# Computational Science: Modeling and Simulation

Roberto M. Cesar Jr.

rmcesar@usp.br

Chapters 7, 11

# RANDOM PROCESSES

# Random experiments

| **Experiment** | $E_1$: Measuring your weight in the morning | $E_2$: Throwing a dice |
|---|---|---|
| **Sample Space** | $S = \{\text{a real number between } 30\,\text{Kg and } 200\,\text{Kg}\}$ | $S = \{1, 2, 3, 4, 5, 6\}$ |
| **Some possible events** | $A_1 = \{71\,\text{Kg}\}$<br>$A_2 = \{\text{weight} \geqslant 70\,\text{Kg}\}$<br>$A_3 = \{65\,\text{Kg} < \text{weight} < 69\,\text{Kg}\}$ | $B_1 = \{1\}$<br>$B_2 = \{2, 3, 4\}$<br>$B_3 = S$ |
| **Outcomes** | One of the infinite elements of $S$ | {1} {2} {3} {4} {5} {6} |

**Table 2.5:** *Elementary concepts in probability.*

$$P(A) = \frac{\text{number of elements in } A}{\text{number of elements in } S}.$$

$$P(B_1) = \frac{1}{6},$$

$$P(B_2) = \frac{3}{6} = \frac{1}{2}, \quad \text{and}$$

$$P(B_3) = \frac{6}{6} = 1.$$

More generally, the probability of an outcome/event is defined as the limit of the relative frequencies of respective observations when the number of observations tends to infinity. A well-defined probabilistic model should satisfy the following axioms:

**Axiom 1:** $P(S) = 1$;

**Axiom 2:** $0 \leqslant P(A) \leqslant 1$;

**Axiom 3:** For two *mutually exclusive events* $A$ and $B$ (i.e., $A \cap B = \emptyset$):

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

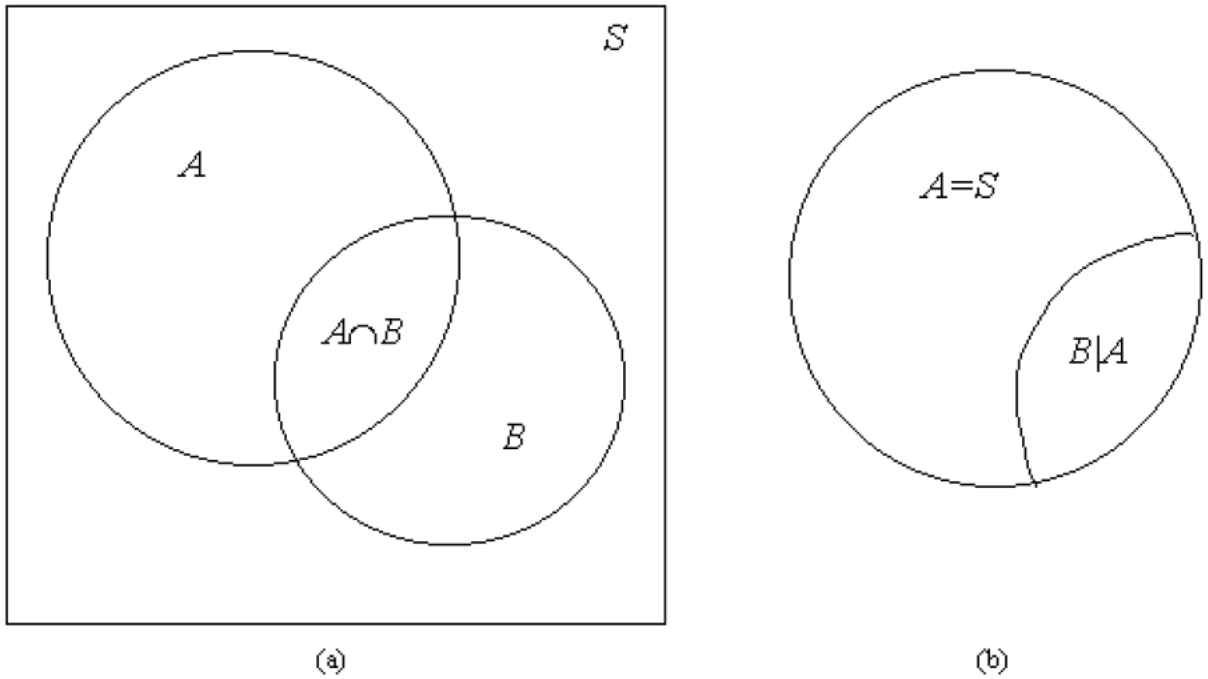



**Figure 2.62:** *Venn diagram illustrating the events A, B, and their intersection* (a). *The fact that A has occurred redefines the sample space* (b).

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

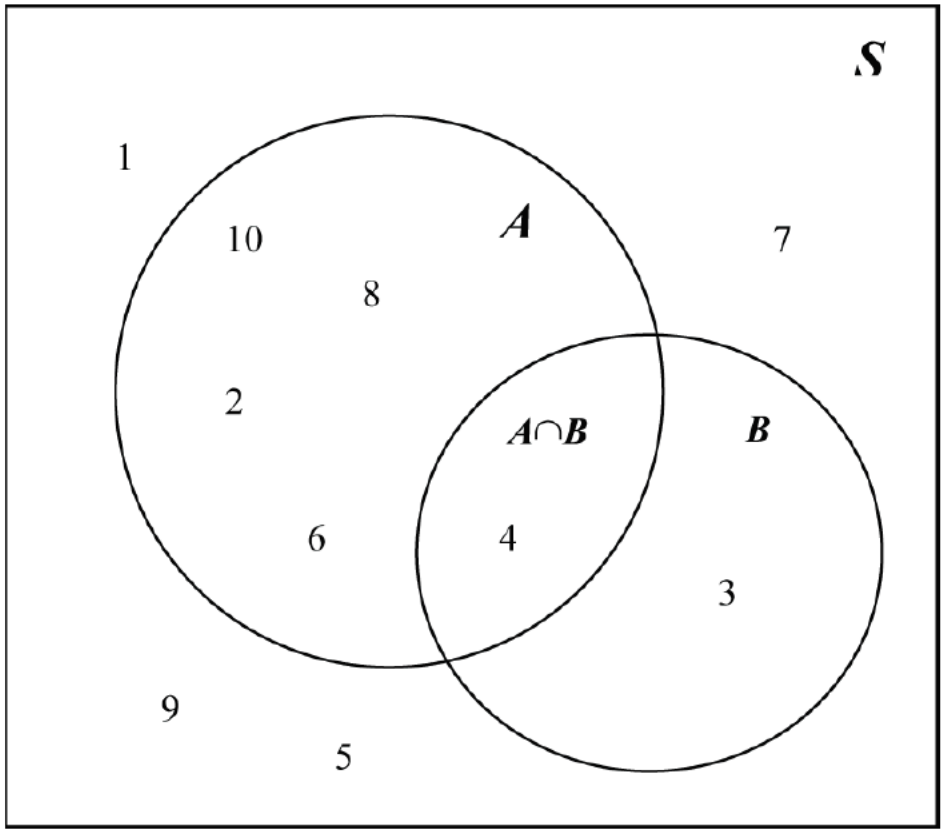



**Figure 2.63:** *Venn diagram illustration of the situation in the above example* (b)*, i.e., extracting balls from an urn. In case the extracted ball is known to be even (represented as the event A), the probability of being the ball number* 4 *is* $P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{1}{5}$.

