Instituto de Astronomía UNAM

The Monte Carlo Method in Astrophysics

(Part 5)

Luis A. Aguilar

aguilar@astrosen.unam.mx

What is a random realization?

Up to now, we have been talking about random realizations, but what exactly is a random realization?

In the first lecture, we gave a definition in terms of statistical tests, however, we did not explain in detail what we meant by this.

So, what exactly do we mean by random realizations?

The clue is in the second part of the definition we gave: "consistent within the Poisson fluctuations expected for the size of the sample".
To understand this in detail, we must talk about the Poisson probability distribution function.

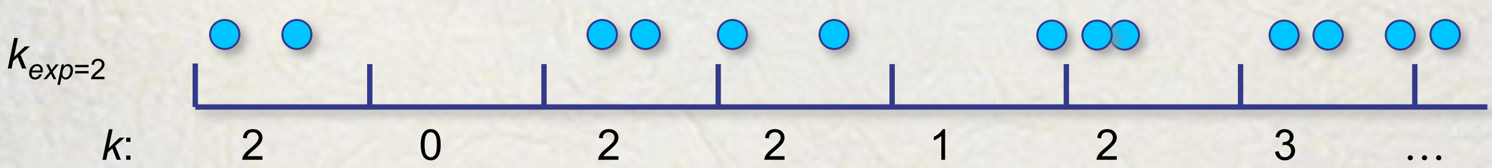
What is a random realization?

The *Poisson probability distribution function*, or Poisson function for short, was introduced by Siméon-Denis Poisson in 1838.

This function gives the probability of a number of events occurring within a fixed interval, if these events occur with a known average number per unit interval and the events are independent of each other.

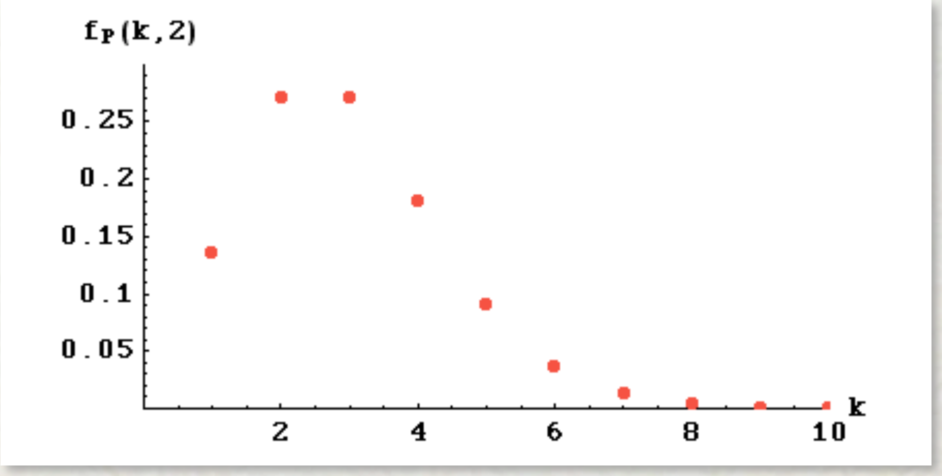
Specifically, the probability of finding exactly k events ($k \geq 0$) during an interval where the expected number of events is k_{exp}, is:

$$f_P(k;k_{\mathrm{exp}}) = \frac{k_{\mathrm{exp}}^k}{k!}\exp(-k_{\mathrm{exp}})$$



What is a random realization?

$$f_P(k;\, k_{\exp}) = \frac{k_{\exp}^{k}}{k!}\exp(-k_{\exp})$$



The mean and variance of this distribution function are: $\langle k \rangle = k_{exp}$, $\sigma^2 = k_{exp}$.

Instituto de Astronomía UNAM

What is a random realization?

In all the procedures to build random realizations that we described before, the randomness comes from the ***ran*( )** function that gives values between *0* and *1*, as equal probability, independent events.

As such, the characteristics of the resulting distribution of values are described by the Poisson distribution function: **independent draws with fixed probability**.

