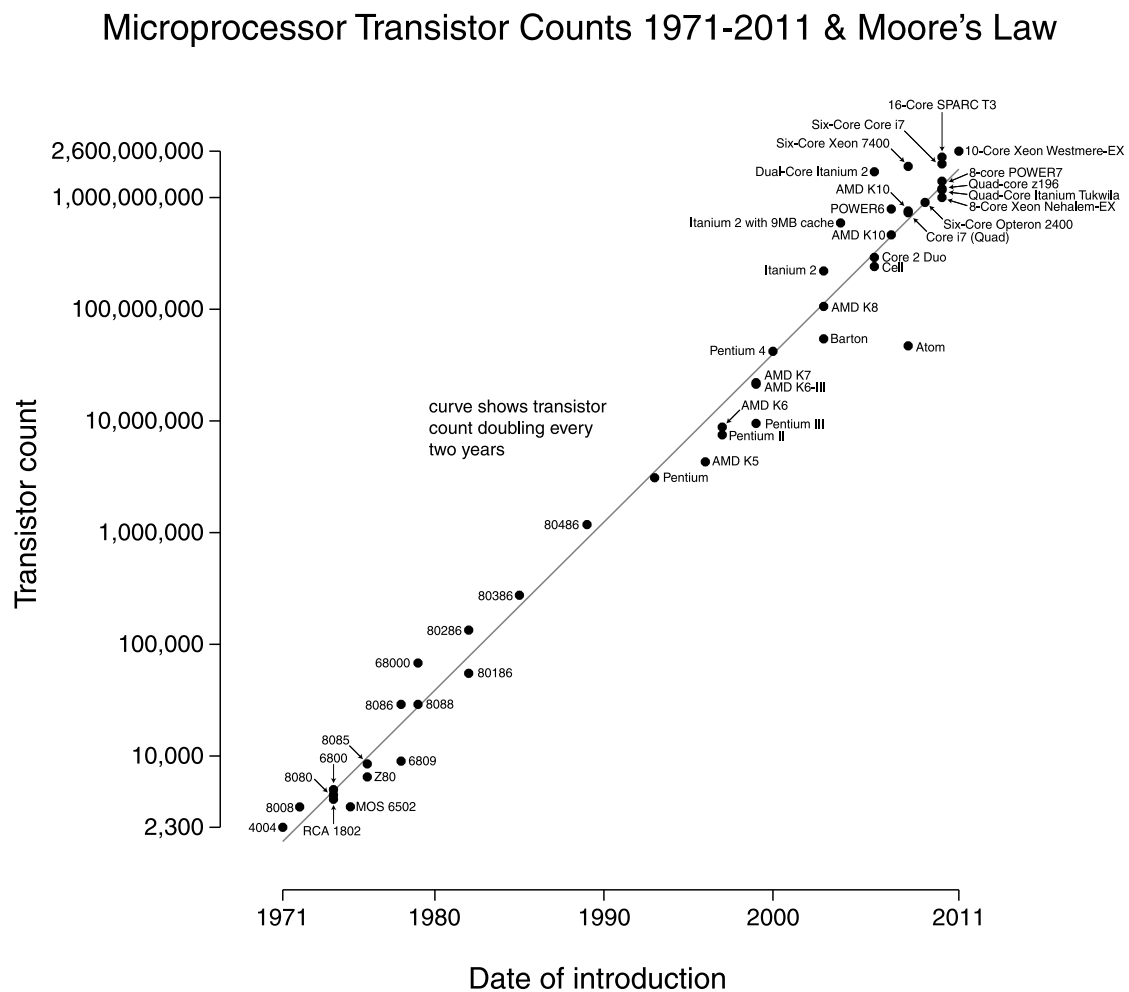


Figure 1.2: Plot of CPU transistor counts versus dates of introduction. The curve shows counts doubling every two years.

1.2 Computer Simulations in Physics

In this section we want to provide a short discussion on why it is useful to use computer simulations in physics and natural sciences in general. As many laws of nature can be expressed in mathematical laws, only few of them can be solved analytically. Analytical tools like differential calculus are best suited to solve *linear* problems, so in most cases one is limited to very simple systems (in fact often *toy models* have to be used) that possess analytical solutions. Analytical solutions, *e. g.* are available for a harmonic oscillation of a mass attached to a spring, the quantum mechanical harmonic oscillator or two-body problems in quantum mechanics.

In fact the majority of natural phenomena are *nonlinear*. For example, typical problems that cannot be solved analytically are multi-particle problems in QM ($N \geq 2$) or classical many body physics. As relatively few nonlinear problems can be solved analytically, the increasing computational power provides a tool to explore these phenomena in more detail.

Typical physical procedures:

- By constructing appropriate theories (i.e. performing perturbation expansions, linearisation of non-linear equations) an analytical solvable problem can be approximated
- Construction of simple models by reducing the degree of complexity of the original problem (*toy models*)
- Alternatively, a numerical approximation can be performed

The important question after performing any approximation is to check if the results produced match experimental data. Also “wrong” theories can match experimental results (especially if many *tunable* parameters are available. An impressive citation on this topic was given by Enrico Fermi [1]:

I remember my friend Johnny von Neumann used to say: “With four parameters I can fit an elephant and with five I can make him wiggle his trunk.”

1.2.1 The New Trinity of Physics

Nowadays computer simulations are sometimes referred to as *computer experiments*. This can be explained by having a look at the historical relations of experiments and theory (see Fig. 1.3). In the classical access to physics, new physical effects are discovered by an appropriate

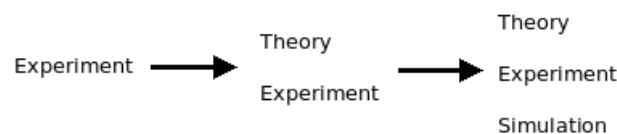


Figure 1.3: Classical relations between experiments, theory and simulations

experiment. Then theory is used to derive and mathematically describe this effect, checking the

accordance with experimental results. In a third step one performs simulations and compare the different results, for example to check if simulation data (which is based on theory) matches experimental results or checking the influence of simplifications made in theory.

Nowadays there is a quite different look at the importance of theory and simulation, referred to

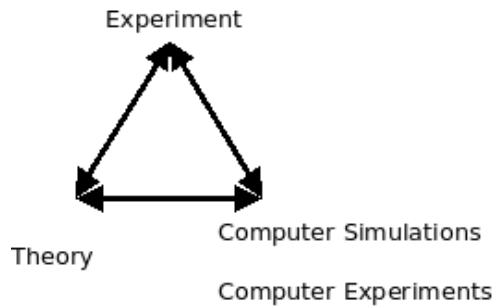


Figure 1.4: The new trinity of physics

as the *new trinity of physics*[2] (Fig. 1.4), where all three parts of physics have equal character and interact with each other:

Theory provides simplified models with exact or approximated solutions.

