# Methods in Computational Science – Stochastic methods (ch.12)

Johan Hoffman

# Probability space

The foundation of probability theory is measures and integration. A *probability space* $(\Omega, \mathcal{F}, P)$ is a type of measure space used to model random processes. Here $\Omega$ is the *sample space*, the set of all possible outcomes $\omega \in \Omega$, and $\mathcal{F}$ is the set of *events*, a $\sigma$-algebra of collections of outcomes. We will here be concerned with the case $\mathcal{F} = 2^{\Omega}$, the $\sigma$-algebra of all subsets of $\Omega$, for which we use the notations $\mathcal{F}$ and $\Omega$ interchangeably. The *probability measure*

$$P : \mathcal{F} \to [0, 1]$$

is defined such that $P(\emptyset) = 0$ and $P(\Omega) = 1$, with the property that if $A \cap B = \emptyset$ then

$$P(A \cup B) = P(A) + P(B).$$

# Independence

We distinguish between *dependent* and *independent events*, where we say that two events $A, B \in \mathcal{F}$ are independent if

$$P(A \cap B) = P(A)P(B).$$

If two event are dependent, we define the *conditional probability* as

$$P(A|B) = \frac{P(A \cap B)}{P(B)},$$

where the conditional probability $P(A|B)$ can be either larger or smaller than $P(A)$. From the fact that $P(A \cap B) = P(B \cap A)$, we obtain *Bayes' theorem* which gives a relation between pairwise conditional probabilities,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

# Independence

**Example 12.1.** The results of two coin tosses is an example of two independent events, since the result of one coin toss does not influence the other. In contrast, the events of two cards drawn from the same deck are dependent, because the second card is drawn from a deck which miss the first card already drawn.

# Random variable

A *random variable* is a measurable function

$$X : \Omega \to S,$$

with $S$ a measurable *state space* consisting of all possible *observations* $x = X(\omega)$, for $\omega \in \Omega$, and any measurable function $f : S \to T$ defines a new random variable $f(X) = f(X(\omega))$ from the sample space $\Omega$ to the state space $T$. The probability distribution of a random variable $X$ is characterized by the *cumulative distribution function* (CDF),

$$F_X(x) = P(X \leq x), \quad x \in R.$$

# Probability mass function

If $S$ is a discrete space we say that $X(\omega) \in S$ is a discrete random variable to which we assign a *probability mass function* $p_X : S \to [0, 1]$, defined by

$$p_X(x_i) = P(\omega \subset \Omega : X(\omega) = x_i) = P(X = x_i),$$

for each observation $x_i \in S$, from which it follows that

$$\sum_{x_i \in S} p_X(x_i) = 1.$$

# Probability density function

When $S$ is a continuous space, $X$ is a continuous random variable. For certain probability distributions we can define a *probability density function* (PDF) $f_X : S \to R$, such that

$$P(a \leq X \leq b) = F_X(b) - F_X(a) = \int_a^b f_X(x)\, dx,$$

where

$$\int_{-\infty}^{\infty} f_X(x)\, dx = 1,$$

and by the fundamental theorem of calculus $f_X = F'_X$.

# Histogram

Given a set of observations $\{x_i\}_{i=1}^N$ we can estimate the PDF of the underlying random variable $X$ by constructing a *histogram*, where the state space is divided into a set of disjoint categories, or *bins*, to which each observation is added, effectively constructing a discrete approximation of the PDF in the form of a probability mass function.