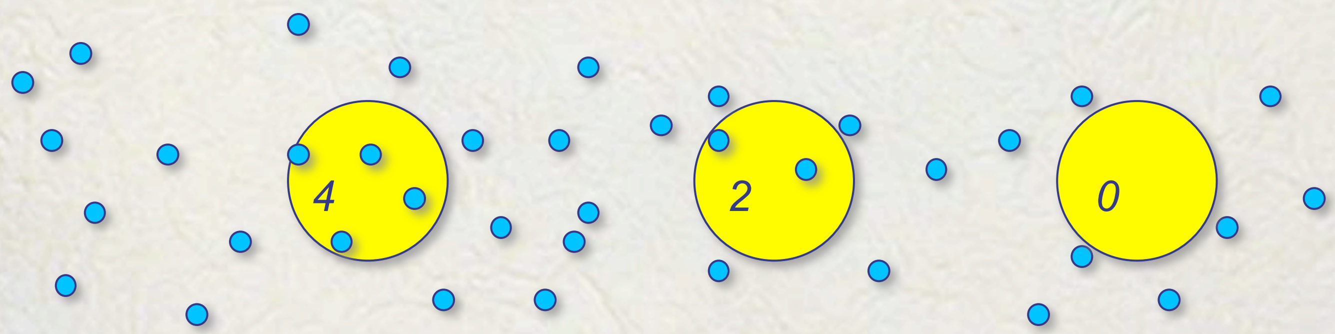
What is a random realization?

To see what exactly we mean by this, let us cast the definition of this function in terms appropriate to the problem of random realizations of a 3D model:

The probability of finding *N*, and only *N* points within a spherical region of radius *r* (our 3D interval), is,

$$f_P(k;\,k_{\mathrm{exp}}) = \frac{k_{\mathrm{exp}}^{k}}{k!}\exp(-k_{\mathrm{exp}}) \quad \Longrightarrow \quad f(N;\,r) = \alpha^{N}\frac{r^{3N}}{N!}\exp(-\alpha r^{3})$$

where the expected number of particles within the region is given by: $N_{exp} = (4/3)\,\pi r^3 \times n = \alpha\, r^3$, n is the number density of points and $\alpha \equiv (4/3)\,\pi n$, can be interpreted as the expected number of points within the unit sphere.



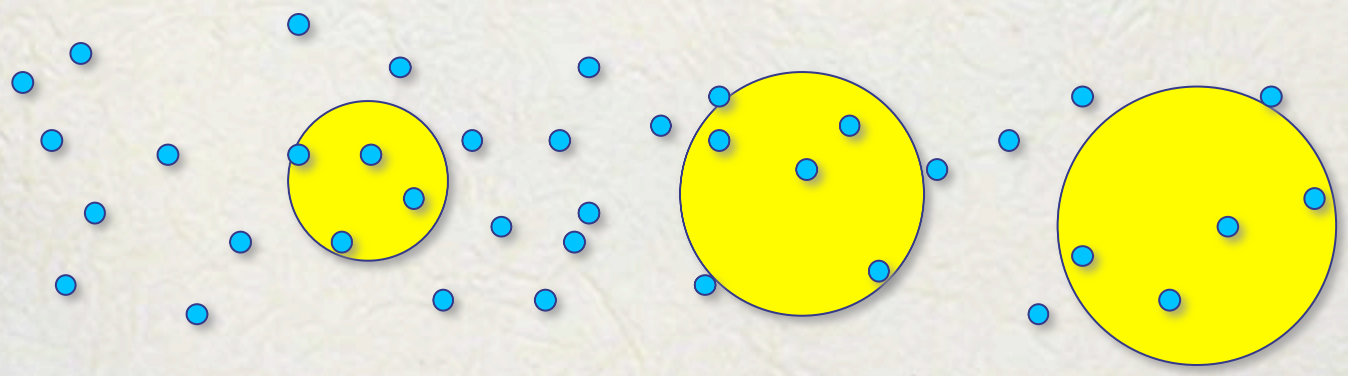
What is a random realization?

We can turn around this function and interpret it as a probability distribution function for *r*, in this case, we get the probability of getting exactly *N* points as a function of the radius of the spherical region being considered:

$$f_N(r) = 3\frac{\alpha^{N+1/3}}{\Gamma(N+1/3)}\, r^{3N} \exp(-\alpha r^3)$$

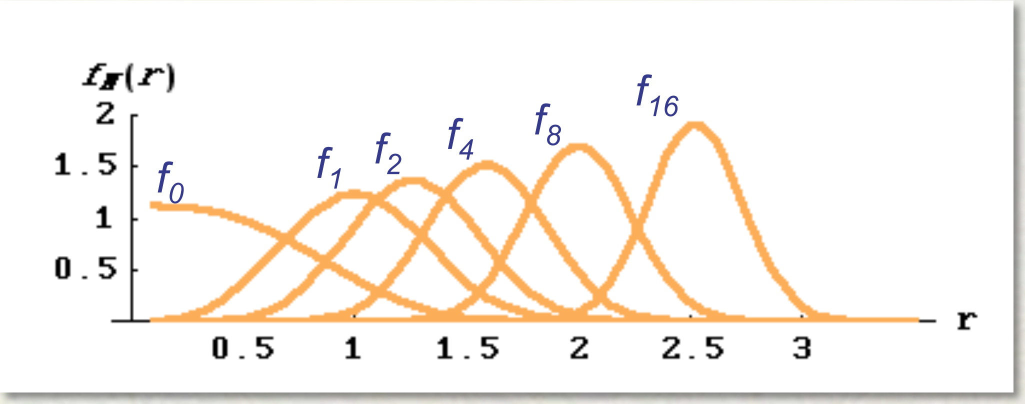
which has been properly normalized as a function of *r* : $\int_0^{\infty} f(N;r)dr = \frac{\Gamma(N+1/3)}{3N!\alpha^{1/3}}$

