

The Monte Carlo Method in Astrophysics *(Part 1)*

Luis A. Aguilar

aguilar@astrosen.unam.mx

Program

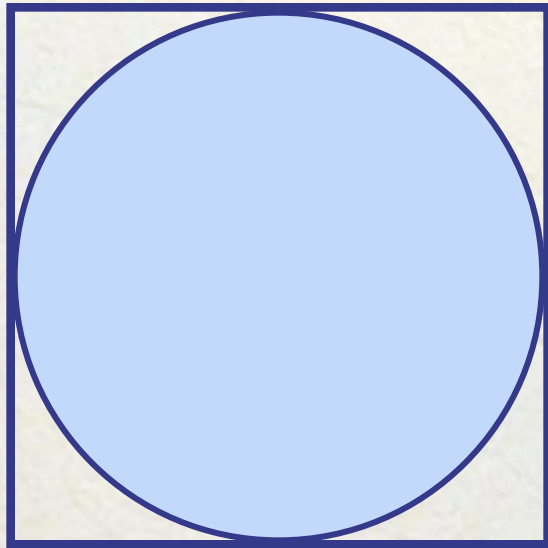
- What is the Monte Carlo Method?
- A bit of history
- Applications
- The core of Monte Carlo: Random realizations
- 1st example: Initial conditions for N-body simulations
- 2nd example: Simulating a proper motion survey

What is the Monte Carlo Method?

- The Monte Carlo Method encompasses a large number of algorithms whose defining characteristic is the use of random numbers.
- As such, these algorithms are non-deterministic, however, their use is not constrained to probabilistic models.
- They can be described as “statistical simulation methods”, because they provide solutions by performing statistical sampling.
- Due to its nature: repetitiveness and large number of iterations, the Monte Carlo method usually requires the use of computers.
- The Monte Carlo method is well suited to solve problems where a large number of parameters are involved, the complexity of the model makes a direct solution method unfeasible, or the elements involved are not well determined.

What is the Monte Carlo Method?

A simple example: Estimating the value of π

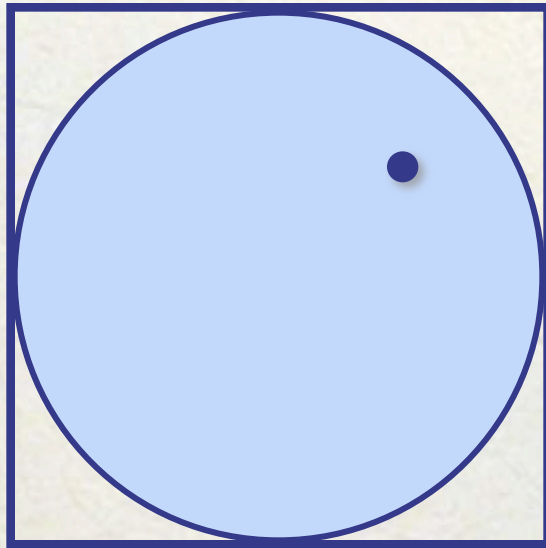


If we circumscribe a circle within a square, it is obvious that the ratio of their areas is proportional to π :

$$(\pi r^2 / 4r^2) = \pi / 4$$

What is the Monte Carlo Method?

A simple example: Estimating the value of π

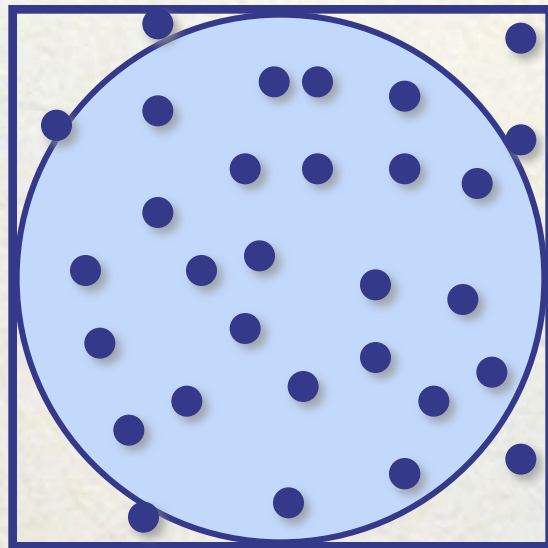


We can turn around this expression to say that, the probability that a randomly chosen point within the square will lie inside the circle, is equal to $\pi/4$.

$$\text{probability} = \pi / 4$$

What is the Monte Carlo Method?

A simple example: Estimating the value of π



This suggests a method to approximate the value of π using random numbers:

1. Reset 2 counters: $m=0$, $n=0$.
2. Increment 1st counter: $m = m+1$.
3. Generate 2 random numbers between 0 and 1: (x_i, y_i) .
4. Check if $(x_i)^2 + (y_i)^2 < 1$; If so, increment 2nd counter: $n = n+1$.
5. Compute ratio n/m as approximation to $\pi/4$.
6. Iterate from step 2 until desired precision is achieved.

Notice that we have used an aleatory algorithm to compute a definite value.

A bit of history

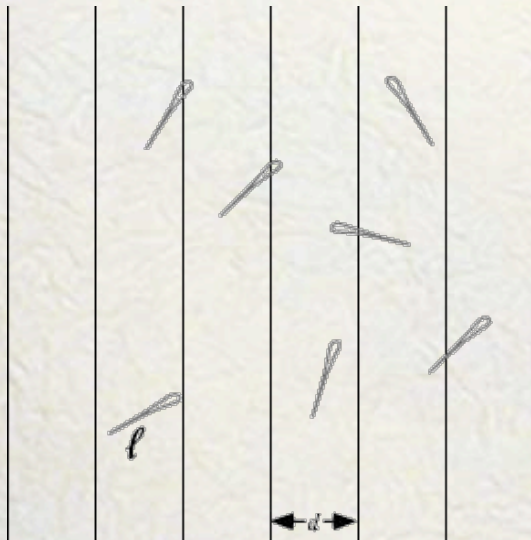
The seeds of the Monte Carlo method lie in the study of statistics and probability, as well as differential equations, in the XIX and early XX centuries.

- G.L. Comte Buffon's needle problem
- Lord Rayleigh study of parabolic differential equations
- W.S. Gosset validation of the t-distribution function.

Buffon's needle problem

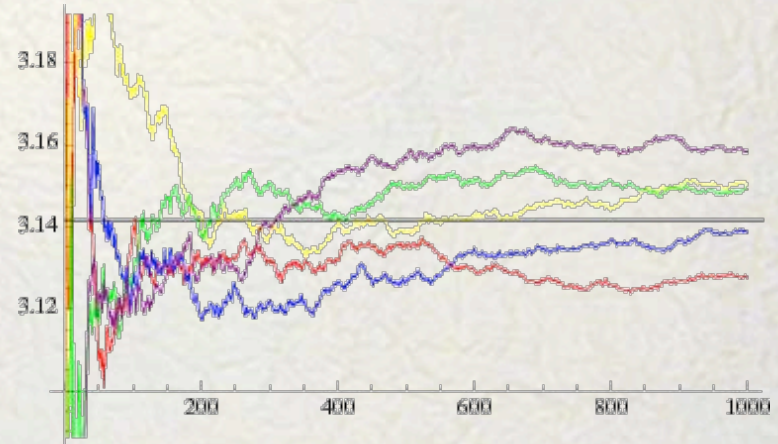
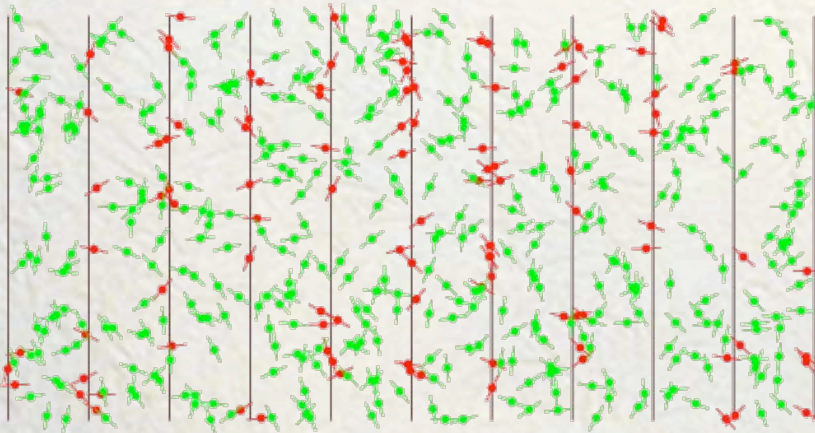
The French aristocrat and scientist Georges-Louis Leclerc, Comte de Buffon, proposed this problem in 1777.

