Chapter 5

Random Numbers

Contents

In this chapter we introduce numerical methods suited to model stochastic systems. Starting point is the Fokker-Planck equation (2.148) within the frame-work of Ito calculus

$$\partial_t\, p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0) \;=\; -\sum_i\, \partial_i\, A_i\; p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0) \;+\; \frac{1}{2}\,\sum_{i,j}\,\partial_i\partial_j\,[\mathbf{B}\cdot\mathbf{B}^{\mathsf{T}}]_{\mathbf{ij}}\;\mathbf{p}(\boldsymbol{x},\mathbf{t}|\boldsymbol{x_0},\mathbf{t_0})\;. \tag{5.1}$$

which proved useful in analytical descriptions of Brownian dynamics. There exist two approaches to solve this equation by numerical means.
On the one hand one can treat the Fokker-Planck equation as an ordinary parabolic partial differential equation and apply numerical tools like finite differencing, spectral or variational methods that are applicable to partial differential equations in general. These methods will be described in a yet unwritten chapter of these notes.
On the other hand one can resort to the stochastic processes that underlie the Fokker-Planck equation. We have shown in chapter 2.5 that the Fokker-Planck equation (2.148, 5.1) corresponds to a stochastic differential equation [c.f., (2.135)]

$$\begin{aligned} d\boldsymbol{x}(t) &= \Big(\boldsymbol{A}[\boldsymbol{x}(t),t] \;+\; \mathbf{B}[\boldsymbol{x}(\mathbf{t}),\mathbf{t}]\cdot\boldsymbol{\xi}(\mathbf{t})\Big)\;\mathbf{dt} \\ &= \boldsymbol{A}[\boldsymbol{x}(t),t]\;dt \;+\; \mathbf{B}[\boldsymbol{x}(\mathbf{t}),\mathbf{t}]\cdot\mathbf{d}\boldsymbol{\omega}(\mathbf{t})\;. \end{aligned} \tag{5.2}$$

Instead of solving (5.1) one may simulate, therefore, stochastic processes that obey the stochastic differential equation (5.2). The probability distribution $p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)$ resulting from an ensemble of simulated stochastic processes starting at $\boldsymbol{x}(t_0)=\boldsymbol{x}_0$ can then be taken as a solution of the Fokker-Planck equation (5.1). This approach is called the *Brownian dynamics* method.
We will address the Brownian dynamics method first. Two ingredients are needed to generate sample trajectories $\boldsymbol{x}(t)$ of stochastic processes as displayed in Figure 5.4.First, one has to generate random numbers to simulate the random variables $\boldsymbol{\xi}(t)$ and $d\boldsymbol{\omega}(t)$. Second, one needs rules to

Figure 5.1: Two dimensional random walk generated by adding up 1000 random number tuples with a standard Gaussion distribution.

translate the stochastic differential equation (5.2) into a dicretized numerical form with which one can generate discretized stochastic trajectories. The generation of random numbers is the subject of the current chapter. In chapter 6 we derive rules for generating numerically stochastic trajectories. In chapter 7 we consider typical applications.
We begin the current chapter about random numbers with a definition of randomness in mathematical terms. We then address the problem of generating random numbers on a digital computer. The chapter closes with a short introduction to the *Monte Carlo* integration method, one of the most prominent random number applications and part in the derivation of the Brownian dynamics method introduced in chapter 6.

5.1 Randomness

Microscopic systems down to the level of small molecules exhibit strong random characteristics when one views selected degrees of freedom or non-conserved physical properties. The laws of statistical mechanics, even though strictly applicable only to very large systems, are realized at the molecular level, exceptions being rare. Underlying this behaviour are seemingly random events connected with transfer of momentum and energy between degress of freedom of strongly coupled classical and quantum mechanical systems. One can describe these events through random processes. We have stated above that solutions to the Fokker-Planck equations which govern the approach to statistical mechanical equilibrium can also be cast in terms of random processes. This leaves one to consider the problem how random events themselves can be mathematically modelled. This is achieved through so-called random numbers.
The concept of randomness and random numbers is intuitive and easily explained. The best known example of randomness is the throw of a dice. With each throw, a dice reveals a random number r between 1 and 6. A dice is thus a random number generator with a random number domain equal to the set $\{1, 2, 3, 4, 5, 6\}$. Other random number generators and domains are of course possible; take a dime, a roulet game, and so on.
Once a random number is obtained it is no longer random. The randomness refers to the process generating the numbers, not to a single number as such. Nevertheless a sequence of random numbers exhibits properties that reflect the generating process. In this section we will introduce

and investigate these properties. In the section thereafter we will address the problem of generating random numbers on a digital computer.
What makes a sequence of numbers $\{r_i\}$ random? Certainly, it is not just the overall probability distribution $p(r)$. A sorted and, hence, non-random list of numbers could easily satisfy this criterium. Instead, we approach randomness via the related notion of *unpredictability*. A sequence $\{r_i\}$ is unpredictable if it is impossible to foretell the value of a subsequent element r_{m+1} based on the occurence of preceding elements $r_l, \ldots, r_m$. This criterium translates into

$$p(r_{m+1}|r_l, \ldots, r_m) \;=\; p(r_{m+1}) \qquad \forall\; 0 \le l \le m < n. \tag{5.3}$$