The following relationship between conditional probabilities is known as *Bayes's rule*:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}. \tag{2.30}$$

# Random variables

- Random experiments
- Random variables
- Probability distributions

## Density Functions

The key concept involved in random variables is that of a *density function*, which is any function $p(x)$, where $x$ stands for observations of a random variable $X$, satisfying the following criteria:

$$p(x) \geqslant 0 \quad \text{for any real } x$$

and

$$\int_{-\infty}^{\infty} p(x)\, \mathrm{d}x = 1.$$

Given a density probability function $p(x)$, the respective *probability distribution* $P(x)$ can be defined as:

$$P(x) = P(X \leqslant x) = \int_{-\infty}^{x} p(s)\, \mathrm{d}s, \tag{2.31}$$

hence

$$p(x) = \frac{dP}{dx} = P'(x).$$

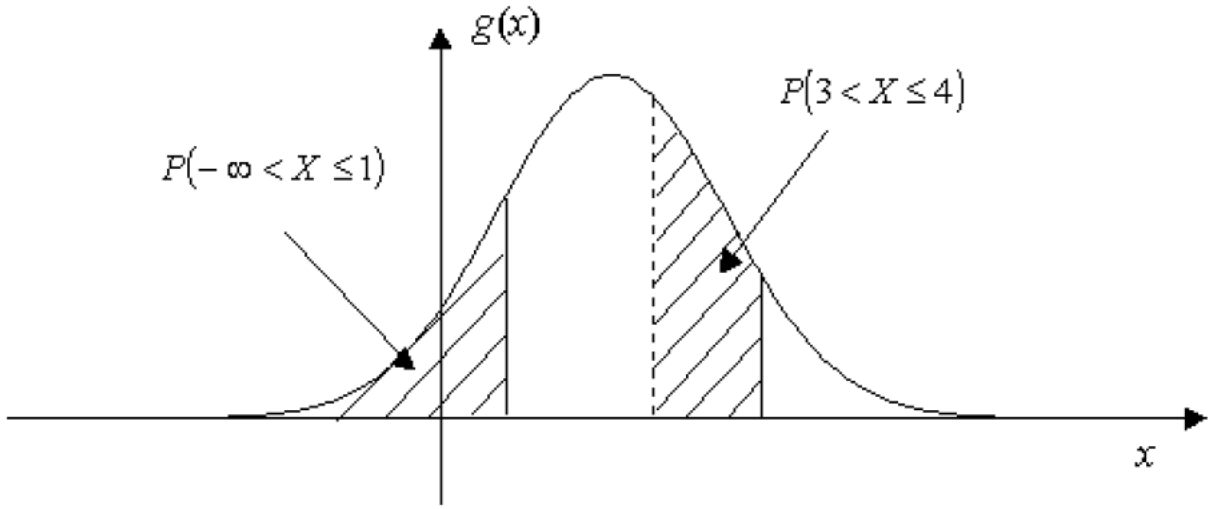


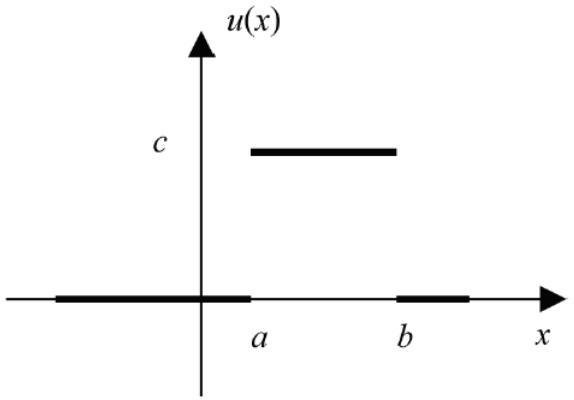

**Figure 2.64:** *A Gaussian density function characterizing the random variable X. The probability of finding an observation comprised in an interval is equal to the area below the function along that interval.*

## Uniform

$$u(x) = \begin{cases} c = \frac{1}{b-a} & a < x \leqslant b \\ 0 & \text{otherwise} \end{cases}$$

$$E(x) = \frac{a+b}{2}$$

$$\text{var}(x) = \frac{(b-a)^2}{12}$$

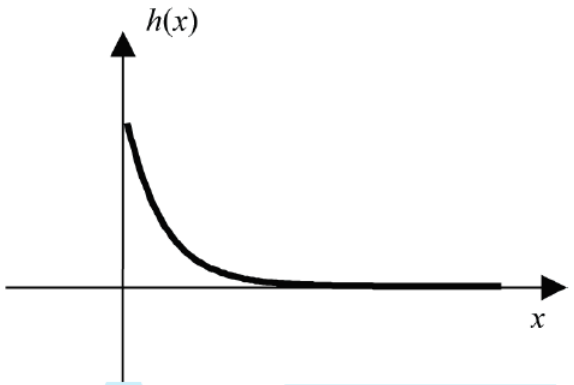



## Exponential

$$h(x) = \alpha \exp(-\alpha x), \quad x > 0$$

$$E(x) = \frac{1}{\alpha}$$

$$\text{var}(x) = \frac{1}{\alpha^2}$$




## Gaussian

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

$$E(x) = \mu$$

$$\text{var}(x) = \sigma^2$$

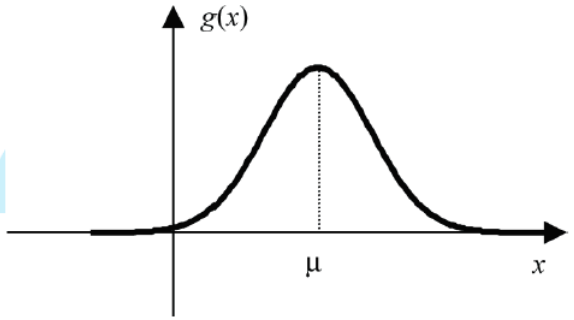



**Table 2.6:** *Important density probability functions.*

## 2.6.3 Random Vectors and Joint Distributions

A *random vector* is nothing else but a vector of random variables. For instance, the temperature ($T$), pressure ($P$) and humidity ($H$) at noon in the North Pole on a certain day can be represented by the following random vector:

$$p = \begin{bmatrix} T \\ P \\ H \end{bmatrix} = (T, P, H)\,.$$

$$\vec{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = (X_1, X_2, \ldots, X_N)\,.$$

The *joint density function* characterizing the behavior of such a random vector has the general form:

$$p\left(\vec{x}\right) = p\left(x_1, x_2, \ldots, x_N\right),$$

An example of joint density function is the multivariate Gaussian, given by the following:

$$p\left(\vec{x}\right) = \frac{1}{(2\pi)^{\frac{N}{2}}\sqrt{\det\left(K\right)}} \exp\left[-\frac{1}{2}\left(\vec{x} - \vec{\mu}_{\vec{X}}\right)^{\mathrm{T}} K^{-1}\left(\vec{x} - \vec{\mu}_{\vec{X}}\right)\right].$$

This density function is completely specified by the mean vector $\vec{\mu}_{\vec{X}}$ and the covariance matrix $K$ (see below).

