8 Random Numbers

Random numbers (RNs) are *large sets of numbers* with given statistical properties. There is no such thing as a single random number!

Random Numbers are being used in physics, chemistry, biology, mathematics and many other disciplines to simulate stochastic processes. Examples

- Classical stochastic processes: Brownian motion, diffusion, particle transport (in heterogeneous environments), ...
- Stochastic processes in biology: chemotaxis, transport of ions through membranes, binding kinetics of substances to proteins, ...
- Interacting systems of statistical physics,measuring equations of state, determining phase diagrams, characterizing non-equilibrium/relaxation phenomena, ...
- Interacting fields, QCD, lattice gauge theory, cross-sections for elementary particle scattering events, masses of particles (baryons, mesons), ...
- Simulation of quantum mechanical processes, (light) scattering, absorption of ionizing radiation , ...

Sources of random numbers

- Throwing dice
- Tables (Edited, e.g. by RAND Corp.)
- From radioactive decay data (time between 'ticks' of a Geiger counter)

- Computer generated *pseudo random numbers*

It is the last method which is currently the method of choice to generate random numbers in sufficient quantity and quality.

Quality is indeed one of the most important issues for computer generated random numbers, because they are generated by deterministic algorithms; so tests are required to show that they have the right statistical properties (desired distribution, independence (absence of correlations)); in fact computer generated random numbers can at best approximate these desired properties.

There are established tests for quality of random numbers (e.g. Kolmogorov-Smirnov, ..) but we shall not discuss them in this lecture (see good Statistics books on this issue)

8.1 Generation of Homogeneously Distributed RNs

Algorithms for generating RNs are usually starting out from methods to generate *uniformly* distributed RNs. Most programming languages/computer-systems (compilers) include subroutines for that task.

• A note of caution: the collection of system routines for generating RNs with bad statistical properties is quite large and contains famous examples (the most infamous being the `RANDU`-generator of IBM (see below))

8.1.1 System-Provided Generators

The GNU-C compiler offers a pair of routines. The declarations are

- `void srand(unsigned int)` for initialization.
  The integer argument `iseed` specifies which particular se-

quence of random numbers is going to be generated when calling the (second) function `rand()`.

- `int rand(void)` for generating homogeneously distributed random numbers `i` of type integer in the range $0 \leq i \leq 2147483647 = 2^{31} - 1$=`RAND_MAX`..

Through the definition `frand()=rand()/(RAND_MAX + 1.)` (preferably in a `#define` statement) one obtains a random number generator that generates real RNs r in the range $0 \leq r < 1$

8.1.2 Linear Congruential Generators (LCGs)

The system-provided generators quite often produce random numbers of inferior quality. This is related to the fact that they are usually so-called Linear congruential generators (LCGs) which generate RNs, starting from a seed I_0, via

$$I_{j+1} = (aI_j + c) \pmod{m} ,$$

where a, c and m are given parameters of the generator. The period of such generators cannot be larger than m; with 'unfortunate' choices for a and c it can turn out considerably shorter.

The I_j are clearly not random, being generated by a simple deterministic map; this become obvious also by looking at the 'return-map' in which I_{j+1} is plotted vs. I_j. The dynamics is chaotic (for $a > 1$, the Ljapunov-exponent is $\lambda = \ln a$) but clearly not random. Besides being chaotic, it is also *mixing*: an order I_j, I'_j of two RNs is *not* preserved under iteration.

LCGs have sequential correlations. If subsequent triples (I_j, I_{j+1}, I_{j+2}) (more generally k-tuples) are plotted as points in

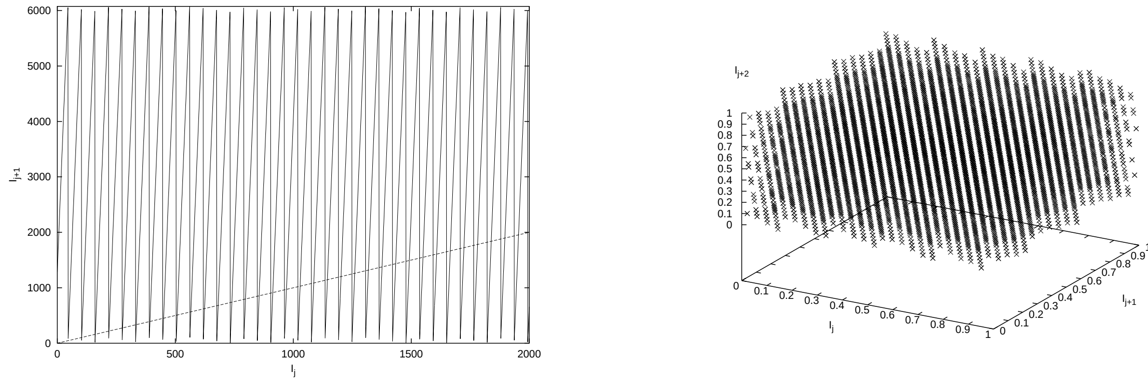

Figure 23: (a) Blow-up of part of the return-map of the generators with $a = 106$, $m = 6075$ and $c = 1283$ from NumRec; the straight line corresponds to $I_{j+1} = I_j$. (b) Triple statistics for that generator.

$\mathbb{R}^k$, they always lie on $k - 1$ dimensional hyper-planes. This was proven by Marsaglia for LCGs in general. The number of different hyper-planes is at most $m^{1/k}$; with an unfortunate choice of the parameters of the random number generator it can be significantly smaller. (Note that for $m = 2^{31} - 1$, even the maximum theoretical number of planes for triples $m^{1/3} = \mathcal{O}(2^{10})$ is not all that impressive, and could be seen with a naked eye when viewed from the right angle! The IBM `RANDU` generator had only 11 planes for triples; the generator was often copied and modified, and in the process not always improved!