The elements $r_l, \ldots, r_m$ must not condition r_{m+1} and, hence, conditional probabilities $p(r_{m+1}|r_l, \ldots, r_m)$ must equal the unconditional probabilities $p(r_{m+1})$. Furthermore, $p(r_{m+1})$ should be the same as the probability distribution $p(r)$ which applies to all elements of $\{r_i\}$.
Since unconditional probabilities can be factorized by conditional probabilties one obtains

$$\begin{aligned} p(r_l, \ldots, r_m, r_{m+1}) &= p(r_{m+1}|r_l, \ldots, r_m)\; p(r_l, \ldots, r_m) \\ &= p(r_{m+1})\; p(r_l, \ldots, r_m)\;, \end{aligned} \tag{5.4}$$

and, since (5.4) holds true for any $l \le m$, one can write

$$p(r_l, \ldots, r_m) \;=\; \prod_{j=\ell}^{m} p(r_j)\;. \tag{5.5}$$

Equation (5.5) provides a criterium for randomness. However, to verify criterium (5.5) for a given number sequence one has to derive a measurable quantity. Note, that the probability distributions $p(r_l, \ldots, r_m)$ are unkown. We begin this endeavor by considering the generating functions $G_{r_l \ldots r_m}(s_l \ldots s_m)$ of the unconditional probabilties $p(r_l, \ldots, r_m)$ and by applying these to equation (5.5).

$$\begin{aligned} G_{r_l \ldots r_m}(s_l \ldots s_m) &= \int \cdots \int \left(\prod_{k=l}^{m} dr_k \right)\; p(r_l, \ldots, r_m)\; \exp\left[i \sum_{k=l}^{m} s_k r_k \right] \\ &= \prod_{k=l}^{m} \int dr_k\; p(r_k)\; e^{i s_k r_k} \\ &= \big(G_r(s)\big)^{m-l+1}\;. \end{aligned} \tag{5.6}$$

Taking the logarithm of equation (5.6) and comparing the coefficients of the Taylor expansion

$$\log[G_{r_l \ldots r_m}(s_l \ldots s_m)] \;=\; \sum_{n_l, \ldots, n_m = 0}^{n} \left\langle\!\!\left\langle r_l^{n_l} \ldots r_k^{n_m} \right\rangle\!\!\right\rangle \frac{(i\, s_l)^{n_l}}{n_l!} \ldots \frac{(i\, s_m)^{n_m}}{n_m!} \tag{5.7}$$

one finds for the cumulants [c.f. Eq. (2.18) and Eq.(2.30)]

$$\left\langle\!\!\left\langle r_l^{n_l} \ldots r_m^{n_m} \right\rangle\!\!\right\rangle \;=\; 0\;, \qquad \text{if } 1 \le n_{(l \le i \le m)} \text{ and } l < m\;. \tag{5.8}$$

One can verify the criteria (5.8) of unpredictability and thus randomness by utilizing the relation

between cumulants and moments

$$\begin{aligned}
\Big\langle\!\Big\langle r_k\, r_l \Big\rangle\!\Big\rangle &= \Big\langle r_k\, r_l \Big\rangle - \Big\langle r_k \Big\rangle \Big\langle r_l \Big\rangle \,, && (5.9)\\
\Big\langle\!\Big\langle r_k\, r_l^2 \Big\rangle\!\Big\rangle &= \Big\langle r_k\, r_l^2 \Big\rangle - \Big\langle r_k \Big\rangle \Big\langle r_l^2 \Big\rangle - 2\Big\langle r_l \Big\rangle \Big\langle r_k r_l \Big\rangle + 2 \Big\langle r_k \Big\rangle \Big\langle r_l \Big\rangle^2 \,, && (5.10)\\
\Big\langle\!\Big\langle r_k\, r_l\, r_m \Big\rangle\!\Big\rangle &= \Big\langle r_k\, r_l\, r_m \Big\rangle - \Big\langle r_k \Big\rangle \Big\langle r_l r_m \Big\rangle - \Big\langle r_l \Big\rangle \Big\langle r_k r_m \Big\rangle - \Big\langle r_m \Big\rangle \Big\langle r_k r_l \Big\rangle \\
&\quad + 2 \Big\langle r_k \Big\rangle \Big\langle r_l \Big\rangle \Big\langle r_m \Big\rangle \,, && (5.11)\\
&\;\;\vdots
\end{aligned}$$

Each moment on the r.h.s. of (5.12) can be determined by taking the arithmetic average of the expression within the brackets $\langle \ldots \rangle$.
However, to take the arithmetic average one needs an ensemble of values. What, if only a single random number sequence is given? In such a case it is often permissable to create an ensemble by shifting the elements of the number sequence in a cyclic fashion. We denote a shift $\mathcal{S}_k$ of a sequence $\{r_i\}$ by k numbers by

$$\mathcal{S}_k\big(\{r_0, r_1, \ldots, r_n\}\big) \;=\; \{r_k, r_{k+1}, \ldots, r_n, r_0, \ldots, r_{k-1}\} \,. \tag{5.12}$$

