Random numbers

Introduction to Numerical Analysis

Riccardo Mannella

February 14, 2008

Preliminaries

What is a *random* number generator? Computers will normally generate pseudo random sequences (but there are entropy algorithms). When two generators give two different results for the same dynamics, it means that at least one of the two is bugged. We need a mean of obtaining "statistically independent" numbers from a "deterministic" machine. Also, how much "random" depends in general on the purpose of the generator and on the physical system.

Bible: DE Knuth, Seminumerical algorithms, vol 2 of The Art of Scientific Computing (Addison-Wesley, 1981)

Basic routines

Bottom line: be suspicious.

Basic routines

Bottom line: be suspicious. Once you have been suspicious, stop and be suspicious again.

Basic routines

Bottom line: be suspicious. Once you have been suspicious, stop and be suspicious again. In judging a routine, there are *many* things to check. Start with a simple algorithm, *linear congruential generator*

$$x_{n+1} = ax_n + c \pmod{m}$$

They are very fast and easy to implement (even at low level).

Basic routines

Bottom line: be suspicious. Once you have been suspicious, stop and be suspicious again. In judging a routine, there are *many* things to check. Start with a simple algorithm, *linear congruential generator*

$$x_{n+1} = ax_n + c \pmod{m}$$

They are very fast and easy to implement (even at low level). But there are correlations; and plotting a k-plet, the generator fills up a $k-1$ D space, not a k D one, in planes, and their number is at most $m^{1/k}$

Basic routines

Bottom line: be suspicious. Once you have been suspicious, stop and be suspicious again. In judging a routine, there are *many* things to check. Start with a simple algorithm, *linear congruential generator*

$$x_{n+1} = ax_n + c \pmod{m}$$

They are very fast and easy to implement (even at low level). But there are correlations; and plotting a k-plet, the generator fills up a $k-1$ D space, not a k D one, in planes, and their number is at most $m^{1/k}$
Take $m = 2^{32}$, $k = 3$, $m^{1/k} \approx 1600$

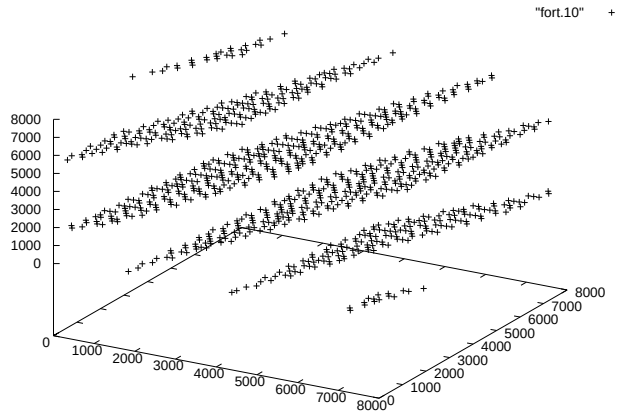
$$x_{n+1} = ax_n + c \pmod{m}$$

Let us see some examples. $k = 3$ always

$$x_{n+1} = ax_n + c \pmod{m}$$

Let us see some examples. $k = 3$ always

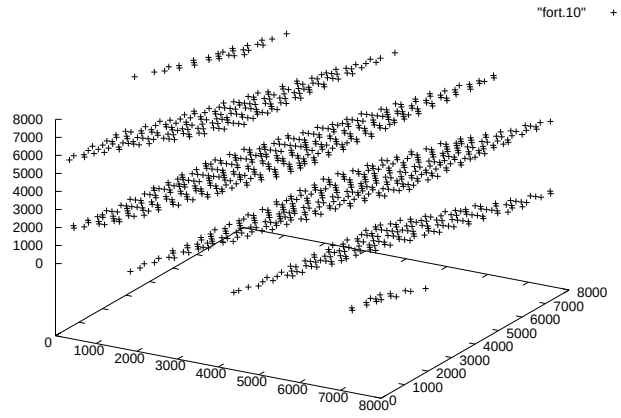
$a = 421$, $c = 1663$ and $m = 7875$, 100 points
$m^{1/k} \approx 19.9$



