Chapter 2

Numerical Integration and Monte Carlo Methods

2.1 Numerical Integration in One Dimension

The basic premise of numerical integration in one dimension is to approximate an integral of the form

$$F = \int_a^b f(x)dx\,, \tag{2.1}$$

where an analytical solution cannot be found.

Note that this is a somewhat different problem that the numerical integration of a differential equation, discussed previously in the context of the 1D diffusion equation (Sec. 1.2.5). The key difference is for differential equations the solution is the function (or set of functions) and only the set of derivatives w.r.t. the different dimensions are known — not the function itself. This will usually require some form of step-wise advancement along the coordinates. For the problem posed by Eq. 2.1 this need not be so: the value of $f(x)$ is already known at any position $x \in [a, b]$, the challenge thus is, to obtain a sufficiently good approximation to the exact solution.

In many cases one will interpret the expression in Eq. 2.1 geometrically as the area under the curve $f(x)$ to obtain a numerical approximation to the integral. One could thereby divide up the interval $[a, b]$ into n equally wide segments of length

$$\Delta x = \frac{b-a}{n} \tag{2.2}$$

and specify a sequence of $n+1$ points and define the positions

$$x_n = x_0 + n\Delta x\,, \tag{2.3}$$

such that $x_0 = a$ and $x_n = b$ (*compare*: Fig. 2.2.1).

2.1.1 Rectangular approximation

In this case, the x-value at the left boundary of each interval is used (Fig. 2.2.1a) and the integral in Eq. 2.1 is approximated as

$$F_n = \sum_{i=0}^{n-1} f(x_i)\Delta x\,. \tag{2.4}$$

To compute the numerical error of this approximation consider the Taylor expansion of the function f around the position x_i,

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2 + \dots \tag{2.5}$$

Inserting Eq. 2.5 into Eq. 2.1 the numerical error arising by integration from x_i to x_{i+1} can be obtained from the integral

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_i)\Delta x + \frac{1}{2}f'(x_i)\Delta x^2 + \frac{1}{6}f''(x_i)\Delta x^3 + \dots\,, \tag{2.6}$$

where $\Delta x \equiv x_{i+1} - x_i$ and where the numerical discretization was assumed by multiplying the Taylor series (Eq. 2.5) by Δx. The numerical error arising from the integration over

the segment from x_i to x_{i+1} can now be defined as the difference between the approximate value and the exact value of the integral in Eq. 2.6, namely

$$\Delta_i \equiv \int_{x_i}^{x_{i+1}} f(x)dx - f(x_i)\Delta x \approx \frac{1}{2}f'(x_i)\Delta x^2 \,. \tag{2.7}$$

According to Eq. 2.2 the integration interval $[a, b]$ was divided into n such segments. Noting further that $\Delta x \sim n^{-1}$, the total error Δ_{rect} will scale as

$$\Delta_{rect} \sim n\Delta_i \sim n\Delta_x^2 \sim n^{-1} \,. \tag{2.8}$$

2.1.2 Trapezoidal approximation.

Instead of using the left boundary of each x-segment, a trapezoid is assumed so that line segments are introduced between the points (x_i, y_i) and (x_{i+1}, y_{i+1}), in effect giving rise to the modified approximation

$$F_n = \left[\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\right]\Delta x \,. \tag{2.9}$$

To estimate the error, note that the exact integral of $f(x)$ over the interval $[x_i, x_{i+1}]$ is

$$\begin{aligned} I = \int_{x_i}^{x_{i+1}} dx\, f(x) &= \int_{x_i}^{x_{i+1}} dx\, \left[f(x_i) + xf'|_{x=x_i} + \frac{x^2}{2}f''|_{x=x_i} + \mathscr{O}(x^3)\right] && (2.10)\\ &= \Delta x f(x_i) + \frac{\Delta x^2}{2}f'|_{x=x_i} + \frac{\Delta x^3}{6}f''|_{x=x_i} + \mathscr{O}(x^4)\,, && (2.11)\\ & && (2.12) \end{aligned}$$