$\Gamma(x)$ is the Gamma, or factorial function.



What is a random realization?

We show below the graph of f_0, f_1, f_2, f_4, f_8 and f_{16}, which represent the probability of finding none, one, two, etc., particles within a sphere of radius *r*, respectively.



We can see that, as *N* increases, the value of *r* is more tightly constrained.

What is a random realization?

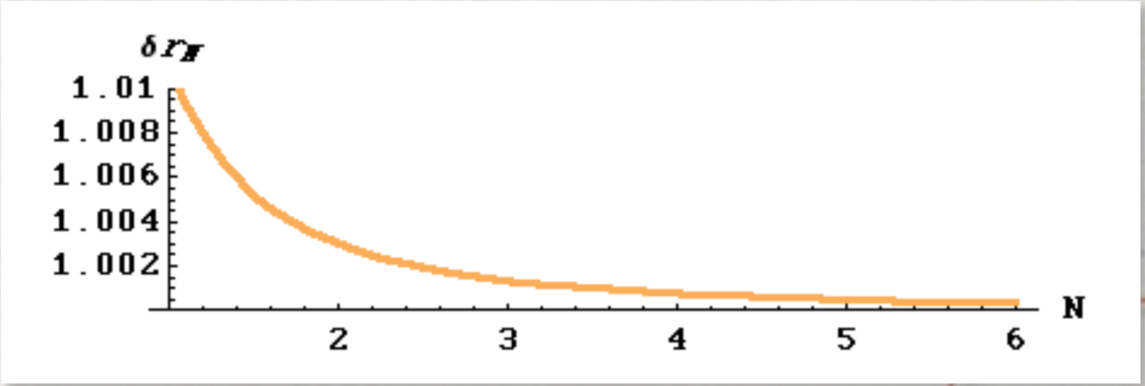
We can compute the mean of these distributions $\langle r_N \rangle$, which represents the mean radius of a sphere that contains exactly *N* points:

$$\langle r_N \rangle = \int_0^\infty r\, f(N; r)\, dr = \frac{\Gamma(N + 2/3)}{\Gamma(N + 1/3)\alpha^{1/3}}$$

It is interesting to compare this expression with the the radius of the sphere that contains exactly *N* points:

$$r_N = (N/\alpha)^{1/3}$$

The plot shows the fractional difference δr_N as a function of *N*:



What is a random realization?

So, this is what a random realization of a uniform probability distribution function is.

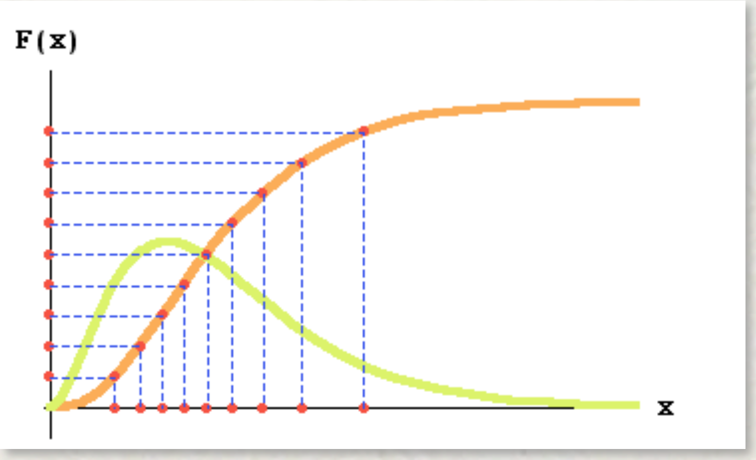
In the case of a model where the density is not uniform, we can shape the basic uniform random realization into whatever is needed, using the methods seen before.

Non-random realizations

Having discussed at length what a random realization is, we now turn to non-random realizations.