$x_{n+1} = ax_n + c \pmod{m}$

Let us see some examples. $k = 3$ always

$a = 421$, $c = 1663$ and $m = 7875$, 100 points
$m^{1/k} \approx 19.9$

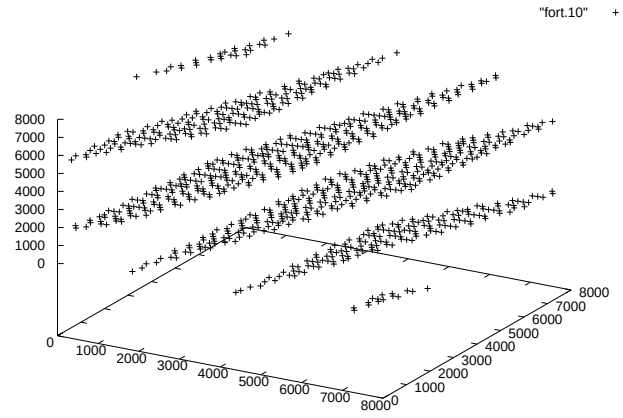


$a = 4096$, $c = 150889$ and $m = 714025$, 10^4 points
$m^{1/k} \approx 63.8$ (Numerical Recipes)

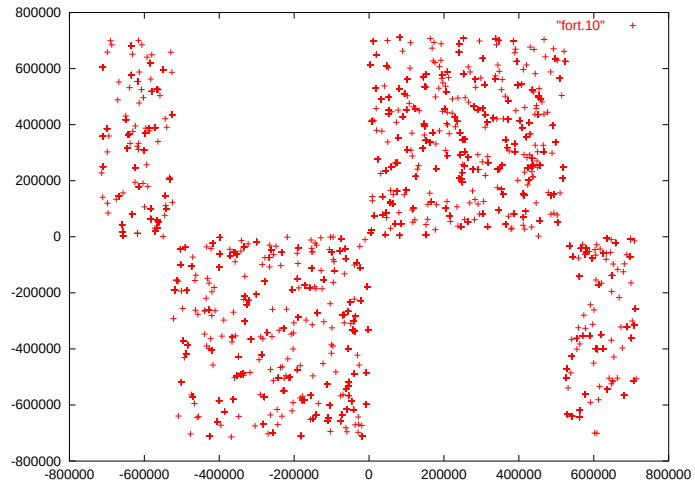
$x_{n+1} = ax_n + c \pmod{m}$

Let us see some examples. $k = 3$ always

$a = 421$, $c = 1663$ and $m = 7875$, 100 points
$m^{1/k} \approx 19.9$



$a = 4096$, $c = 150889$ and $m = 714025$, 10^4 points
$m^{1/k} \approx 63.8$ (Numerical Recipes)



Hopeless?

Well, combining several congruential methods? `RAN1.F` combines 3 methods for most significant, least significat bits and for reshuffling. If speed is the concern, use only two congruential methods, but bear in mind that the generator will sample "less" of the phase space (`RAN2`).

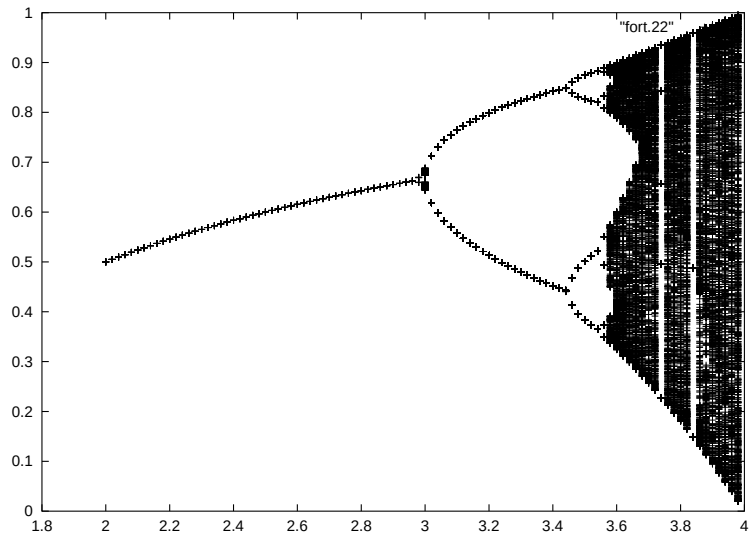
Hopeless?