where $f(x)$ was expanded in a Taylor series around x_i and then integrated term by term. On the other hand, the approximate integral I_{trap} for the trapezoidal method is

$$\begin{aligned} I_{trap} &= \frac{\Delta x}{2}\left(f(x_i) + f(x_{i+1})\right) && (2.13)\\ &= \frac{\Delta x}{2}\left(f(x_i) + f(x_i) + \Delta x f'|_{x=x_i} + \frac{\Delta x^2}{2}f''|_{x=x_i} + \mathscr{O}(\Delta x^3)\right) && (2.14)\\ &= \Delta x f(x_i) + \frac{\Delta x^2}{2}f'|_{x=x_i} + \frac{\Delta x^3}{4}f''|_{x=x_i} + \mathscr{O}(\Delta x^4))\,. && (2.15) \end{aligned}$$

Hence, the numerical discrepancy between I and I_t yields the leading order error

$$|I - I_{trap}| = \frac{\Delta x^3}{12}f''|_{x=x_i}\,, \tag{2.16}$$

which is $\mathscr{O}(\Delta x^3)$ and is bounded by the product of the maximum of $|f''(x)|$ within $[x_{i-1}, x_i]$ and the interval $(x_i - x_{i-1})^3$.

Given that there are n discrete intervals within the integration range $[a, b]$ the total total error for the trapezoidal method Δ_{trap} becomes $\Delta_{trap} \sim n\Delta x^3 = n \cdot n^{-3} = n^{-2}$.

2.1.3 Quadratic interpolation (Simpson's rule)

It is often a better approximation to allow for a polynomial representation of the function $f(x)$. In the case of a parabola, each set of neighboring triplets, say (x_0, y_0), (x_1, y_1), (x_2, y_2), would define a parabola of the form

$$y(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2\,. \quad (2.17)$$

The integral

$$F_0 \equiv \int_{x=x_0}^{x=x_2} dx\, y(x) \quad (2.18)$$

$$= \frac{1}{3}(y_0 + 4y_1 + y_2)\,\Delta x\,, \quad (2.19)$$

where $\Delta x = x_1 - x_0 = x_2 - x_1$. As a sum of parabolic segments, the total integral in Eq. 2.1 then evaluates to

$$F_n = \frac{1}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right]\Delta x\,, \quad (2.20)$$

an approximation requiring n to be even and known as *Simpson's rule*. Regarding the numerical error $|I_{simp} - I|$, consider the points x_0, x_1 and x_2, and expand $f(x)$ around the point x_1 in the integral of Eq. 2.1, namely

$$\begin{aligned}
\int_{x_0}^{x_2} dx\, f(x) &= \int_{x_0}^{x_2} dx\,[f(x_1) + f'(x_1)(x-x_1) + \\
&\quad f''(x_1)(x-x_1)^2 + f'''(x_1)(x-x_1)^3 + f^{(4)}(x_1)(x-x_1)^4] \\
&= f(x_1)(x_2-x_0) + \frac{f'(x_1)}{2}[(x_2-x_1)^2 - (x_0-x_1)^2] + \\
&\quad \frac{f''(x_1)}{6}[(x_2-x_1)^3 - (x_0-x_1)^3] \\
&\quad \frac{f'''(x_1)}{24}[(x_2-x_1)^4 - (x_0-x_1)^4] + \frac{f^{(4)}(\xi)}{120}[(x_2-x_1)^5 - (x_0-x_1)^5] \\
&= 2\Delta x f(x_1) + \frac{f'(x_1)}{2}[\Delta x^2 - \Delta x^2] + \frac{f''(x_1)}{6}[\Delta x^3 + \Delta x^3] + \\
&= \frac{f'''(x_1)}{24}[\Delta x^4 - \Delta x^4] + \frac{f^{(4)}(x_1)}{120}[\Delta x^5 + \Delta x^5] \\
&= 2\Delta x f(x_1) + \Delta x^3 \frac{f''(x_1)}{3} + \Delta x^5 \frac{f^{(4)}(\xi)}{60}\,. \quad (2.21)
\end{aligned}$$