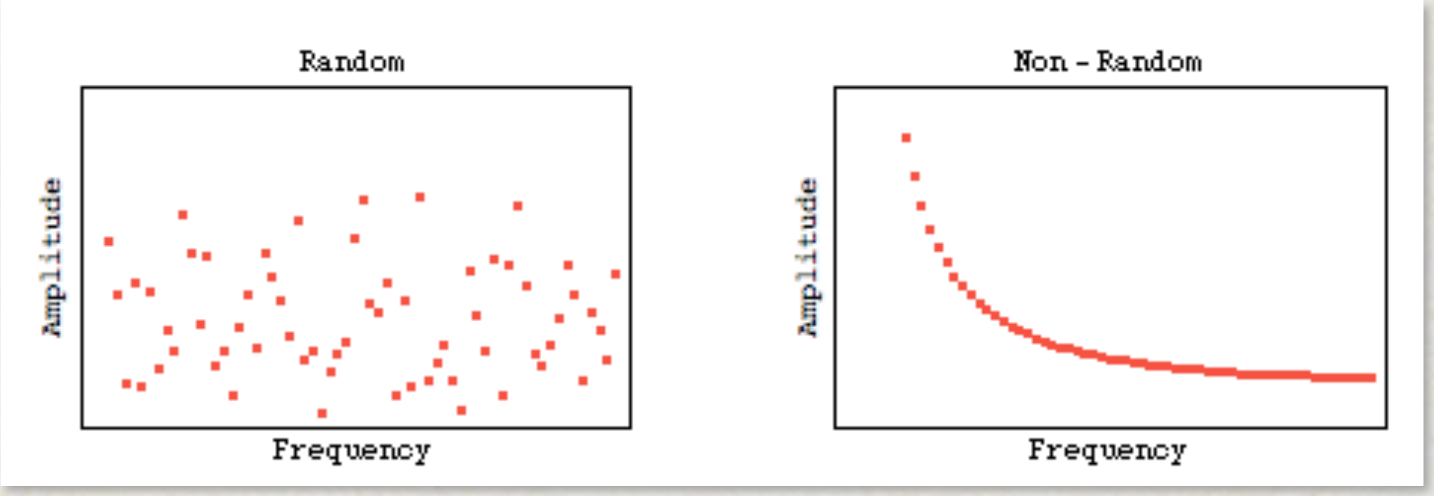
The easiest one to describe is what we will call a *grid realization*. In the case of a 1-dimensional, uniform distribution function, this is just N points equally spaced at intervals $\delta x = (x_{Max}-x_{Min})/N$.

Using the method of the cumulative distribution function, we can also generate a non-random realization of any arbitrary 1-dimensional distribution function.



Non-random realizations

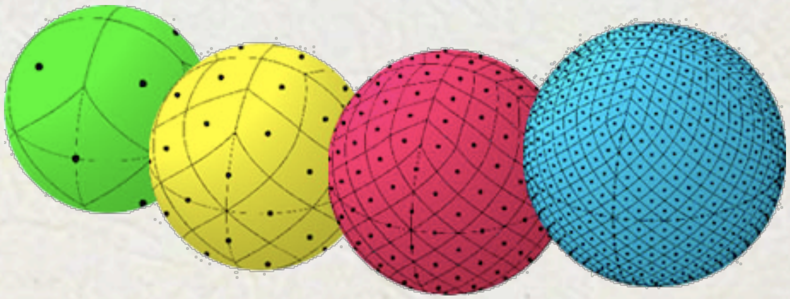
The properties of a random and a grid realization are quite different, although both are consistent with the continuous model from which they have been drawn, the fluctuations in the local density with respect to the average density are not the same, this is shown below by comparing the power Fourier spectrum of a random and a grid realization of a uniform distribution function:



Non-random realizations

In two or more dimensions, when the distribution function can be separated, we can apply the previous technique for each random variable. However, it is not trivial how to combine these coordinate values to produce a multi-dimensional realization that minimizes the frequency range of the fluctuations.

There is, however, an astronomically interesting 2-dimensional case that has been solved: that of a grid realization of a uniform density function on a spherical shell. Gorski, Hivon and Wandelt (1999) have developed the HEALPIX package that distributes points in the unit sphere maximizing the angular distance between them while maintaining equal solid angles in all the quadrangles formed by every four neighbooring points.
See: *http:/healpix.jpl.nasa.gov/*.