Well, combining several congruential methods? `RAN1.F` combines 3 methods for most significant, least significat bits and for reshuffling. If speed is the concern, use only two congruential methods, but bear in mind that the generator will sample "less" of the phase space (`RAN2`). But there are more elegant approaches: take a chaotic dynamical system and use it.

Problems? Well, yes! Take a prototype system:

$$x_{n+1} = \lambda x_n (1 - x_n)$$

How chaotic is chaos?

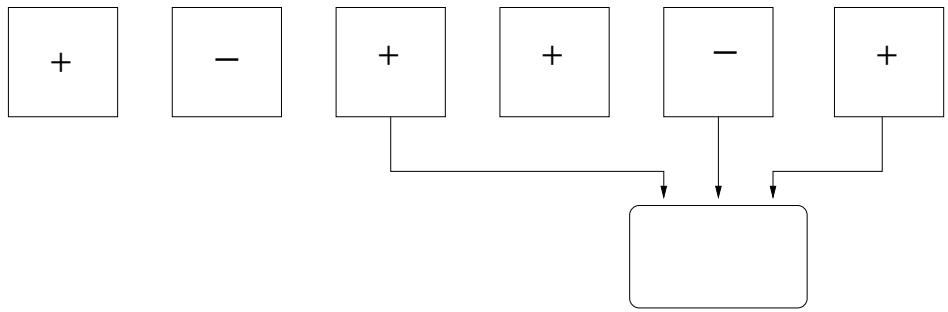


Dynamical system

The idea is to build a dynamical system which has controlled chaotic behaviour: this can be done with a "many states" machine:

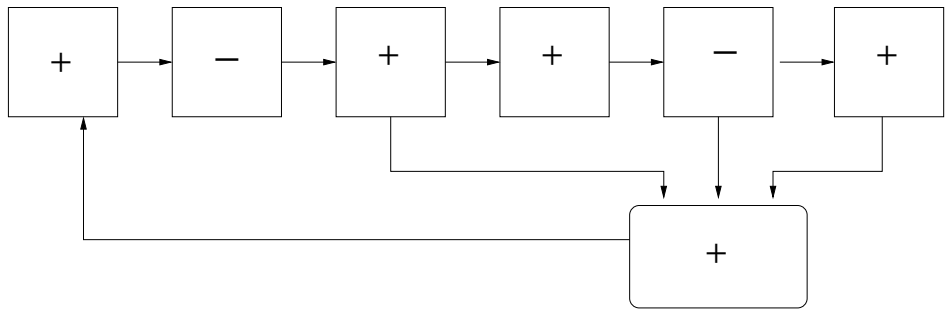
Dynamical system

The idea is to build a dynamical system which has controlled chaotic behaviour: this can be done with a "many states" machine:



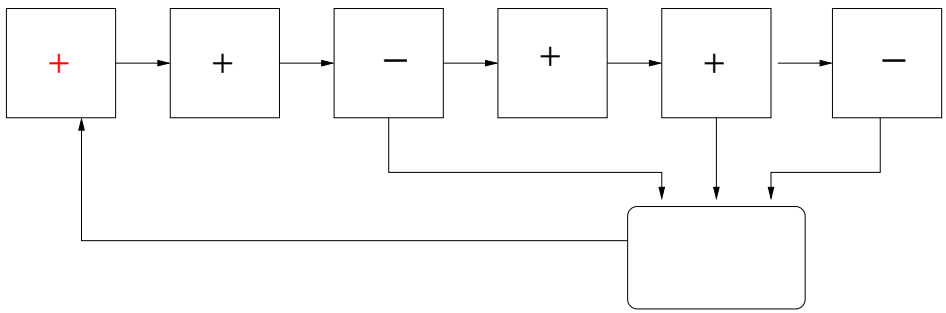
Dynamical system

The idea is to build a dynamical system which has controlled chaotic behaviour: this can be done with a "many states" machine:



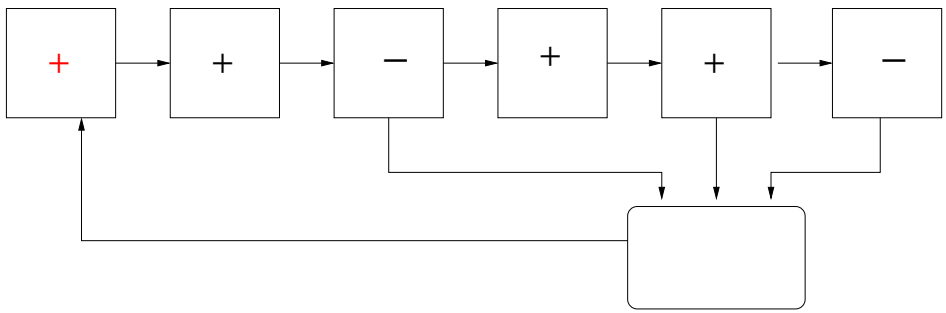
Dynamical system

The idea is to build a dynamical system which has controlled chaotic behaviour: this can be done with a "many states" machine:



Dynamical system

The idea is to build a dynamical system which has controlled chaotic behaviour: this can be done with a "many states" machine:



This is implemented in "add and carry" or "subtract and carry" routines, like `RAN3`, or `rcarry`. Very good algorithms, large repetition times, well behaved (no spectral theorem proof, however, unlike linear congruent ones)

Distribution

How do we generate distributions $p(x)dx$, beside the uniform one? There are different methods to obtain non uniform distributions. Suppose $p(x)$ is a uniform distribution, but we need a distribution $p(y)$. It is always true that $|p(y)dy = p(x)dx|$. From this it follows that

$$p(y) = p(x)\left|\frac{dx}{dy}\right|$$

The difficulty is to find the appropriate function to carry out the transformation. This is related to solving:

$$\frac{dx}{dy} = f(y)$$

Distribution

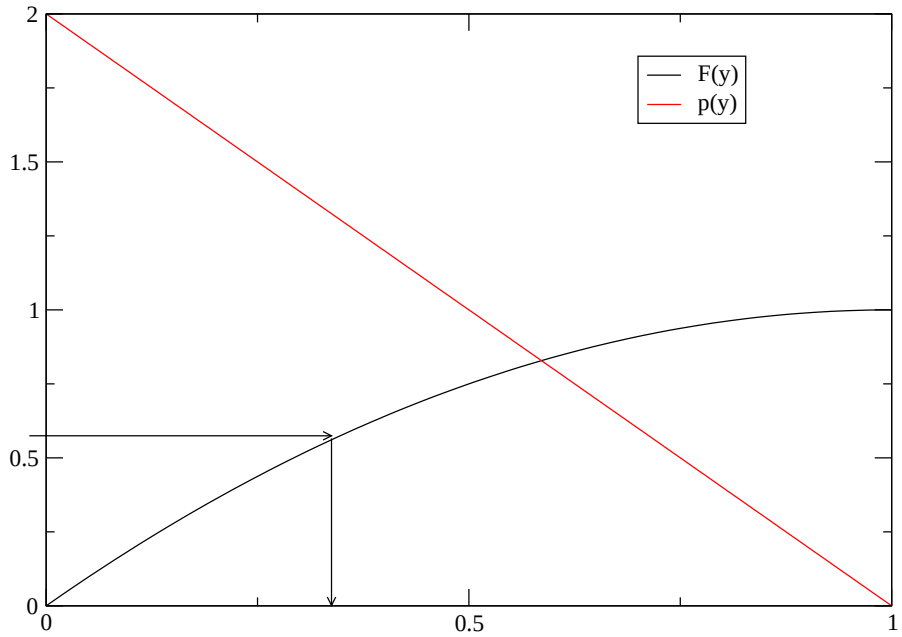
Remember that $dx/dy = f(y)$, and solving this differential equation, $x = F(y)$, where $F(y) = \int p(y)dy$. Finally,