# Random models in physics

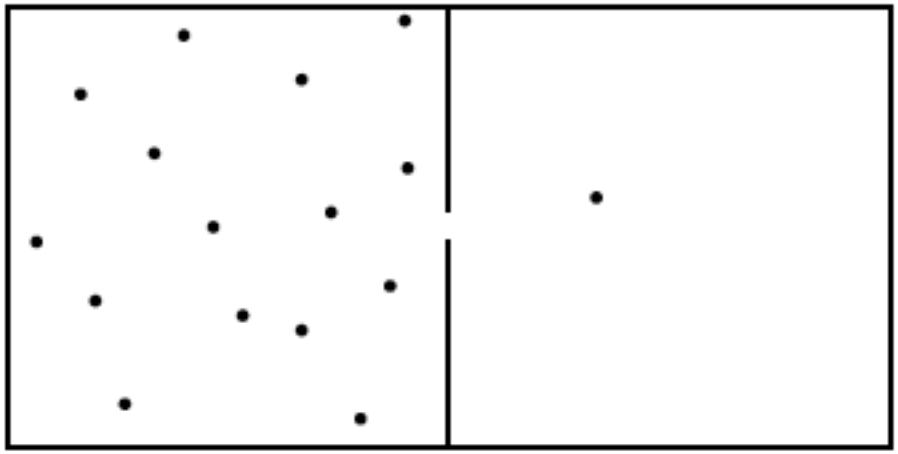


Figure 7.1: A box is divided into two equal halves by a partition. After a small hole is opened in the partition, one particle can pass through the hole per unit time.

1. Use a random number generator to choose a particle at random.
2. Move this particle to the other side of the box.
3. Give the particle a random position on the new side of the box. This step is for visualization purposes only.
4. Increase the "time" by unity.

How long does it take for the system to reach equilibrium? How does this time depend on the number of particles? After the system reaches equilibrium, what is the magnitude of the fluctuations? How do the fluctuations depend on the number of particles? Problems 7.2 and 7.3 explore such questions.

Java - osp_csm/src/org/opensourcephysics

Partitioned box

File Edit Display Tools Help

Fermat.java

Capturing Animated GIF (.gif)

Animated GIF (.gif) fps: 30 Loop

045 Capture Clear Save As...

Box data

File Edit Display Tools Views Help

number on left

time

BoxApp Controller

File Edit Display Help

Input Parameters

| Name | Value |
|---|---|
| Number of particles | 64 |
| steps per display | 10 |

Start Step New

Messages

Problems

BoxApp [Jav

**Problem 7.2.** Approach to equilibrium

a. Use `BoxApp` and `Box` and describe the nature of the evolution of $n$, the number of particles on the left side of the box. Choose the total number of particles $N$ to be $N = 8$, 16, 64, 400, 800, and 3600. Does the system reach equilibrium? What is your qualitative criterion for equilibrium? Does $n$, the number of particles on the left-hand side, change when the system is in equilibrium?

b. The algorithm we have used is needlessly cumbersome, because our only interest is the number of particles on each side. We used the positions only for visualization purposes. Because each particle has the same chance to go through the hole, the probability per unit time that a particle moves from left to right equals the number of particles on the left divided by the total number of particles, that is, $p = n/N$. Modify the program so that the following algorithm is implemented.

   (i) Generate a random number $r$ from a uniformly distributed set of random numbers in the interval $0 \leq r < 1$.

   (ii) If $r \leq p = n/N$, move a particle from left to right, that is $n \to n-1$; otherwise, $n \to n+1$.

c. Does the time dependence of $n$ appear to be deterministic for sufficiently large $N$? What is the qualitative behavior of $n(t)$? Estimate the time for the system to reach equilibrium from the plots. How does this time depend on $N$?

**Problem 7.3.** Equilibrium fluctuations

a. As a rough measure of the equilibrium fluctuations, visually estimate the deviation of $n(t)$ from $N/2$ for $N = 16$, 64, 400, 800, and 3600? Choose a time interval that is bigger than the time needed to reach equilibrium. How do your results for the deviation depend on $N$?

b. A better measure of the equilibrium fluctuations is the mean square fluctuations $\Delta n^2$, which is defined as

$$\Delta n^2 = \langle (n - \langle n \rangle)^2 \rangle = \langle n^2 \rangle - 2\langle n \langle n \rangle \rangle + \langle n \rangle^2 = \langle n^2 \rangle - 2\langle n \rangle^2 + \langle n \rangle^2 = \langle n^2 \rangle - \langle n \rangle^2. \qquad (7.1)$$

The brackets, $\langle \cdots \rangle$, denote an average taken after the system has reached equilibrium. The relative magnitude of the fluctuations is $\Delta n / \langle n \rangle$. Modify your program so that averages are taken after equilibrium has been reached. Run for a time that is long enough to obtain meaningful results. Compute the mean square fluctuations $\Delta n^2$ for the same values of $N$ considered in part (a). How do the relative fluctuations, $\Delta n / \langle n \rangle$, depend on $N$? (You might find it helpful to see how averages are computed in Listings 7.3 and 7.4.)

From Problem 7.2 we see that $n(t)$ decreases in time from its initial value to its equilibrium value in an almost deterministic manner if $N >> 1$. It is instructive to derive the time dependence of $n(t)$ to show explicitly how chance can generate deterministic behavior. If there are $n(t)$ particles on the left side after $t$ moves, then the change in $\langle n \rangle(t)$ in the time interval $\Delta t$ is given by

$$\Delta\langle n\rangle = [\frac{-\langle n\rangle(t)}{N} + \frac{N - \langle n\rangle(t)}{N}]\Delta t. \tag{7.2}$$

(We defined the time so that the time interval $\Delta t = 1$ in our simulations.) What is the meaning of the two terms in (7.2)? If we treat $\langle n \rangle$ and $t$ as continuous variables and take the limit $\Delta t \to 0$, we have

$$\frac{\Delta\langle n\rangle}{\Delta t} \to \frac{d\langle n\rangle}{dt} = 1 - \frac{2\langle n\rangle(t)}{N}. \tag{7.3}$$

The solution of the differential equation (7.3) is

$$\langle n\rangle(t) = \frac{N}{2}[1 + e^{-2t/N}], \tag{7.4}$$

where we have used the initial condition $\langle n \rangle(t = 0) = N$. Note that $\langle n \rangle(t)$ decays exponentially to its equilibrium value $N/2$. How does this form (7.4) compare to your simulation results for various values of $N$? We can define a *relaxation time* $\tau$ as the time it takes the difference $[\langle n \rangle(t) - N/2]$ to decrease to $1/e$ of its initial value. How does $\tau$ depend on $N$? Does this prediction for $\tau$ agree with your results from Problem 7.2?