The problem consists in finding the probability that a needle of a given length, will cross a line on a floor with a set of parallel lines at equal intervals.



Buffon's needle problem

This was one of the the first geometric probability problem to be solved.



Source: <http://mathworld.wolfram.com>

A bit of history

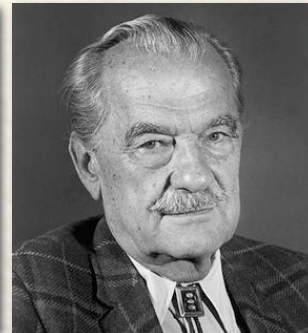
The Monte Carlo method first appears in the mid XX century with the work of Ulam, von Neumann and Metropolis.



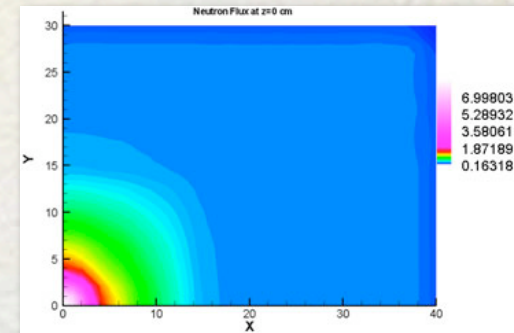
Stanislaw Ulam



John von Neumann



Nicholas Metropolis



Neutron diffusion calculation

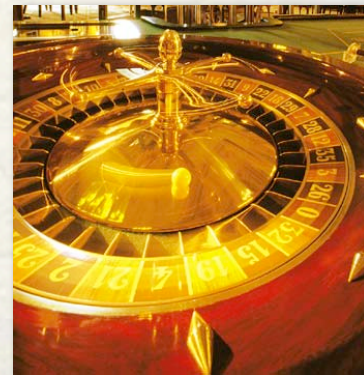
The first modern use was in the simulation of neutron diffusion, within the program to develop the hydrogen bomb at Los Alamos.

A bit of history

The method was named after a famous casino in Monaco.



Casino in Monte Carlo



Roulette

The method occurred to Ulam while he was convalescing and pondering probability problems related to the “solitaire” card game. Metropolis suggested the name, because of its aleatory and repetitive nature.

Applications

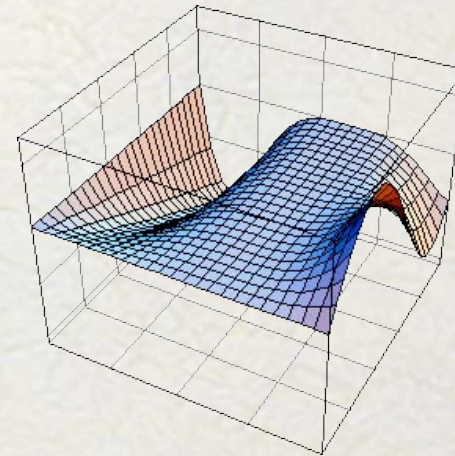
The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

- In **Mathematics** it is used to solve multi-dimensional integrals with complex boundary conditions

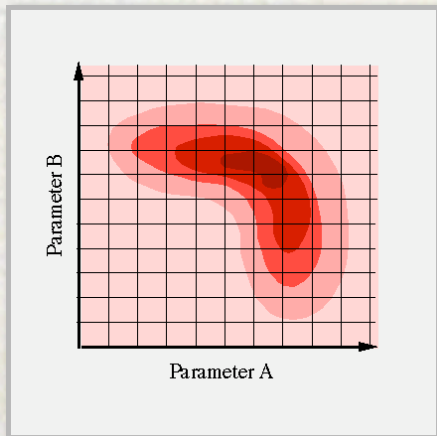
In an m -dimensional domain, the error in the approximation decreases as $n^{-1/2}$, where n is the number of sampling points used, rather than as $n^{-1/m}$, as do most other methods in the absence of exploitable features.



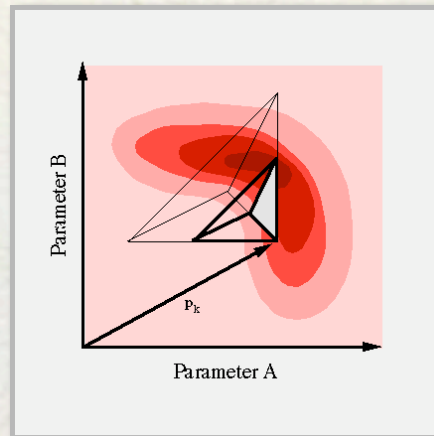
Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

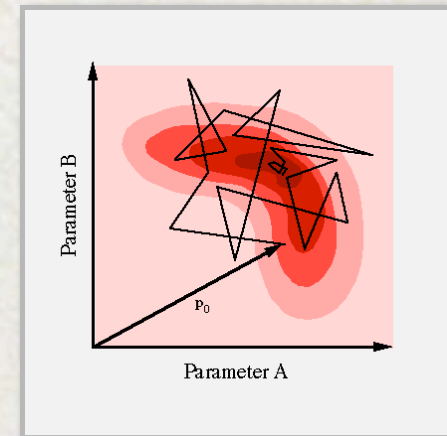
- In **Mathematics** it is also widely used to solve multi-dimensional optimization problems



Grid search



Simplex algorithm



Simulated Annealing

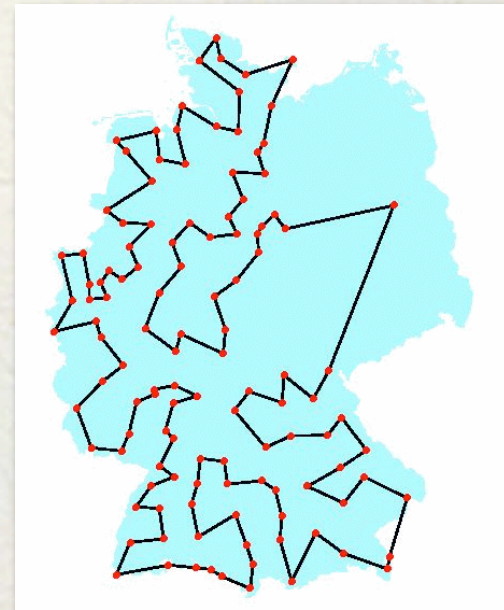
Source: esd.lbl.gov/iTOUGH2/Minimization

Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

- In **Statistics** there are problems that are only tractable with a Monte Carlo approach (*NP-hard*), like the “traveling salesman problem”