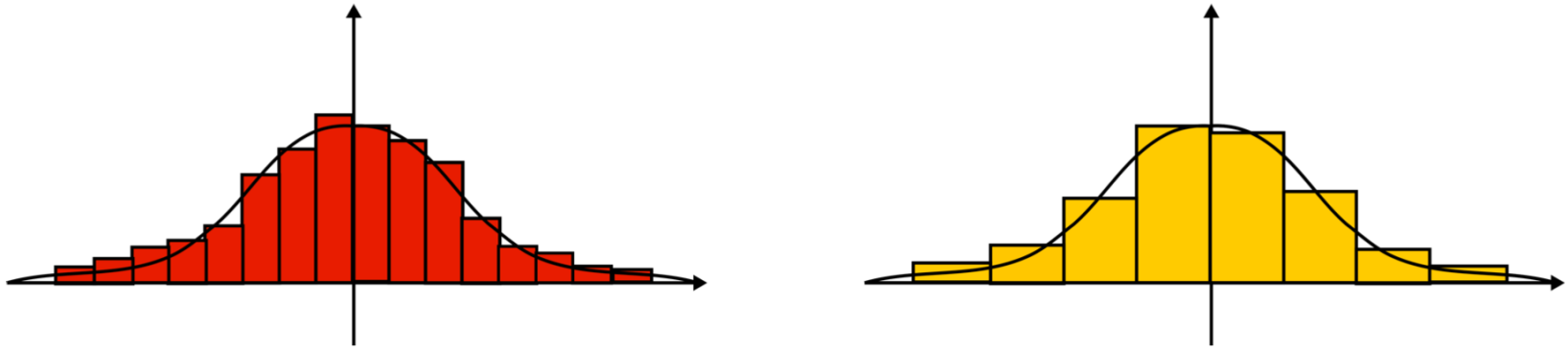


**Figure 12.1.** *Histogram approximations of a PDF, constructed from a set of observations* $\{x_i\}_{i=1}^N$ *of the underlying random variable* $X$*. The histograms represent the number of observations for each interval of two different discretizations of the state space.*

# Joint CDF

For two random variables $X$ and $Y$ with CDFs $F_X$ and $F_Y$, we write their joint CDF as

$$F_{X,Y}(x,y) = P(X \leq x \cap Y \leq y), \quad x, y \in R,$$

from which it follows that if $X$ and $Y$ are independent, in the sense that the events $X \leq x$ and $Y \leq y$ are independent, then

$$F_{X,Y}(x,y) = F_X(x)F_Y(y).$$

# Conditional CDF

If $X$ and $Y$ are not independent, we define the conditional CDF by

$$F_{X|Y}(x|y) = \frac{F_{X,Y}(x,y)}{F_Y(y)},$$

with the corresponding conditional probability mass function

$$p_{X|Y}(x|y) = P(X = x|Y = y) = \frac{P(X = x \cap Y = y)}{P(Y = y)} = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$

for discrete probability distributions, and the conditional probability density function

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

for continuous probability distributions, where $f_{X,Y} = F'_{X,Y}$. Since $F_{X,Y}$ is symmetric in the two random variables, we have that

$$f_{X|Y}(x|y)f_Y(y) = f_{Y|X}(y|x)f_X(x), \quad p_{X|Y}(x|y)p_Y(y) = p_{Y|X}(y|x)p_X(x).$$

# Expected value

We define the *expected value* of a random variable $X$ as

$$E[X] = \int_{\Omega} X(\omega) dP(\omega),$$

which in the case of a discrete random variable corresponds to

$$E[X] = \sum_{x_i \in S} X p_X(x_i),$$

and for a continuous random variable for which a PDF exists,

$$E[X] = \int_{-\infty}^{\infty} x f(x)\, dx.$$

A random variable with zero expected value is said to be *centered*.

# Variance and covariance

The *standard deviation* $\sigma_X$ is the square root of the *variance*,

$$Var[X] = \sigma_X^2 = E[(X - E[X])^2],$$

and the *covariance* between two random variables $X$ and $Y$ is defined as

$$Cov[X, Y] = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y],$$

so that $Var[X] = Cov[X, X]$. The expected value of a product of two independent random variables $X, Y$ is equal to the product of their expected values,

$$E[XY] = E[X]E[Y],$$

which follows from the definition of the expected value in terms of the probability measure.