Simulations can implement these models to compute numerical estimates to the solutions.

So if there is a theory with an exact analytical solution, it can be used to test simulation algorithms, whereas if there is only an approximated solution to theory, simulation algorithms can be used to test the theoretical approximation (like velocity range etc.).

Possible interactions between the new trinity:

- If a theory does not match experimental results,
 - ...the theory may be wrong (i.e. starting from erroneous assumption or model)
 - ... the approximations are wrong
- If a simulation does not match experimental results,
 - ...the simulation may be wrong (i.e. badly performed = *stupid simulator*)
 - ...the model may not be applicable
- Whereas if the simulation does not match theory,
 - ...the simulation is wrong
 - ... the theory is wrong

As discussed above Computer Simulations can provide (quasi-)exact results to test theoretical approaches. By performing such simulations one has a powerful tool to promote better theories and reject the bad ones. In addition one is able to generate ranges of validity for theories, which

individually can depend on built-in parameters of each theory.

But surely each simulation has its limits, from which here we only can name a few. For example there is a limitation by the fact that we can only simulate small system sizes as well as finite time scales.

Of course any simulation can only be as exact as the accuracy of machine it is performed on. But compared to above limitations this effect will be of minor order.

1.2.2 Optimization of simulation performance

1. Optimization by Moore's law

As Gordon Moore, a co-founder of Intel, maintained in 1965 (see 1.1.3) the number F of *FLOPS* (*Floating Point Operations per Second*) a computer can perform will double every 1 to 2 years. As F is assumed to be directly proportional to the number of transistors per unit area that actually doubled every 18 months over about the last 40 years, one can find the following relation (see Fig. 1.2):

$$F \sim 2^{\text{months}/18} \Rightarrow \ln F \sim \text{months} \cdot \frac{\ln 2}{18} \quad (1.1)$$

However, it can be imagined that there are limits that avoid the ongoing of this scaling in the near future. Typical limitations include the effects of miniaturization on nano-scales or limited memory band width,

2. Algorithmic improvement

The speed-up for common large data problems rivals Moore's law. Due to novel developed Simulation Methods (*Algorithms*) one can achieve even a better improvement than by the development of computers

In fact, nowadays often a combination of both methods as modern *multi-core* processors is getting more and more important. So we need algorithms which take most advantage of the hardware.

1.2.3 Optimization of simulation algorithms

The more powerful modern computers are, the greater is the advantage of optimal algorithms.

Example Consider the following two algorithms (N is the input size):

$$\text{ALG1} \quad T_{CPU} \sim N^2$$

$$\text{ALG2} \quad T_{CPU} \sim N$$

In a certain time (neglecting the proportionality factors) using ALG1 we can increase our system size 100 times larger compared to 10,000 times larger by using ALG2 - or in other words: ALG1 takes 100 times longer!

To illustrate the power of optimal algorithms we will analyze the numerical solution of Poisson's equation on a cube of size $N = n^3$ (table 1.1):

$$\nabla^2 u = f \quad (1.2)$$

Year	Method	Storage	Flops
1947	Gauss elimination	n^5	n^7
1950	Optimal Successive Overrelaxation (SOR)	n^3	$n^3 \lg n$
1971	Conjugated Gradient (CG)	n^3	$n^3.5 \lg n$
1984	Full Multigrid	n^3	n^3

Table 1.1: Algorithms for solving Poisson's equation on a cube $N = n^3$

This means that for a cube of size $n = 64$ the optimization of the algorithm will give us a reduction of flops in the order of $n^4 \sim 17 \cdot 10^6$.

1.2.4 Computational approaches for different length scales

It can be seen (Fig. 1.5), that computer simulations cover a broad range of length and time scales. Even in astrophysical problems, computer simulations can be also applied. In the course of this lecture, several methods will be introduced and in detail regarded. Much of the simulation methods are nowadays typically included in modern textbooks. We recommend the reader to consult the following publications as their first reference choice:

- [3] D. Frenkel and B. J. Smit: *Understanding molecular simulation - From algorithms to applications*. Academic Press, 2nd ed. San Diego, USA (2001)
- [2] Michael P. Allen and Dominic Tildesley: *Computer simulation of liquids*. Oxford Science Publications, Oxford, UK (1986)
- [4] Andrew R. Leach: *Molecular modeling - Principles and applications*. World Press, Singapore (2001).