The key is now to express the value $f''(x_1)$ in terms of the values in Eq. 2.19, by Taylor expanding $f(x)$ around x_1 and evaluating it both at x_0 and x_2:

$$f(x_1 \pm \Delta x) = f(x_1) \pm \Delta x f'(x_1) + \frac{\Delta x^2}{2}f''(x_1) \pm \frac{\Delta x^3}{6}f'''(x_1) + \frac{\Delta x^4}{24}f^{(4)}(x_1)\,. \quad (2.22)$$

Thus, by adding $f(x_1 - \Delta x)$ and $f(x_1 + \Delta x)$ and re-arranging we obtain

$$f''(x_1) = \frac{f(x_0) - 2f(x_1) + f(x_2)}{\Delta x^2} - \frac{\Delta x^4}{12}f^{(4)}(\xi)\,. \quad (2.23)$$

Inserting into Eq. 2.21, we have

$$\int_{x_0}^{x_2} dx\, f(x) = \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{\Delta x^5}{90} f^{(4)}(\xi)\,, \tag{2.24}$$

where the final term is the error, i.e., the discrepancy to Simpson's approximate formula. Hence the total error becomes

$$\Delta_{simp} \sim n\Delta x^5 = n^{-4}\,. \tag{2.25}$$

2.2 Higher-dimensional numerical integration.

2.2.1 Rectangular approximation in 2D

Let us now consider higher-dimensional integration, which is quite common in many numerical integration problems. To better understand the scaling of the error with dimension, let us return to the simple rectangular approximation. The two-dimensional rectangular approximation would divide up the integration domain into a 2D-grid of equally spaced points, spacings of Δx and Δy. The average value of the function $f(x, y)$ within the interval $[x_i, x_i + \Delta x] \times [y_i, y_i + \Delta y]$ would be approximated by the value of f at one of the corners of the corresponding grid box multiplied by the area of the gridbox, yielding $f(x_i, y_i)\Delta x\Delta y$.

On the other hand, using the notation $f_x(x_0, y_0) \equiv \partial f(x, y)/\partial x|_{x_0,y_0}$, near an arbitrary grid point (x_i, y_i) the two-dimensional Taylor series has the form

$$f(x, y) = f(x_i, y_i) + f_x(x_i, y_i)(x - x_i) + f_y(x_i, y_i)(y - y_i) + \dots\,. \tag{2.26}$$

The numerical error can thus be computed as

$$\begin{aligned}
\Delta_i &= \int_{x_i}^{x_i+\Delta x}\int_{y_i}^{y_i+\Delta y} f(x, y)\, dx\, dy - f(x_i, y_i)\Delta x\Delta y \qquad (2.27)\\
&= \int_{x_i}^{x_i+\Delta x}\int_{y_i}^{y_i+\Delta y} \left(f(x_i, y_i) + f_x(x_i, y_i)(x - x_i) + f_y(x_i, y_i)(y - y_i)\right)\, dx\, dy\\
&\quad -f(x_i, y_i)\Delta x\Delta y\\
&= f(x_i, y_i)\Delta x\Delta y + f_x(x_i, y_i)\frac{\Delta x^2}{2}\Delta y + f_y(x_i, y_i)\Delta x\frac{\Delta y^2}{2}\\
&\quad -f(x_i, y_i)\Delta x\Delta y\\
&= f_x(x_i, y_i)\frac{\Delta x^2}{2}\Delta y + f_y(x_i, y_i)\Delta x\frac{\Delta y^2}{2}\,. \qquad (2.28)
\end{aligned}$$

Hence, for $\Delta x = \Delta y$, $\Delta_i \sim \Delta x^3$. With $n \sim \Delta x^{-2}$, we have that $\Delta_{rect,2D} \sim n\Delta x^3 \sim \Delta n^{-1/2}$.