The notation for a corresponding shift of a single numbers r_i is

$$\mathcal{S}_k\big(r_i\big) \;=\; r_{i+k} \,. \tag{5.13}$$

It is permissable to create an ensembles of random number sequences through operation $\mathcal{S}_k$ if one investigates a sequence that stems from an iterative generating process. This is usually the case when working with random number generators on digital computers. Such routines start with an initial number, a seed, and then generate a list of numbers by iteratively applying a mapping over and over again. The resulting number sequence starts to repeat as soon as the routine returns to the initial seed, thus forming a number cylce. This is inevitable for mappings that operate in a discrete and finite number domain. To avoid short repetitious number cycles, good number generators exhibit just one long number cycle that completely covers the available number domain. No matter which seed one chooses, the routine produces the same cycle of numbers simply shifted by a certain number of elements. Hence, applying the shift operation $\mathcal{S}_k$ j times with j different k's is equivalent to generating an ensemble of j sequences with j different seeds. One can thus write for a statisitcal moment in (5.12)

$$\Big\langle r_l^{n_l} \ldots r_m^{n_m} \Big\rangle \;=\; \frac{1}{j} \sum_{k=0}^{j-1} \big(\mathcal{S}_k\,(r_l)\big)^{n_l} \ldots \big(\mathcal{S}_k\,(r_m)\big)^{n_m} \,. \tag{5.14}$$

To verify if a number sequence is truely random, one has to check all cumulants $\big\langle\!\big\langle r_l^{n_l} \ldots r_m^{n_m} \big\rangle\!\big\rangle$ of all orders $(n_l, \ldots, n_k)$ in (5.8). These correlations should be approximately zero. Of course, due to statistical error, a variance around zero of the order of $(1/\sqrt{n}) \big\langle\!\big\langle r_l^2 \big\rangle\!\big\rangle \ldots \big\langle\!\big\langle r_m^2 \big\rangle\!\big\rangle$ is to be expected.
In practice cumulants of higher order are laborious to caluculate. One therefore performs the verification of randomness (5.8) for low orders only. We will see that a correlation check of low order is sometimes insufficient. We will give an example by applying criteria (5.8) to a linear congruential random generator, the kind of generator that we are going to introduce next.

5.2 Random Number Generators

At the begining of this chapter we already encountered a random number generator; the dice. Obviously it is not feasible to role a dice to generate a large sequence of random numbers. To automate the generation process one uses digital computers instead. A computer, however, is a deterministic machine and, thus, cannot provide truely random numbers. Nevertheless, deterministic programs, so-called random number generators, provide a good substitute.
Any program that creates a sequence of numbers $\{r_i\}$, $i = 0, 1, 2, \ldots, n$ which appear to be random with respect to the test (5.8) derived in the previous section can serve as a random number generator. In this section we introduce some of the standard random number generating programs. We begin with a mechanism that creates random number sequences with a uniform distribution in a given interval. In the paragraph thereafter we outline techniques to generate sequences with different probability distributions, in particular the Gaussian distribution. For further reading in this matter we refer the reader to chapter 3.5 of [20] and to [37].

5.2.1 Homogeneous Distribution

The best known random number generators are so-called *linear congruential generators*. They produce random numbers with a homogeneous probability distribution. Random number generators which emulate other probability distributions are, in general, based on the method introduced here. Linear congruential generators produce integer number sequences $\{r_j\}$ with a homogeneous probability distribution between 0 and some maximum number m using the recurrence relation

$$r_{i+1} \;=\; (a\, r_i \;+\; c) \mod m\,. \tag{5.15}$$

a and c are positive integers called *multiplier* and *increment*. A sequence $\{r_i\}$ starts with an arbitrarily chosen seed r_0. The linear congruential generators exhibit features common to most random number generators:

1. The sequence of random numbers is deterministic and depends on an intial value (or list of values), the seed r_0. Hence, the random number sequence is reproducible.

2. The random number generator is a mapping within a finite number range (or finite region of number tuples). Such a generator can only produce a finite sequence of random numbers and will eventually repeat that sequence all over again, thus, forming a number cycle.

3. The random number sequence tends to exhibit some sequential correlations.

Hence, before employing a random number generator one should check the following criteria.
To avoid a repitition of random numbers one should make sure that the random number cycle produced by the generator contains more elements than the random number sequence that one intends to use. A large value for m and carefully chosen parameters a and c can produce a nonrepetitious sequence of up to m random numbers. A feasible set of constants is for example $m = 2^{31} - 1$, $a = 7^5$ and $c = 0$ [44].
One should verify, if sequential correlations in a random number sequence influence the result of the calculation. One can do so by monitoring the cumulants $\langle\!\langle r_l^{n_l} \ldots r_m^{n_m} \rangle\!\rangle$ of (5.8) or by applying different random number generators and comparing the results. If needed, one can suppress sequential correlations by reshuffling a random number sequence, by merging two sequences or by similar techniques [37].
So far we can generate homogeneously distributed positive random integers on an interval $[0, m]$. One can transform these integers r into fractions or floating point numbers with a homogenous