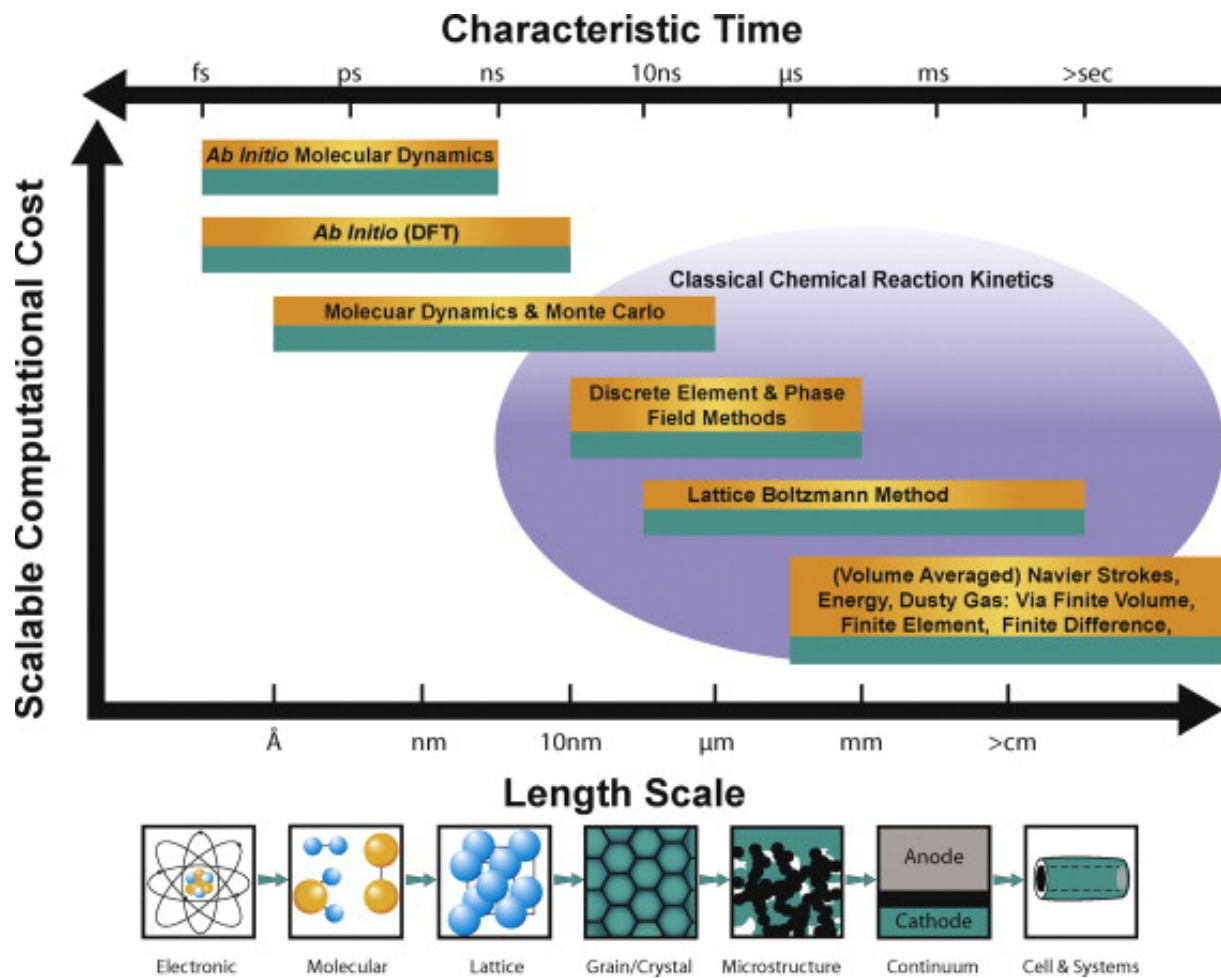


Figure 1.5: Applicability of simulation methods for time and length scales. Taken from [5].

2 Molecular Dynamics integrators

2.1 Principles of Molecular Dynamics

The *Molecular Dynamics Method* is a very general method for modeling systems. It was first introduced by Alder and Wainwright [6] in 1959.

In the simplest case, i.e. **Newton's equations**

$$\vec{F}_i = m\vec{a}_i \quad (2.1)$$

$$\text{or : } \quad \ddot{\vec{x}}_i = - \left(\frac{\partial \vec{x}_i V_i(x)}{m} \right) \quad (2.2)$$

can be used to integrate the *classical equations of motion* that each particle in the system obeys (in case we do have particles).

For example this method is applicable to molecular liquids, noble gases (like argon; already about 30 atoms are sufficient for determining thermodynamic observables), polymers, liquid crystals, granular matter, simulating galaxies in planetary motion, cannonballs, etc.

There are two typical fields of interest:

- Dynamical information: by keeping track of each individual simulation step one can collect information about process flow, aggregation or nucleation rates, calculate diffusion constants or other transport coefficients or simply determine a particles trajectory
- Calculating thermodynamic averages: for a sufficient number of particles, the ergodicity hypothesis is assumed to be fulfilled, i.e. the time average of simulated particles is the same as the ensemble average. This is a very important foundation of thermodynamics and statistical physics and can be read about in textbooks of statistical mechanics.

2.1.1 Ergodicity

With the ergodic hypothesis, it is stated that a time-averaged observable equals an ensemble averaged observable. This hypothesis is very important for connecting Molecular Dynamics to Monte Carlo methods. A timely averaged observable can be expressed by

$$\bar{A}_t = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t dt' A(\Gamma, t) \quad (2.3)$$

where $\Gamma = (q^{3N}, p^{3N})$ denotes a point in a $6N$ -dimensional phase space of particle position and momentum. A statistically averaged observable can be expressed by

$$\langle A \rangle = \frac{\int d\Gamma e^{-\beta \mathcal{H}(\Gamma)} A(\Gamma)}{\int d\Gamma e^{-\beta \mathcal{H}(\Gamma)}} \quad (2.4)$$

where $\mathcal{H}(\Gamma)$ denotes the Hamilton-Function of a system and $\beta = 1/k_B T$ the inverse thermal energy.

Combining both equations gives

$$\bar{A}_t = \langle A \rangle \quad (2.5)$$

which is the ergodic hypothesis. The connection between Molecular Dynamics and Monte Carlo methods is given by the fact that observables in Molecular Dynamics can be interpreted as timely averaged. Monte Carlo methods instead focus on an ensemble average due to the presence of a pseudo dynamic behavior.

However, several remarks on this hypothesis have to be added:

- Strict ergodicity can be only proven for a very small number of simple systems
- It breaks down for some systems like glasses and metastable phases

2.2 Integration of equations of motion

2.2.1 Algorithms

The basic idea in numerical integration of the equations of motion is the discretization of time. Therefore we use finite differences, i.e. we approximate

$$\frac{\partial f(x)}{\partial x} = \frac{f(x + \varepsilon) - f(x)}{\varepsilon} \quad (2.6)$$

This is almost equivalent to a Taylor expansion in an appropriate order of accuracy:

$$x(t + \Delta t) = x(t) + v(t) \cdot \Delta t + \frac{f(x(t))}{m} \cdot \Delta t^2 + \mathcal{O}(\Delta t^3) \quad (2.7)$$

Such an integration scheme where terms of higher order than Δt^2 are truncated is called the *Euler Scheme*. It is one of the simplest methods for numerical integration but also one of the worst [2, 3].

Properties of a **good** algorithm are [3]:

- It should be easy to implement, low memory needs and be fast in execution
- It should allow to use large time steps Δt and to simulate a long time span
- The trajectory should be reproducible and therefore reversible under time reversal (in case of classical motion, depending on implemented algorithm)
- Conservation of total energy, linear and angular momentum should be fulfilled.

2.2.2 Estimation of time step

In principle, the length of the time step determines the numerical accuracy. However, often specific processes like time-resolved collisions between particles are of interest. In one single time step, a maximal penetration depth of about 1/10 of the particle's diameter d should be simulated as a rule of thumb:

$$\Delta t \approx \frac{d}{10} \cdot \frac{1}{v_{max}}. \quad (2.8)$$

Another limitation is given by time-resolved molecular or electronic motion. Vibrations of a covalently bonded hydrogen atom occur on a picosecond time scale. Thus, the minimally range of the time step in an atomistic molecular dynamics simulation is also in the range of picoseconds (10^{-12} s). Due to the very heavy influence of integrators on the acceptable size of the time step and therefore on the precision and success of a simulation, detailed integration schemes have been introduced. We recommend the reader to [2, 3, 4] for a further reading and a more detailed explanation and derivation.