Generalizing to d dimensions, the expression Eq. 2.28 would involve terms of the form

$$f(x_1, \dots, x_p, \dots, x_d)\Delta x_p^2 \prod_{q\neq p} \Delta x_q \sim \Delta x^{d+1} \sim \Delta_i\,, \tag{2.29}$$

whereas $n \sim \Delta x^{-d}$, hence $\Delta_{rect,d-dim} \sim \Delta x^{d+1} n \sim \Delta x^{d+1}\Delta x^{-d} = \Delta x = n^{-1/d}$.

Following an analogous analysis for the trapezoidal and Simpson approximations, the corresponding errors can be shown to be

$$\Delta_{trap,2D} = n^{-1}\,, \Delta_{simp,2D} = n^{-2}\,. \tag{2.30}$$

Again generalizing to higher dimensions d, the relation

$$\Delta_d \sim n^{-a/d}\,, \tag{2.31}$$

can be derived, where a takes the values 1, 2, and 4 for rectangular, trapezoidal and Simpson approximations, respectively. Notably, with increasing dimension d the decay of the error becomes exceedingly slow – even substantial increase in the number of gridpoints n will not lead to large reductions in the error.

In the following we will see that for the so-called *Monte Carlo (MC) technique* the error scales in a form that is independent of dimension:

$$\Delta_{MC} \sim n^{-1/2}\,. \tag{2.32}$$

2.2.2 Monte Carlo integration.

The mean-value theorem of integral calculus states that the definite integral (Eq. 2.1) can be evaluated when knowing the average value of the integrand $f(x)$ in the range $\leq x \leq b$, hence

$$F = \int_a^b f(x)dx = (b-a)\langle f\rangle\,. \tag{2.33}$$

Earlier, the values x_i at which $f(x)$ was sampled were chosen in well-defined intervals. For a Monte Carlo integration the values x_i are simply chosen at random, assuming a uniform distribution within the bounds of the integral. Hence, the Monte Carlo approximation is

$$F_n = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i) \approx (b-a)\langle f\rangle\,. \tag{2.34}$$

From the Central Limit Theorem (Sec. A3) we have learned that for a large number n of randomly drawn numbers the average of these n random numbers fluctuates as $n^{-1/2}$ and is distributed as a Gaussian, hence the expected error of the estimate of $\langle f\rangle$ approaches zero as $n^{-1/2}$ for large n.

How is the error in a Monte Carlo integration estimated, when the exact value of the average is not known? In this case one can resort to the *sample variance* $\tilde{\sigma}^2$, which is defined as

$$\tilde{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}[f(x_i) - \langle f\rangle]^2\,, \tag{2.35}$$

where

$$\langle f\rangle \equiv \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{2.36}$$

is the sample average. For large n we can replace this expression by that of the variance σ^2, namely