Quasi-random realizations

The issue of density fluctuations is quite important.
The expected relative fluctuations in a random sample of size N are $\delta N/N \propto 1/\sqrt{N}$ (Poisson distribution).
If we simulate a system of N_s stars using only N particles, we will be overestimating the density fluctuations by a factor:

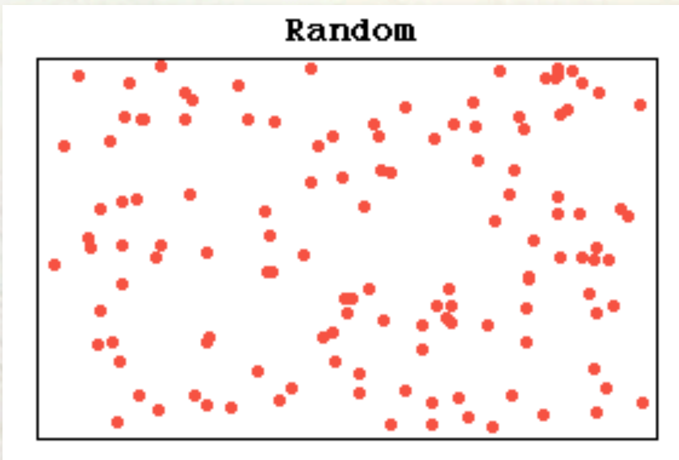
$$\frac{(\delta N/N)}{(\delta N_s/N_s)} = \sqrt{N_s/N}$$

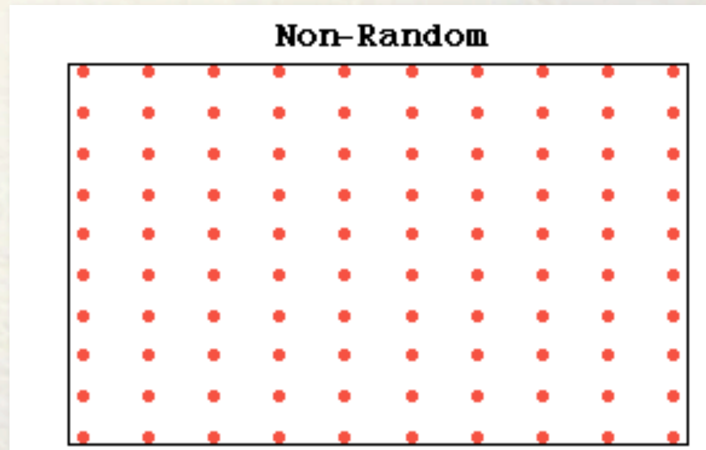
If the dynamical phenomenon we are trying to study depends critically on the size of the fluctuations, we may be using initial conditions that are "too noisy".

It is then highly desirable to find "*quasi-random*" realizations that fall in between the random and grid realizations of the previous two sections.

Quasi-random realizations

Can we find realizations whose points fill the space more uniformly than random, without having the fixed spacing of a grid?





The answer is yes!

They are called “*quasi-random realizations*”.
They have the property of “avoiding” each other and introduce progressively higher spatial frequencies as the size of the sample increases.

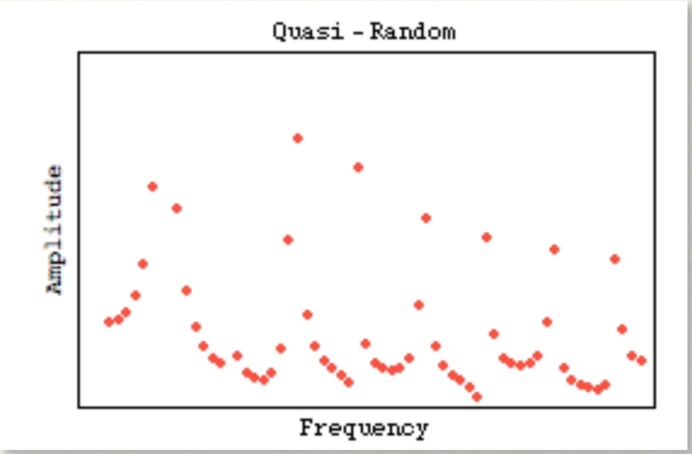
Quasi-random realizations

We will use a simple quasi-random sequence to illustrate the concept:
Halton's sequence: H_1, H_2, …

Here's the recipe to obtain the j^{th} term in the sequence:

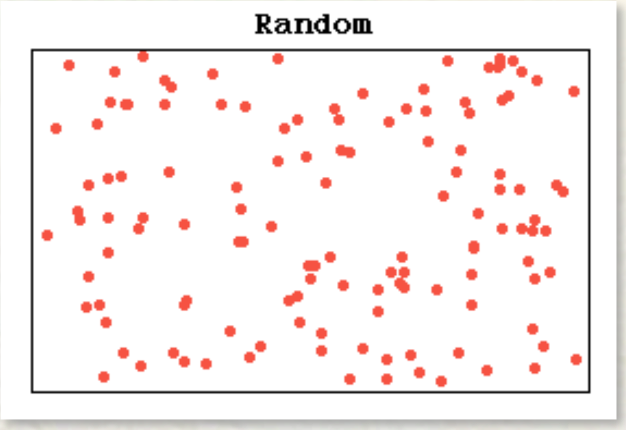
- Change j to a representation base b (prime): $j=19$, $b=3 \Rightarrow 201_3$
- Reverse the order of digits and put a radix point: 0.102_3
- Convert back to decimal representation: 0.407407

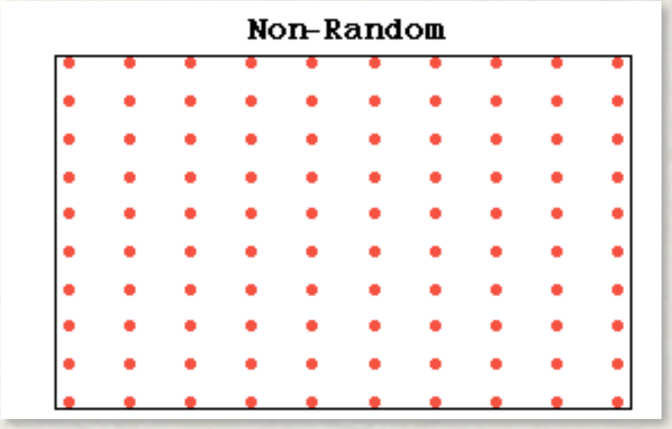
The power spectrum is intermediate between those of the grid and random samples.

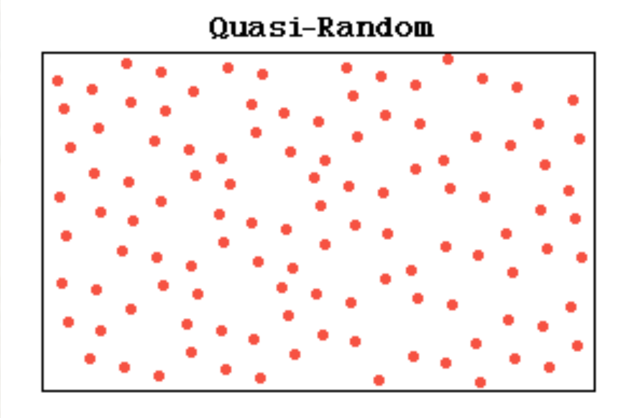


Quasi-random realizations

To generate multidimensional points, we use different prime bases *b* for each coordinate.





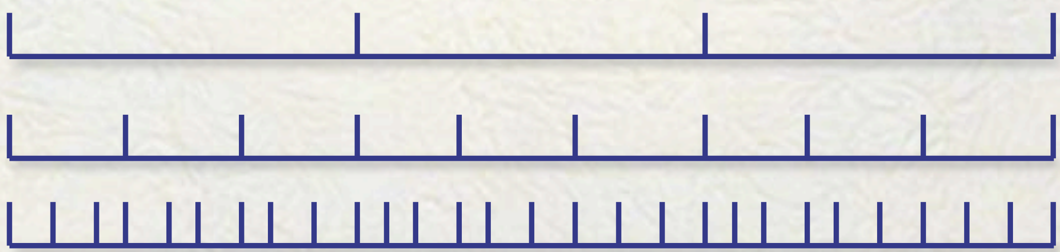


Quasi-random realizations

How does *Halton's* procedure works?

Every time the number of digits increases by one, the reversed-order digit sequence becomes a factor *b* finer meshed:

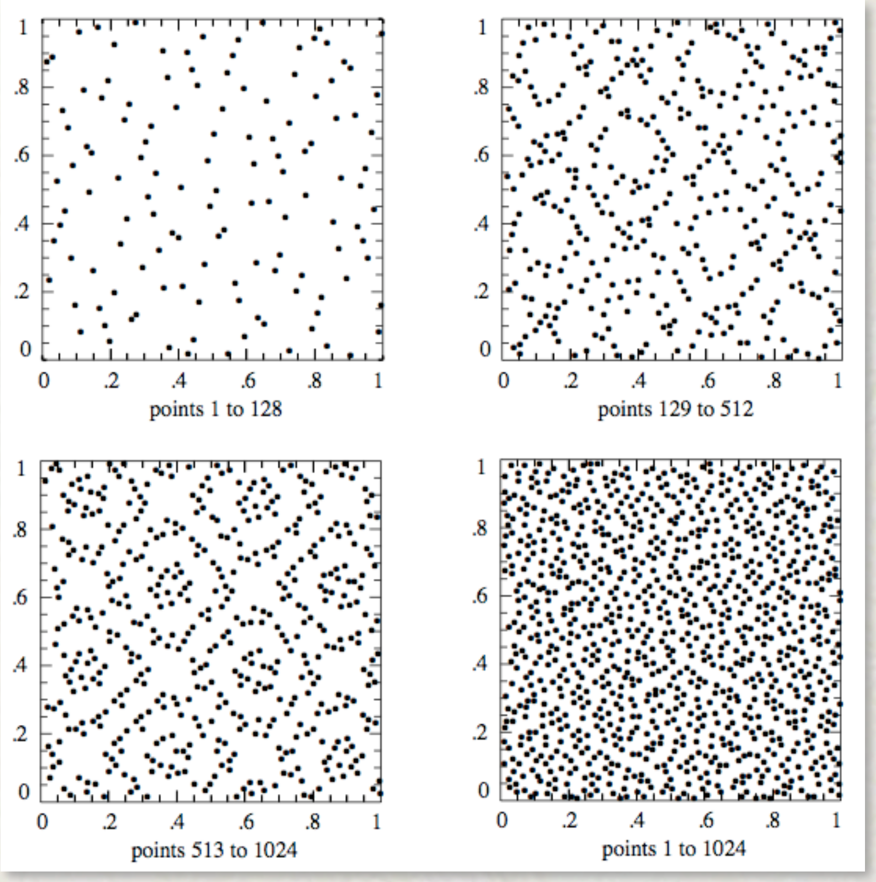
$$j = \ \ 8 = \ \ 22_3 \rightarrow \ \ 0.22_3 = 0.888889$$
$$j = \ \ 9 = 100_3 \rightarrow 0.001_3 = 0.037037$$



The process is one of filling the domain on a sequence of finer and finer meshes in a maximally spread-out order on each mesh: The most rapidly changing digit in *j* controls the most significant digit of the fraction.

Quasi-random realizations

There are other, more elaborated sequences, like *Sobol sequences*, that can be used (Press et al.; *Numerical Recipes*)



Random number generators

Up to now we have been using a function "***ran( )***" that somehow produces "random" numbers with equal probability between 0 and 1.
How does this works?

Strictly speaking, we shouldn't call the numbers produced by *ran()* "random". They are produced by a deterministic machine following an algorithm, so they can't be really "random".

A truly random sequence would be the outcome of a random physical process, like the time intervals between nuclei decay in a radioactive sample.

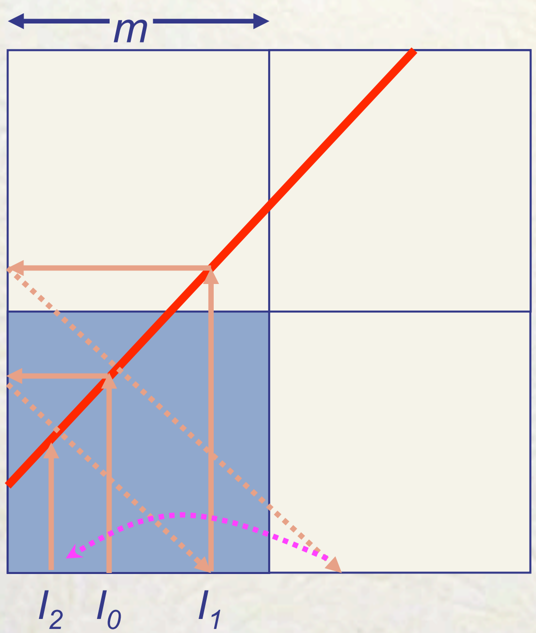
There are, however, algorithms that can produce sequences that, *within some precise limits in dynamical range and size of sample*, are described by the Poisson distribution we discussed earlier.

We will call these sequences "*pseudorandom*".