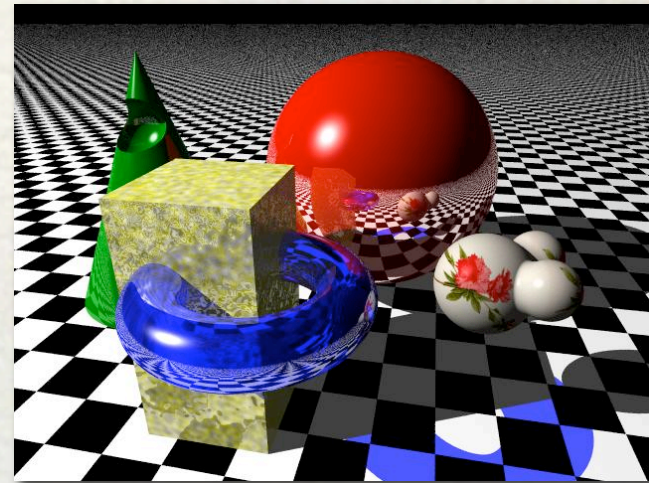
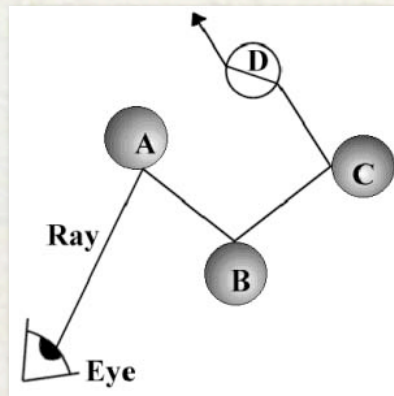
Given a number of cities and the costs of traveling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city?



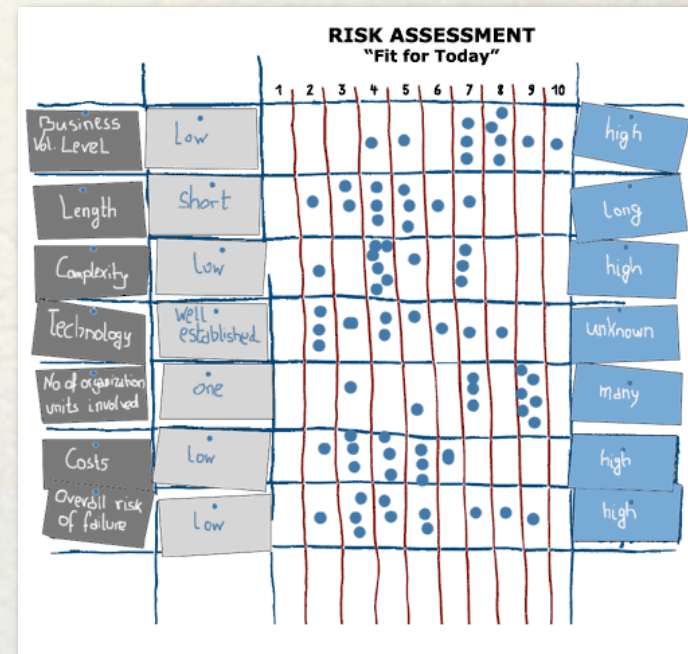
Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

- In **Computer Graphics** they are used in the “ray tracing” rendering technique



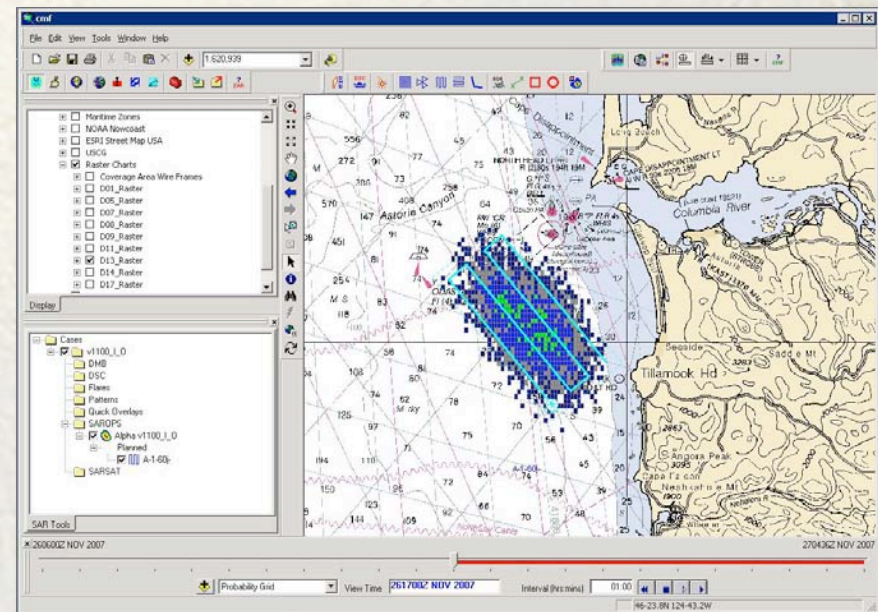
The idea is to trace back, from the pixels in the final image, the light rays back in time, taking into account multiple dispersions and reflections



Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

- A practical, and very useful application in **Risk Analysis**, is the program SAROPS, used by the U.S. Coast Guard to predict the likely position of debris and survivors of a maritime accident.



Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

- In **Astronomy** it has many applications. One example is modeling light scattering by dust grains.

Revista Mexicana de Astronomía y Astrofísica, **37**, 221–236 (2001)

AN EFFICIENT MONTE CARLO ALGORITHM FOR A RESTRICTED CLASS OF SCATTERING PROBLEMS IN RADIATION TRANSFER

A. M. Watson and W. J. Henney

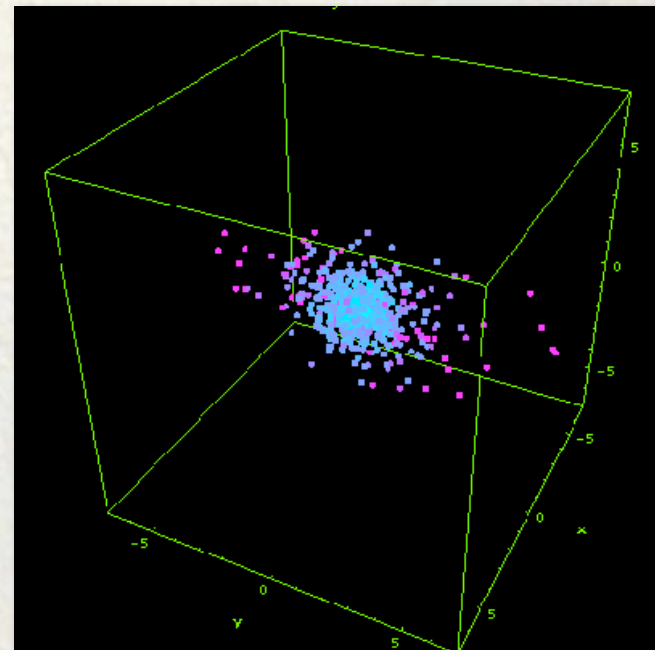
Instituto de Astronomía
Universidad Nacional Autónoma de México, Campus Morelia

Received 2001 July 2; accepted 2001 August 15

Applications

The Monte Carlo method is used in a very large number of fields, to solve a great diversity of problems.

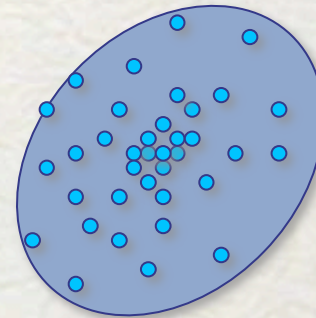
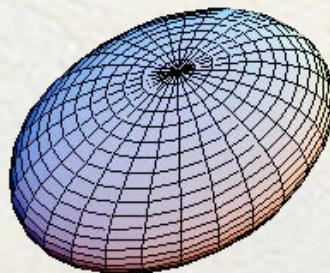
- Another astronomical application is generation of initial conditions for N-body simulations.



Random realizations

Our basic task is to build what is called a *random realization* of a given distribution function of one, or possibly, many variables.