$$\sigma^2 = \langle f^2\rangle - \langle f\rangle^2\,, \tag{2.37}$$

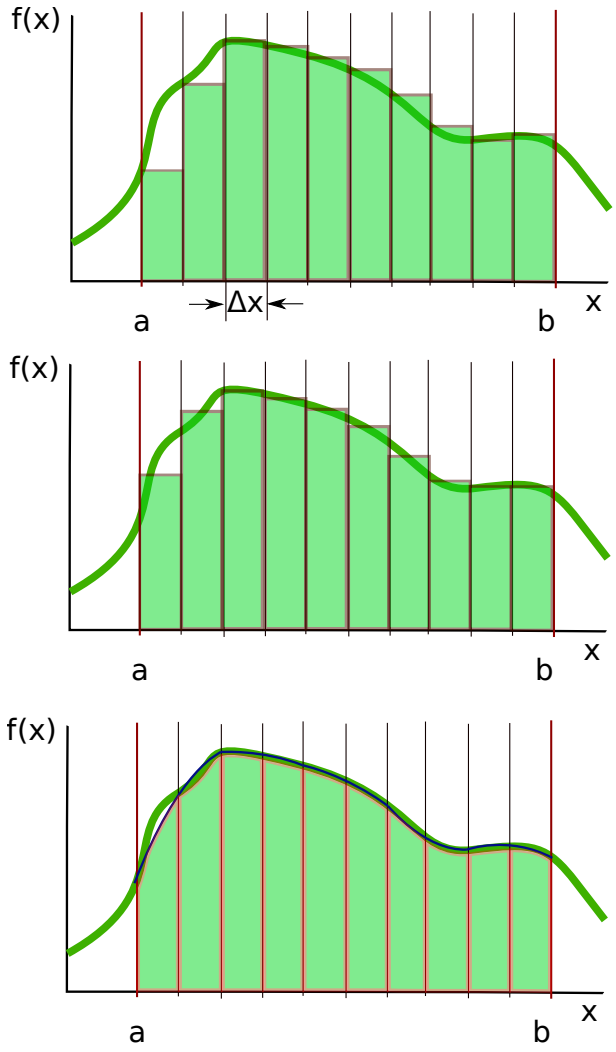

Figure 2.1: **Various simple approximations.** Schematic showing numerical integration by the rectangular, trapezoidal, and quadratic methods.

where

$$\langle f^2 \rangle = \frac{1}{n} \sum_{i=1}^{n} (f(x_i))^2 \,. \tag{2.38}$$

Whereas σ^2 does not depend systematically on n, the *standard error of the means*

$$\sigma_m \equiv \frac{\sigma}{\sqrt{n-1}} \approx \frac{\sigma}{\sqrt{n}} \tag{2.39}$$

does, Notably, Eq. 2.39 assumes that the samples are all independent. Hence, even when $\langle f \rangle$ is not known, plotting the value of σ_m against n gives an estimate on the scaling of the uncertainty regarding the mean with the number of samples.

Another way to understand the scaling $\sim n^{-1/2}$ for the Monte Carlo technique is to consider m sets of measurements with n trials each, thus requiring a total of $m\,n$ trials. For one measurement the mean is M_q is

$$M_q = \frac{1}{n} \sum_{i=1}^{n} x_{q,i} \,, \tag{2.40}$$

hence the total mean of the m measurements is

$$\langle M \rangle = \frac{1}{m} \sum_{q=1}^{m} M_q \tag{2.41}$$

$$= \frac{1}{n\,m} \sum_{q,i} x_{q,i} \,. \tag{2.42}$$

The variance of the means is

$$\sigma_m^2 = \frac{1}{m} \sum_{q=1}^{m} (M_q - \langle M \rangle)^2 \,, \tag{2.43}$$

defining $d_{q,i} =_{q,i} - \langle M \rangle$, the variance of the $n\,m$ trials can be obtained as

$$\sigma_m^2 = \frac{1}{m} \sum_{q} \left(\frac{1}{n} \sum_{i} d_{q,i} \right) \left(\frac{1}{n} \sum_{j} d_{q,j} \right) \tag{2.44}$$

$$= \frac{1}{m} \sum_{q} \frac{1}{n^2} \left(\sum_{i} d_{q,i}^2 + \sum_{i \neq j} d_{q,i} d_{q,j} \right) \,. \tag{2.45}$$

For large n only terms with $i = j$ will survive and

$$\sigma_m^2 = \frac{1}{m\,n^2} \sum_{q,i} d_{q,i}^2 = \frac{1}{n} \sigma^2 \tag{2.46}$$

hence $\sigma_m = \sigma n^{-1/2}$.

Comparing this scaling for the Monte Carlo technique with that of the Simpson approximation in d dimensions (Eq. 2.31), the perhaps startling result is that the Monte

Carlo technique allows for substantially lower computational effort — already in moderate numbers of dimensions ($d > 8$). Apart from the preferable scaling properties the Monte Carlo method offers additional advantages: it can easily be implemented, as it, at each step, only requires to draw a random coordinate and evaluate the function $f(\mathbf{x})$ for the corresponding d-dimensional coordinate $\mathbf{x}$. When a given number of evaluations have been made and the approximation is still considered too crude, then it is simple to extend the Monte Carlo integration to include further data points. This is different for the structured methods, e.g. Simpson's rule, where it is not straightforward to add data points in an unbiased way. In summary, the Simpson rule will often be the method of choice in dimensions $d < 8$ but for higher dimensional integrals the Monte Carlo method will be preferable.