Random number generators

One such algorithm is the *Linear Congruential Method* (LCM).
It works by applying a simple scaling and shift on an integer that acts as a “seed”, or initial member of the sequence: I_0, I_1, I_2, …

$$I_{j+1} = (a \times I_j) + c, \quad \mathrm{Mod}\ m$$

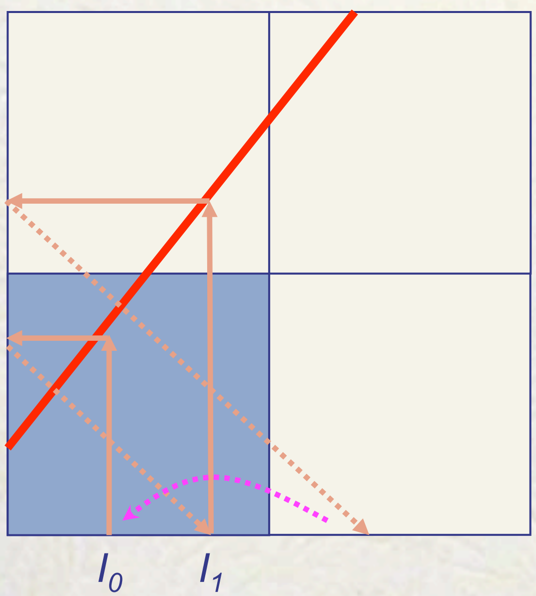


This is a popular algorithm that is used in many of the run-of-the-mill random number generators provided in computers.

We will use this algorithm to illustrate two important features of random number generators that we must be aware of: periodicity and serial correlation.

Random number generators

We begin with periodicity: The sequence produced by LCM is periodic with a period that can not be longer than *m*. If we make bad choices for the parameters that define the algorithm, we can end up with an implementation whose period is far shorter than this.



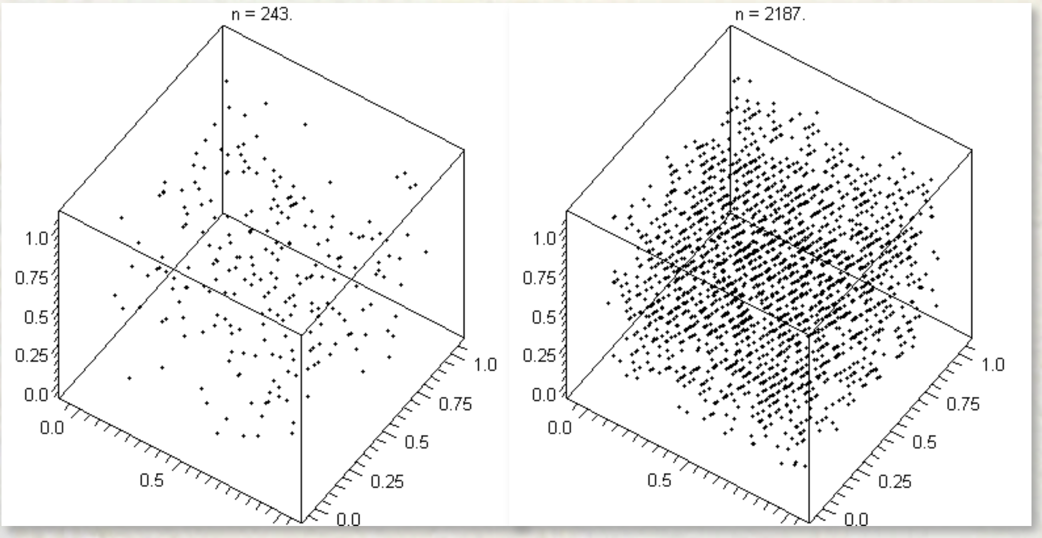
Since random number generators have periodicity, it is important to make sure that the generator we are using has a period substantially longer than the number of times we expect to call it.

In the case of the Von Neumann Rejection Technique, this number can easily exceed the number of points in our initial condition by a very large factor.

Instituto de Astronomía UNAM

Random number generators

Serial correlation refers to sequential correlation on successive calls, this is bad when we generate *n*-tuples as coordinates for points in *n*-dimensional space.



In *n*-dimensional space, points will tend to lie on (*n-1*) dimensional hyper-planes.
It can be proved that at most, there will be $m^{1/n}$ such planes.
If the values of *a*, *c* and *m* are poorly chosen, there could be a lot less planes.

Random number generators

Two very desirable properties we should look for in random number generators are: *portability* and *repeatability*.

Portability refers to the property of being able to be ported to different computers. This implies implementation on a higher level computer language with no system-specific functions.

Repeatability refers to the property of being able to repeat exactly the same random sequence when initialized with the same seed.

Portability and repeatability are very important to be able to reproduce exactly the same numerical experiments on the same, or a different machine. If some odd behavior is found in some particular run, we can reproduce it to analyze it in detail.
It is very important to include the value of the seed used in initial condition files.

Random number generators

An excellent reference for these issues is:

Press, W., et al. (1992): *Numerical Recipes: The Art of Scientific Computing*.
Cambridge University Press
ISBN 0-521-43064
http://www.nr.com

Homework

5th Assignment

- There is no assignment!

Keep working on the mock proper motion survey!

End of fifth lecture