For instance, we may want to create a list of values x_i for $i=1, N$, distributed according to a distribution function $f(x)$, or we may want to create a list of 3-D coordinates (x_i, y_i, z_i) , $i=1, N$, distributed according to some density function $\rho(x, y, z)$.



Random realizations

Now, **what exactly is meant by a random realization of a continuous model?** After all, a discrete collection of values is quite different from a continuous function.

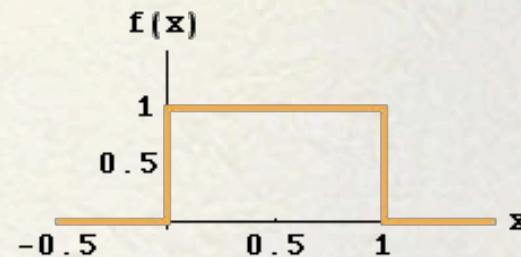
What we mean is that if we apply statistical tests on the discrete values, like grouping them together in bins to perform a χ^2 test, or compute the cumulative distribution to perform a *Kolmogorov-Smirnov* test, we will find that the discrete set of values is consistent, within the Poisson fluctuations expected for the size of the sample, with the continuous model.

We will refine this definition later on, for now this will suffice.

The constant probability function

Let us illustrate this with a very simple distribution: the constant probability distribution function:

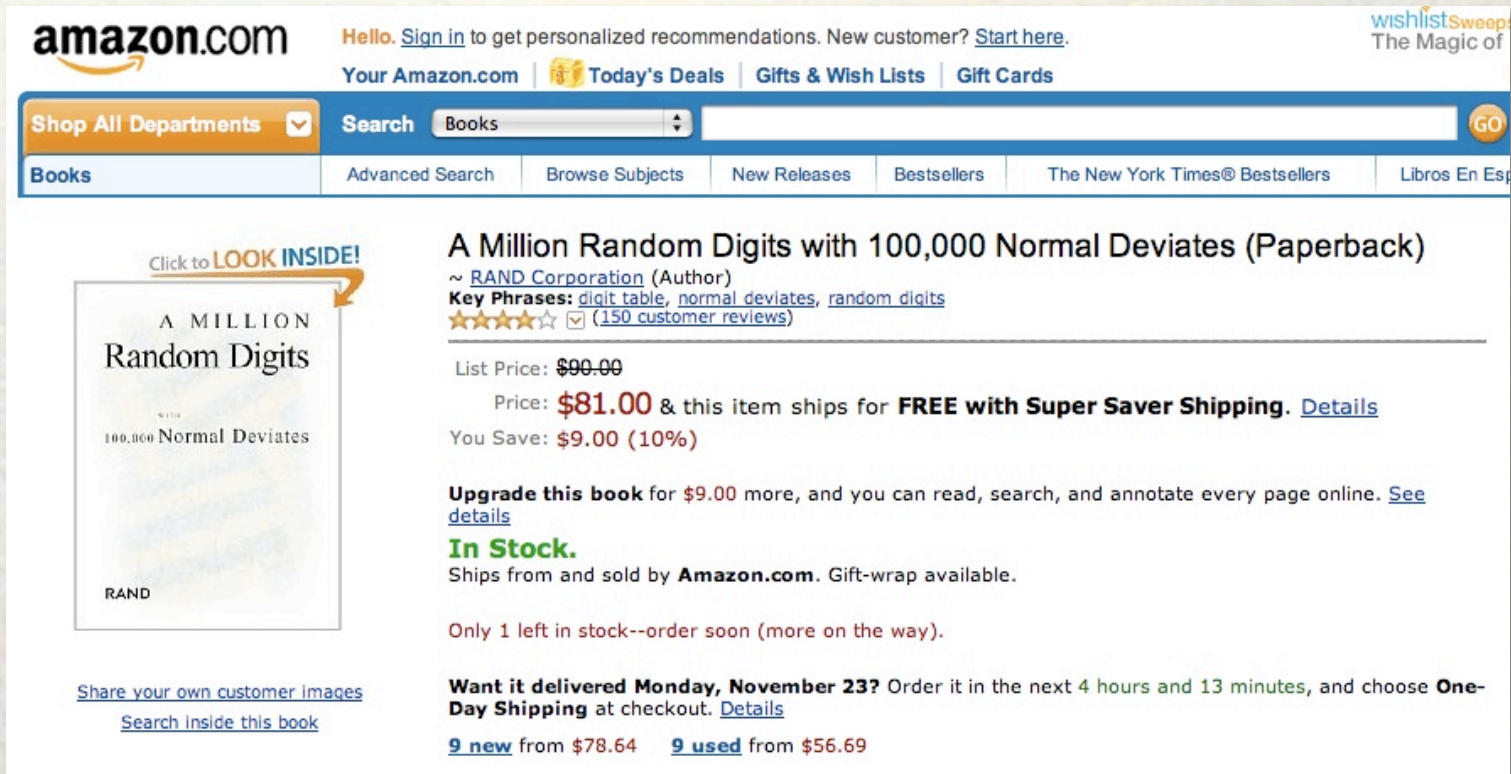
$$f(x) = \begin{cases} 1, & 0 < x < 1, \\ 0, & x < 0, \text{ or } x > 1 \end{cases}$$



It is clear that in this case a random, equal probability list of independent values between 0 and 1 will constitute a random realization of this, trivial, continuous function.

The constant probability function

But how do we get a list of random, independent values between 0 and 1?



The screenshot shows the Amazon.com interface for the book "A Million Random Digits with 100,000 Normal Deviates" by RAND Corporation. The page includes the Amazon logo, navigation links, and product details. The book is priced at \$81.00, down from \$90.00, and is available for free shipping with Super Saver Shipping. It has a 4.5-star rating from 150 customer reviews. The page also features a "LOOK INSIDE!" button and a "Share your own customer images" link.

amazon.com Hello. [Sign in](#) to get personalized recommendations. New customer? [Start here](#). [wishlistSweep](#) The Magic of

Your Amazon.com | [Today's Deals](#) | [Gifts & Wish Lists](#) | [Gift Cards](#)

[Shop All Departments](#) Search [GO](#)

[Books](#) [Advanced Search](#) [Browse Subjects](#) [New Releases](#) [Bestsellers](#) [The New York Times® Bestsellers](#) [Libros En Esp](#)

[Click to LOOK INSIDE!](#)

A MILLION Random Digits
100,000 Normal Deviates
RAND

A Million Random Digits with 100,000 Normal Deviates (Paperback)
~ [RAND Corporation](#) (Author)
Key Phrases: [digit table](#), [normal deviates](#), [random digits](#)
★★★★☆ ☒ (150 customer reviews)

List Price: ~~\$90.00~~
Price: **\$81.00** & this item ships for **FREE with Super Saver Shipping**. [Details](#)
You Save: **\$9.00 (10%)**

Upgrade this book for \$9.00 more, and you can read, search, and annotate every page online. [See details](#)

In Stock.
Ships from and sold by **Amazon.com**. Gift-wrap available.

Only 1 left in stock--order soon (more on the way).