Bootstrap technique. Consider that we want to estimate the error in some statistic from a given distribution function. The statistic could be the mean or variance, but also more complicated measured, e.g. a certain percentile of the distribution function. The distribution function itself could be any general distribution function or simply a set of data. For example, one might have n measurement pairs (x_i, y_i) and want to fit a best straight line to the data. The statistic of interest could be the slope G of the best fit. To estimate the error the fit parameters, one can *re-sample* n pairs of values from the sample *with replacement*, such that some samples will be selected multiple times at the expense of others not being selected at all. Each of the sub-samples k would give a new estimate on the slope G, termed G_k and, when repeating many times n_r, the set of all values G_k, $k \in \{1, \ldots, n_r\}$ will allow us to compute the variance of the estimate on G, namely

$$\sigma_G = \frac{1}{n_r - 1} \sum_{k=1}^{n_r} [G_k - \langle G_k \rangle]^2 \,, \tag{2.47}$$

with the mean

$$\langle G_k \rangle = \frac{1}{n_r} \sum_{k=1}^{n_r} G_k \,. \tag{2.48}$$

The bootstrap method is an approximation which is often used, when it is not feasible to simply repeat the experiment a larger number of times and thus obtain a clean set of samples each time. The random sampling with replacement in the bootstrap method will lead to systematic estimation errors, e.g. the variance of the distribution is often underestimated.

2.2.3 Importance sampling

In the Monte Carlo approximation the error was found to be proportional to the standard deviation σ of the integrand (Eq. 2.32). Hence, if one could reduce σ and replace f by a function of lower standard deviation, one could reduce the error. This can often be achieved by defining a positive function $p(x)$ such that

$$\int_a^b p(x) dx = 1 \,, \tag{2.49}$$

Figure 2.2: **Computational effort for Monte Carlo vs. Simpson technique.**

and then re-writing the integral (Eq. 2.1) as

$$F = \int_a^b \left[\frac{f(x)}{p(x)}\right] p(x)dx\,. \tag{2.50}$$

The integral (Eq. 2.1) can hence be evaluated by

$$F_n = \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}\,. \tag{2.51}$$

In the case of a uniform distribution $p(x) = 1/(b-a)$ the approximation in Eq. 2.33 is recovered. The aim is to minimize the variance of the ratio $f(x)/p(x)$. Clearly, the optimal function $p(x)$ for this purpose is generally not known, hence one must try to qualitatively choose a function $p(x)$ that reduces the variance. For example, given that $f(x) = \exp(-x^2)$ one may consider a function $p(x) = A\exp(-x)$, which is positive and has a decaying behavior with increasing x. Even with sub-optimal choices like this one can often substantially reduce the computational effort.

Metropolis Algorithm. Many integrals in physics, e.g. statistical mechanics, have the form

$$\langle f\rangle = \frac{\int f(x)p(x)dx}{\int p(x)dx}\,. \tag{2.52}$$

In statistical mechanics, $p(x)$ thereby could be the Boltzmann distribution

$$p(x) \propto \exp(-\beta x)\,, \tag{2.53}$$

with $\beta = 1/k_BT$ the inverse temperature, k_B the Boltzmann constant, and x the energy of a state. The denominator of Eq. 2.52 would then be referred to as the partition function

$$Z \equiv \int p(x)dx\,. \tag{2.54}$$

In statistical physics, the problem is often that while the proportionality in Eq. 2.53 is known, the coefficient Z cannot be computed — hence the normalization of $p(x)$ is unknown.

The aim of the Metropolis algorithm is to produce a random walk between values x such that, asymptotically, a set of values x will be chosen so that $p(x)$ is obeyed, that is, the distribution of the samples values of x approaches $p(x)$. A requirement is that each value of x should be accessible in finite time.