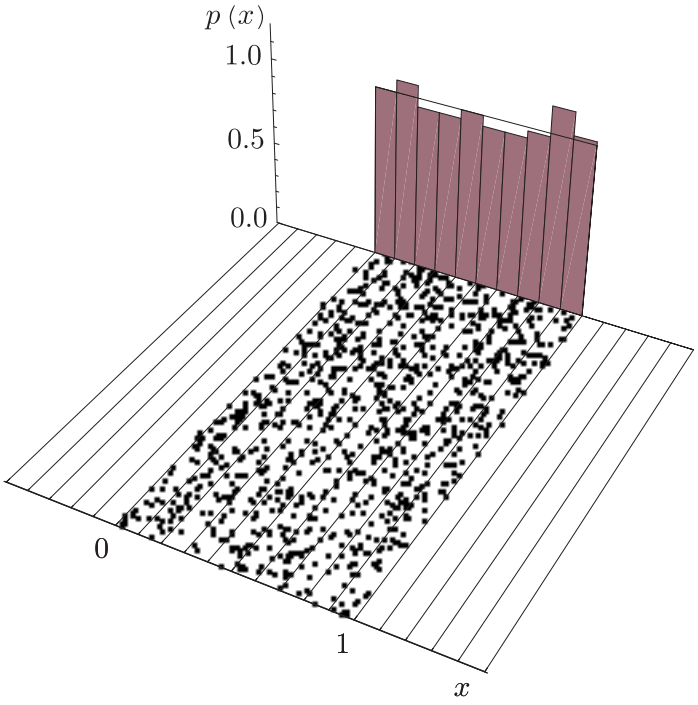

Figure 5.2: Random number sequence with a homogenous distribution in $[0,1]$ generated using (5.15) with $m = 2^{31}$, $a = 65539$, $c = 0$, and $r_0 = 1$.

distribution on an arbitrary interval $[l,u]$ by applying the linear mapping $f(r) = \frac{u-l}{m}\, r + l$. Thus, we can assume from now on to have available a basic random number generator with real-valued (or rather floating point) numbers homogeneously distributed in the interval $[0,1]$.
Before one employs any random number generator needs to be concerned about its quality. Are the numbers generated truely random? To demonstrate typical problems we will consider a pathological example which arises if one chooses in (5.15) $m = 2^{31}$, $a = 65539$ and $c = 0$. Figure 5.2 demonstrates that the linear congruential generator (5.15) produces actually for the present parameters a homogeneous distribution in the interval $[0,1]$. This figure displays a random number sequence of 1000 elements starting in the front and proceeding in 1000 steps to the rear. To verify the probability distribution a bin count with a bin width of 0.1 is displayed in the background. The bar hights represent the normalized probability density in each bin. The line superimposed on the bar chart depicts the ideal theoretical distribution.
The scattered plot in Fig. 5.3 showing the distribution of adjacent random number pairs in $\mathbb{R} \times \mathbb{R}$ allows one to detect statistical correlations $\langle\!\langle r_i\, r_{i+1} \rangle\!\rangle$ of second order. The homogeneous distribution of points in the square $[0,1] \times [0,1]$ indicates that such correlations do not exist in the present case. We can support the graphical correlation check in Fig. 5.3 numerically by calculating the correlation coefficients of order k.

$$C^{(k)}_{(n_1,n_2,\ldots,n_k)}\big(\{r_i\}\big) \;=\; \frac{\langle\!\langle r_{n_1} \ldots r_{n_k} \rangle\!\rangle}{\sqrt{\langle\!\langle r^2_{n_1} \rangle\!\rangle \cdots \langle\!\langle r^2_{n_k} \rangle\!\rangle}}\,. \tag{5.16}$$

The correlation coefficients may be viewed as normalized correlations. We have seen in (5.8) that one can detect correlations or rather the lack thereof by verifying if the cumulants $\langle\!\langle r_{n_1} \ldots r_{n_k} \rangle\!\rangle$ are zero. However, these verifications are subject to statistical errors that not only depend on the size

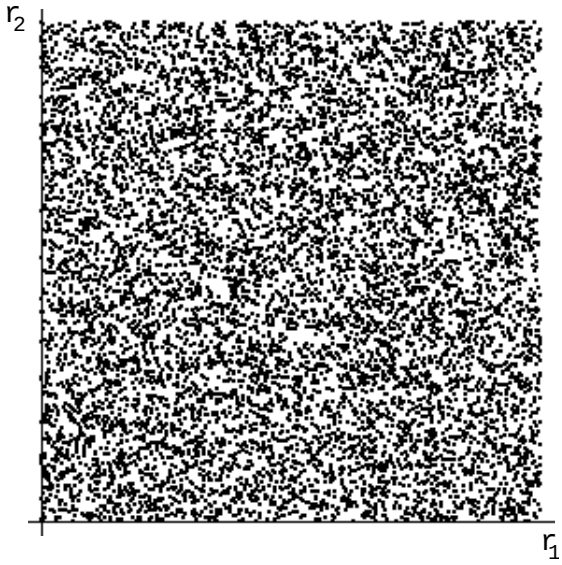

Figure 5.3: Scattered plot of 10000 adjacent random number pairs generated as in Fig. 5.2.