https://en.wikipedia.org/wiki/Monte_Carlo_method

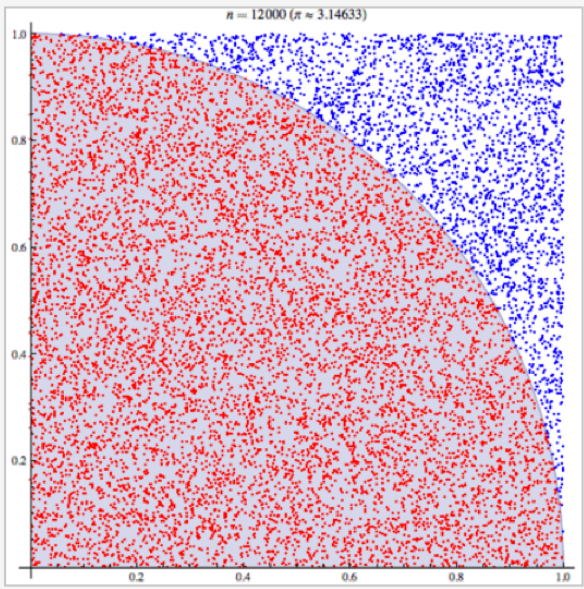
# Introduction [ edit ]

Monte Carlo methods vary, but tend to follow a particular pattern:

1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain.
3. Perform a deterministic computation on the inputs.
4. Aggregate the results.

For example, consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is π/4, the value of π can be approximated using a Monte Carlo method:[6]

1. Draw a square on the ground, then inscribe a circle within it.
2. Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.
3. Count the number of objects inside the circle and the total number of objects.
4. The ratio of the two counts is an estimate of the ratio of the two areas, which is π/4. Multiply the result by 4 to estimate π.




Monte Carlo method applied to approximating the value of π. After placing 30000 random points, the estimate for π is within 0.07% of the actual value. This happens with an approximate probability of 20%.

https://en.wikipedia.org/wiki/Monte_Carlo_method

Being secret, the work of von Neumann and Ulam required a code name.[*citation needed*] A colleague of von Neumann and Ulam, Nicholas Metropolis, suggested using the name *Monte Carlo*, which refers to the Monte Carlo Casino in Monaco where Ulam's uncle would borrow money from relatives to gamble.[7] Using lists of "truly random" random numbers was extremely slow, but von Neumann developed a way to calculate pseudorandom numbers, using the middle-square method. Though this method has been criticized as crude, von Neumann was aware of this: he justified it as being faster than any other method at his disposal, and also noted that when it went awry it did so obviously, unlike methods that could be subtly incorrect.

http://jackman.stanford.edu/mcmc/metropolis1.pdf

John von Neumann saw the relevance of Ulam's suggestion and, on March 11, 1947, sent a handwritten letter to Robert Richtmyer, the Theoretical Division leader (see "Stan Ulam, John von Neumann, and the Monte Carlo Method"). His letter included a detailed outline of a possible statistical approach to solving the problem of neutron diffusion in fissionable material.

Johnny's interest in the method was contagious and inspiring. His seemingly relaxed attitude belied an intense interest and a well-disguised impatient drive. His talents were so obvious and his cooperative spirit so stimulating that he garnered the interest of many of us. It was at that time that I suggested an obvious name for the statistical method—a suggestion not unrelated to the fact that Stan had an uncle who would borrow money from relatives because he "just had to go to Monte Carlo." The name seems to have endured.

The spirit of Monte Carlo is best conveyed by the example discussed in von Neumann's letter to Richtmyer. Consider a spherical core of fissionable material surrounded by a shell of tamper material. Assume some initial distribution of neu-

determine the new momentum of the neu-

**John von Neumann**

tron. When the neutron crosses a material boundary, the parameters and characteristics of the new medium are taken into account. Thus, a genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated.

numbers.)

Once one has an algorithm for generating a uniformly distributed set of random numbers, these numbers must be transformed into the nonuniform distribution $g$ desired for the property of interest. It can be shown that the function $f$ needed to achieve this transformation is just the inverse of the nonuniform distribution function, that is, $f = g^{-1}$. For example, neutron physics shows us that the distribution of free paths—that is, how far neutrons of a given energy in a given material go before colliding with a nucleus—decreases exponentially in the interval $(0, \infty)$. If $x$ is uniformly distributed in the open interval $(0, 1)$, then $f = -\ln x$ will give us a nonuniform distribution $g$ with just those properties.

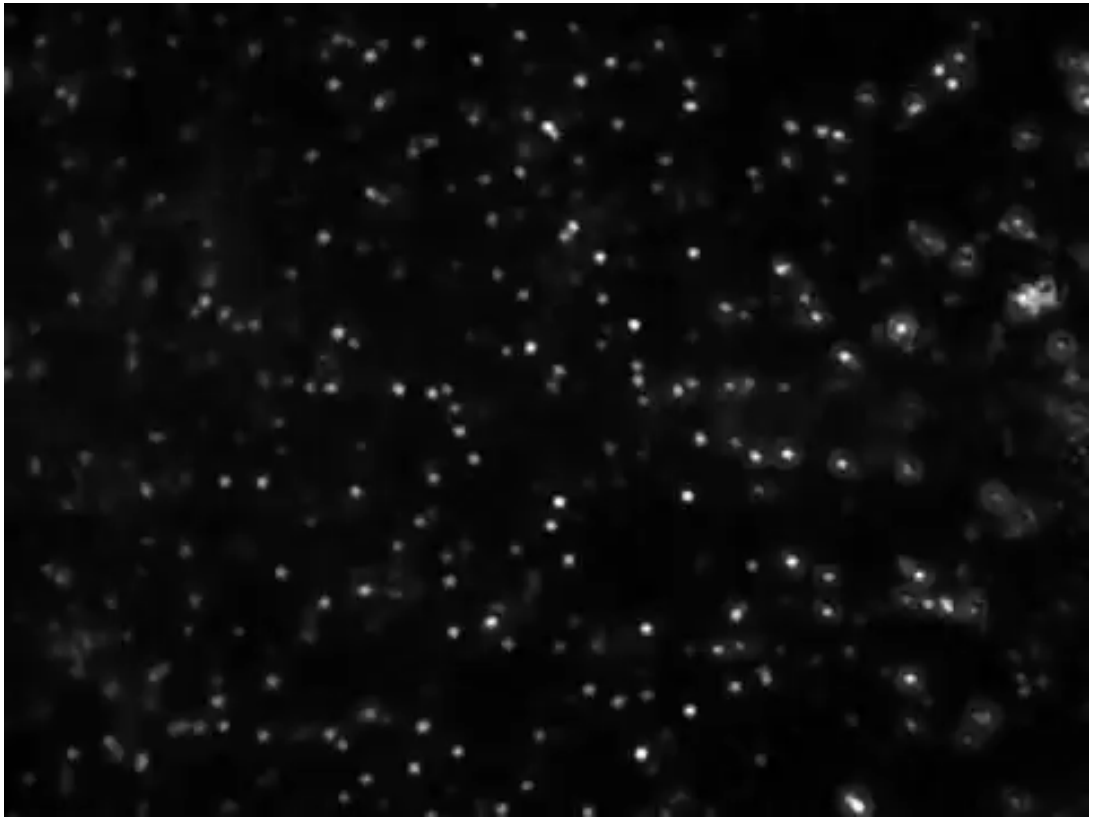
The reader will appreciate many of the advantages of the Monte Carlo method compared to the methods of differential equations. For example, a neutron-velocity spectrum with various peaks and valleys is difficult to handle mathematically. For Monte Carlo one needs only to mirror the velocity spectrum in the probability distribution. Also, the Monte Carlo method is sufficiently flexible to ac-