# Uniform distributions

**Example 12.4 (Discrete uniform distribution).** The *discrete uniform distribution* is defined by the probability mass function

$$p_X(x_i) = \frac{1}{n},$$

for each $x_i \in S$ with $n$ the number of outcomes in the state space $S$. It is a model of an experiment with a finite number of possible outcomes, each with the same probability to be realized. A typical example is a coin toss, with $n = 2$ and $S = \{heads, tails\}$.

**Example 12.5 (Continuous uniform distribution).** For the sample space $S = [a, b]$, we define the *continuous uniform distribution* by the probability density function

$$f_X(x) = \frac{\chi_{[a,b]}}{b-a},$$

with the *indicator function* for a set $A$ defined by

$$\chi_A(x) \begin{cases} 1, & x \in A, \\ 0, & x \notin A. \end{cases}$$

# Normal distribution

**Example 12.6 (Normal distribution).** We denote by $X \sim N(\mu, \sigma^2)$ a random variable $X$ which has a *normal*, or *Gaussian*, probability distribution with expected value, or mean, $\mu$ and variance $\sigma^2$. A normal distribution is determined by the probability density function

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

The *standard normal distribution* is denoted by $N(0,1)$, and if $Z \sim N(0,1)$ then the random variable $X = a + bZ \sim N(a, b^2)$, for real numbers $a$ and $b \geq 0$. The probability density function $f_X$ can be expressed in terms of the standard normal distribution with probability density function $\varphi$, stretched by the standard deviation $\sigma$ and translated by the mean $\mu$,

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), \quad f_X(x) = \frac{1}{\sigma}\varphi\left(\frac{x-\mu}{\sigma}\right).$$