$$
\begin{array}{rcrcl@{\qquad}rcrcl}
C^{(2)}_{(0,0)}(\{r_i\}) &=& 1.00000 &=& 1\,, & C^{(4)}_{(0,0,0,0)}(\{r_i\}) &=& -1.19844 &\sim& -\frac{6}{5}\,, \\
C^{(2)}_{(0,1)}(\{r_i\}) &=& 0.01352 &\sim& 0\,, & C^{(4)}_{(0,0,0,1)}(\{r_i\}) &=& -0.02488 &\sim& 0\,, \\
C^{(3)}_{(0,0,0,)}(\{r_i\}) &=& -0.00099 &\sim& 0\,, & C^{(4)}_{(0,0,1,1)}(\{r_i\}) &=& 0.00658 &\sim& 0\,, \\
C^{(3)}_{(0,0,1)}(\{r_i\}) &=& -0.01335 &\sim& 0\,, & C^{(4)}_{(0,0,1,2)}(\{r_i\}) &=& 0.00695 &\sim& 0\,, \\
C^{(3)}_{(0,1,2)}(\{r_i\}) &=& -0.00670 &\sim& 0\,, & C^{(4)}_{(0,1,2,3)}(\{r_i\}) &=& -0.00674 &\sim& 0\,.
\end{array}
$$

Table 5.1: The table lists the correlation coefficients of adjacent random numbers in a sequence $\{r_i\}$ of 10000 elements generated by a linear congruential generator of equation (5.15) with $m = 2^{31}$, $a = 65539$, $c = 0$, and $r_0 = 1$.

of the ensemble of random number sequences, but are also influenced by the range of the random number domain. A scaling of the random numbers by a factor s would result in a scaling of the above cumulant by a factor s^k. Hence, to achieve comparable results one divides the cumulant by the square root of the variance of each random number. One normalizes the random number domain according to the definition (5.16).

Table 5.2.1 lists all correlation coefficients up to fourth order for adjacent random numbers of a sequence of 10000 elements. The results are compared with the ideal values of an ideal random number sequence.

All simulated correlations coefficients of two or more different sequences are roughly zero, thus satisfying the criterium in equation (5.8). To prove true randomness one would have to proceed this way and determine all correlation coefficients of all orders. Since this is an impossible endeavor one truncates the test at some low order. This, however, can be dangerous.

Figure 5.4 presents random number triplets as Figure 5.3 presented pairs of random numbers. At first sight the left scatter plot displays a perfectly homogeneous distribution indicating perfect randomness. However, rotating the coordinate system slightly reveals a different picture as shown on the right side of Fig. 5.4. We can discern that the random number triplets gather on 15 planes.

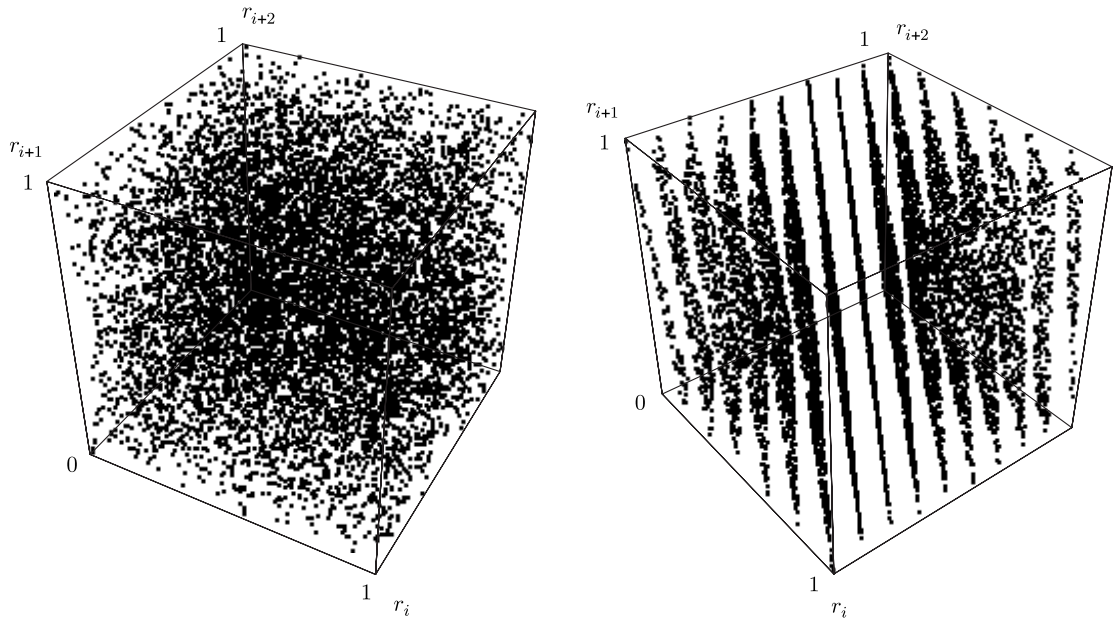

Figure 5.4: The left and right scatter plots display the same three-dimensional distribution of 10000 adjacent random number triplets as generated by the linear congruential generator used in Fig. 5.2. The right results from the left plot through rotation around the r_{i+2}-axis.