Some examples of random number generators, given is (m, a, c, I_0):

- RANDU used by IBM 1970 in FORTRAN: $(2^{31}, 2^{16} + 3 = 65539, 0, 1)$. RANDU was widely used but is “really horrible” (Donald Knuth). It does not have the maximum possible period, among other problems.

- ANSIC used by ANSI C function `rand()` $(2^{31}, 1103515245, 12345, 12345)$. Lower bits of bad statistical quality. Has been superseded by

- DRAND48 used by many Unix implementations `drand48()`

$(2^{48}, 1575931494, 11, 12345)$.

There are many more different linear and non-linear random number generators; look for example in the Web for the Mersenne twister's, based on Mersenne's prime numbers. It is claimed to be of period $2^{19937}-1$, which is a number which exceeds by far the number of particles in the universe...

8.1.3 Portable Generators

Portable generators are written in a higher programming language and mostly of LCG-type or derived from LCG tye generators

- `float ran0(long *idum)`: Park-Miller generator
  The Park-Miller generator produces integer RNs according to a simplified LCG-principle:

  $$I_{j+1} = aI_j \pmod{m} \qquad \text{mit} \quad a = 7^5\,, m = 2^{31}-1 = 2147483647$$

  The difficulty of the implementation is to avoid overflow or other exceptions due to multiplication of large numbers. This is accomplished through the so-called Schrage algorithm, based on the decomposition $m = aq + r$, where $q = [m/a]$ and square brackets denote the integer part of the contents. In NumRec, this generator with $q = 127773$ and $r = 2386$ is implemented as `ran0`, with integer values finally converted to float. Initialization requires a *negative* `idum`. (Rem.: as the generator has no offset, it must exhibit sequential correlations: for small I_j, I_{j+1} will be small as well.) It is not recommended to use `ran0` by itself in serious applications.

- `float ran1(long *idum)`: Park-Miller generator with mixing via a table

Combines the `ran0` generator with a mixing algorithm via a table.

- Initialization upon calling with `idum`< 0. After a warmup-phase a table of length `NTAB` is filled with RNs generated by `ran0`.
- Further calls `ran0(&idum)` use `ran0` to compute a random address in the table, read and return the table entry as RN and refill that table entry with a new RN generated by `ran0`.

The generator has good statistical properties, as long as sequences of RNs are required which are shorter than $m = 2^{31} - 1 = 2147483647 \sim 10^9$.

- `float ran2(long *idum)`: Park-Miller generator with mixing via a table
Function is as in `ran1`, however different generators for computing random table addresses and for filling table entries are used. The period is estimated to be $\mathcal{O}(10^{18})$. The generator has good statistical properties

- Other generators:
  - `float ran3(long *idum)` after D. Knuth. from NumRec. (not based on LCG principle); Initialization by call with neg. `idum`.
  - home-made 'quick and dirty' LCGs; suggestions for parameters `a, b, m` can be found in NumRec. Statistical properties are often mediocre; generators are however simple and can easily be coded *in-line* and thus be made fast; sufficient for simple 'randomization problems'.

8.2 Generation of RNs with Prescribed Probability Density

One is often interested in RNs which are not uniform over an interval but follow a prescribed probability density. Two main methods will be discussed: (i) the transformation-method (a special variant being known as Box-Muller algorithm for generating Gaussian RNs), and (ii) the rejection-method.

8.2.1 Transformation-Method

This method is based on a standard identity for the transformation of probability density functions (PDFs). Let $p(x)$ denote a PDF for the realizations x of a random variable X and let $x = x(y)$ be a (monotoneous) function of y; then

$$p(x)\mathrm{d}x = p(x(y)) \left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| \ \mathrm{d}y \ .$$

This defines the PDF for y via

$$f(y) = p(x(y)) \left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| \ .$$

This identity suggests a method to generate RNs y with prescribed PDF $f(y)$, starting from homogeneously distributed RNs x in [0,1) with $p(x) \equiv 1$. One simply has to require that

$$\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = f(y) \ .$$

Assuming for simplicity $\frac{\mathrm{d}x}{\mathrm{d}y} > 0$, one obtains the ODE

$$\frac{\mathrm{d}x}{\mathrm{d}y} = f(y)$$

and by integration

$$x = F(y) = \int_{y_{\min}}^{y} \mathrm{d}y' f(y') \qquad \Longleftrightarrow \qquad y = F^{-1}(x)$$

That is, F is the integral of f and is the probability distribution (in German also: kumulierte Wahrscheinlichkeitsdichte) of y, with initial condition $F(y_{\min}) = 0$ leading to $F(y_{\max}) = 1$. (The case $\frac{\mathrm{d}x}{\mathrm{d}y} < 0$ is treated analogously, requiring just a change of initial conditions when integrating the ODE.)

RNs with PDF $f(y)$ are thus generated by generating homogeneously distributed RNs x in [0,1), and using these to compute the y's according to $y = F^{-1}(x)$. The geometrical interpretation of this map is indicated in the following figure.

For this method to be sufficient it is necessary to have a simple way to compute the inverse function F^{-1} of the y-distribution.

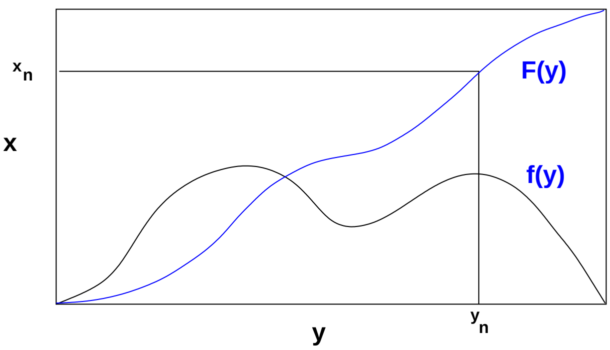

Figure 24: Transformation-method for generating RNs with density $f(y)$.