2.2.3 Verlet Algorithm

The *Verlet integrator* [7, 3, 2, 4] offers greater stability than the much simpler Euler method and - even more important - is preserving physical properties of the system. It was first introduced by Loup Verlet in 1967. In general, Verlet-like integrators are called to be *symplectic*. A symplectic integrator includes the preservation of phase space and time-reversibility. Although it was assumed for a long time that the Verlet integrator should be symplectic, a general proof was first derived in the 90's of the last century [3]. It considers the time evolution of the Liouville equation and explicitly takes into account the non-commutability of the Liouville operators by a Trotter expansion. The original idea by Verlet [7] considers a Taylor expansion of the position coordinate in two different directions of time:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\partial^3 x}{3!\partial t^3}\Delta t^3 + O(\Delta t^4) \quad (2.9)$$

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\partial^3 x}{3!\partial t^3}\Delta t^3 + O(\Delta t^4) \quad (2.10)$$

By summing up Eqn. (2.9) and (2.10) we get the Verlet integration step for the position:

$$(2.9) + (2.10): \quad x(t + \Delta t) = 2x(t) - x(t - \Delta t) + \frac{f(t)}{2m}\Delta t^2 + O(\Delta t^4) \quad (2.11)$$

It has to be noticed, that no velocities are needed for the computation of the new positions. However, if a velocity is needed (for example velocity-dependent forces or the determination of the system temperature), it can be derived by

$$(2.9) - (2.10): \quad x(t + \Delta t) - x(t - \Delta t) = 2v(t)\Delta t + O(\Delta t^2) \quad (2.12)$$

$$\Rightarrow v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + O(\Delta t^2) \quad (2.13)$$

The velocities calculated via the Verlet scheme are less accurate than the positions, but for conserving the kinetic energy in terms of the temperature, mainly sufficient.

Advantages: Algorithm is straight forward, needs only modest storage and is time-reversible.

Disadvantages: Reaches only moderate precision and is not self-starting (we need an initialization step for $x(t - \Delta t)$).

2.2.4 Velocity Verlet Algorithm

A related, and more commonly used, algorithm is the Velocity Verlet algorithm first introduced in 1982 [8]. Being mathematically equivalent to the Verlet scheme, the derivation approach is quite similar but explicitly incorporates velocity, solving the first-timestep problem in the basic Verlet algorithm as discussed above:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 \quad (2.14)$$

$$v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t \quad (2.15)$$

It can be shown that the error on the Velocity Verlet is of the same order as the basic Verlet algorithm. Note that the Velocity algorithm is not necessarily more memory consuming, because it's not necessary to keep track of the velocity at every time step during the simulation. The standard implementation scheme of this algorithm is shown in figure 2.1:

1. Start from system's initial setup.
2. Calculate new position

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$

3. Calculate the intermediate velocity

$$v(t + \frac{1}{2}\Delta t) = v(t) + \frac{1}{2}a(t)\Delta t$$

4. Compute new acceleration $a(t + \Delta t)$ (since now we know $r(t + \Delta t)$)
5. Calculate new velocity

$$v(t + \Delta t) = v(t + \frac{1}{2}\Delta t) + \frac{1}{2}a(t + \Delta t)\Delta t$$

In case of velocity dependent forces, the new velocities $v(t + \Delta t)$ are also needed for performing step 3 \rightarrow 4 (dotted line in figure 2.1).

Advantages of the Velocity Verlet form are:

- It is simple and easy to implement.
- The accuracy reaches up to the order of $\mathcal{O}(\Delta t^4)$.

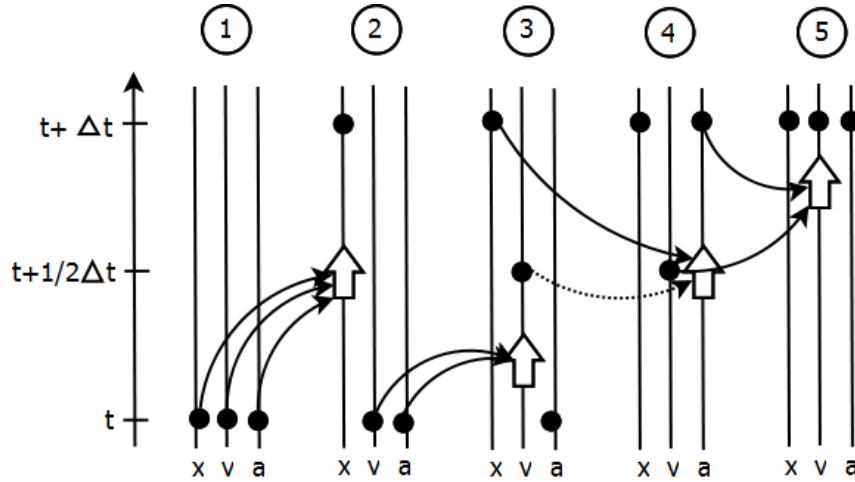


Figure 2.1: Illustration of the Velocity Verlet integration scheme

- The integration is unitary, i.e. phase space volume is preserved.
→ Important for ergodicity $\langle A \rangle_{ensemble} = \langle A \rangle_{time}$
- Time-reversibility is fundamental for conservation laws.
- The mathematical derivation is based on Liouville factorization property (\sim preserves a *shadow Hamiltonian*, a Hamiltonian that describes the influence of energy drift due to integration artifacts in addition to the systems Hamiltonian).

2.2.5 Leapfrog method

The *Leapfrog integration method* [3] is mathematical equivalent to Verlet & Velocity Verlet algorithms. Its name comes from the leapfrog-looking integration scheme as seen in figure 2.2.

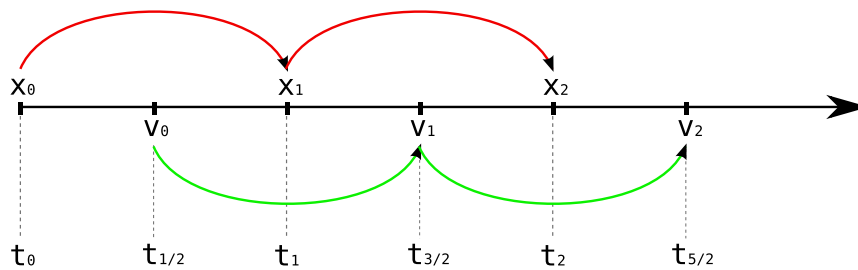


Figure 2.2: Illustration of the Leapfrog integration method

The basic idea is to give up the synchronous modification of position and velocity data. We first calculate the velocity at $t + \frac{1}{2}\Delta t$

$$v\left(t + \frac{1}{2}\Delta t\right) = v(t) + a(t)\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2) \quad (2.16)$$

and then we update the velocities asynchronously, but by whole time steps:

$$x(t + \Delta t) = x(t) + v\left(t + \frac{1}{2}\Delta t\right)\Delta t + \mathcal{O}(\Delta t^2) \quad (2.17)$$

$$v\left(t + \frac{3}{2}\Delta t\right) = v\left(t + \frac{1}{2}\Delta t\right) + \left(\frac{F(t + \Delta t)}{m}\right)\Delta t + \mathcal{O}(\Delta t^2) \quad (2.18)$$

This algorithm has the advantage of being totally symmetric and therefore time-reversible and also avoids the initialization problem.