Hence, the numbers in these triplets are not completely independent of each other and, therefore, not truely random.
The lack of randomness may or may not have influenced the result of a calculation. Imagine sampling a three-dimenional density function using the pseudo random sampling points as displayed in Figure 5.4. All features of the density function that lie inbetween those 15 planes would go undetected. However, sampling a two-dimensional function with the same random numbers as displayed in Figure 5.3 would be sufficient.
Unfortunately, it is impossible to give general guidelines for the quality and feasibility of random number generators.

5.2.2 Gaussian Distribution

Random numbers with a homogeneous distribution are fairly easy to create, but for our purposes, the simulation of random processes, random numbers with a Gaussian distribution are more important. Remember that the source of randomness in the stochastic equation (5.2) is the random variable $d\omega$ which exhibits a Gaussian and not a homogeneous distribution. Hence, we have to introduce techniques to convert a random number sequence with homogeneous distribution into a sequence with a different probability distribution, e.g., a Gaussian distribution.
Given a real valued random number sequence $\{r_i\}$ with a normalized, uniform probability distribution in the interval $[0, 1]$

$$p(r)\,dr \quad = \quad \begin{cases} dr & \text{for } 0 \le r \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

one can create a new random number sequence $\{s_i\}$ by mapping a strictly monotonous function $f(r)$ onto the sequence $\{r_i\}$. The probability distribution of the new sequence $\{s_i\} = \{f(r_i)\}$ is

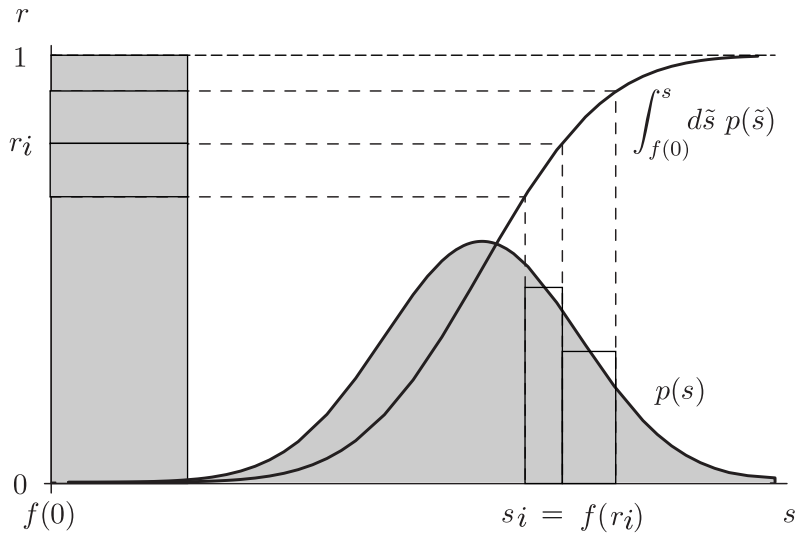

Figure 5.5: The transformation f of homogeneous random numbers r_i into random numbers s_i with a probability distribution $p(s)$.

then given by

$$p(s)\,ds \;=\; p(r)\left|\frac{\partial r}{\partial s}\right| ds\ . \tag{5.18}$$

To find the function $f(r)$ for a desired probability distribution $p(s)$ one integrates both sides of equation (5.18) over the interval $s \in [f(0), f(1)]$ or $s \in [f(1), f(0)]$ depending on $f(r)$ being a montonously increasing (i.e., $\frac{\partial f(r)}{\partial r} > 0$) or decreasing (i.e., $\frac{\partial f(r)}{\partial r} < 0$) function. We assume here for simplicity $\frac{\partial f(r)}{\partial r} > 0$. One obtains

$$\begin{aligned}\int_{f(0)}^{s} d\tilde{s}\ p(\tilde{s}) &= \int_{f(0)}^{s} d\tilde{s}\ p(r)\ \frac{\partial r}{\partial \tilde{s}} \\ &= \int_{f(0)}^{s} d\tilde{s}\ \frac{\partial f^{(-1)}(\tilde{s})}{\partial \tilde{s}} \\ &= f^{(-1)}(\tilde{s})\ \Big|_{f(0)}^{s}\end{aligned}$$

and, consequently,

$$f^{(-1)}(s) \;=\; \int_{f(0)}^{s} d\tilde{s}\ p(\tilde{s})\ . \tag{5.19}$$

The inverse of equation (5.19) renders $f(r)$. The above calculation is depicted in Fig. 5.5. The homogeneous distribution of r on the interval $[0, 1]$ is placed on the vertial axes on the left. Each (infinitesimal) bin dr is mapped by the function $f(r)$ defined in (5.19) onto the horizontal s-axes. Depending on the slope of $f(r)$ the width of the bins on the s-axes increases or decreases. However, the probability for each bin depicted by the area of the rectangles is conserved resulting in a new probability density distribution $p(s)$.
The method described here fails if one cannot find a closed or at least numerically feasable form for $f(r)$. Unfortunately, this is the case for the Gaussian distribution. Fortunately one can resort to a similar, two-dimensional approach.