8.2.2 Example: Exponentially Distributed RNs

To generate RNs with exponential PDF $f(y) = \mathrm{e}^{-y}$, one uses the transformation

$$y = -\ln(1 - x) \qquad \Longleftrightarrow \qquad x = \mathrm{e}^{-y}$$

- Exponentially distributed RNs, NumRec routine: `float expdev(long *idum)`
  The exponential distribution occurs as distribution of waiting times between independent Poissonian random events (occurring randomly at a given rate, such as in radioactive decay).

$f(y) = \mathrm{e}^{-y}$ is the PDF of waiting times y for a process with a rate of 1 event per unit time.

8.2.3 Gaussian RNs – Box-Muller Algorithm

The identity for transforming PDFs given above generalizes to joint PDFs of several random variables. Let $x_1, x_2, \ldots, x_n$ be distributed according to the PDF $p(x_1, x_2, \ldots, x_n)$ and let $x_1 = x_1(y_1, y_2, \ldots, y_n), x_2 = x_2(y_1, y_2, \ldots, y_n), \ldots, x_n = x_n(y_1, y_2, \ldots, y_n)$ be n functions of the n variables $y_1, y_2, \ldots, y_n$. Then

$$f(y_1, y_2, \ldots, y_n) = p(x_1, x_2, \ldots, x_n) \left| \frac{\partial(x_1, x_2, \ldots, x_n)}{\partial(y_1, y_2, \ldots, y_n)} \right|$$

is the PDF of the y_i, where $\frac{\partial(x_1, x_2, \ldots, x_n)}{\partial(y_1, y_2, \ldots, y_n)}$ denotes the Jacobi-determinant of the transformation $\{y_i\} \to \{x_j\}$.
A special case of this identity entails the Box-Muller method for generating Gaussian (or normally) distributed RNs. Starting from pairs of independent uniformly distributed RNs (x_1, x_2) in [0,1), one defines

$$\begin{aligned} y_1 &= \sqrt{-2\ln x_1}\cos(2\pi x_2) \\ y_2 &= \sqrt{-2\ln x_1}\sin(2\pi x_2) \end{aligned}$$

with inverse transformation

$$\begin{aligned} x_1 &= \exp\left\{-\frac{1}{2}(y_1^2 + y_2^2)\right\} \\ x_2 &= \frac{1}{2\pi}\arctan\left(\frac{y_2}{y_1}\right) \end{aligned}$$

The Jacobi-determinant of this transformation is

$$\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} = -\left[\frac{\mathrm{e}^{-y_1^2/2}}{\sqrt{2\pi}}\right]\left[\frac{\mathrm{e}^{-y_2^2/2}}{\sqrt{2\pi}}\right] ,$$

so

$$f(y_1, y_2) = f(y_1)\ f(y_2) \qquad \text{with} \qquad f(y) = \frac{\mathrm{e}^{-y^2/2}}{\sqrt{2\pi}}$$

The y_i so defined are independent and normally distributed with mean $\langle y_i \rangle = 0$ and variance $\langle y_i^2 \rangle - \langle y_i \rangle^2 = 1$.

- Normally distributed RNs in NumRec: `float gasdev(long *idum)`
  Normally distributed RNs frequently appear in science as a consequence of the *Central Limit Theorem*: If $\{x_i\}$ is a collection of independent RNs of mean 0 and variance 1[2], then

  $$y = \frac{1}{N} \sum_{i=1}^{N} x_i$$

  is normally distributed in the limit of large N — *independently* of the nature of the individual distributions of the x_i.

8.2.4 Rejection-Method

The rejection method is a variant of the transformation-method. Unlike the latter it does not, however, require that the inverse F^{-1} of the distribution corresponding to the PDF $f(y)$ is readily obtainable.

Instead, one only requires that a function $\tilde{f}(y)$ majoring $f(y)$, i.e. $\tilde{f}(y) > f(y)$, exists with indeterminate integral (Stammfunktion)

$$\tilde{F}(y) = \int_{y_{\min}}^{y} \mathrm{d}y' \tilde{f}(y')$$

for which $\tilde{F}(y_{\min}) = 0$ — on choosing the integration constant such that $\tilde{F}(y_{\max}) = 1$ — the inverse function $\tilde{F}^{-1}(y)$ be readily computable. The rejection method is then based on the geometrical argument illustrated in the following figure.

[2] It is sufficient that the variances of the x_i-distributions all be finite so that variables can be rescaled.

1. Generate RNs x homogeneously distributed in $[0, 1)$, i.e. with $p(x) = 1$. For $x = \tilde{F}(y) \Leftrightarrow y = \tilde{F}^{-1}(x)$ then, y is distributed according to the PDF $\tilde{f}(y)$.
2. For each x (and thus for each y) generated this way, one generates a second RN x' homogeneously distributed in the interval $[0, 1)$. One accepts y as random number if $x' < f(y)/\tilde{f}(y)$, that is, with probability $f(y)/\tilde{f}(y)$; otherwise y is rejected and a new x is sampled. The y's that are finally accepted are thus distributed according to $f(y)$

For reasons of efficiency one should make sure that the majorizing function $\tilde{f}(y)$ is always rather close to $f(y)$; otherwise too many attempts get rejected.