2.2.6 Runge-Kutta method

The *Runge-Kutta method* [9] named after Carl Runge and Martin Wilhelm Kutta is relevant for integrating equations of the type $\dot{y}_i(t) = g(t, y_1, \dots, y_n)$. In contrast to the *Euler method* [3] one takes a trial step in the middle of each time interval before calculating the value after a full step:

$$y_i(t + \frac{1}{2}\Delta t) = y_i(t) + \frac{1}{2}\Delta t g(t, y_j(t)) \quad j = 1, \dots, N \quad (2.19)$$

$$y_i(t + \Delta t) = y_i(t) + \Delta t g(t + \frac{1}{2}\Delta t, y_j(t + \frac{1}{2}\Delta t)) \quad (2.20)$$

Applying to the Newton equation (2.1) we get:

$$v_1 = v(t) + \frac{1}{2}\Delta t \frac{f(x(t))}{m} \quad (2.21)$$

$$x(t + \Delta t) = x(t) + \Delta t v_1 \quad (2.22)$$

For a further improvement one can use even more trial steps within a time step Δt . The implementation commonly used is the Runge-Kutta integrator of 4th order:

$$\begin{aligned} v_1 &= v(t) + \frac{1}{2}\Delta t \frac{f(x(t))}{m} \\ x_1 &= x(t) + \Delta t v(t) \\ v_2 &= v(t) + \Delta t \frac{f(x(t) + \frac{x_1 - x(t)}{2})}{m} \\ x_2 &= x(t) + \Delta t v_1 \\ v_3 &= v(t) + \Delta t \frac{f(x(t) + \frac{x_2 - x(t)}{2})}{m} \\ x_3 &= x(t) + \Delta t v_2 \\ x_4 &= x(t) + \Delta t v_3 \end{aligned}$$

The new position then is given by:

$$x(t + \Delta t) = x(t) + \frac{x_1}{6} + \frac{x_2}{3} + \frac{x_3}{3} + \frac{x_4}{6} + \mathcal{O}(\Delta t^5) \quad (2.23)$$

2.2.7 Predictor-Corrector methods

Predictor-corrector methods proceed in two steps; first a rough approximation is calculated before second the corrector step refines this approximation. Such an integrator in general is **not** time-reversible but in some systems, e.g. in systems with constraints, it is easier (and faster) to implement. There exist a lot of variations and different implementations of which we only want to discuss a general implementation in 4th order [10]. Here the performed predictor steps are:

$$\begin{aligned} x^p(t + \Delta t) &= x(t) + \Delta t v(t) + \frac{\Delta t^2}{2} a(t) + \frac{\Delta t^3}{6} \ddot{v}(t) + \mathcal{O}(\Delta t^4) \\ v^p(t + \Delta t) &= v(t) + \Delta t a(t) + \frac{\Delta t^2}{2} \ddot{v}(t) \\ a^p(t + \Delta t) &= a(t) + \Delta t \ddot{v}(t) \\ \ddot{v}^p(t + \Delta t) &= \ddot{v}(t) \end{aligned}$$

Now the corrected acceleration can be computed via Newton's law using the positions at time $t + \Delta t$:

$$\dot{v}_i^c(t + \Delta t) = \frac{F(t + \Delta t)}{m_i} = g + \frac{1}{m_i} \sum_j f_{ij} \left(x_i^p(t + \Delta t) - x_j^p(t + \Delta t) \right) \quad (2.24)$$

Again the convention $\dot{y}_i(t) = g(t, y_1, \dots, y_n)$ is used and the last terms are *local* forces acting on the particle i . Usually this yields values different from the $a^p(t)$ calculated previously, so a correction is defined $\delta_i(t + \Delta t) = \dot{v}_i^c(t + \Delta t) - \dot{v}_i^p(t + \Delta t)$ such that the corrections in lowest order can be written as:

$$\begin{aligned} x_i^c(t + \Delta t) &= x_i^p(t + \Delta t) + c_0 \delta_i(t + \Delta t) \\ v_i^c(t + \Delta t) &= v_i^p(t + \Delta t) + c_1 \delta_i(t + \Delta t) \\ \dot{v}_i^c(t + \Delta t) &= \dot{v}_i^p(t + \Delta t) + c_2 \delta_i(t + \Delta t) \\ \ddot{v}_i^c(t + \Delta t) &= \ddot{v}_i^p(t + \Delta t) + c_3 \delta_i(t + \Delta t) \end{aligned}$$

This way an optimally predicted constant c_i is achievable via the *variational principle* ([10]):

$$c_0 = 1/5 \quad c_1 = 5/6 \quad c_2 = 1 \quad c_3 = 1/3$$

2.2.8 Ljapunov-Characteristics

As one can easily imagine there are many other integration schemes available, each having advantages and disadvantages. For a suitable choice of an integrator for a specific problem one has to answer the following question: Is the exact deterministic trajectory of physical importance for the specific observables we are interested in?

For example, if we want to simulate the shot of a satellite into earth's orbit and we want to avoid loosing the satellite, we definitely need to take care of the exact deterministic trajectory. Unlike if we want to generate data only to sample dynamical or statistical information about a system, the exact trajectory has no additional information. In this case the influence of chaos comes into play, i.e. the sensitivity of individual trajectories to small perturbations.

Consider a system of N particles, where $\vec{q}(t)$ gives us the position of one individual particle at a

time t . Then this position-vector is depending on the initial conditions (position and momentum) of all N particles:

$$\vec{r}(t) = f[\vec{q}^N(0), \vec{p}^N(0), t] \quad (2.25)$$

Now imagine a small perturbation by a small amount $\vec{\varepsilon}$ of these initial conditions, for example:

$$\vec{q}'(t) = f[\vec{q}^N(0), \vec{q}^N(0) + \vec{\varepsilon}, t]$$

For a quantitative discussion we regard the change in position $\Delta\vec{r}(t) = \vec{r}'(t) - \vec{r}(t)$ caused by this perturbation. Note that this discussion of stability only regards small changes in the initial conditions. The form of the equations of motion as well as environmental influences are not considered!