Want it delivered Monday, November 23? Order it in the next 4 hours and 13 minutes, and choose **One-Day Shipping** at checkout. [Details](#)

[9 new](#) from \$78.64 [9 used](#) from \$56.69

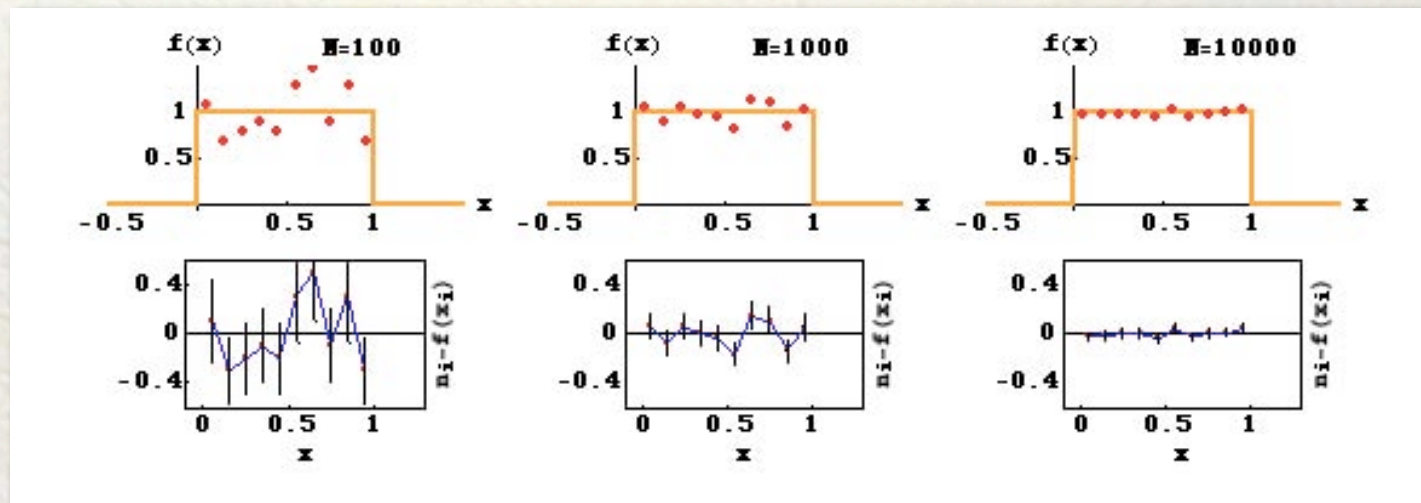
[Share your own customer images](#)
[Search inside this book](#)

Not precisely a bestseller!

The constant probability function

Most computer languages have an intrinsic function with a name like “*ran*” that returns a random real number between 0 and 1:

$$x_i = \text{ran}()$$

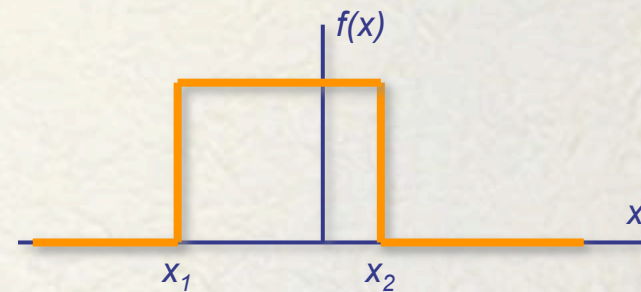


As we can see, the size of the fluctuations decreases as the size of the random realization increases, in proportion to the expected poissonian fluctuations. So, all these samples are random realizations of a constant probability function defined between 0 and 1.

The constant probability function

Now, what do we do for a constant probability distribution function defined between two arbitrary points?, say,

$$f(x) = \begin{cases} 1, & x_1 < x < x_2, \\ 0, & x < x_1, \text{ or } x > x_2 \end{cases}$$



In this case, it is obvious that the simple transformation:

$$x_i = [(x_2 - x_1) \times \text{ran}(\)] - x_1$$

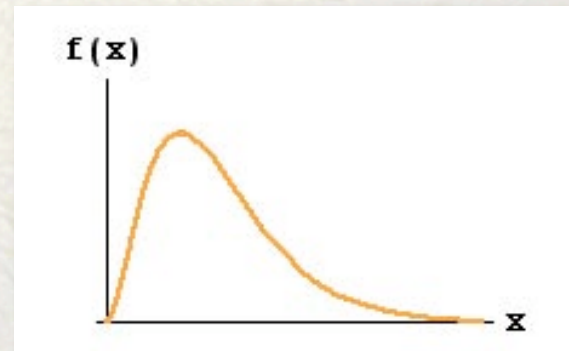
will do the trick, as it maps the unit interval onto the desired interval.

1-Dimensional case

Let us now turn to arbitrary 1-dimensional distribution functions.

There are two basic conditions that a good distribution function must meet: it must be positive everywhere and its integral over the entire range of the independent variable must be finite, and in fact, we will assume that the function has been normalized to unit area, so we can interpret it as a **probability density distribution function**.

$$f(x) > 0, \quad \text{and} \quad \int_{-\infty}^{\infty} f(x') dx' = 1.$$

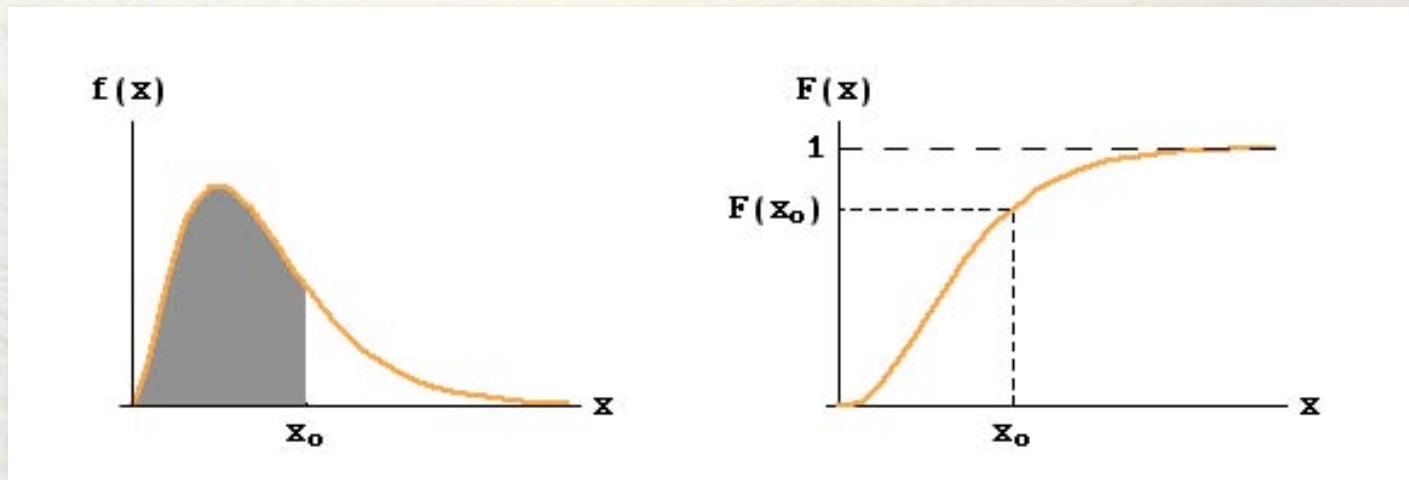


The question is, how to generate a random realization of this arbitrary function?

1-Dimensional case

Before addressing this problem, let us define the **cumulative distribution function** $F(x)$ as,

$$F(x) \equiv \int_{-\infty}^x f(x') dx'$$



Since we assume that $f(x)$ has been normalized to unit area, it is clear that:

$$\lim_{x \rightarrow -\infty} F(x) = 0, \quad \text{and} \quad \lim_{x \rightarrow \infty} F(x) = 1.$$

It is also clear that $F(x)$ is a monotonically increasing function, since $f(x)$ is positive.

1-Dimensional case

Let us look at a particular interval at $x = x_i$ of length dx . The expected number of points within this interval in a random realization of size N of this distribution function is:

$$n_i = N f(x_i) dx,$$

Now, from its definition, it is clear that $f(x)$ is the derivative of its cumulative function: $f(x) = (dF/dx)$.