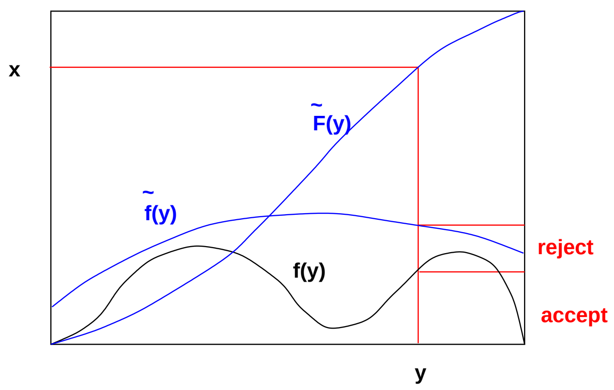

Figure 25: Rejection-method for generating RNs with density $f(y)$.

For generating RNs with unimodal PDF (one maximum) one may choose a non-normalized Lorentz-curve as

$$\tilde{f}(y) = \frac{c_0}{1 + (y - y_0)^2/a_0^2}$$

as majorizing function; its peak height at y_0 is c_0, its full width at half maximum (FWHM) is $2a_0$. The indeterminate integral is

$$\tilde{F}(y) = a_0 c_0 \arctan((y - y_0)/a_0) + c$$

and is easily inverted. The task is to choose the parameters a_0, c_0 and y_0 such that $f(y)$ is majorized as tightly as possible. If the y-domain is finite, y is determined from

$$x = \tilde{F}(y) - \tilde{F}(y_{\min})$$

which on inversion gives

$$y = y_0 + a_0 \tan\left(\frac{x}{a_0 c_0} + \arctan(\frac{y_{\min} - y_0}{a_0})\right) .$$

The variable x must be chosen homogeneously distributed in $[0, 1)$. The rejection method (with majorizing functions of this type) is used to generate RNs with a variety of different PDFs, among them

- Binomially distributed RNs, NumRec: `float bnldev(float q, int n, long *idum)`,
  defined on the integers $\{0, 1, \ldots, n\}$

  $$p_{q,n,j} = \binom{n}{j} q^j (1-q)^{n-j}$$

  gives the probability of j successes in n trials for probability q of success in a single trial.

- Poissonian RNs, NumRec: `float poidev(float x, long *idum)`,
  defined on the positive integers,

  $$p_{x,j} = \frac{x^j \mathrm{e}^{-x}}{j!}$$

  The parameter x defines the mean: $\langle j \rangle = x$. The Poisson distribution gives the probability for the occurrence of j events per unit interval for a Poisson process with rate x (alternatively in an interval x for a rate-1 process), and formally as the limit $q = \frac{x}{n}$ and $n \to \infty$ of the Binomial distribution.

- Gamma distributed RNs, NumRec: `float gamdev(int a, long *idum)`
  defined on the positive real numbers.

  $$p_a(x) = \frac{x^{a-1}\mathrm{e}^{-x}}{\Gamma(a)}$$

  The parameter a defines the mean: $\langle x \rangle = a$. The Gamma distribution (with integer parameter a) describes the probability that the waiting time up to the a-th event in a rate-1 Posisson-process is x.

9 Monte Carlo Simulation

Monte Carlo simulation is a method to simulate stochastic processses or the stochastic composition of systems in chemistry biology, physics and other disciplines with the help of random numbers Random numbers are being used

- to directly simulate (imitate) *stochasticity of processes*. Brownian motion, diffusion, dendritic growth (diffusion limited aggregation), dynamics of stock-prices, of neural nets, scattering of light, ...
- to generate a heterogeneous medium, in which processes are then studied which may or may not themselves be stochastic, ground-water aquifers (transport of pollutants therein), alloys (electrical transport, dynamical properties, magnetism), population dynamics (spread of epidemics, fire in woods), percolation-problems, ...
- to compute expectations of functions or random variables Monte Carlo integration (the result aimed for is non-random), statistical mechanics,...

9.1 Monte Carlo Integration: Evaluation of Integrals and Expectations

The Monte Carlo method may be used as a tool for integrating functions, or for evaluating averages of functions of random variables.

Let $p(\mathbf{x})$ denote a PDF for random-vectors $\mathbf{x}$ in a volume V. The expectation of the Function f over p is defined as

$$\langle f \rangle = \int_V \mathrm{d}\mathbf{x}\ p(\mathbf{x})\ f(\mathbf{x})\ .$$

By the law of large numbers, the expectation $\langle f \rangle$ is approximated by a sample mean: Drawing N points $\{\mathbf{x}_i\}$ at random according to the PDF $p(\mathbf{x})$ one obtains the sample mean and second (uncentered) moment as

$$\overline{f_N} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \qquad , \qquad \overline{f_N^2} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)^2 \; .$$

The empirical mean is denoted as an estimator of the expectation value. The sample variance (or second centered moment) is

$$s_N^2 = \frac{1}{N}\sum_{i=1}^{N} \left(f(\mathbf{x}_i) - \overline{f_N}\right)^2 = \overline{f_N^2} - \overline{f_N}^2$$

The "true" variance of our distribution is $s^2 = \langle f^2 \rangle - \langle f \rangle^2$; According to statistics the sample variance converges to the "true" variance as

$$\langle s_N^2 \rangle = \frac{N-1}{N} s^2$$

If we define

$$\sigma_N(f) = \sqrt{\frac{\overline{f_N^2} - \overline{f_N}^2}{N-1}} \; .$$

as an estimator of the error of our sample, it follows

$$\left\langle \sigma_N^2(f) \right\rangle = \frac{1}{N-1}[\overline{f_N^2} - \overline{f_N}^2] \approx \frac{1}{N}[\left\langle f^2 \right\rangle - \langle f \rangle^2]$$

Note that for *finite* N, both the empirical mean $\overline{f_N}$ and the empirical error $\sigma_N(f)$ are random-variables themselves. We also say that σ_N is an unbiased estimator of the error of our sample, i.e.

$$\langle f \rangle = \int_V \mathrm{d}\mathbf{x} \; p(\mathbf{x}) \; f(\mathbf{x}) = \overline{f_N} \pm \sigma_N(f)$$

Rem.:

- The empirical error decreases only like $N^{-1/2}$ with the sample size N. That is MC is worse than all other integrators discussed so far!