For sufficient small values $|\vec{\varepsilon}|$ this change is nearly linear in ε : $|\Delta\vec{r}(t)| \propto |\vec{\varepsilon}|$

However the relation is even more exact approximated by

$$|\Delta\vec{r}(t)| \approx \varepsilon \exp(\lambda t) \quad (2.26)$$

where λ is the *Ljapunow* exponent [11]. So the trajectories will diverge exponentially. The maximum acceptable value for λ for trajectories bound in an interval Δ_{max} during a time-step $0 \leq t \leq t_{max}$ is given by

$$\Delta_{max} \leq |\Delta\vec{r}(t)| \approx \varepsilon \exp(\lambda t) \quad \Rightarrow \quad \varepsilon \approx \Delta_{max} \exp(-\lambda t). \quad (2.27)$$

It can be seen for 2.27, that the acceptable error to keep the trajectories bound, i.e. $\Delta_{max} \leq |\vec{r}(t)|$, decreases exponentially with the simulation time t_{max} .

A possible test is to do two different simulations on a typical system of 1000 particles. At time $t = 0$ we change two velocities by an amount of 10^{-10} and -10^{-10} respectively (in reduced units) and record the total squared displacement $\sum_{i=1}^N |\vec{r}_i(t) - \vec{r}'_i(t)|^2$ of the trajectories (Fig. 2.3). In this example after already 1100 time steps the trajectories have become independent.

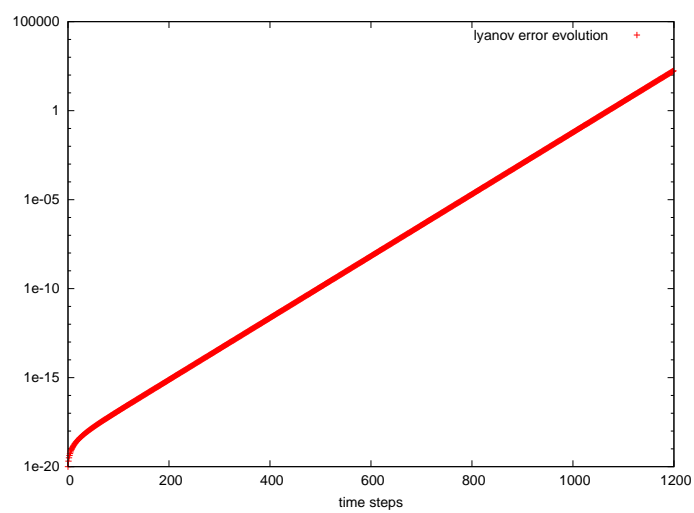


Figure 2.3: Evolution of the Ljapunow error caused by a velocity displacement ($\lambda = 0.02$)

3 Statistical mechanics - Quick & Dirty

The aim of numerical simulations is often the determination of observables like temperature, pressure etc. Compared to real systems, only a small amount of up to (10^6) particles can be simulated which is nothing compared to the number of particles $\mathcal{O}(10^{23})$ present in a mol. The large number allows the presence of the *Law of large numbers* which directly connects the systems properties to statistical averages. Therefore an overwhelming amount of individual microscopic states gives rise to a few macroscopic observables. For the determination of these valuables, a suitable averaging is needed. The topic of this chapter is by far too complex to be explained on a few pages. We therefore recommend the reader to consult basic textbooks like [12, 13, 14] for a further reading.

3.1 Microstates

In quantum mechanics, the properties of a system are discussed in terms of eigenstates of spectators [15], like

$$\hat{H}|i\rangle = E_i|i\rangle \quad (3.1)$$

which determines the eigen energies of a state i by applying the Hamilton operator \hat{H} . For classical systems, states are also characterized by a position q and a momentum p which form the $6N$ dimensional phase space described by $\Gamma = (q^{3N}, p^{3N})$ where N denotes the particle number in the system. Thus, a suitable description of classical dynamics like in quantum mechanics is also given in terms of Hamiltonian Dynamics [14]. The states in classical dynamics are typically called *microstates*.

The basic assumption in statistical mechanics is given by

All microstates are equally probable.

It has to be remarked, that for systems with a large number of degrees of freedom, the same microstate reflecting a macroscopic observable can appear more than once, due to a phenomenon called degeneracy.

3.1.1 Distribution functions

Consider a binary model system of a spin magnet. Spins in these system can only have two states: up (u) or down (d).

Thus a many spin system consists of product states

- 1 spin: either (u) or (d)

- 2 spins: (uu) , (dd) or (du) and (ud) which gives $(u + d)^2$ states
- ...
- N spins: $(u + d)^N$.

Now the magnetization can be either $+m$ (associated with (u)) or $-m$ which is related to (d) . The total magnetization is then given by

$$M = \sum_i^{N_{spins}} m_i \quad (3.2)$$

which gives for a two-spin system the following values: $M = +m, -m, 0$ where the realization for $M = 0$ occurs twice with (ud) and (du) .

For N spins, the set of values for M is realized by

$$M = Nm, (N - 2)m, (N - 4)m, \dots, 0, \dots, -(N - 2)m, -Nm \quad (3.3)$$

which gives 2^N microstates with many degenerated states. It has to be noticed that the largest degeneracy is given for $M = 0$.

In the next step, consider the number of spins N to be even. The number of up-spins (down-spins) is denoted by N_u (N_d). The spin excess s is then defined as $N_u - N_d = 2s$. Therefore one can write

$$(u + d)^N \sim \sum_s \frac{N!}{(1/2N + s)!(1/2N - s)!} \quad (3.4)$$

in terms of the binomial coefficient.

The multiplicity function $\rho(N, s)$ can be also written as

$$\rho(N, s) = \frac{N!}{(1/2N + s)!(1/2N - s)!} = \frac{N!}{N_u!N_d!} \quad (3.5)$$

where for large values of N the central limit theorem applies which gives

$$\rho(N, s) \approx \left(\frac{2}{\pi N}\right)^{1/2} 2^N \exp\left(-\frac{2s^2}{N}\right) \quad (3.6)$$

which is a Gaussian distribution centered around 0 and with variance $\sigma^2 = N/4$. A typical Gaussian distribution function is shown in Fig. 3.1. The the number of states is given by

$$\int_{-\infty}^{+\infty} \rho(N, s) ds = 2^N. \quad (3.7)$$

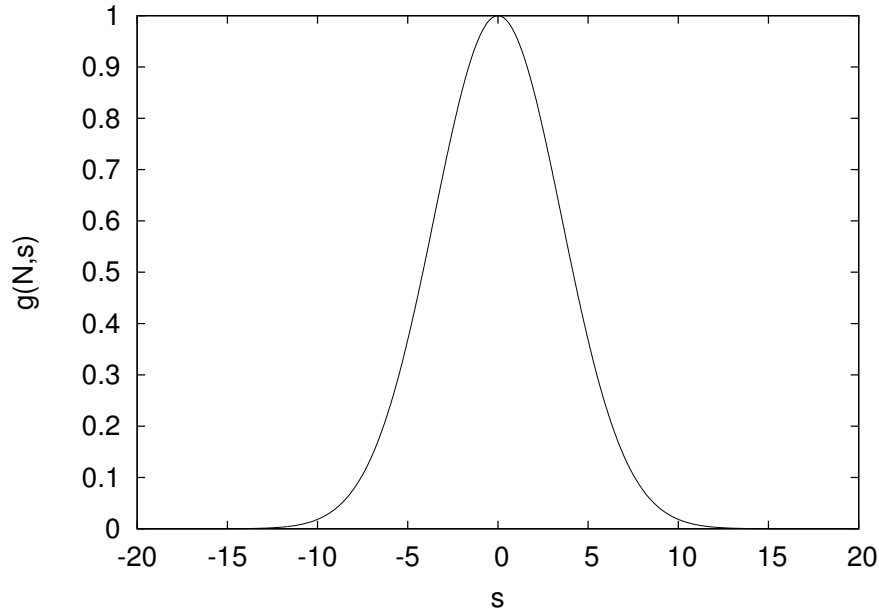


Figure 3.1: A typical Gaussian distribution function centered around 0.