# Normal distribution

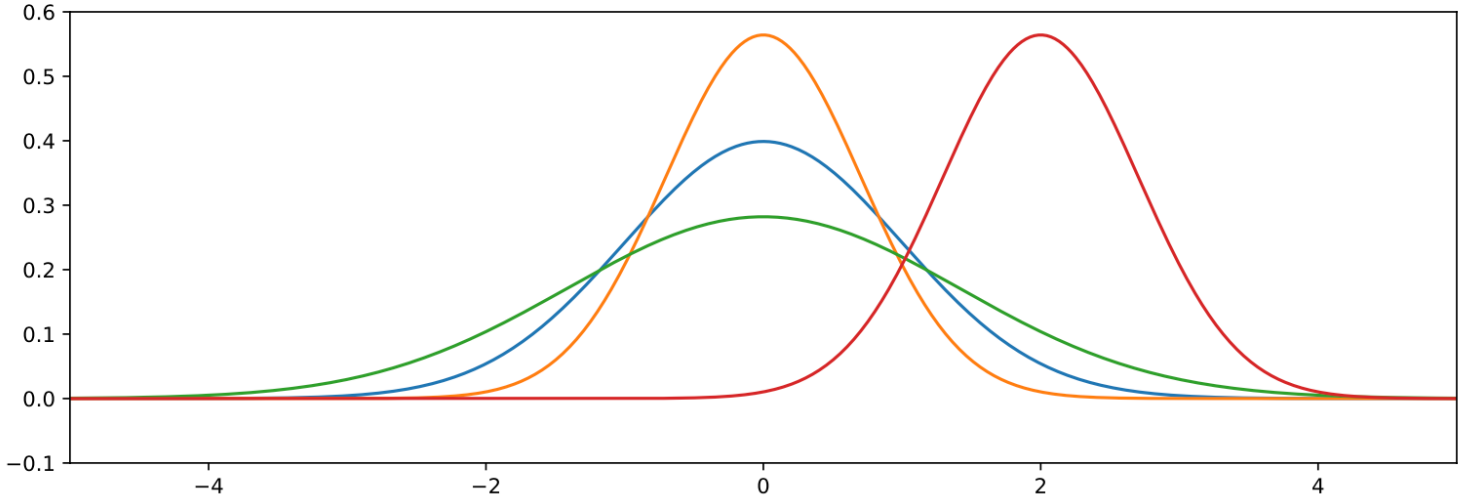



**Figure 12.2.** *PDFs for* $N(0,1)$ *(blue),* $N(0,0.5)$ *(orange),* $N(0,2)$ *(green) and* $N(2,0.5)$ *(red).*

# Stochastic process

A *stochastic process* is a collection of random variables $\{X_t\}_{t \in T}$, indexed by some index set $T$. For the index set $T = \{1, ..., n\}$ we get a *random vector* $(X_1, ..., X_n)^T$, and if $T = [0, \infty)$ or $T = \{1, 2, 3, ...\}$ the index set may represent continuous or discrete time. *Random fields* are stochastic processes for which the index set does not have a sequential order, instead some other notion of order between the random variables is needed, for example, with $T = R^n$ we can use the Cartesian coordinates. If the random variables are independent and have the same probability distribution we say that they are *independent identically distributed* (i.i.d.), else the probability distribution of one random variable is conditional on the other random variables.

# Markov chain

A *Markov chain* is a stochastic process with a sequential order for which the probability distribution of each random variable $X_t$ depends only on the previous random variable in the order, or chain, of the stochastic process. This enables simulation of a Markov process with algorithms similar to the time stepping of Algorithm 3.6.

**Example 12.7 (Random walk).** A *random walk* $\{S_n\}_{n=1}^{\infty}$ is an example of a discrete Markov chain, defined by

$$S_n = \sum_{s=1}^{n} X_s,$$

where $\{X_s\}_{s=1}^{n}$ is a set of i.i.d. random variables. Here $S_{n+1}$ is dependent on $S_n$ but independent of all $S_m$ for which $m < n$,

$$P(S_{n+1} = s | S_1 = s_1 \cap ... \cap S_n = s_n) = P(S_{n+1} = s | S_n = s_n).$$

# Random walk

**ALGORITHM 12.1. S = random_walk(A, S, Ns).**
Input: transition probability matrix **A**, initial state **S**, number of steps **Ns**.
Output: final state **S**.

```
1: n = 1
2: while n < Ns+1 do
3:     xs = sample(A, S)
4:     S = S + xs
5:     n = n + 1
6: end while
7: return S
```

# Wiener process

**Example 12.8 (Wiener process).** The *Wiener process* is a continuous Markov chain $\{W_t\}_{t \in T}$ for which each increment of step size $\Delta t$,

$$W_{t+\Delta t} - W_t,$$

has a normal distribution $N(0, \Delta t)$. Under appropriate conditions, the Wiener process can be interpreted as the limit of a random walk with vanishingly small steps. *Brownian motion* describes physical phenomena where a quantity undergoes small random fluctuations, such as microscopic particles suspended in a liquid, and can be modelled by a Wiener process, or a random walk.