- On the positive side, note that the method is robust and simple to use.

- A special case is obtained for homogeneously distributed $\mathbf{x}$, $p(\mathbf{x}) = V^{-1}$. This gives rise to the estimate

$$\int_V \mathrm{d}\mathbf{x}\ f(\mathbf{x}) = V\ \left(\overline{f_N} \pm \sigma_N(f)\right)$$

  for ordinary integrals.

9.1.1 Metropolis Algorithm

The method for integrating functions which we just described is very inefficient if significant contributions to the integral come only from a small subset of the volume V, where $p(\mathbf{x})$ differs significantly from zero — a region which may, moreover, be of a shape which is not easily characterized. A MC integrator based on homogeneously distributed random vectors will be useless, as it keeps adding terms to the integral which are insignificantly small (this phenomenon occurs quite frequently in high-dimensional integration)

In such a situation one uses the idea of *Importance Sampling*, which we shall encounter again below when discussing problems of statistical mechanics.

The basic idea is simple: Suppose we want to evaluate an integral of the form

$$\langle f \rangle = \int_V \mathrm{d}\mathbf{x}\ p(\mathbf{x})\ f(\mathbf{x})$$

for a PDF which is significantly different from zero only in certain parts of V, for which, moreover a simple analytical characterization is not available. In such a situation, neither a transformation-method nor a rejection-method could be efficiently constructed to generate random numbers following the PDF $p(\mathbf{x})$ in question. In such a situation one constructs a *stochastic process* which has $p(\mathbf{x})$ as its *equilibrium distribution.*

The stochastic process will be constructed as a so-called Markov-process, i.e. a process without memory that is completely characterized by a matrix of $W(\mathbf{x}, \mathbf{x}')$ of transition rates between states (for transitions $\mathbf{x}' \to \mathbf{x}$)

In terms of the transition rates, a master equation of the following form

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \sum_{\mathbf{x}'} [W(\mathbf{x}, \mathbf{x}')p(\mathbf{x}', t) - W(\mathbf{x}', \mathbf{x})p(\mathbf{x}, t)]$$

can be formulated. It is a continuity-equation for probability densities: The change of the probability density at $\mathbf{x}$ is the net result of inflowing and outflowing probability currents. (In the continuum case, the $\mathbf{x}'$–sum in the above equation is to be read as an integral.)

In equilibrium the right hand side of the master equation vanishes. One now constructs matrix of transition rates in such a way that the probability density $p(\mathbf{x})$ is the equilibrium density, satisfying in particular the special condition of *detailed balance*: in that case the right hand side of the master equation vanishes term by term, so that for all $\mathbf{x}$ and $\mathbf{x}'$ the condition

$$W(\mathbf{x}, \mathbf{x}')p(\mathbf{x}') = W(\mathbf{x}', \mathbf{x})p(\mathbf{x})$$

holds.

A canonical proposal for the generation of appropriate matrices of transition rates is due to *Metropolis, Rosenbluth, Rosenbluth, Teller and Teller*. It constructs W as product of a proposal-probability and an acceptance-rate as follows

$$W(\mathbf{x}, \mathbf{x}') = \gamma\Theta(\delta - |\mathbf{x} - \mathbf{x}'|)\ \min\left\{1, \frac{p(\mathbf{x})}{p(\mathbf{x}')}\right\}$$

$$W(\mathbf{x}', \mathbf{x}) = \gamma\Theta(\delta - |\mathbf{x} - \mathbf{x}'|) \ \min\left\{1, \frac{p(\mathbf{x}')}{p(\mathbf{x})}\right\}$$

The proposal-probability $\gamma\Theta(\delta - |\mathbf{x} - \mathbf{x}'|)$ for moves in either direction restricts allowed moves to stay below a certain maximum distance, but have an otherwise uniform probability to be proposed. The factor γ can be absorbed in the time scale. One checks by explicit calculation that the Metropolis et al. proposal does satisfy a detailed balance condition with respect to the probability density $p(\mathbf{x})$.

• The folllowing fact is crucial: as long as W is constructed such that every state can be reached from every other state via finitely many transitions, the equilibrium density is *unique*. Thus, the Metropolis-algorithm converges to an equilibrium characterized by $p(\mathbf{x})$. Thus, the desired integral can – after a suitable equilibration phase – be evaluated via

$$\langle f \rangle = \int_V \mathrm{d}\mathbf{x}\ p(\mathbf{x})\ f(\mathbf{x}) \simeq \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)$$

with $\mathbf{x}_i$ generated via the Metropolis-algorithm.

9.1.2 Discrete Systems as Example

A typical example are spin systems as a models of magnetisation phenomena. The simplest model is the Ising model, defined by the energy function

$$H(\mathbf{S}) = -J \sum_{(i,j)} S_i S_j - h \sum_i S_i \ .$$

The S_i can take the values ± 1. The coupling J is a so-called exchange coupling, h represents an external magnetic field. If $J > 0$, ferromagnetic order is preferred, if $J < 0$ it is anti-ferromagnetic order. By (i, j) we denote nearest neighbour pairs on a regular lattice.