3.1.2 Average values and thermodynamic limit

The average value for an observable f can be calculated by

$$\langle f \rangle = \sum_s f(s)P(s) \quad (3.8)$$

where $P(s)$ is a (normalized) probability distribution with $\sum_s P(s) = 1$. Regarding the example above, the normalized probability distribution is given by $\rho(N, s)/2^N$. If now the *mean square spin excess* is defined by $f(s) = (2s)^2$, the following average value can be determined $\langle (2s)^2 \rangle = N$.

Finally, the fractional fluctuation $2s$ is defined as

$$\mathcal{F} \equiv \frac{\sqrt{\langle (2s)^2 \rangle}}{N} = \frac{1}{\sqrt{N}} \quad (3.9)$$

which gives a value of $1/\sqrt{N}$. Thus, for large N this value tends to zero, resulting in a sharply peaked distribution around 0. The limit of large particle numbers N is the so-called thermodynamic limit where thermodynamic relations are applicable [12, 13, 14]. Finally, the large number of probabilities that occur at that limit result in one macroscopic value.

Since the spin excess can be measured ($\sim M$), a macroscopic magnetization of zero can be observed at nearly every time:

$$N = 10^{20} \rightarrow \mathcal{F} = 10^{-10} \quad (3.10)$$

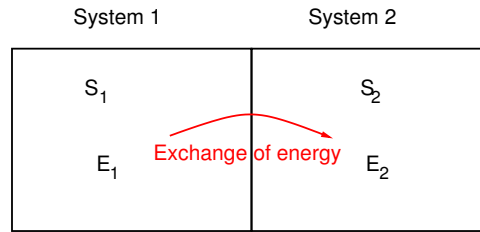
which results in non-detectable fluctuations.

3.2 Definition of Entropy and Temperature

The number of eigenstates (microstates) is denoted by $\Omega(E, V, N)$ with energy E of a system with N particles in a volume V . Regarding the example described in section 3.1.1, this corresponds to $\rho(N, s)$ for the spin system. The basic assumption of statistical mechanics is then given by:

A system with fixed values for N, V, E is equally likely to be found in any of its $\Omega(E)$ eigenstates.

If two systems S_1 and S_2 are coupled together such that they can exchange energy with $E_1 + E_2 = E = \text{const.}$ a situation which is illustrated in Fig. 3.2, then the number of degenerated states of the system is multiplicative with $\Omega_1(E_1) \times \Omega_2(E_2)$. It has to be noted, that there are many



1

Figure 3.2: Two systems with energies E_1 and E_2 are coupled with each other and exchange energy.

ways to distribute the energy under the constraint $E_1 + E_2 = E$.

In order to have a quantitative measure, the natural logarithm of the degeneracy gives

$$\log \Omega(E_1, E - E_1) = \log \Omega_1(E_1) + \log \Omega_2(E - E_1) \quad (3.11)$$

which is additive. Every energy eigenstate in this system is equally likely, but if we look at the distribution, there is one distribution which has the highest probability *i. e.* the largest number of realizations. The maximum of this distribution can be found by

$$\left(\frac{\partial \log \Omega(E_1, E - E_1)}{\partial E_1} \right)_{N, V, E} = 0 \quad (3.12)$$

which is equal to

$$\left(\frac{\partial \log \Omega_1(E_1)}{\partial E_1} \right)_{N_1, V_1} = \left(\frac{\partial \log \Omega_2(E_2)}{\partial E_2} \right)_{N_2, V_2} \quad (3.13)$$

where one can use the short symbol notation $\beta(E_1, V_1, N_1) = \beta(E_2, V_2, N_2)$. The identity ensures that both systems are in equilibrium.

Due to the above mentioned advantages like additivity, the entropy is defined by

$$S(N, V, E) \equiv k_B \log \Omega(N, V, E) \quad (3.14)$$

with the Boltzmann constant $k_B = 1.38 \times 10^{-23} \text{ J/K}$. Thus the following relations in equilibrium hold:

all microstates (eigenstates) are equally likely \rightarrow the multiplicity is maximum in equilibrium \rightarrow the entropy is maximum in equilibrium

The definition of the entropy further allows an absolute thermodynamic definition of the temperature:

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E} \right)_{N, V} \quad (3.15)$$

which can be defined as $\beta = 1/k_B T$.

3.3 Boltzmann distribution

If now two systems with total energy E are connected with each other (Fig. 3.3), the energy of the heat bath in B is $E_B = E - E_i$ if it is assumed that A is in a specific state $E_A = E_i$. The

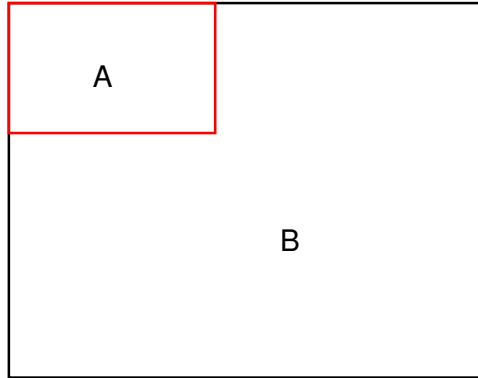


Figure 3.3: System A is connected to system B with total energy E .

degeneracy of the heat bath is then consequently $\Omega_B(E - E_i)$. This determines the probability P_i to find the system A in state i to

$$P_i = \frac{\Omega_B(E - E_i)}{\sum_j \Omega_B(E - E_j)}. \quad (3.16)$$

To compute $\Omega_B(E - E_i)$, the logarithm is expanded by

$$\log \Omega_B(E - E_i) = \log \Omega_B(E) - E_i \frac{\partial \log \Omega_B(E)}{\partial E} + \mathcal{O}(1/E) \quad (3.17)$$

where $\partial_E \log \Omega_B(E) = 1/k_B T$. Thus the expression can be rewritten by

$$\log \Omega_B(E - E_i) = \log \Omega_B(E) - \frac{E_i}{k_B T} + \mathcal{O}(1/E). \quad (3.18)$$

If now Eqn. 3.18 is inserted into Eqn. 3.16, the following expression can be derived

$$P_i = \frac{\Omega_B(E) e^{-E_i/k_B T}}{\sum_j \Omega_B(E) e^{-E_j/k_B T}} = \frac{e^{-E_i/k_B T}}{\sum_j e^{-E_j/k_B T}} \quad (3.19)$$

which is the *Boltzmann distribution*.