The central notion is that an update rule is defined so that asymptotically the distribution $p(x)$ will be achieved by the sampled set of values $\{x_0, x_1, \dots\}$. In equilibrium, the condition

$$\sum_{x_j} \left[p(x_i)w_{x_i\to x_j} - p(x_j)w_{x_j\to x_i}\right] = 0 \tag{2.55}$$

should be fulfilled, where $w_{x_i\to x_j}$ denotes the probability to transition from position x_i to x_j and the sum runs over all possible positions. Eq. 2.55 can be interpreted as the *probability flux* leaving x_i equals the flux returning to x_i. The condition in Eq. 2.55 can be achieved in many possible ways. Physically, the simplest case may be the most plausible, namely that each term equal zero:

$$p(x_i)w_{x_i\to x_j} - p(x_j)w_{x_j\to x_i} = 0\,, \tag{2.56}$$

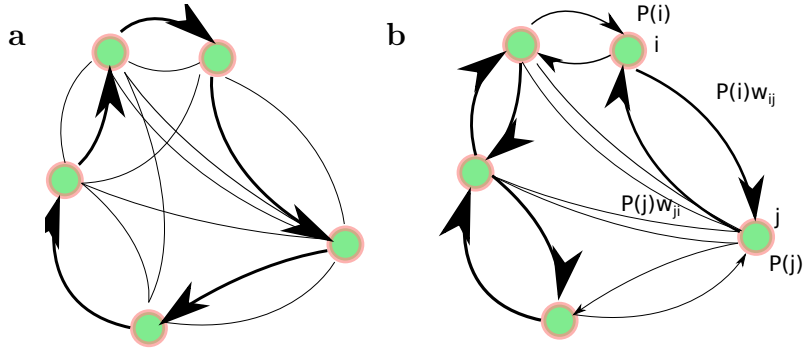

Figure 2.3: **Cartoon of transitions in the Metropolis algorithm.** Schematic showing five states and possible transitions between them (lines and arrows). **a**, States with balanced fluxes for all states: there is one large flux leaving and an equally large flux entering each nodes, thus the fluxes are balanced. **b**, Detailed balance: the net flux between *any two states* must be zero. This is depicted by similarly thick arrows connecting any two nodes back and forth.

a condition known as *detailed balance*, because the probability flux between any two positions is balanced. The condition in Eq. 2.56 can be re-expressed as

$$\frac{p(x_i)}{p(x_j)} = \frac{w_{x_i \to x_j}}{w_{x_j \to x_i}}\,, \tag{2.57}$$

hence, the normalization coefficient Z drops out of this relation, since it applies equally to $p(x_i)$ and $p(x_j)$. The remaining objective is to choose the transition rates $w_{x_i \to x_j}$ such that the proper ratio $r \equiv \frac{p(x_i)}{p(x_j)}$ is obtained — a choice that is not unique. As an ansatz, $w_{x_i \to x_j} = F(r)$ is often made, whereby, by symmetry, $w_{x_j \to x_i} = F(1/r)$. It can be verified that any function $F(r)$ of this type will fulfill the requirement of Eq. 2.57. A popular choice for $F(r)$ is the so-called *Metropolis algorithm* where

$$w_{x_i \to x_j} = \min\left[1, \frac{p(x_j)}{p(x_i)}\right]\,. \tag{2.58}$$

Implementing the Metropolis Algorithm

1. define the probability function $p(x)$
2. define an initial position x_0. set $x_{old} = x_0$
3. choose a new candidate value x_{new}
4. compute the ratio $r = \frac{p(x_{new})}{p(x_{old})}$
5. if $r \geq 1$, move to the new position, hence $x_{old} = x_{new}$, else ($r < 1$) move only at probability $r < 1$. In order to determine numerically, whether the move is made, draw a random number r' from a uniform distribution within $[0, 1]$ and perform the move only if $r' < r$ is fulfilled. If this is not fulfilled, then x_{old} is maintained.
6. by recording $f(x_i)$ at each update step i, a sequence of values $f(x_i)$ is obtained.

The algorithm consists of two stages: (i) a transient stage consisting if n_0 steps, which is used to approach equilibrium; (ii) a steady-state stage ($n > n_0$) where samples are recorded and added to the integral.