Journal of Statistical Software

February 2005, Volume 12, Issue 7. http://www.jstatsoft.org/

A Comment on the Implementation of the Ziggurat Method

Philip H.W. Leong
Chinese University of Hong Kong

Guanglie Zhang
Chinese University of Hong Kong

Dong-U Lee
Imperial College London

Wayne Luk
Imperial College London

John D. Villasenor
University of California
Los Angeles

Abstract

We show that the short period of the uniform random number generator in the published implementation of Marsaglia and Tsang's Ziggurat method for generating random deviates can lead to poor distributions. Changing the uniform random number generator used in its implementation fixes this issue.

Keywords: random number generators, normal distribution, Ziggurat method, Xorshift RNGs.

Marsaglia and Tsang (2000) proposed the Ziggurat method for generating a random variable from a given decreasing probability density. This method is one of the fastest available for generating normal variates. We applied the chi-square (χ^2) goodness-of-fit test (Knuth (1997)) to check the distribution of random variables generated using an implementation available from Marsaglia and Tsang (2000).

The χ^2 test involves quantizing the horizontal axis of the probability density function (pdf) into k bins, determining the actual and expected number of samples appearing in each bin, and using the results to derive a single number that serves as an overall quality metric. Let t be the number of observations, p_i be the probability that each observation falls into the category i and Y_i be the number of observations that actually do fall into category i. The χ^2_{k-1} statistic is given by $\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(Y_i - tp_i)^2}{tp_i}$.

The Ziggurat method was used to generate 20 billion normal random variates, and the distribution tested using the χ^2 test based on 200 bins spaced uniformly over $[-7, 7]$. The χ^2_{199} statistics for five different trials with different initial seeds are shown in Table 1. As a comparison, the same test was repeated using (a) the GNU Scientific Library's implementation of

Method	Trial					Speed
	1	2	3	4	5	
Ziggurat (original)	1123	1119	1165	1167	1149	21.7
Polar	191	216	223	170	207	4.2
Ziggurat (modified)	155	214	174	196	198	18.8

Table 1: χ^2_{199} values for 5 experiments involving the generation of 20 billion normally distributed random variates. Speed is in million random numbers per second.

the polar method[1] (Knuth (1997), Project (2004)) and (b) a modified implementation of the Ziggurat method which uses a different uniform random number generator (RNG). For a 95% level of confidence, the critical value for the χ^2_{199} test is 233. Since the values obtained for the original implementation of the Ziggurat method are all well above this value, we conclude that the generated values are not normally distributed. The polar method and modified Ziggurat method give χ^2_{199} statistics below the critical value. Figure 1 shows a plot of the χ^2_{199} value versus the iteration number for a typical trial. As can be seen, the critical value of 233 is exceeded when the uniform random number generator repeats after approximately 2^{32} (4 billion) iterations.

The original implementation of the Ziggurat method uses an xorshift RNG (Marsaglia (2003)) implemented via the C macro:

```
#define SHR3 (jz=jsr, jsr^=(jsr<<13), jsr^=(jsr>>17), jsr^=(jsr<<5),jz+jsr)
```

The modified Ziggurat RNG uses the KISS uniform RNG (Marsaglia (1999)) with a longer period. KISS combines the output of a SHR3 generator with that of two multiply-with-carry generators MWC, and a congruential generator CONG as follows:

```
#define znew   (z=36969*(z&65535)+(z>>16))
#define wnew   (w=18000*(w&65535)+(w>>16))
#define MWC    ((znew<<16)+wnew )
#define SHR3 (jz=jsr, jsr^=(jsr<<13), jsr^=(jsr>>17), jsr^=(jsr<<5),jz+jsr)
#define CONG  (jcong=69069*jcong+1234567)
#define KISS  ((MWC^CONG)+SHR3)
```

The rightmost column of Table 1 shows the number of normal random variates generated per second on a 2.66 GHz Intel Pentium 4 machine using GNU gcc version 3.2 with -O3 optimization. It can be seen that the modified Ziggurat RNG is 13% slower than the original version. Many other equally suitable choices for a uniform RNG are available, some of which may offer higher speed.

To summarize, the Ziggurat normal RNG available from (Marsaglia and Tsang (2000)) does not pass a χ^2 test for a normal distribution due to the short period of the SHR3 uniform RNG used in its implementation. A modified version which uses the KISS uniform RNG is presented which addresses this issue with a minor performance penalty. It would be prudent

[1] gsl_ran_gaussian() from gsl version 1.4 with the KISS uniform random number generator (described below).

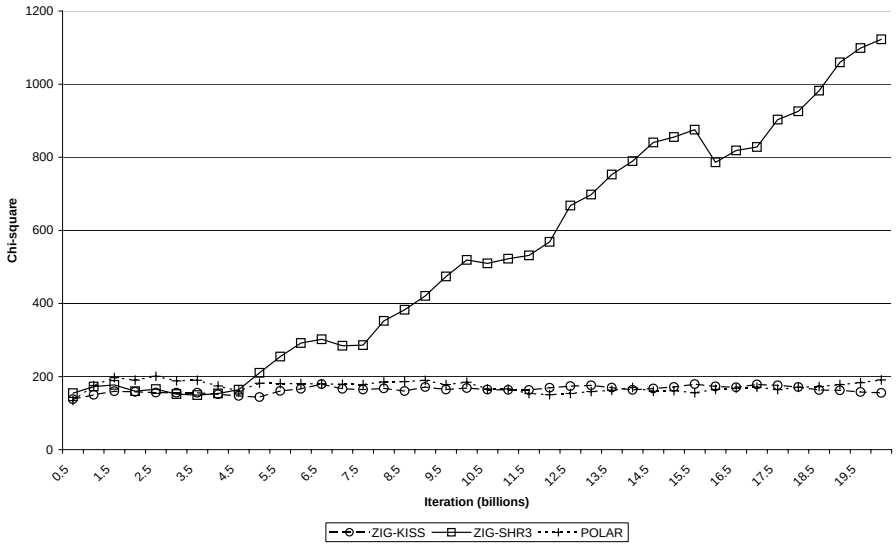

Figure 1: Plot of the χ^2_{199} value versus the iteration number for a typical trial.

to test simulations with several different uniform RNGs (Marsaglia (1999), Knuth (1997)) to ensure consistent results.

Acknowledgments

The authors gratefully acknowledge support from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4333/02E).

References

Knuth DE (1997). *The Art of Computer Programming. Volume 2: Seminumerical Algorithms*. Addison-Wesley, 2 edition.

Marsaglia G (1999). "Random Numbers for C: End, at last?" *Posting to sci.stat.math*. URL http://groups.google.com/groups?selm=36A5BB98.F2561DFF%40stat.fsu.edu.

Marsaglia G (2003). "Xorshift RNGs." *Journal of Statistical Software*, **8**(14). URL http://www.jstatsoft.org/v08/i14/.

Marsaglia G, Tsang WW (2000). "The Ziggurat Method for Generating Random Variables." *Journal of Statistical Software*, **5**(8). URL http://www.jstatsoft.org/v05/i08/.

Project G (2004). "GNU Scientific Library." URL http://www.gnu.org/software/gsl/.

Affiliation:

Philip Leong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
E-mail: phwl@cse.cuhk.edu.hk
URL: http://www.cse.cuhk.edu.hk/~phwl/

Journal of Statistical Software
February 2005, Volume 12, Issue 7.
http://www.jstatsoft.org/

Submitted: 2004-11-06
Accepted: 2005-02-08