Chapter 7

# RANDOM WALKS

# Random walks

- Drunken sailor
- Brownian motion



https://www.youtube.com/watch?v=cDcprgWiQEY

We first consider an idealized example of a random walker that can move only along a line. Suppose that the walker begins at $x = 0$ and that each step is of equal length $a$. At each time interval the walker has a probability $p$ of a step to the right and a probability $q = 1 - p$ of a step to the left. The direction of each step is independent of the preceding one. After $N$ steps the displacement $x$ of the walker from the origin is given by

$$x_N = \sum_{i=1}^{N} s_i, \tag{7.5}$$

where $s_i = \pm a$. For $p = 1/2$ we can generate one walk of $N$ steps by flipping a coin $N$ times and increasing $x$ by $a$ each time the coin is heads and decreasing $x$ by $a$ each time the coin is tails.

$$x_N^2 = [\sum_{i=1}^{N} s_i]^2.$$

We can gain more insight into the nature of random walks by doing a Monte Carlo simulation, that is, by using a computer to "flip coins." The implementation of the random walk algorithm is simple, for example,

```
if (p < Math.random()) {
    x++;
}
else {
    x--;
}
```

Clearly we have to sample many $N$ step walks because in general, each walk will give a different outcome. We need to do a Monte Carlo simulation many times and average over the results to obtain meaningful averages. Each $N$-step walk is called a *trial*. How do we know how many trials to use? The simple answer is to average over more and more trials until the average results don't change within the desired accuracy. The more sophisticated answer is to do an error

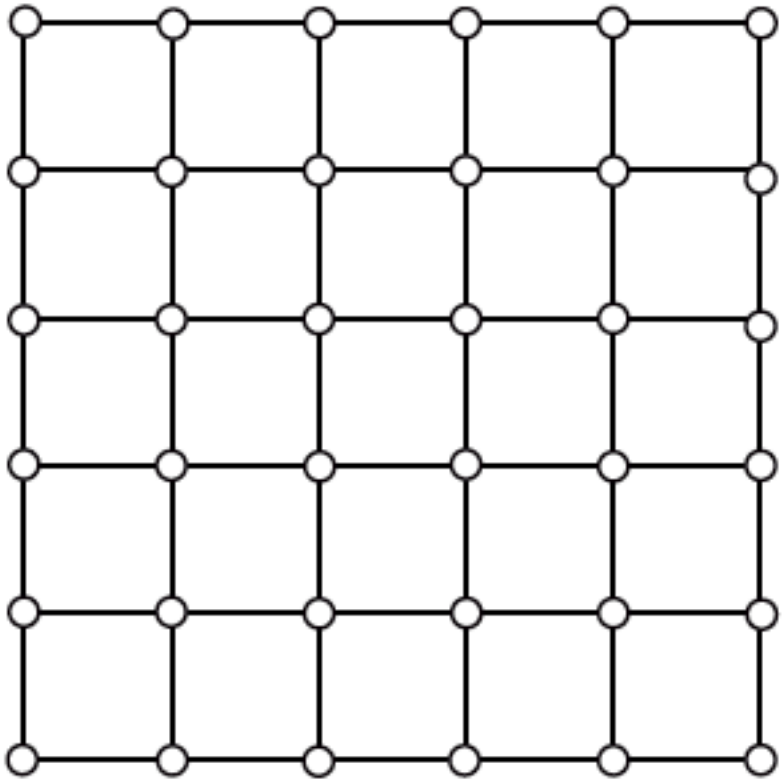


Figure 7.2: An example of a $6 \times 6$ square lattice. Note that each site or node has four nearest neighbors.

**Problem 7.8.** A random walk in two dimensions

a. Consider a collection of walkers initially at the origin of a square lattice (see Figure 7.2). At each unit of time, each of the walkers moves at random with equal probability in one of the four possible directions. Create a drawable class, `Walker2D`, which contains the positions of $M$ walkers moving in two dimensions and draws their location, and modify `WalkerApp`. Unlike `WalkerApp`, this new class need not specify the maximum number of steps. Instead the number of walkers should be specified.

b. Run your application with the number of walkers $M \geq 1000$ and allow the walkers to take at least 500 steps. If each walker represents a bee, what is the qualitative nature of the shape of the swarm of bees? Describe the qualitative nature of the surface of the swarm as a function of the number of steps, $N$. Is the surface jagged or smooth?

c. Compute the quantities $\langle x \rangle$, $\langle y \rangle$, $\Delta x^2$, and $\Delta y^2$ as a function of $N$. The average is over the $M$ walkers. Also compute the mean square displacement $R^2$ given by

$$R^2 = \langle x^2 \rangle - \langle x \rangle^2 + \langle y^2 \rangle - \langle y \rangle^2 = \Delta x^2 + \Delta y^2. \tag{7.11}$$

What is the dependence of each quantity on $N$? (As before, we will frequently write $R^2$ instead of $R_N^2$.)

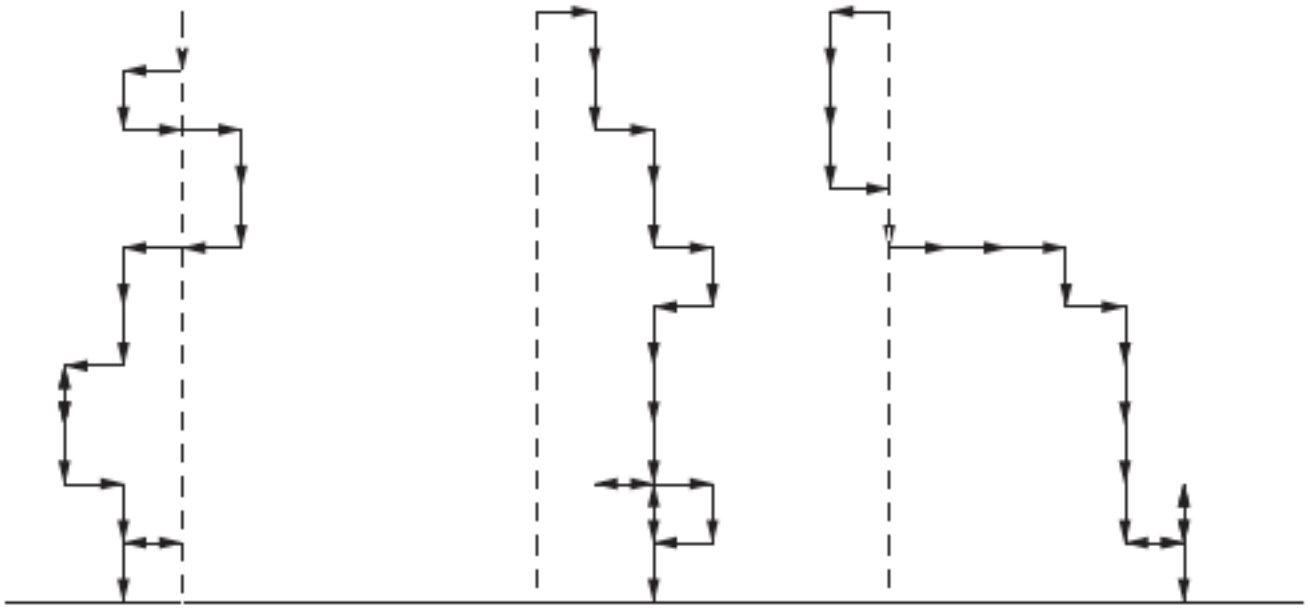


Figure 7.3: Examples of the random path of a raindrop to the ground. The step probabilities are given in Problem 7.9. The walker starts at $x = 0$, $y = h$.