$$y(x) = F^{-1}(x)$$

So, as long as the *inverse of the integral of* $p(y)$ is numerically easy to compute, we have a way of obtaining a generic distribution. Example: $y(x) = -\ln x$. We have

$$p(y)dy = p(x)\left|\frac{dx}{dy}\right| dy = e^{-y}dy$$

Geometrical interpretation



$$y(x) = F^{-1}(x) \quad F(y) = \int p(y)dy$$

Many dimensions

Transformation methods work in more than 1 D:

$$p(y_1, y_2, ...)dy_1 dy_2... = \left| \frac{\partial(x_1, x_2..)}{\partial(y_1, y_2, ..)} \right| p(x_1, x_2...)dy_1 dy_2...$$

Many dimensions

Transformation methods work in more than 1 D:

$$p(y_1, y_2, ...)dy_1 dy_2... = \left| \frac{\partial(x_1, x_2..)}{\partial(y_1, y_2, ..)} \right| p(x_1, x_2...)dy_1 dy_2...$$

Gaussian distributions! Consider

$$y_1 = \sqrt{-2 \ln x_1} \cos 2\pi x_2$$

$$y_2 = \sqrt{-2 \ln x_1} \sin 2\pi x_2$$

Many dimensions

Transformation methods work in more than 1 D:

$$p(y_1, y_2, ...)dy_1 dy_2... = \left| \frac{\partial(x_1, x_2..)}{\partial(y_1, y_2, ..)} \right| p(x_1, x_2...)dy_1 dy_2...$$

Gaussian distributions! Consider (Box-Muller)

$$y_1 = \sqrt{-2 \ln x_1} \cos 2\pi x_2$$

$$y_2 = \sqrt{-2 \ln x_1} \sin 2\pi x_2$$

Work out the Jacobian, the result is:

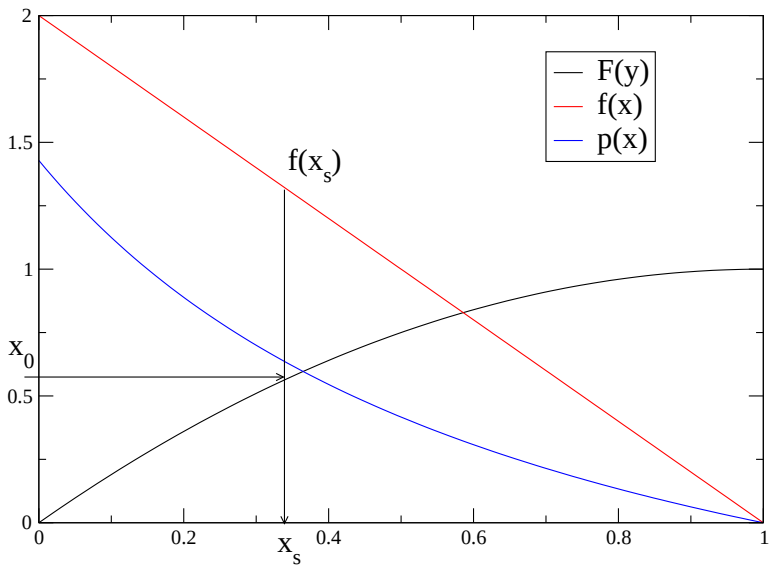
$$P(y_1, y_2) = \exp(-y_1^2/2 - y_2^2/2)/\sqrt{2\pi}$$

Rejection method

Not required an explicit integral function. Take $p(x)$, the desired probability function, and $f(x)$, which is such that $f(x) > p(x) \forall x$. Then using a couple of random deviates, sample for $p(x)$.