We can then write the expected number of points as,

$$n_i = N dF.$$

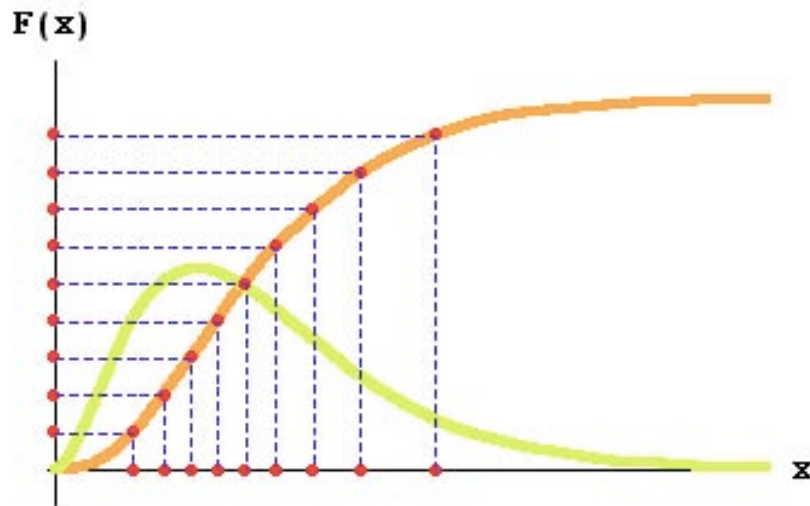
Let us now compare the previous two expressions for n_i .

In the first one, the expected number of points varies as x changes, even if we keep the size of the dx interval fixed, through the function $f(x)$.

In the second expression, however, the variation has been absorbed by dF , so if we fix the length of the interval dF , **the expected number of points is constant!**

1-Dimensional case

And this is the trick, we should create a random realization of a uniform distribution in the F variable and then project back onto x by using the inverse function $F^{-1}(x)$:



Notice how a uniform distribution of points on F maps onto a varying distribution in x that follows the $f(x)$ distribution function: Where $f(x)$ has a large value, F climbs fast and a given interval dF maps to narrow intervals in x . Where $f(x)$ has a small value, F flattens out and the same interval dF now stretches to a large interval in x .

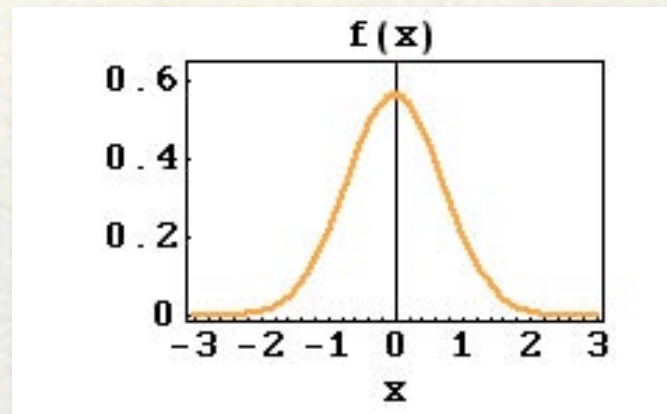
So the general recipe to create a random realization of an arbitrary distribution function of one variable $f(x)$ is:

$$x_i = F^{-1}[\text{ran}(\)]$$

A 1-dimensional example

Let's work a specific example:

$$f(x) = (1 / \sqrt{\pi}) \exp(-x^2)$$

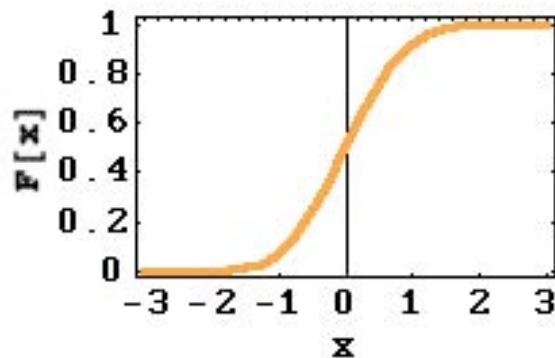


This is an even, normalized function, whose maximum is at the origin and quickly decreases at both sides.

A 1-dimensional example

The first step is to find the cumulative distribution function:

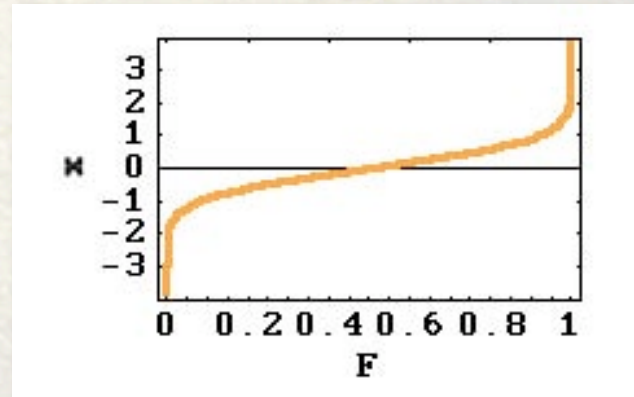
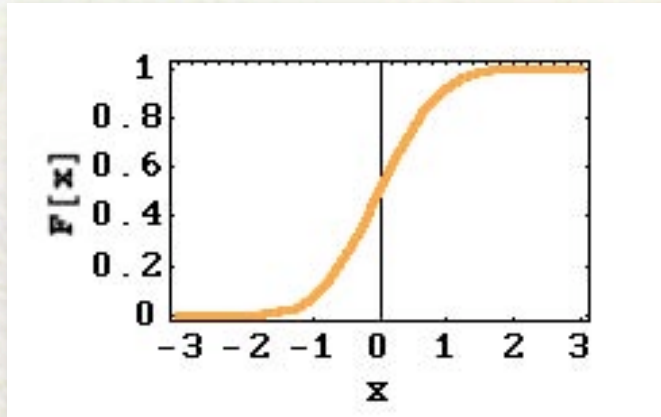
$$F(x) = \int_{-\infty}^x f(x') dx' = (1 / \sqrt{\pi}) \int_{-\infty}^x \exp(-x'^2) dx' = (1 / 2)[1 + \text{Erf}(x)]$$



A 1-dimensional example

The first step is to find the cumulative distribution function:

$$F(x) = \int_{-\infty}^x f(x') dx' = (1 / \sqrt{\pi}) \int_{-\infty}^x \exp(-x'^2) dx' = (1 / 2)[1 + \text{Erf}(x)]$$