The Ising model can be solved analytically in $d = 1$ by means of the transfer matrix method (i.e. free energy and expectations of observables can be evaluated in closed form). In $d = 2$, a solution has so far been obtained only without field ($h = 0$). In $d \geq 3$, no analytic solution is known, but approximate results have been obtained in various ways.

To check the quality of approximations, one simulates the system using Monte Carlo methods.

For N spins one has 2^N possible spin configurations of the system. The numerical evaluation of the partition sum is therefore plagued by very much the same problems as the general continuum case discussed earlier. Randomly chosen spin configurations with $S_i = \pm 1$ having equal a-priori probability have an energy

$$H(\mathbf{S}) = \mathcal{O}(1/\sqrt{N})$$

due to the central limit theorem (on a cubic lattice with N vertices in d dimensions one has dN pairs (i, j), and $S_i S_j = \pm 1$ with equal probability.) Thermodynamically relevant states, however, have energies $H(\mathbf{S}) = -\mathcal{O}(N)$; as in the continuum case there is an energy difference $\mathcal{O}(N)$ between thermodynamic and random states, (the latter are thermodynamically relevant in the high temperature limit, $T \to \infty$).

Using the Metropolis-Algorithm one constructs a stochastic process, which converges to equilibrium with spin-configurations $\mathbf{S}$ distributed according to the Gibbs-Boltzmann distribution

$$p(\mathbf{S}) = \frac{\mathrm{e}^{-\beta H(\mathbf{S})}}{\sum_{\mathbf{S}} \mathrm{e}^{-\beta H(\mathbf{S})}} \ .$$

Quantities of Interest

- Partition sum

$$Z_N = \sum_{\mathbf{S}} \mathrm{e}^{-\beta H}$$

- Free energy (per spin)

$$f_N = -(\beta N)^{-1} \ln Z_N$$

- Internal energy (per spin)

$$u = \frac{\mathrm{d}}{\mathrm{d}\beta} \left.(\beta f)\right|_h = \frac{1}{N} \langle H(\mathbf{S}) \rangle$$

Here $\langle (\ldots) \rangle$ denotes the average over the Gibbs-Boltzmann distribution

- Magnetization (per spin)

$$m = -\frac{\mathrm{d}}{\mathrm{d}h} \left. f_N \right|_\beta = \frac{1}{N} \sum_i \langle S_i \rangle$$

- Specific heat

$$c = \frac{\mathrm{d}}{\mathrm{d}T} \left. u \right|_h = -k_{\mathrm{B}} \beta^2 \frac{\mathrm{d}}{\mathrm{d}\beta} \left. u \right|_h = \frac{k_{\mathrm{B}} \beta^2}{N} \left[\left\langle H(\mathbf{S})^2 \right\rangle - \langle H(\mathbf{S}) \rangle^2 \right] \geq 0$$

The specific heat is related to fluctuations in energy and so must be positive.

- Susceptibility

$$\chi = \frac{\mathrm{d}}{\mathrm{d}h} \left. m \right|_\beta = \frac{\beta}{N} \sum_{i,j} \left[\langle S_i S_j \rangle - \langle S_i \rangle \langle S_j \rangle \right] = \frac{\beta}{N} \left\langle \left[\sum_i (S_i - \langle S_i \rangle) \right]^2 \right\rangle \geq 0$$

The susceptibility is related to fluctuations in magnetization and so must be positive as well.

- Correlation functions: On a microscopic level one looks at correlations between spins. The correlation function

$$G(\mathbf{r}) = \langle S_{\mathbf{i}} S_{\mathbf{i}+\mathbf{r}} \rangle - \langle S_{\mathbf{i}} \rangle \langle S_{\mathbf{i}+\mathbf{r}} \rangle \sim \frac{\mathrm{e}^{-r/\xi}}{r^{d-2+\eta}} \qquad r \gg 1$$

is characterized by two quantities, the correlation length ξ, which determines the scale on which correlations decay (exponentially), and an exponent η quantifying power law corrections (see below). One notes that the susceptibility is basically a sum of correlation functions.

Evaluation of expectations within a Monte Carlo simulation using the Metropolis algorithm.

$$W(\mathbf{S}', \mathbf{S}) = \Theta(1 - |\mathbf{S} - \mathbf{S}'|) \ \min\left\{1, \mathrm{e}^{-\beta \Delta H}\right\}$$

with

$$\Delta H = H(\mathbf{S}') - H(\mathbf{S})$$

Properties of the system

- In $d \geq 2$ and in the thermodynamic limit $N \to \infty$, the system (at $h = 0$) exhibits a phase transition from a disordered phase with $m = 0$ into an orderd phase with non-zereo spontaneous magnetisation $m = \pm m_0(T)$. On the square lattice the critical point is at

$$\sinh(2\beta_c J) = 1 \Leftrightarrow \beta_c J = 0.4406868\ldots \Leftrightarrow \frac{T_c}{J} = 2.269185\ldots$$

- In the vicinity of the critical point $(T_c, h = 0)$ various thermodynamic functions exhibit algebraic singularities, which can be characterized by critical exponents. With $t = (T - T_c)/T_c$ we have (for $|t| \ll 1$)

– Specific heat

$$c \sim \ln|t|$$

a behaviour which is often characterized by writing $c \sim |t|^{-\alpha}$ with $\alpha = 0(\ln)$.