Rejection method

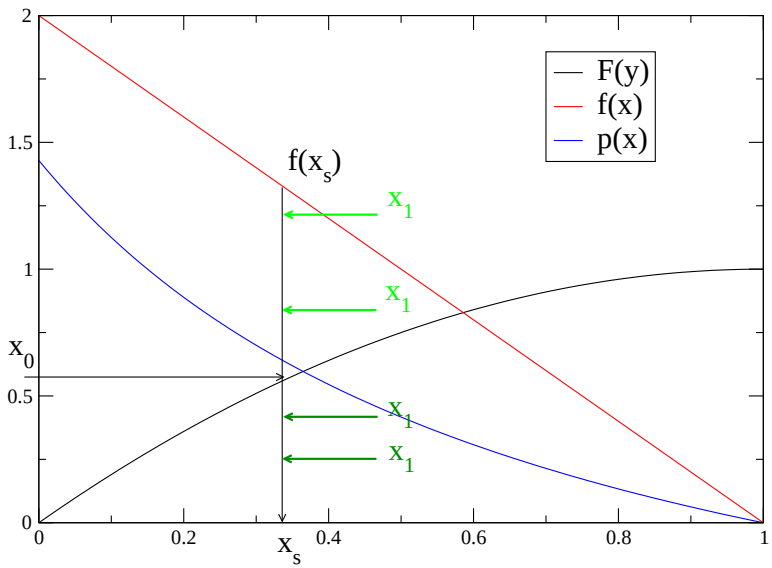
Not required an explicit integral function. Take $p(x)$, the desired probability function, and $f(x)$, which is such that $f(x) > p(x) \forall x$. Then using a couple of random deviates, sample for $p(x)$.



x_0 is a uniform deviate, between 0 and max $F(y)$.

Rejection method

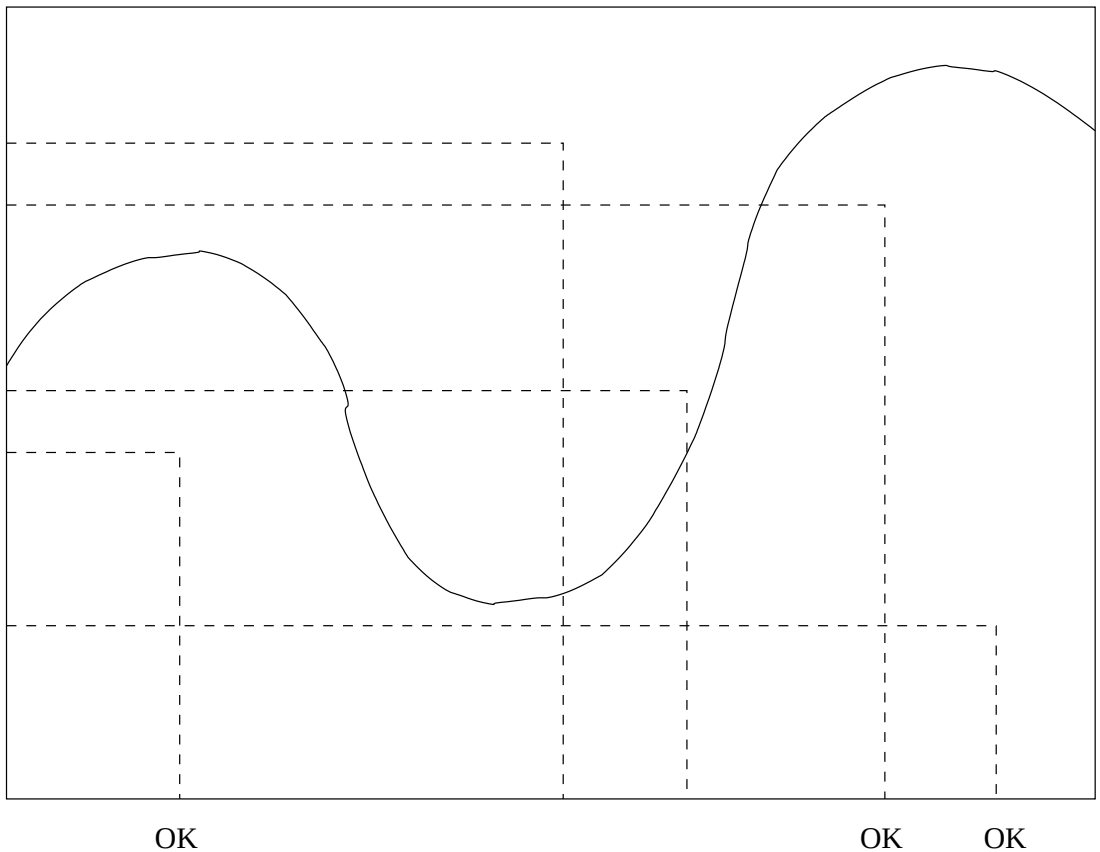
Not required an explicit integral function. Take $p(x)$, the desired probability function, and $f(x)$, which is such that $f(x) > p(x) \forall x$. Then using a couple of random deviates, sample for $p(x)$.



