

# FYI: Something I saw on generating random-numbers

---

*Source:* <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2006-09/msg01520.html>

---

- *From:* [dkcombs@xxxxxxxxxx](mailto:dkcombs@xxxxxxxxxx) (David Combs)
  - *Date:* 23 Sep 2006 21:57:32 -0400
- 

I like to lurk over at comp.graphics.algorithms -- extremely good math stuff, and explanations, there.

(Not that I understand any of it!)

Especially long posts by this guy/gal who goes by "d'FAQS".

Since generation of random numbers is often talked about here in this group, for those who like that kind of stuff, [even you likely know most if not all of these methods] I post this (and which I just saw):

From nobody-here@xxxxxxx Sat Sep 23 21:49:00 EDT 2006

Article: 165648 of comp.graphics.algorithms  
NNTP-Posting-Date: Sat, 02 Sep 2006 20:02:38 -0500  
From: Just d' FAQs <nobody-here@xxxxxxx>  
Newsgroups: comp.graphics.algorithms  
Subject: Re: Stochastic Positioning of A Point in A 3D Gaussian Distribution  
Xref: panix comp.graphics.algorithms:165648

On 1 Sep 2006 09:56:43 -0700, hoffmann@xxxxxxxxxxxxx wrote:

1. Why is the Box-Muller Transform, which converts two PRNG (pseudo random number generator) numbers  $u_1, u_2$  with uniform distributions into a geometrical pair  $x, y$  with Gaussian distributions, better than a transform which converts simply one PRNG number with uniform distribution into one number with Gaussian distribution ?

[http://en.wikipedia.org/wiki/Box-Muller\\_transform](http://en.wikipedia.org/wiki/Box-Muller_transform)

The OP needs obviously THREE numbers for  $x, y, z$ .

2. Numbers by a PRNG with uniform distribution can be transformed into numbers with Gaussian distribution either by adding  $N$  numbers or by averaging  $N$  numbers.

E.g. one can find that  $N=6$  is reasonable and  $N=12$  is near to

## FYI: Something I saw on generating random-numbers

perfect.

IMO this is perhaps faster than using Box-Muller.

Is any comparison of quality available ?

There is no reason anyone should need to write their own pseudo-random number generator, whether for uniform or Gaussian distributions. It's really easy to create a bad uniform generator, and even the venerable Box-Muller generator for Gaussians has two major variations, one of which is troublesome.

Four possibilities are:

- \* Box-Muller: this gives two independent variates for each call, and is still fast even if one is discarded.
- \* Ratio: This is a method using rejection, and the ellipse regions of Leva make it fast as well as brief.
- \* Averaging uniform: This is both slow and inaccurate, it is not to be recommended even with some known adjustments.
- \* Ziggurat: A variation of the "rectangle-wedge-tail" approach, this tends to be the fastest, but requires a large table.

Two good sources to consult are Luc Devroye,

<<http://cg.scs.carleton.ca/~luc/rng.html>>

(including his book, a free download),

<<http://cg.scs.carleton.ca/~luc/books-luc.html>>

and Knuth,

Knuth, D. The Art of Computer Programming, Vol. II., 3/e.

Or just grab an implementation, such as winrand,

<<http://crypto.mat.sbg.ac.at/ftp/pub/data/winrand.zip>>

or the GNU Scientific Library (GSL).

<<http://www.gnu.org/software/gsl/>>

I hope it turns out of interest.

David

FYI: Something I saw on generating random-numbers