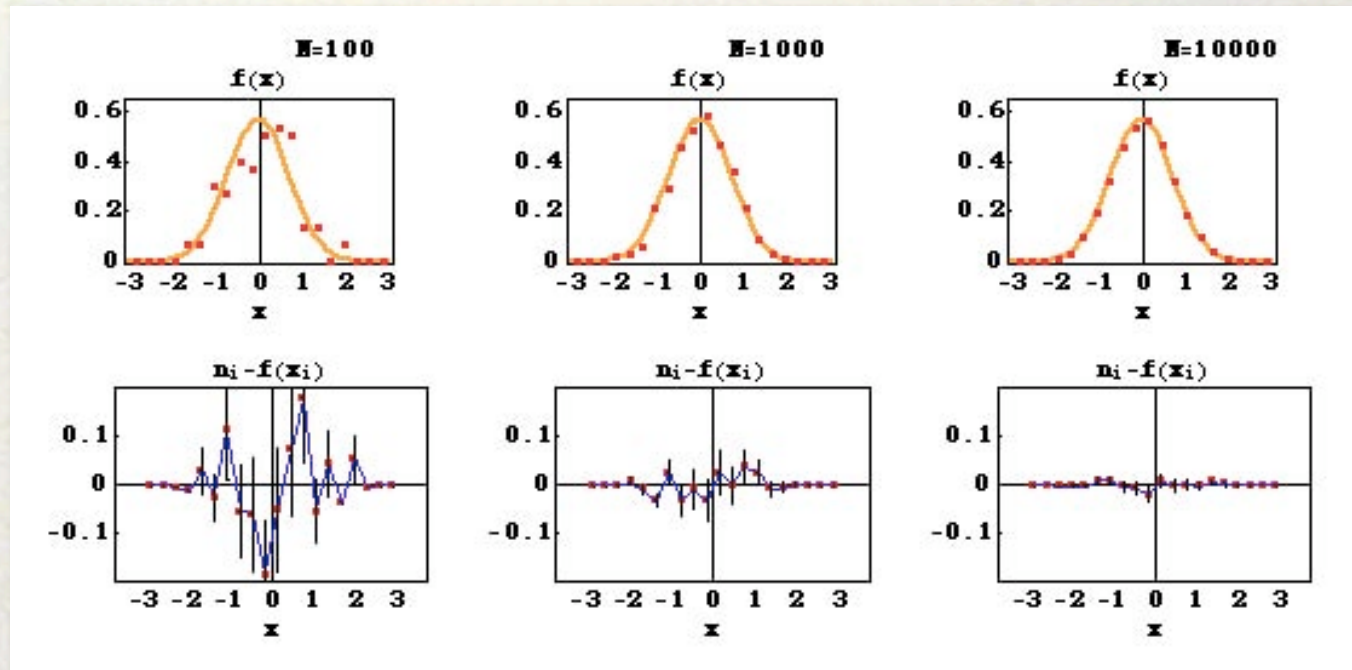
The next step is to find the inverse of the cumulative distribution function:

$$x = \text{Erf}^{-1}(2F - 1)$$

A 1-dimensional example

We can now use F^{-1} to generate random realizations of the function we are using as our example:

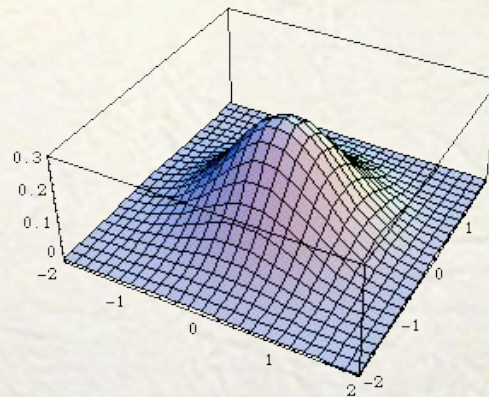
$$x_i = \text{Erf}^{-1}[2 \times \text{ran}() - 1]$$



2-Dimensional case

We begin with a simple 2-dimensional example:

$$\Sigma(x,y) = (1/\pi)\exp[-(x^2 + y^2)]$$



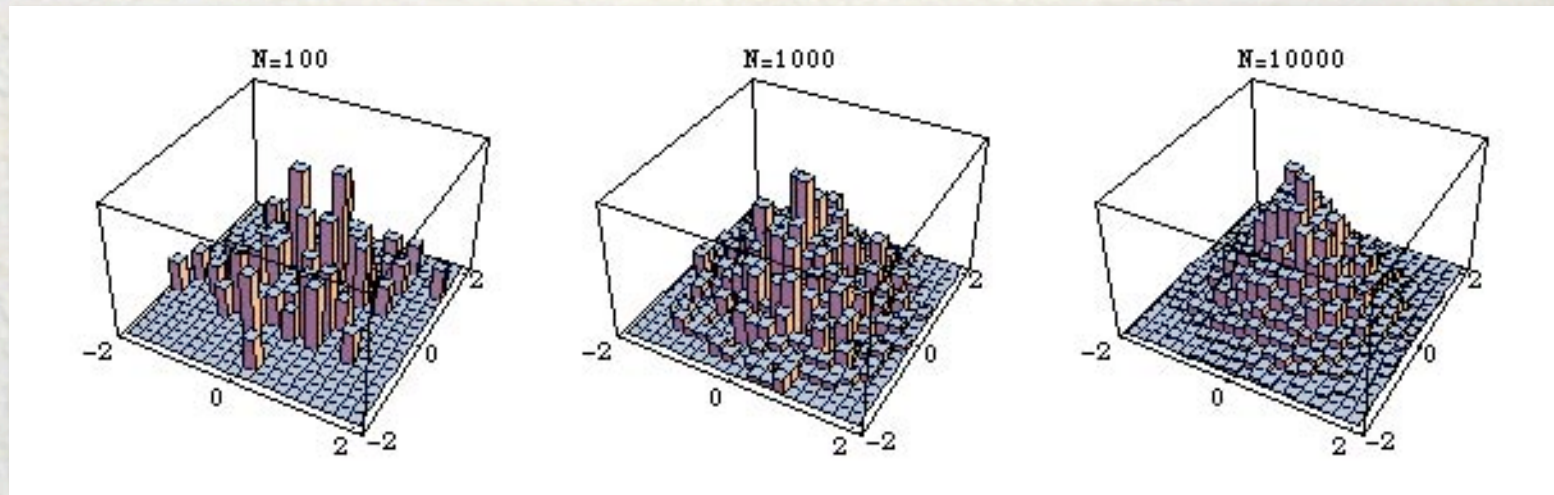
We notice that this function can be separated as the product of two functions, one for each variable, with no cross term:

$$\Sigma(x,y) = (1/\sqrt{\pi})\exp(-x^2) \times (1/\sqrt{\pi})\exp(-y^2)$$

2-Dimensional case

This means that each variable is independent of the other and so, we can reduce this problem to two, 1-dimensional problems, and use the methods of last section.

$$(x_i, y_i) = \left(\text{Erf}^{-1}[2 \times \text{ran}() - 1], \text{Erf}^{-1}[2 \times \text{ran}() - 1] \right)$$



Changing coordinates

Some times it is more convenient to use coordinates other than Cartesian to generate the random realization.

For instance, in the previous example, the 1-dimensional cumulative function involves the inverse of the error function, which has to be approximated numerically.

Although there are very efficient routines to compute it, it is better to avoid it, because as the size of the random realization increases, its repeated evaluation will slow down the computations.

It is better to look for a more efficient approach.

Changing coordinates

We will use polar coordinates (R, θ) to simplify the previous example.

Now, since **the expected number of points in a given region of the domain has to be the same, regardless of the coordinate system used**, it is clear that:

$$\Sigma(x,y)dxdy = \Sigma(R,\theta)RdRd\theta = \Sigma'(R,\theta)dRd\theta$$

where $RdRd\theta$ is the differential element of area that corresponds to $dxdy$. Notice that Σ' has absorbed the extra R .

The distribution function in polar coordinates is then,

$$\Sigma'(R,\theta) = (R / \pi) \exp(-R^2).$$

Since this function has no dependency on θ , it is convenient to integrate it azimuthally:

$$\Sigma^*(R) = \int_0^{2\pi} \Sigma'(R,\theta)d\theta = 2R \exp(-R^2).$$

Changing coordinates

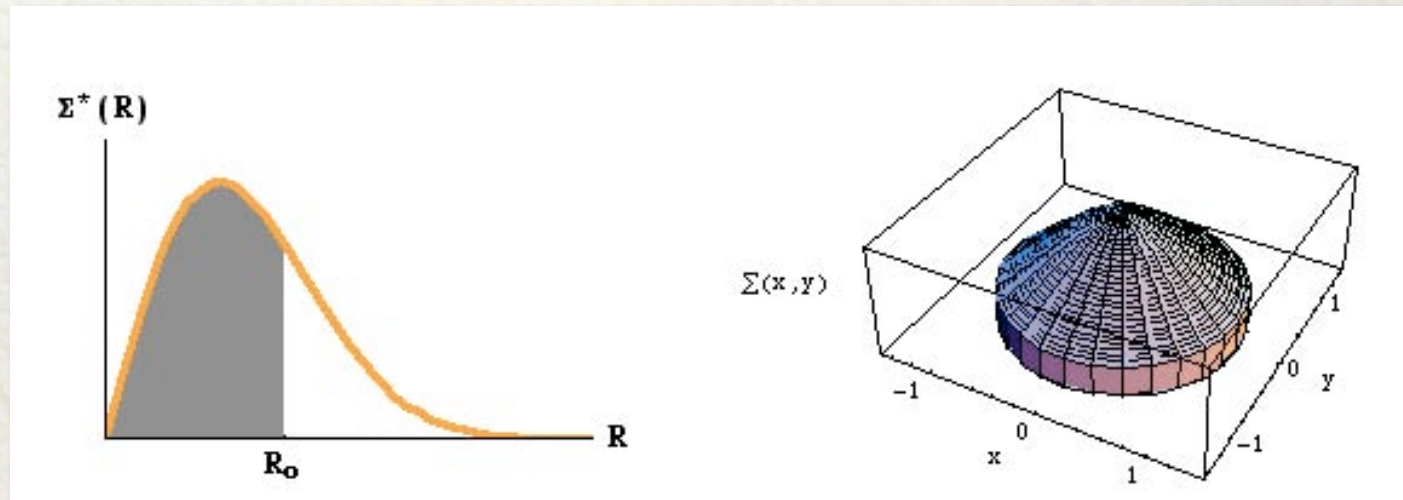
It is important to realize that $\Sigma'(R, \theta)$ and $\Sigma^*(R)$ are not the same, the former represents probability per unit area $dRd\theta$, while the latter represents probability per unit annulus of radius R and width dR . They are quite different.

So, we have turned the problem of creating a random realization of a 2-dimensional function, into two 1-dimensional problems in polar coordinates: we must create a random realization of $\Sigma^*(R)$, and since the distribution function has no azimuthal dependence, for θ it will be a matter of drawing random values between 0 and 2π .

Changing coordinates

First, we find the cumulative distribution function for $\Sigma^*(R)$:

$$Z(R) = \int_0^R \Sigma^*(R') dR' = 2 \int_0^R R' \exp(-R'^2) dR' = 1 - \exp(-R^2).$$



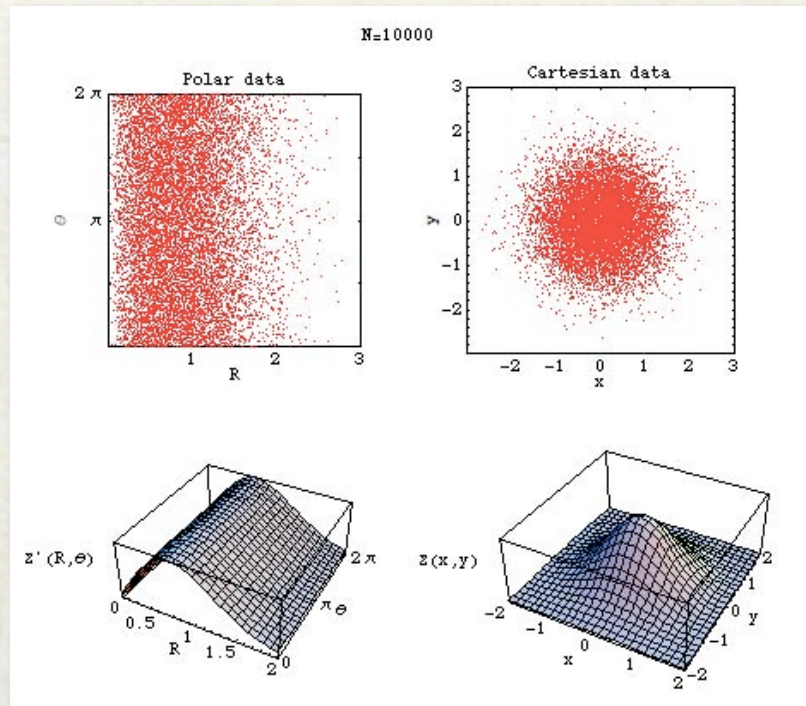
Notice that $Z(R)$ represents the fraction of the sample contained within a circle of radius R . It is normalized, as it should be, since we have obtained it ensuring that the probabilities per unit area are conserved.

Changing coordinates

The inverse function is simply: $R = \sqrt{-\ln(1 - Z)}$.

We can now create a random realization of this function as follows:

$$(R_i, \theta_i) = \left(\sqrt{-\ln[1 - \text{ran}(\)]}, 2\pi \times \text{ran}(\) \right)$$



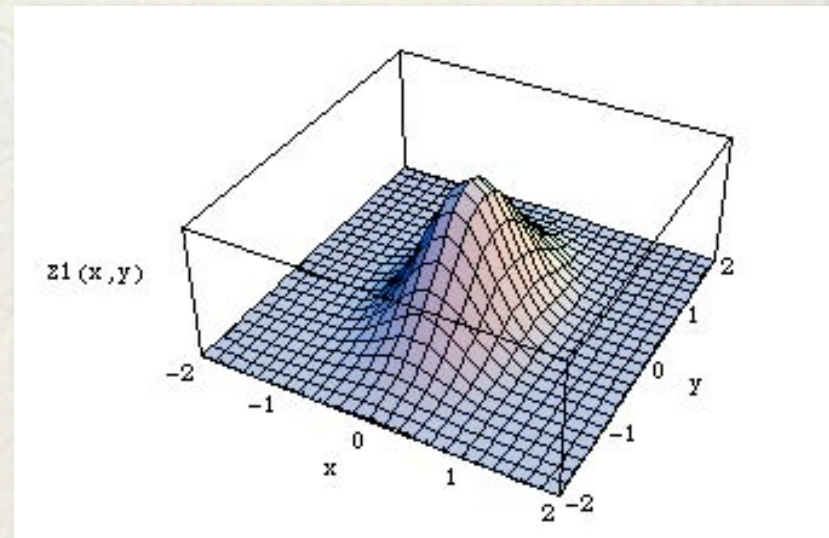
Notice that, in this particular example, by going to polar coordinates, we have avoided the inverse error function, and ended up with a faster algorithm.

Changing variables

Our previous example had azimuthal symmetry, and so it was obvious that polar coordinates would simplify the problem, as the resulting distribution function would be the product of two independent functions, and the one for the azimuth would be trivial.

What happens if this symmetry is lost by a simple scaling along the principal axes of the distribution?

$$\Sigma_1(x,y) = \left(\sqrt{\alpha\beta} / \pi \right) \exp[-(\alpha x^2 + \beta y^2)]$$



Changing variables

It is clear that a simple change of variables can result in azimuthal symmetry:

$$s = \sqrt{\alpha} x, \quad t = \sqrt{\beta} y, \quad \Rightarrow \quad \exp[-(\alpha x^2 + \beta y^2)] = \exp[-(s^2 + t^2)]$$

In order to transform the distribution function to the new coordinates, we must remember the basic principle that the expected fraction of a random realization, on equivalent domain regions in the old and new variables, is the same:

$$\Sigma_1(x, y) dx dy = \Sigma_1(s, t) \frac{ds dt}{\sqrt{\alpha \beta}} = \Sigma'_1(s, t) ds dt$$

The distribution function in the new variables is then,

$$\Sigma'_1(s, t) = (1 / \pi) \exp[-(s^2 + t^2)]$$

which is exactly our previous example.

Changing variables

Distribution functions can be transformed changing variables, or using different coordinate systems. But in doing so, we should always use the basic principle that, **the fraction of the sample on equivalent domain regions must be the same.**

Homework

1st Assignment

- Write a program that approximates the value of π using the method described in the lecture.
- Make a plot of the fractional error as a function of the number of iterations.
- Determine the dependency of the fractional error on the number of iterations.

End of first lecture