– Magnetization

$$m \sim (-t)^{\beta} \qquad \text{with} \qquad \beta = \frac{1}{8}$$

– Susceptibility

$$\chi \sim |t|^{-\gamma} \qquad \text{with} \qquad \gamma = \frac{7}{4}$$

– Magnetization at T_c

$$m(T_c, h) \sim \text{sgn }(h)|h|^{1/\delta} \qquad \text{with} \qquad \delta = 15$$

– Correlation length

$$\xi \sim |t|^{-\nu} \qquad \text{with} \qquad \nu = 1$$

– Critical correlation function: At criticality the correlation length is infinite, and the correlation function exhibits slow algebraic decay of the form

$$G(r)|_{T_c} \sim \frac{1}{r^{d-2+\eta}} \qquad \text{with} \qquad \eta = \frac{1}{4}$$

Remarkable properties of critical exponents

Critical exponents have two remarkable properties :

1. They are *universal* in the sense that they depend only on very few details of the system under study: (i) the spatial dimension, (ii) the dimension of the 'order parameter' — the quantitiy that characterizes the ordered phase; here the magnetization,

which in the case at hand is a scalar, so one-dimensional. Magnets exist, in which the magnetization can point anywhere in a plane oder into any spatial direction; the order parameter dimensions would be 2 resp. 3 in these cases, (iii) whether the interaction has long or short range (criterium for short range is $\Sigma_j |J_{ij}| < \infty$). All other aspects are irrelevant (e.g. . lattice type).

2. The critical exponents are not all independent, rather for continuous phase transitions as described above they (almost) always satisfy the following 4 scaling relations

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2 \qquad \text{Rushbrooke} \\ \beta(\delta - 1) &= \gamma \qquad \text{Widom} \\ \nu(2 - \eta) &= \gamma \qquad \text{Fisher} \\ 2 - d\nu &= \alpha \qquad \text{Josephson} \end{aligned}$$

The generally accepted theoretical understanding of these properties has been provided through the renormalization goup theory of critical phenomena (K. Wilson, 1970, Nobelpreis 19xx).

9.1.3 Aspects of the Simulation

(i) Simulation with cyclic boundray conditions to reduce surface effecs. In $d = 2$ this means that the lattice is wrapped on a torus.

(ii) Realization in C: Spins of a $L \times L$ lattice stored in an array `int S[L-1][L-1]`. To compute corrdinates of nearest neigbours, C offers the modulo-operation: neighbouring spins of `S[i][j]` are `S[i±1 % L][j]` and `S[i][j±1 % L]`.

(iii) Elementary step: spin reversion of a single spin: $S_{\mathbf{i}} \to S'_{\mathbf{i}} = -S_{\mathbf{i}}$. Energy change for Metropolis decision

$$\Delta H = H(\mathbf{S}') - H(\mathbf{S}) = 2S_{\mathbf{i}} \left[J \sum_{\mathbf{j} \in n(\mathbf{i})} S_{\mathbf{j}} + h \right]$$

Here $n(\mathbf{i})$ denotes the set of nearest neighbours of $\mathbf{i}$. In $d = 2$ the *positive* energy changes (für $J > 0$) can only be $\Delta H = 2[2J + h]$ and $\Delta H = 2[4J + h]$. Instead of computing $p_\Delta = \exp\{-\beta \Delta H\}$ for the acceptance criterion for $\Delta H > 0$ each time anew, one stores the two possible values and uses ΔH just as an index pointing to the right probability.

(iv) In the vicinity of T_c and for $T > T_c$ one rather computes the absolute value of the magnetization $\langle | \sum_i S_i | \rangle$ – this allows to use the statistics of configuration, even if during a simulation the system has switched between states of negative and positive magnetization (the times it takes for such transition are often negligibly short.

(v) The evaluation of thermodynamic functions in the vicinity of T_c requires a finite-size scaling analysis. E. g. for the susceptibility one has ($h = 0$)

$$\chi_L(T) = \left| \frac{T - T_c}{T_c} \right|^{-\gamma} \mathcal{F} \left(L^{1/\nu} \frac{T - T_c}{T_c} \right)$$

Plotting $\chi_L(T) \left| \frac{T-T_c}{T_c} \right|^{\gamma}$ vs. $L^{1/\nu} \frac{T-T_c}{T_c}$ will for properly chosen values of T_c, γ and ν result in curves which will fall on top of each other for different system sizes. An estimate of T_c can be obtained from the 'pseudo-critical' temperature $T_{\max}(L)$, for which at given L a maximum of $\chi_L(T)$ is observed. FSS holds that

$$T_{\max}(L) = T_c + aL^{-1/\nu} \qquad L \gg 1 \ .$$

A plot of $T_{\max}(L)$ vs. $1/L^a$ should for large L produce a straigth line, if $a = 1/\nu$ is properly chosen; this allows to locate T_c and to determine ν.

9.1.4 Estimation of Errors – The Role of Dynamics and critical Slowing Down

The Metropolis–Algorithm generates spin configurations which are not independent within a time step of one Monte Carlo step per spin (MCS) In fact, one observes temporal correlations of the type

$$C(t) = \frac{1}{N}\sum_i [\langle S_i(t_0)S_i(t_0+t)\rangle - \langle S_i(t_0)\rangle \langle S_i(t_0+t)\rangle \sim \mathrm{e}^{-t/\tau}$$

(times being measured in units of 1 MCS gemessen). Here τ is called relaxation time (of the spin-correlation function). This quantity is important in judging the true statistical error of the Monte Carlo evaluation of expectations. We have

$$\langle f(\mathbf{S})\rangle = \frac{1}{K}\sum_{k=1}^{K} f(\mathbf{s}^{(k)} \pm \mathcal{O}\left(\sqrt{\frac{\overline{f^2} - \overline{f}^2}{K_{\text{eff}}}}\right)$$

with $\overline{f^n} = \frac{1}{K}\sum_{k=1}^{K} f^n(\mathbf{s}^{(k)}$, where K_{eff} is the number of effectively independent configurations in the sample of K measurements. It is reasonably estimated to be $K_{\text{eff}} = K/\tau$.