# Modified random walks

- Memory
- Multiple random walkers
- Restricted random walks and traps
- Persistent random walk
- Synchronized random walks
- Steps of variable length
- Central limit theorem

**Problem 7.11.** A persistent random walk

a. In a persistent random walk, the *transition* or jump probability depends on the previous step. Consider a walk on a one-dimensional lattice, and suppose that step $N-1$ has been made. Then step $N$ is made in the same direction with probability $\alpha$; a step in the opposite direction occurs with probability $1-\alpha$. Write a program to do a Monte Carlo simulation of the persistent random walk in one dimension. Estimate $\langle x \rangle$, $\Delta x^2$, and $P_N(x)$. Note that it is necessary to specify both the initial position and an initial direction of the walker. What is the $\alpha = 1/2$ limit of the persistent random walk?

**Problem 7.12.** Synchronized random walks

a. Randomly place two walkers on a one-dimensional lattice of $L$ sites, so that both walkers are not at the same site. At each time step randomly choose whether the walkers move to the left or to the right. Both walkers move in the same direction. If a walker cannot move in the chosen direction because it is at a boundary, then this walker remains at the same site for this time step. A trial ends when both walkers are at the same site. Write a program to determine the mean time and the mean square fluctuations of the time for two walkers to reach the same site. This model is relevant to a method of doing cryptography using neural networks (see Rutter et al.).

# Application

- Propose a game using random walkers

# Modified random walks

**Problem 7.16.** Generation of the Gaussian distribution

*__Problem 7.17.__ Random walk on lattices containing random traps

# Method of Least-squares

$$y = mx + b.$$

$$d_i = y_i - mx_i - b,$$

$$\chi^2 = \sum_{i=1}^{n} (y_i - mx_i - b)^2.$$

$$\frac{\partial \chi}{\partial m} = -2 \sum_{i=1}^{n} x_i (y_i - mx_i - b) = 0,$$

$$\frac{\partial \chi}{\partial b} = -2 \sum_{i=1}^{n} (y_i - mx_i - b) = 0.$$

# Method of Least-squares

$$m\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i$$
$$m\sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i.$$

$$\langle c\rangle = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$\langle y\rangle = \frac{1}{n}\sum_{i=1}^{n} y_i$$
$$\langle xy\rangle = \frac{1}{n}\sum_{i=1}^{n} x_i y_i,$$

$$m\langle x^2\rangle + b\langle x\rangle = \langle xy\rangle,$$
$$m\langle x\rangle + b = \langle y\rangle.$$

$$m = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sigma_x^2}$$
$$b = \langle y\rangle - m\langle x\rangle,$$

$$\sigma_x^2 = \langle x^2\rangle - \langle x\rangle^2.$$

| $N$ | $\Delta x^2$ |
|---|---|
| 8 | 19.43 |
| 16 | 37.65 |
| 32 | 76.98 |
| 64 | 160.38 |

Table 7.2: Computed values of the mean square displacement $\Delta x^2$ as a function of the total number of steps $N$. The mean square displacement was averaged over 1000 trials. The one-dimensional random walker takes steps of length 1 or 2 with equal probability, and the direction of the step is random with $p = 1/2$.

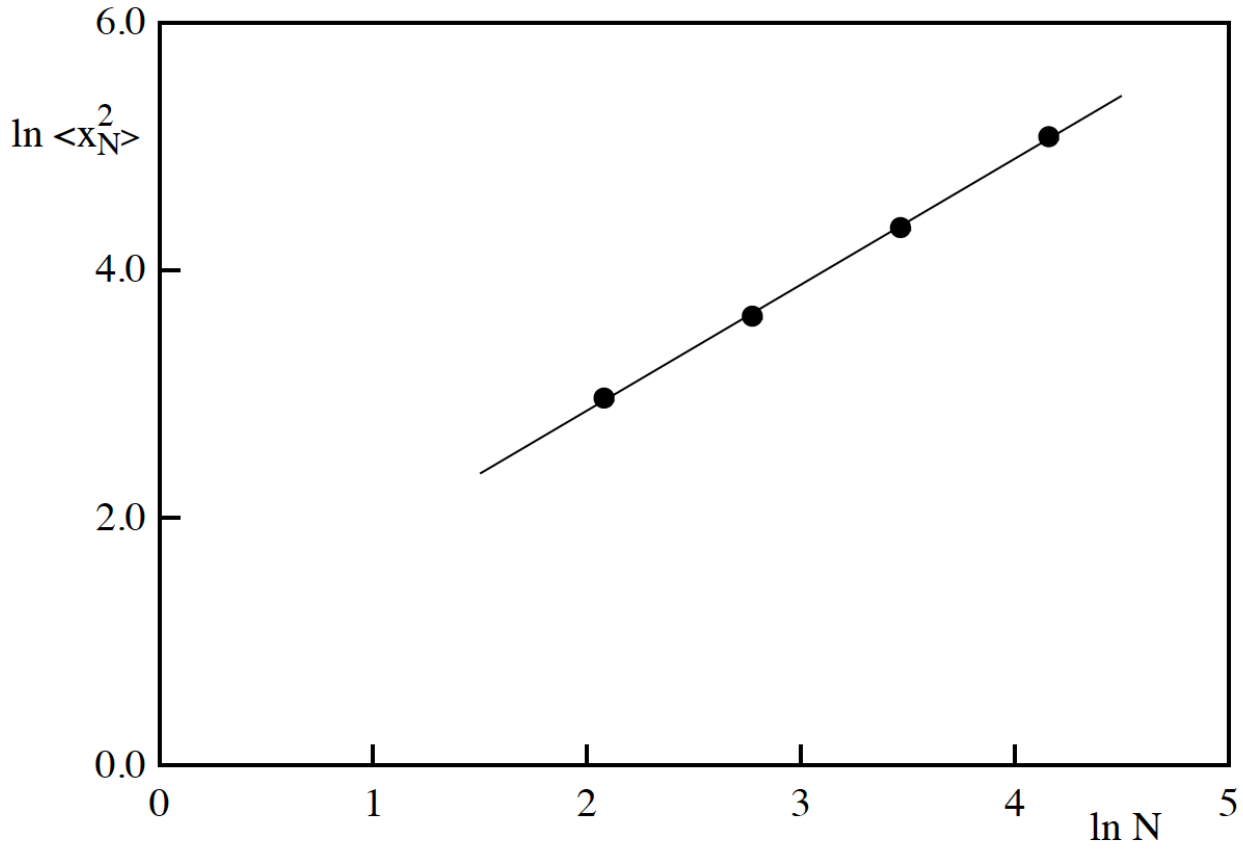



Figure 7.5: Plot of $\Delta x^2$ versus $\ln N$ for the data listed in Table 7.2. The straight line $y = 1.02x + 0.83$ through the points is found by minimizing the sum (7.33).