Equation (5.18) reads in a multi-dimensional case

$$p(s_1, s_2, \dots)\, ds_1\, ds_2\, \dots \;=\; p(r_1, r_2, \dots) \left| \frac{\partial(r_1, r_2, \dots)}{\partial(s_1, s_2, \dots)} \right| ds_1\, ds_2\, \dots \;, \tag{5.20}$$

where $|\partial(\ )/\partial(\ )|$ is the Jacobian determinant. One obtains Gaussian-distributed random numbers through the following algorithm. One first generates two random numbers r_1 and r_2 uniformly distributed in the interval $[0, 1]$. The functions

$$\begin{aligned} s_1 &= \sqrt{-2\, \ln r_1}\; \sin[2\pi\, r_2] \\ s_2 &= \sqrt{-2\, \ln r_1}\; \cos[2\pi\, r_2] \end{aligned} \tag{5.21}$$

render then two Gaussian-distributed numbers s_1 and s_2. To verify this claim, one notes that the inverse of (5.21) is

$$\begin{aligned} r_1 &= \exp\left[-\frac{1}{2}\, \left(s_1^2 + s_2^2 \right) \right] \;, \\ r_2 &= \frac{1}{2\pi}\, \arctan \frac{s_1}{s_2} \;. \end{aligned} \tag{5.22}$$

Applying (6.57) one obtains

$$\begin{aligned} p(s_1, s_2)\, ds_1\, ds_2 &= p(r_1, r_2) \begin{vmatrix} \partial r_1/\partial s_1 & \partial r_1/\partial s_2 \\ \partial r_2/\partial s_1 & \partial r_2/\partial s_2 \end{vmatrix} ds_1\, ds_2 \\ &= p(r_1, r_2) \begin{vmatrix} -s_1\, e^{-\frac{1}{2}(s_1^2+s_2^2)} & -s_2\, e^{-\frac{1}{2}(s_1^2+s_2^2)} \\ \frac{1}{2\pi} \frac{s_2}{s_1^2+s_2^2} & -\frac{1}{2\pi} \frac{s_1}{s_1^2+s_2^2} \end{vmatrix} ds_1\, ds_2 \\ &= \frac{1}{2\pi} \left(\frac{s_1^2}{s_1^2 + s_2^2} + \frac{s_2^2}{s_1^2 + s_2^2} \right) e^{-\frac{1}{2}(s_1^2+s_2^2)}\; ds_1\, ds_2 \\ &= \left(\frac{1}{\sqrt{2\pi}}\; e^{-s_1^2/2}\, ds_1 \right) \left(\frac{1}{\sqrt{2\pi}}\; e^{-s_2^2/2}\, ds_2 \right) \;. \end{aligned} \tag{5.23}$$

This shows that s_1 and s_2 are independent Gaussian distributed numbers. Hence, one can employ (5.21) to produce Gaussian random numbers, actually, two at a time.
Figure 5.6 displays a sequence of 1000 Gaussian random numbers generated with the algorithm outlined above. The Gaussian random numbers around 0 with a standard deviation of 1 are displayed by points starting in the front and proceeding to the rear. To verify the distribution, a bin count with a bin width of 0.3 is displayed in the background. The bar hights represent the normalized probability density in each bin. The line depicts the ideal theoretical distribution as in (5.23).

5.3 Monte Carlo integration

The most prominent application of random numbers is the Monto Carlo integration method. The concept is very simple. To evaluate an integral

$$\int_\Omega d\boldsymbol{x}\; f(\boldsymbol{x}) \tag{5.24}$$

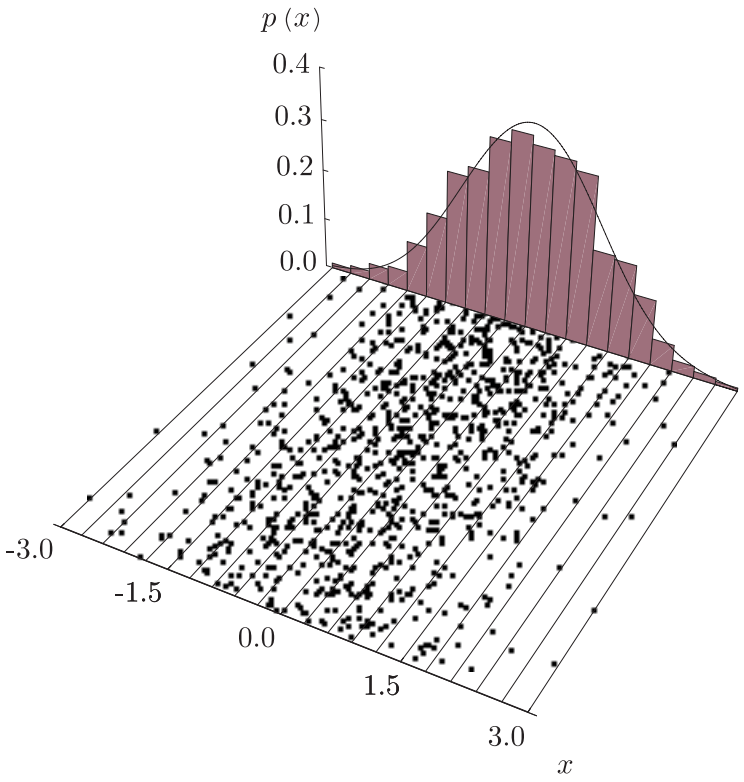

Figure 5.6: Gaussian random number sequence around 0 with a standard deviation of 1.