3.3.1 Total energy and free energy

By using the Boltzmann distribution, the average energy $\langle E \rangle$ can be easily computed by $\langle E \rangle = \sum_i E_i P_i$. This value can be macroscopically measured and which is in thermodynamics called the inner energy

$$U = \langle E \rangle = \frac{\sum_i E_i e^{-E_i/k_B T}}{\mathcal{Z}} \quad (3.20)$$

where $\mathcal{Z} = \sum_i e^{-E_i/k_B T}$ denotes the partition function.

Thus the total inner energy can be written as

$$U = -\partial_\beta \log \sum_i e^{-E_i/k_B T} = -\partial_\beta \log \mathcal{Z} \quad (3.21)$$

where $\partial_\beta = \partial/\partial\beta$. From thermodynamics, the relation

$$U = \frac{\partial \beta F}{\partial \beta} \quad (3.22)$$

can be remembered, where F denotes the free energy of the system. Taking into account the above relations, the following expressions can be obtained

$$-\frac{\partial \log \mathcal{Z}}{\partial \beta} = \frac{\partial \beta F}{\partial \beta} \quad (3.23)$$

which is identical to

$$\beta F = -\log \mathcal{Z} \quad (3.24)$$

which gives a clear definition of the free energy in terms of the partition function by

$$F = -k_B T \log \mathcal{Z} \quad (3.25)$$

as derived from microscopic states. The free energy is a thermodynamic potential[12, 13, 14] and can be seen as the work horse of equilibrium statistical mechanics. It also connects thermodynamics to microstate analysis.

3.4 Quantum mechanics and classical partition functions

For quantum mechanical systems, the partition function is an operator which acts on the eigenstates and a quantum mechanical average can be written as

$$\langle \hat{A} \rangle = \frac{\sum_i^{\text{all states}} \langle i | e^{-\beta \hat{H}} \hat{A} | i \rangle}{\sum_j \langle j | e^{-\beta \hat{H}} | j \rangle} = \frac{\text{Tr}(e^{-\beta \hat{H}} \hat{A})}{\text{Tr}(e^{-\beta \hat{H}})} \quad (3.26)$$

where $\text{Tr}(\dots)$ denotes a trace operation. For classical mechanics, the trace operation is replaced by an integral over phase space:

$$\text{Tr}(e^{-\beta \hat{H}}) \approx \frac{1}{h^{3N} N!} \int \int \int d\vec{p}^N d\vec{q}^N e^{-\beta \left(\sum_{i=1}^N \frac{p_i^2}{2m_i} + V(\vec{q}^N) \right)} \quad (3.27)$$

where h^{3N} denotes the Planck constant with $N!$ permutations integrated over $6N$ dimensions. The relation to quantum mechanics is given by the Planck constant which also gives the correct dimension of the elementary phase space volume. Eqn. 3.27 finally gives

$$\langle A \rangle = \frac{\int d\vec{p}^N d\vec{q}^N e^{-\beta \left(\sum_{i=1}^N \frac{p_i^2}{2m_i} + V(\vec{q}^N) \right)} \hat{A}(\vec{q}^N, \vec{p}^N)}{\int d\vec{p}^N d\vec{q}^N e^{-\beta \left(\sum_{i=1}^N \frac{p_i^2}{2m_i} + V(\vec{q}^N) \right)}} \quad (3.28)$$

as an ensemble average denoted by $\langle A \rangle$. The above expression can be seen as the starting point for all classical simulations of many-body systems.

3.5 Ensemble averages

An ensemble is a representative realization of a statistical system, compatible with the system parameters. For deriving the Boltzmann factor, (N, V, T) are kept fixed which means that only energy exchanges between two systems are allowed. This is by definition the *canonical ensemble*. Further ensembles:

- microcanonical ensemble (N, V, E)
- grand-canonical ensemble (energy and particle exchange) (μ, p, T)
- the (N, p, T) ensemble

All ensembles are equivalent in the thermodynamic limit $N \rightarrow \infty$. It has to be noticed, that all thermodynamic relations can be derived from derivatives of the fundamental potentials like F, G, H or others.

Bibliography

- [1] Freeman Dyson. A meeting with Enrico Fermi. *Nature*, 427:297, 2004.
- [2] Michael P. Allen and Dominic J. Tildesley. *Computer simulations of liquids*. Oxford Science Publications, Oxford, UK, 1986.
- [3] Daan Frenkel and Berend Smit. *Understanding molecular simulation - From algorithms to applications*. Academic Press, San Diego, USA, 2001.
- [4] Andrew R. Leach. *Molecular modelling: Principles and applications*. World Press, Singapore, 2001.
- [5] Kyle N. Grewa and Wilson K.S. Chiu. A review of modeling and simulation techniques across the length scales for the solid oxide fuel cell. *Journal of Power Sources*, 199:1–13, 2012.
- [6] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. i. general method. *J. Chem. Phys.*, 31:459, 1959.
- [7] Loup Verlet. Computer ”experiments” on classical fluids. i. thermodynamical properties of lennard - jones molecules. *Phys. Rev.*, 159:98–103, 1967.
- [8] K. R. Wilson W. C. Swope, H. C. Andersen. P. H. Berens. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76:648, 1982.
- [9] John C. Butcher. *The Numerical Analysis of Ordinary Differential Equations. Runge-Kutta and General Linear Methods*. Wiley, Chichester, UK, 1987.
- [10] C. W. Gear. Numerical initial value problems in ordinary differential equations. *Englewood Cliffs: Prentice Hall*, 1971.
- [11] Lyapunov A.M. *Stability of motion*. Academic Press, New York and London, 1966.
- [12] L. D. Landau and E. M. Lifshitz. *Statistical Physics*. Pergamon Press, New York, USA, 1969.
- [13] F. Reif. *Fundamentals of statistical and thermal physics*. McGraw-Hill, New York, USA, 1965.
- [14] K. Huang. *Statistical Mechanics*. Wiley, New York, USA, 1987.
- [15] B. Diu iand F. Laloe C. Cohen-Tannoudji. *Quantum Mechanics (Vol. 1)*. Wiley, New York, USA, 1992.