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes

# Pseudorandom number generator

From Wikipedia, the free encyclopedia

This article **has an unclear citation style**. The references used may be made clearer with a different or consistent style of citation, footnoting, or external linking. *(September 2009)*

A **pseudorandom number generator** (**PRNG**), also known as a **deterministic random bit generator** (**DRBG**),[1] is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values, called the PRNG's *seed* (which may include truly random values). Although sequences that are closer to truly random can be generated using hardware random number generators, *pseudorandom* number generators are important in practice for their speed in number generation and their reproducibility.[2]

# Random Number Sequences

The most widely used random number generator is based on the *linear congruential* method. One advantage of the linear congruential method is that it is very fast. For a given seed $x_0$, each number in the sequence is determined by the one-dimensional map

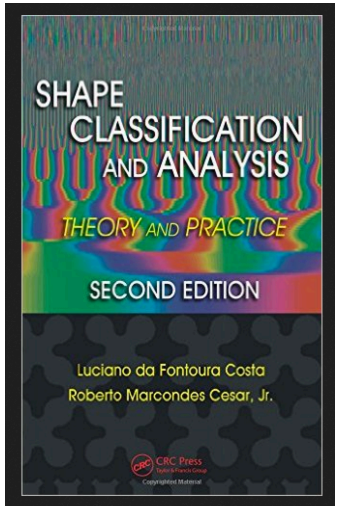
$$x_n = (ax_{n-1} + c) \bmod \mathrm{m}, \tag{7.58}$$

Another popular random number generator is the *generalized feedback shift register* method which uses bit manipulation (see Sections 15.1 and 15.6). Every integer is represented as a series of 1s and 0s called bits. These bits can be shuffled by using the bitwise `exclusive or` operator $\oplus$ (xor) defined by $a \oplus b = 1$ if the bits $a \neq b$; $a \oplus b = 0$ if $a = b$. The $n$th member of the sequence is given by

$$x_n = x_{n-p} \oplus x_{n-q}, \tag{7.59}$$

# Material de trabalho

- Leia o Capítulo 7 do livro texto.
- Resolva os exercícios desse capítulo.
- Material adicional:
- Capítulo 2:
- http://www.crcnetbase.com/isbn/9781420037555



Chapter 11

# NUMERICAL AND MONTE CARLO METHODS

# Numerical Integration Methods in One Dimension

$$F = \int_a^b f(x)\,dx.$$

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

$$x_n = x_0 + n\,\Delta x.$$

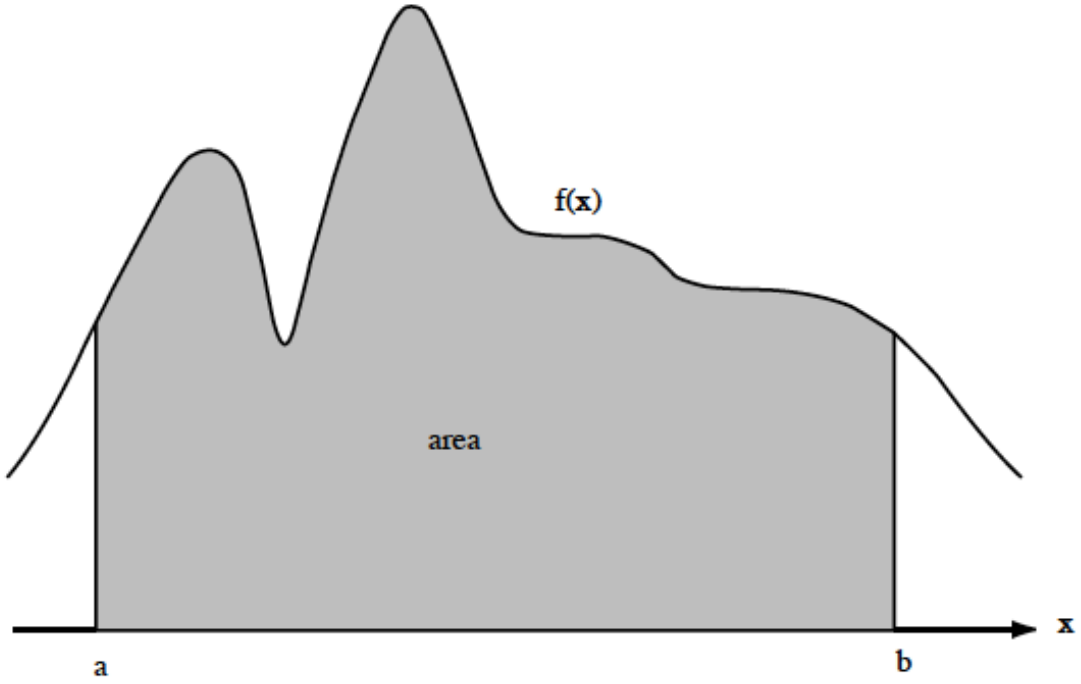



Figure 11.1: The integral $F$ equals the area under the curve $f(x)$.

$$F_n = \sum_{i=0}^{n-1} f(x_i)\Delta x. \qquad \text{(rectangular approximation)}$$

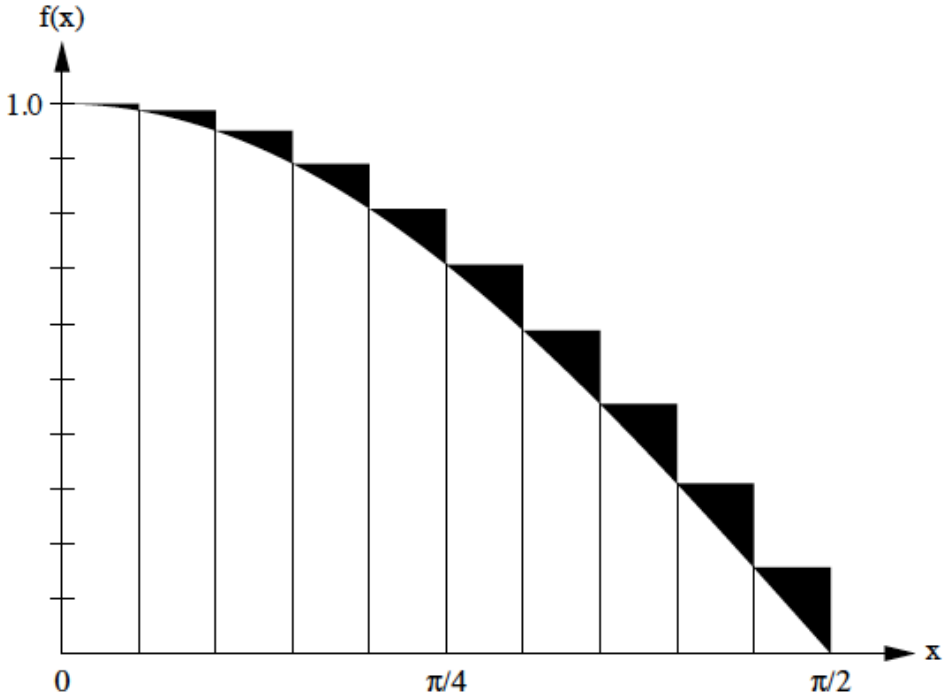



Figure 11.2: The rectangular approximation for $f(x) = \cos x$ for $0 \leq x \leq \pi/2$. The error is shaded. The error for various values of the number of intervals $n$ is given in Table 11.1.