# Simulations and experiments

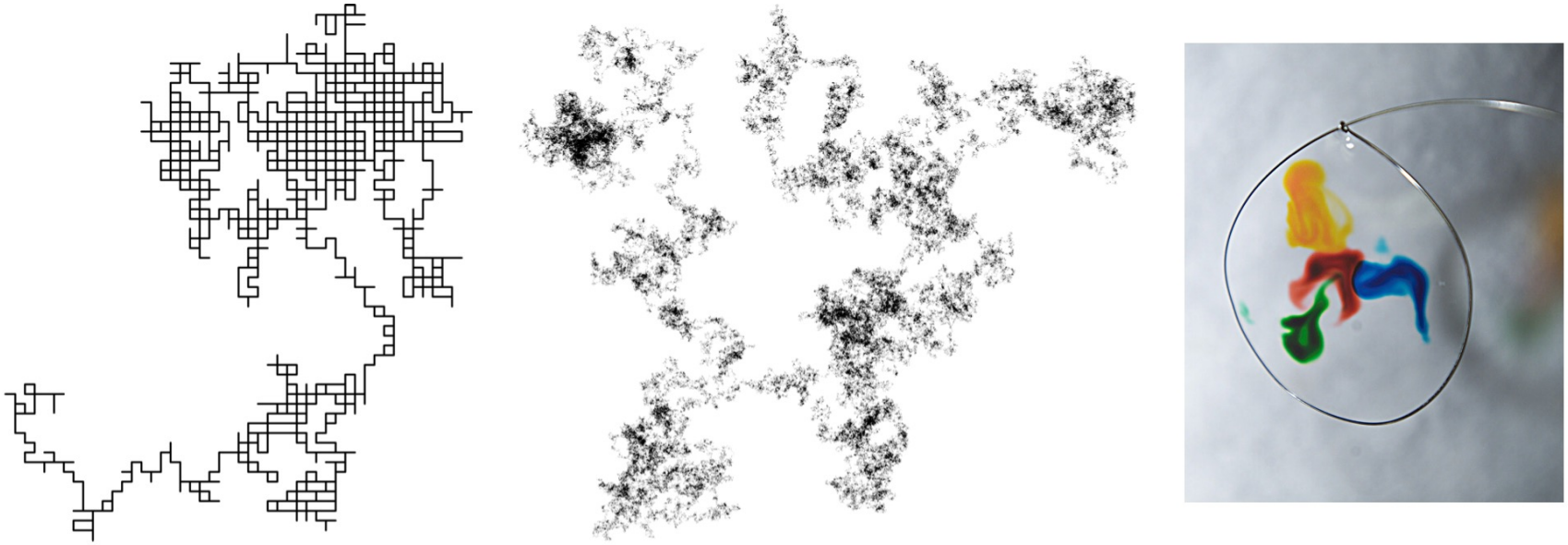


**Figure 12.3.** *Simulations of a random walk (left) and a Wiener process (center), compared to Brownian motion in the form of diffusion of food coloring.*

# Random sample

Given a random variable $X : \Omega \rightarrow S$, a *random sample* is a set of $n$ randomly generated observations $\{x_1, ..., x_n\} \subset S$. The random sample can also be viewed as a stochastic process, in the form of a set of i.i.d. random variables $\{X_1, ..., X_n\}$. The specific meaning should be clear from the context. In this section we present two fundamental theoretical results regarding random samples, a law of large numbers and a central limit theorem.

# Invers transform sampling

By *inverse transform sampling*, we can use the uniform distribution $U(0,1)$ to generate samples from other probability distributions. The main idea is the observation that $Y = F_X(X)$ can be viewed as a random variable with the state space $S = [0,1]$. Assuming the CDF $F_X$ is strictly increasing this is also true for $F_X^{-1}$, so that

$$F_X(X) \leq y \Rightarrow X \leq F_X^{-1}(y),$$

and

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P(F_X(X) \leq y) \\ &= P(X \leq F_X^{-1}(y)) = F_X(F_X^{-1}(y)) = y, \end{aligned}$$

which leads to the conclusion that $Y \sim U(0,1)$. For example, to sample a random variable $X \sim N(\mu, \sigma^2)$ we generate a sample $y$ from the random variable $Y \sim U(0,1)$, to compute $x = F_X^{-1}(y)$.

# Invers transform sampling

**ALGORITHM 12.2. x = inverse_sampling(F_inv).**
Input: inverse CDF function **F_inv** for random variable X.
Output: random sample **x** from random variable X.

```
1: y = uniform_sample()
2: x = F_inv(y)
3: return x
```

# Law of large numbers

**Theorem 12.9 (Weak law of large numbers).** *Let $X_1, ..., X_n$ be a sequence of i.i.d. random variables with $E[X_i] = \mu$ and $Var[X_i] = \sigma^2$, for all $i = 1, ..., n$. Then for any $\epsilon > 0$,*

$$\lim_{n \to \infty} P(|\bar{X} - \mu| \geq \epsilon) = 0.$$

# Central limit theorem

**Theorem 12.10 (Central limit theorem).** *Let $X_1, ..., X_n$ be independent random variables with $E[X_i] = \mu_i < \infty$, and $Var[X_i] = \sigma_i^2 < \infty$. Then the random variable*

$$Z_n = \frac{\sum_{i=1}^n X_i - \sum_{i=1}^n \mu_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}}$$