x_0 is a uniform deviate, between 0 and max $F(y)$. x_1 is a second deviate, between 0 and $f(x_s)$. If $x_1 < p(x_s)$, accept x_s.

Rejection: an example

Let us see a simpler example.



The points marked OK are accepted.

Rejection: Gamma Distribution/1

The Gamma distribution of integer order $a > 0$ is the waiting time distribution for the ath Poisson random process of unit average:

$$p_a(x)dx = \frac{x^{a-1}e^{-x}}{\Gamma(a)} \quad a > 0$$

For small a, generate directly a exponential deviates, and sum them. This is equivalent to computing the product of a random deviates, and then taking the log.
More interesting for larger a. It becomes "bell" shaped, with a peak around a and width $\approx \sqrt{a}$.

Rejection: Gamma Distribution/1

A handy comparison function is the Lorentzian, $p(y)dy = \frac{1}{\pi(1+y^2)}dy$. We integrate it,

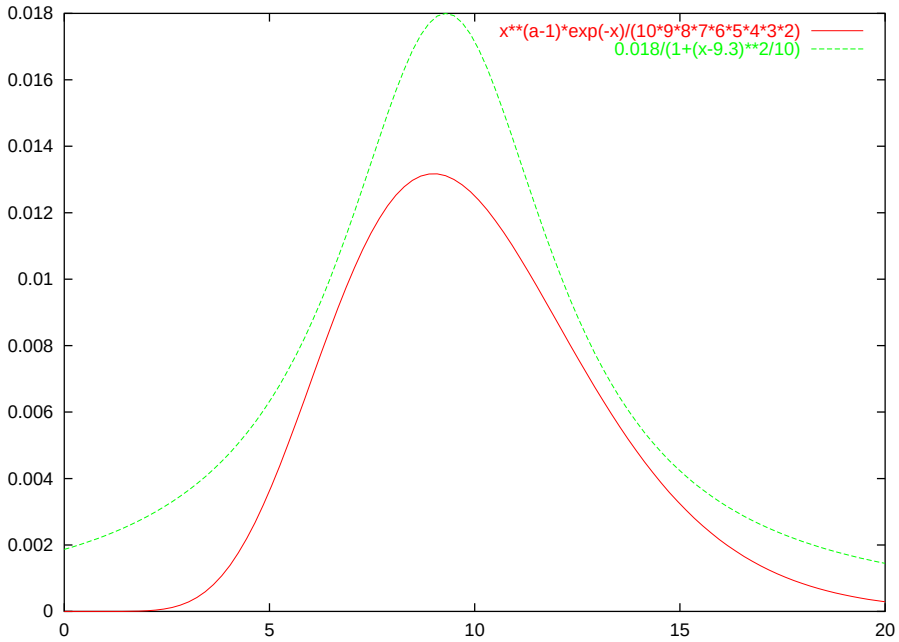
$$F(y) = \int p(y)dy = \frac{1}{\pi}\arctan(y)$$

We maximise the Gamma with the function

$$f(x) = \frac{c_0}{1 + (x - x_0)^2/a_0^2}$$

and optimize the rejection rate to generate a Gamma deviate making $a_0 c_0$ as small as possible, while $f(x)$ is larger than the Gamma function.