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

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

$$F_0 = \frac{1}{3}(y_0 + 4y_1 + y_2)\Delta x,$$

$$F_n = \frac{1}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]\Delta x. \qquad \text{(Simpson's rule)}$$

# Using Differential Equations

$$F(x) = \int_a^x f(t)\,dt,$$

$$\frac{dF(x)}{dx} = f(x).$$

# Using Monte Carlo Evaluation

$$F_n = A\frac{n_s}{n}, \qquad \text{(hit or miss method)}$$

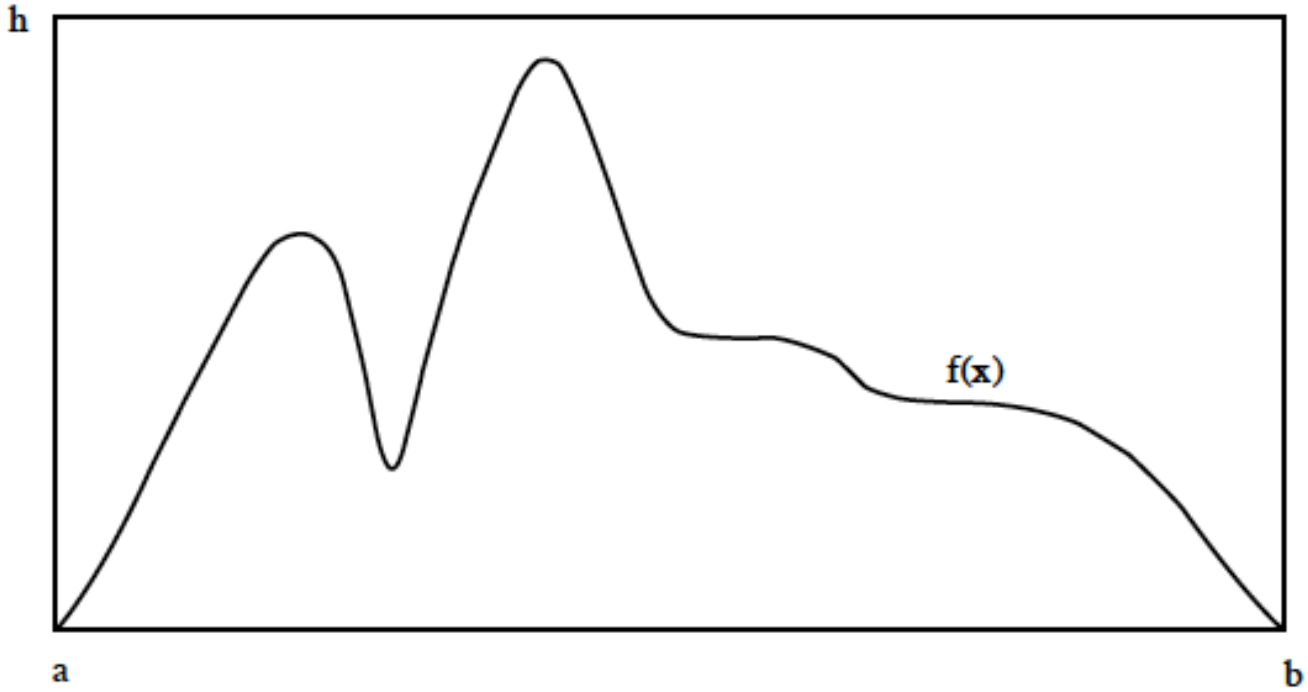



Figure 11.3: The function $f(x)$ is in the domain determined by the rectangle of height $H$ and width $(b-a)$.

$$F = \int_a^b f(x)dx = (b-a)\langle f \rangle.$$

$$F_n = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i) \approx (b-a)\langle f \rangle. \qquad \text{(sample mean method)}$$

# Material de trabalho

- Leia o Capítulo 11 do livro texto.
- Resolva os exercícios desse capítulo.
- Procure o monitor ou o professor para suas dúvidas.

# Probability problems

**Problem 7.21.** Three boxes: stick or switch?

**Problem 7.22.** Conditional probability

**Problem 7.23.** The roll of the dice

**Problem 7.25.** Money exchange

# Variational Methods

The speed of light in a medium can be expressed in terms of $c$, the speed of light in a vacuum, and the index of refraction $n$ of the medium:

$$v = \frac{c}{n}. \tag{7.62}$$

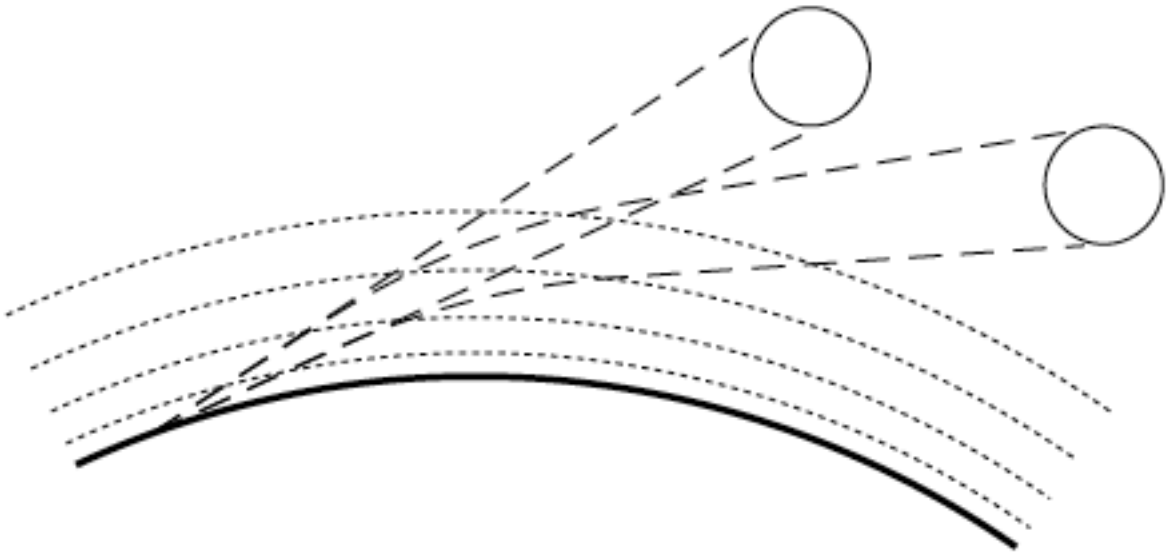


Figure 7.13: Near the horizon, the apparent (exaggerated) position of the sun is higher than the true position of the sun. Note that the light rays from the true sun are curved due to refraction.