*has a standard normal distribution $N(0,1)$ in the asymptotic limit,*

$$\lim_{n\to\infty} P(Z_n \leq x) = \Phi(x).$$

*If the random variables $X_1, ..., X_n$ are i.i.d. with $E[X_i] = \mu < \infty$ and $Var[X_i] = \sigma^2 < \infty$, then for the random variable*

$$Z_n = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}},$$

$$\lim_{n\to\infty} P(Z_n \leq x) = \Phi(x).$$

# Monte Carlo integration

$$\bar{g} = \frac{1}{n}\sum_{i=1}^{n} g(x_i),$$

$$E[g] = \frac{1}{|D|}\int_D g(x)\,dx.$$

By the law of large numbers, for large $n$ $\bar{g} \approx E[g]$, which suggests a quadrature rule in the form of the sample mean of a 1-point quadrature rule based on random sampling of quadrature points $x_i$ from the uniform distribution $U(D)$,

$$\int_D g(x)\,dx \approx \frac{1}{n}\sum_{i=1}^{n} g(x_i)|D| = \frac{|D|}{n}\sum_{i=1}^{n} g(x_i).$$

# Monte Carlo integration

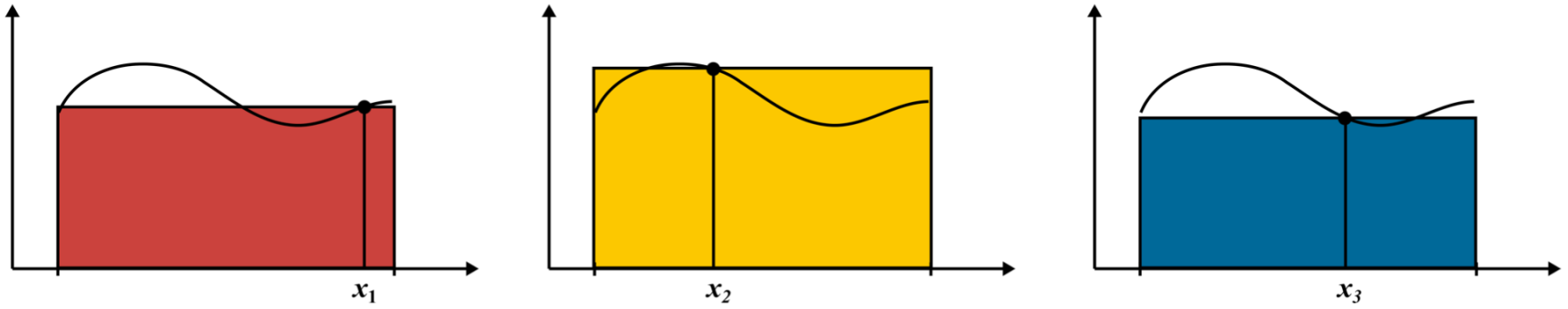



**Figure 12.4.** *Monte Carlo integration over the domain $D \subset R$, with three quadrature points $x_1, x_2, x_3 \in D$, randomly sampled from the uniform distribution $U(D)$. The Monte Carlo approximation is the sample mean of the corresponding 1-point quadrature rules.*

# Monte Carlo integration

$$\left| \int_D g(x)\,dx - \frac{|D|}{n} \sum_{i=1}^{n} g(x_i) \right| = \sigma_g \frac{|D|}{\sqrt{n}} \left| \frac{E[g] - \bar{g}}{\sigma_g / \sqrt{n}} \right|$$

with $\sigma_g^2$ the variance of $g$. Hence, if we assume that the variance $\sigma_g^2$ is bounded the approximation converges at a rate $1/\sqrt{n}$ in a statistical sense, independent of the dimension $d$ of the domain of integration $D \subset R^d$, since the random variable

$$\frac{\bar{g} - E[g]}{\sigma_g / \sqrt{n}}$$

in the limit is distributed as $N(0, 1)$. To accelerate the convergence, variance reduction methods are developed, seeking to reduce $\sigma_g$ by clever selections of the sampled random variables $X_i$.

