Re: Gaussian random variable

Source: http://coding.derkeiler.com/Archive/General/comp.programming/2007−03/msg00621.html

- *From*: "user923005" <dcorbit@xxxxxxxxx>
- *Date*: 30 Mar 2007 14:41:55 −0700

On Mar 30, 6:22 am, Lionel B <m...@xxxxxxxxxxx> wrote:

> On Fri, 30 Mar 2007 08:16:26 +0000, Mark P wrote:
>
> > b...@xxxxxxxxxxxxxx wrote:
> >
> > > I was wondering if anyone can recommend a way to generate
> > > a Gaussian
> > > random variable with standard deviation 1.
> > >
> > > Thank you.
> >
> > Nothing C++ about this question. Followups set to comp.programming.
> >
> > Here's one very general approach that can also be applied to your question:
> >
> > For a general distribution D(x) consider the integral of that
> > distribution G(x) = integral_{−infinity}^x D(y) dy. The range of G(x)
> > is then [0,1]. Pick a value z uniformly at random in [0,1] and consider
> > the inverse function G^{−1}(z). Then the distribution of values of
> > G^{−1}(z) obeys the original distribution D.
> >
> > That may sound complicated but it's really not. You have a standard
> > normal distribution N(x). Integrate that distribution to get:
> >
> > f(x) = 1/2 (1 + erf(x/sqrt(2))).

> > Now pick values uniformly at random in [0,1], which ought to be easy
> > (approximately) in most languages. Then compute f−inverse of those
> > values to get values obeying a Gaussian distribution. The hard part
> > here will be inverting the erf function. You'll probably have to rely
> > on some sort of library for this (or numerically integrate N(x) yourself
> > and create a lookup table of values of f(x)).
>
> That is likely to be an /extremely/ inefficient way to do it.
>
> Simple to code up and pretty fast is the standard Box−Muller method:
>
> http://en.wikipedia.org/wiki/Box−Muller_transform

Here is an implementation from the C−FAQ:
13.20: How can I generate random numbers with a normal or Gaussian distribution?

A: Here is one method, recommended by Knuth and due originally to Marsaglia:

```
#include <stdlib.h>
#include <math.h>

double gaussrand()
{
static double V1, V2, S;
static int phase = 0;
double X;

if(phase == 0) {
do {
double U1 = (double)rand() / RAND_MAX;
double U2 = (double)rand() / RAND_MAX;

V1 = 2 * U1 − 1;
V2 = 2 * U2 − 1;
S = V1 * V1 + V2 * V2;
} while(S >= 1 || S == 0);

X = V1 * sqrt(−2 * log(S) / S);
} else
X = V2 * sqrt(−2 * log(S) / S);

phase = 1 − phase;

return X;
}
```

See the extended versions of this list (see question 20.40) for other ideas.

References: Knuth Sec. 3.4.1 p. 117; Marsaglia and Bray, "A Convenient Method for Generating Normal Variables"; Press et al., _Numerical Recipes in C_ Sec. 7.2 pp. 288−290.

.