Rejection: Gamma Distribution/2



$$f(x) = \frac{c_0}{1 + (x - x_0)^2/a_0^2}$$

$$g(x) = \frac{x^{a-1}e^{-x}}{\Gamma(a)}$$

We proceed generating (u is uniform in $[0, 1]$)

$$x = a_0 \tan(\pi u) + x_0 \left(= f^{-1}(y)\right)$$

and check if acceptable. See the routine `GAMDEV`.

MonteCarlo integration

Closely related to the rejection method. Allows to compute integrals. In an n D space, given N points x_i,

$$\int_V f dv \approx V\langle f\rangle \pm V\sqrt{\frac{\langle f^2\rangle - \langle f\rangle^2}{N}}$$

where

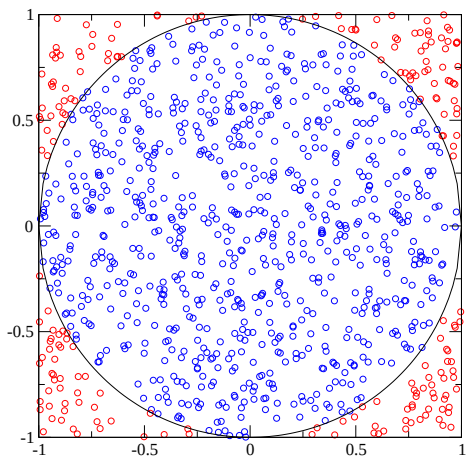
$$\langle f^n\rangle = \frac{1}{N}\sum_{i=1}^{N} f^n(x_i)$$

Embed a "difficult to sample" region into an easy one, sample, and integrate.

MC: example

Area of a circle, set $f = 1$ inside, 0 outside. $V = 4$

$$\int_V f dv \approx V\langle f\rangle \pm V\sqrt{\frac{\langle f^2\rangle - \langle f\rangle^2}{N}}$$



$N = 1000 \;\; \langle f\rangle = 0.797$

$\text{area} = 3.19 \pm 0.05$

$N = 10^6 \;\; \langle f\rangle = 0.78616$

$\text{area} = 3.145 \pm 0.005$

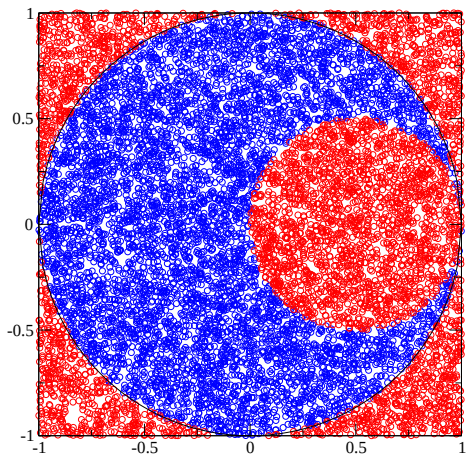
Evaluate the error for increasing N. (`mc1.f`)

MC: example 2

Take a disk, radius 1, with a hole in (0.5,0) of radius 0.5. Compute centre of mass and momentum of inertia with respect to the point (0,0). The integrals are:
$x_{cm} = \int x dV \quad I_{0,0} = \int (x^2 + y^2) dV$

MC: example 2

Take a disk, radius 1, with a hole in (0.5,0) of radius 0.5. Compute centre of mass and momentum of inertia with respect to the point (0,0). The integrals are:
$x_{cm} = \int x dV \quad I_{0,0} = \int (x^2 + y^2) dV$



$$N = 10^6$$

$$x_{cm} = -0.393 \pm 0.004$$

$$y_{cm} = -0.000 \pm 0.005$$

$$I_{0,0} = 1.275...$$

Of course, $f_x = x$ if inside the blue region, zero outside, $f_y^2 = y^2$ etc.