# Metropolis algorithm

**Example 12.11 (Metropolis algorithm).** At step $t$ of a MCMC algorithm we have generated a sample $x_t$, and from a *proposal probability density function*, or *jump distribution*, $g(x|x_t)$ we generate a candidate $x$ for the next step. We assume that the jump distribution is symmetric $g(x|x_t) = g(x_t|x)$. A common choice is to use a normal distribution centered at $x_t$,

$$g(x|x_t) = N(x_t, \sigma^2),$$

where the variance $\sigma^2$ is a parameter of the method. If the target probability density function is $f_X$, then we can formulate a rejection sampling algorithm based on the acceptance criterion $\alpha = f_X(x)/f_X(x_t)$, where we draw a uniformly distributed sample $u \in [0, 1]$ and accept $x$ as $x_{t+1}$ if $u \leq \alpha$, or else we let $x_{t+1} = x_t$. This is the *Metropolis algorithm.*

# Latin hypercube sampling

**Example 12.12 (Latin hypercube sampling).** When sampling the random variable $X_i \sim U(D)$ to compute the Monte Carlo approximation (12.4), we may restrict the random samples to better cover the domain $D \subset R^d$. If $d = 1$, we can distribute the samples evenly over subintervals of $D \subset R$, corresponding to a composite quadrature rule. If the domain has a tensor product structure,

$$D = [a_1, b_1] \times \ldots \times [a_d, b_d],$$

then we can discretize the domain by a structured grid, constructed as a tensor product of each discretized interval $[a_i, b_i]$. *Latin hypercube sampling* is based on such a structured grid, where only a fixed number of samples per row of cells in the grid is allowed in each dimension, see Figure 12.5 for an example in $R^2$ where one sample per row is allowed.

# Latin hypercube sampling

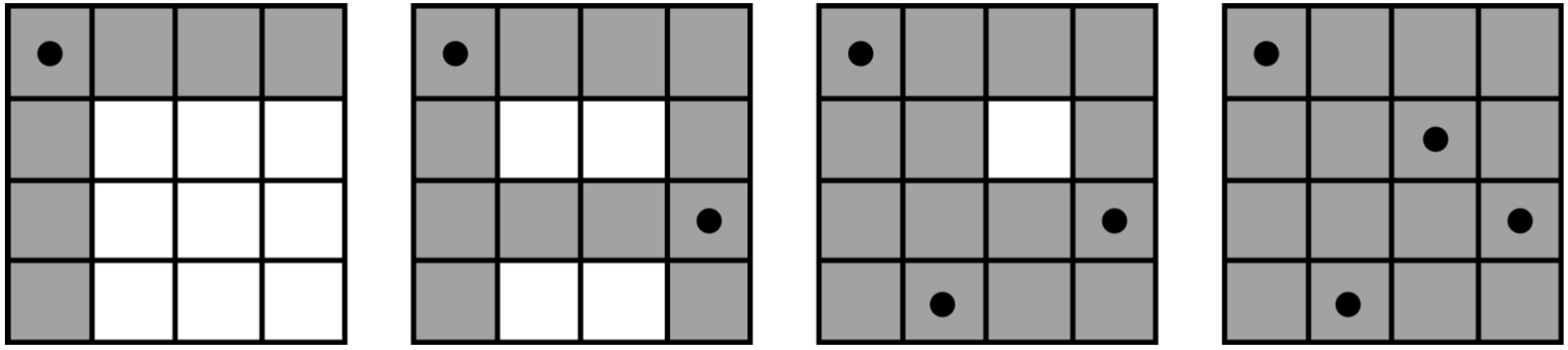


**Figure 12.5.** *Latin hypercube sampling over a tensor product domain in $R^2$, which is discretized by a structured grid. One sample per row is allowed in each dimension, so that for each new sample an increasing part of the domain is closed (shaded).*