with the Monto Carlo method one samples the function $f(\boldsymbol{x})$ at M homogeneously distributed random points $\boldsymbol{r}_k$ in the integration domain Ω. The average of the function values at these random points times the volume $|\Omega|$ of the integration domain Ω can be taken as an estimate for the integral (5.24), as shown in Figure 5.7

$$\int_\Omega d\boldsymbol{x}\ f(\boldsymbol{x}) \quad = \quad \frac{|\Omega|}{M} \sum_{k=1}^{M} f(\boldsymbol{r}_k) \quad + \quad \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) . \tag{5.25}$$

The more function values $f(\boldsymbol{r}_k)$ are taken into account the more accurate the Monte Carlo method becomes. The average $\langle f(x) \rangle$ exhibits a statistical errors proportional to $1/\sqrt{M}$. Thus the error of the numerical integration result is of the order of $\mathcal{O}(1/\sqrt{M})$.
The integration by random sampling seems rather inaccurate at first. Systematic integration methods like the trapezoidal rule (see right Figure 5.7) appear more precise and faster. This is true in many, but not all cases.
The trapezoidal rule approximates a function $f(x)$ linearly. An approximation over intervals of length h between sampling points is thus correct up to the order of $f''(x)\,h^2$. In one dimension the length of the integration step h is given by the number of sampling points M and the length of the integration domain Ω according to $h = |\Omega|/(M+1)$. Hence, the trapezoidal rule is an approximation up to the order of $\mathcal{O}(1/M^2)$. Other systematic integration methods exhibit errors of similar polynomial order. Obviously, systematic numerical integration techniques are superior to the Monte Carlo method introduced above. However, the rating is different for integrals on higher dimensional domains.
Consider an integration domain of n-dimensions. A systematic sampling would be done over an

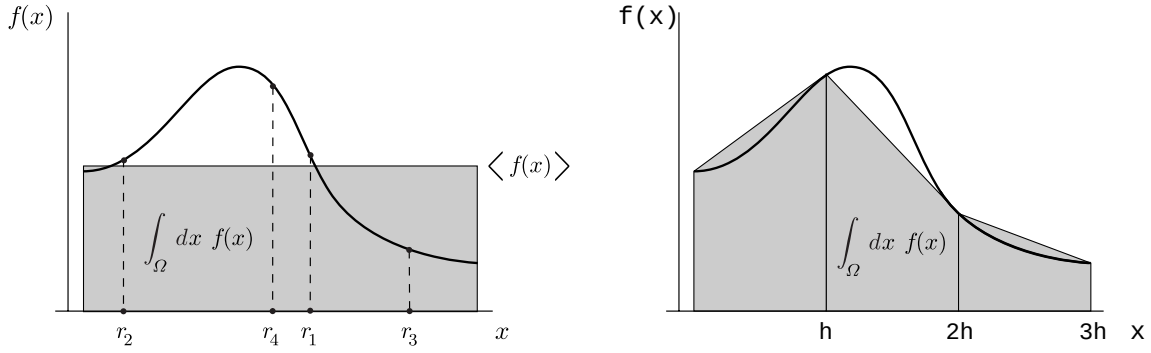

Figure 5.7: Examples for the numerical integration of $f(x)$ according to the Monte Carlo method (left) and the trapezoidal rule (right).

n-dimensional grid. A total of M sampling points would result in $\sqrt[n]{M}$ sampling points in each grid dimension. The trapezoidal rule would, thus, be correct up to the order of $\mathcal{O}(M^{-2/n})$. A random sampling, however, is not affected by dimensional properties of the sampling domain. The Monte Carlo precision remains in the order of $\mathcal{O}(M^{-1/2})$. Hence, we see that the Monte Carlo method becomes feasible for $(n = 4)$-dimensions and that it is superior for high-dimensional integration domains Ω.
One can modify the straight forward Monte Carlo integration method (5.25). Suppose one uses random points $\tilde{\boldsymbol{r}}_k$ with a normalized probability distribution $p(\boldsymbol{x})$ in Ω. Instead of evaluating integral (5.25) with a homogeneous distribution of $1/|\Omega|$ one would approximate

$$\int_\Omega d\boldsymbol{x}\ f(\boldsymbol{x})\, p(\boldsymbol{x}) \quad \sim \quad \frac{1}{M} \sum_{k=1}^{M} f(\tilde{\boldsymbol{r}}_k)\ . \tag{5.26}$$

Modification (5.26) is appropriate if a factor of the integrand happens to be a probability density distribution, for which one can generate random numbers $\tilde{\boldsymbol{r}}_k$. This will be the case in the next chapter where we will show how the Monte Carlo integration method (5.26) is incorporated in the simulation of stochastic processes.