A problem here is that the relaxation time τ itself diverges at the critical point T_c like

$$\tau \sim \left|\frac{T - T_c}{T_c}\right|^{-z\nu}$$

where $z \simeq 2.1$ for the Ising model in $d = 2$. This phenomenon is known as critical slow-down (kritische Verlangsamung) of the dynamics. As a consequence simulations near T_c must be carried out for ever longer times, in order to obtain results with reasonably small errors.

9.1.5 Random Walk

Random motion of particles on a hypercubic d-dimensional lattice: In each time-step Δt the walker takes a step to a neighbouring site at a distance Δx in of the $2d$ possible directions with probability $p = \frac{1}{2d}$. For an ensemble of random walkers let $u(\mathbf{x}, t)$ denote the probability to be on the site $\mathbf{x}$ at time t; the follwing master equation holds for the ensemble of walkers

$$u(\mathbf{x}, t + \Delta t) - u(\mathbf{x}, t) = \frac{1}{2d} \sum_{i=1}^{2d} \left[u(\mathbf{x} + \Delta \mathbf{x}_i, t) - u(\mathbf{x}, t)\right]$$

Dividing by Δt and assuming the scaling $D\Delta t = \frac{1}{2d}|\Delta \mathbf{x}_i|^2$ resp. $|\Delta \mathbf{x}_i| = \sqrt{2dD\Delta t}$, one obtains a diffusion equation in the limit $\Delta t \to 0$:

$$\partial_t u(\mathbf{x}, t) = D\Delta u(\mathbf{x}, t)$$

Its solution for initial condition $(\mathbf{x}, t_0) = \delta(\mathbf{x} - \mathbf{x}_0)$ and boundary condition $u \to 0$ for $|\mathbf{x}| \to \infty$ is a broadening Gaussian probability density function,

$$u(\mathbf{x}, t) = \frac{1}{(4\pi D(t - t_0)^{d/2}} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_0|^2}{4D(t - t_0)}\right) .$$

The mean square displacement from the starting point follows the diffusion law

$$\left\langle (\mathbf{x} - \mathbf{x}_0)^2 \right\rangle = 2dD(t - t_0)$$

Simulating on a lattice with finite lattice constant, one will see diffusive behaviour only at large length and time scales. Only in that limit will the probability density function be *isotropic*, despite the underlying hypercubic lattice.

The diffusion scaling may alternatively understood by considering the position of the random walker after N elementary steps taken as a sum of independent increments $\Delta\mathbf{x}_i$, where $\Delta\mathbf{x}_i = \pm\Delta x\mathbf{e}_\mu$ $\mu = 1, \ldots, d$ with equal probability $(2d)^{-1}$, or for a single component of the increment, $\Delta x_{i\mu} = 0$ with prob. $1 - 1/2d$ and $\pm\Delta x$ with prob. $1/2d$.

Then the (random) location after N steps is

$$\mathbf{x}(N\Delta t) = \sum_{i=1}^{N} \Delta\mathbf{x}_i$$

As the distribution of increments does not single out a direction, one has

$$\langle x_\mu(N)\rangle = 0\ ;$$

because of to the independence of increments for $i \neq j$ then

$$\left\langle x_\mu^2(N\Delta t)\right\rangle = \sum_{i,j=1}^{N} \langle \Delta x_{i\mu}\Delta x_{j\mu}\rangle = \sum_{i=1}^{N} \left\langle \Delta x_{i\mu}^2 \right\rangle = \frac{N}{d}\Delta x^2 = \frac{N\Delta t}{d}\frac{\Delta x^2}{\Delta t}$$

A continuum description is obtained by taking increments and time steps as infinitesimal, so that $N\Delta t = t - t_0$. To obtain a finite variance for finite makroskopic time difference reqires $\frac{\Delta x^2}{\Delta t} = \mathcal{O}(1)$ für $\Delta t \sim N^{-1} \to 0$.

Yet another way to rationalize the scaling comes from the central limit theorem, which states that a sum of N independent increments, if scaled by $1/\sqrt{N}$

$$\mathbf{S}(N\Delta t) = \frac{1}{\sqrt{N}}\mathbf{x}(N\Delta t) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \Delta\mathbf{x}_i$$

converges to a Gaussian distributed random variable in the limit of large N.

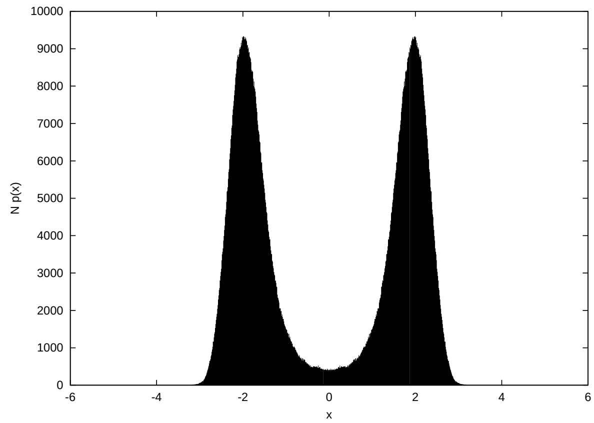

Figure 26: Determining $p(x) = C\exp(-\beta(x - x_0)^2)$ for $\beta = 0.2$ and $x_0 = 2.0$ using a Metropolis algorithm; shown is a symmetrized result, as transitions left/right are rare, to suppress fluctuations of the total weights of left vs. right peak for the simulation of the given size.