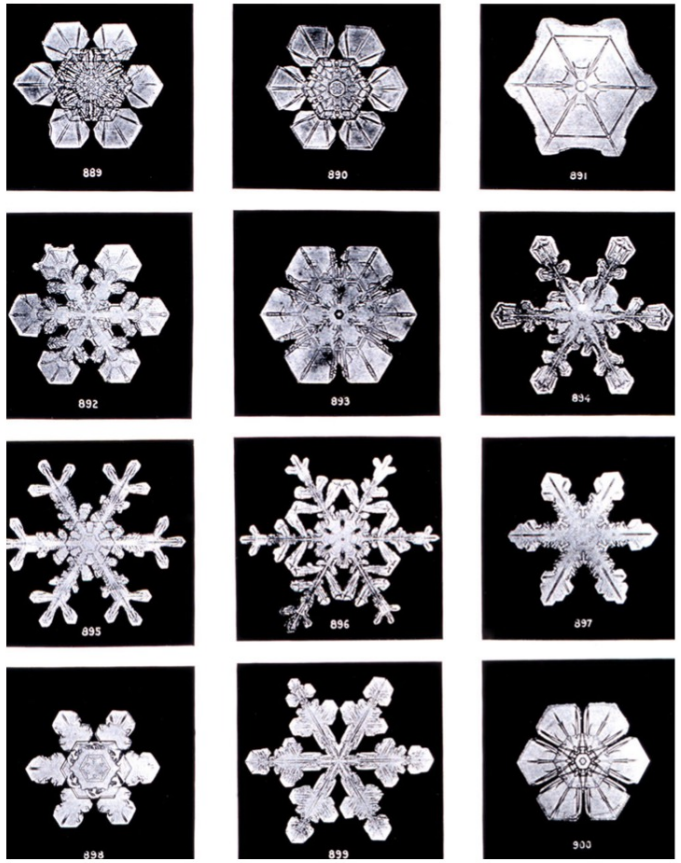
# Agent-based models

An agent-based model can take the form of a *cellular automaton*, a filter kernel acting on a structured grid as a Boolean function that produces one of two states, black or white, based on the neighborhood of each cell in the grid. Cellular automata have been used to model, for example, the structure of a snowflake. With a Moore neighborhood on a 2D grid, there are $2^9 = 512$ possible patterns of black and white cells that represent the domain of the Boolean function.

**Example 12.13 (Game of Life).** The *Game of Life* represents a cellular automaton defined over a 2D structured grid, where a white cell is interpreted as alive whereas a black cell is dead. With different initial conditions over the grid, the following rules generate the game: (i) if a live cell has two or three live neighbors it survives, (ii) any dead cell with three live neighbors becomes alive, and (iii) all other dead or alive cells remain in their states.

Game of Life
Snowflake CA

# Agent-based models



An agent-based model can take the form of acting on a structured grid as a Boolean function that produc ased on the neighborhood of each cell in the grid. Cellular a r example, the structure of a snowflake. With a Moore ne $2^9 = 512$ possible patterns of black and white cells that re nction.

**Example 12.13 (Game of Life).** The *Game of l* efined over a 2D structured grid, where a white cell is interp dead. With different initial conditions over the grid, the foll a live cell has two or three live neighbors it survives, (ii) a s becomes alive, and (iii) all other dead or alive cells remai

# Agent-based models

**Example 12.14 (Boids).** The artificial life computer program *Boids* implements an agent-based model of flocking birds, where each bird defines its neighbors through a metric corresponding to the Euclidian distance. A simple form of the model can be constructed as a combination of the following rules: (i) avoid collision with neighbors, (ii) align with the average direction of neighbors, and (iii) steer towards the average position of all neighbors.

Boids

# Agent-based models

**Example 12.14 (Boids).** model of flocking birds, to the Euclidian distance. the following rules: (i) a neighbors, and (iii) steer t

ı agent-based
orresponding
mbination of
: direction of