Subject: Re: How good is Randomu?
Posted by Brian Larsen on Fri, 04 Jan 2008 17:55:07 GMT
View Forum Message <> Reply to Message

From the IDL help on randomu

""The random number generator is taken from: "Random Number Generators: Good Ones are Hard to Find", Park and Miller, Communications of the ACM, Oct 1988, Vol 31, No. 10, p. 1192. To remove low-order serial correlations, a Bays-Durham shuffle is added, resulting in a random number generator similar to ran1() in Section 7.1 of Numerical Recipes in C: The Art of Scientific Computing (Second Edition), published by Cambridge University Press.""

I don't have my copy of Numerical Recipes handy but it is also online so you can read how good ran1() is. My gut feeling is that if you want 10^9 random numbers randmonu() is probably not good (or fast) enough. My gut says to use ran3() from Numerical Recipes build in C can called via dlm for the large period and speed.

Cheers,

Brian

---------- ----------
Brian Larsen
Boston University
Center for Space Physics

Subject: Re: How good is Randomu?
Posted by Rick Towler on Fri, 04 Jan 2008 18:24:23 GMT
View Forum Message <> Reply to Message

I'm not a statistician, but as far as I can tell ran1 from NRC and thus randomu have a period of ~10^9. This is very short and surprises me... I guess that is why you can buy the IDL Analyst add on. More info on ran1 (and thus randomu) is here:

http://www.nrbook.com/a/bookcpdf/c7-1.pdf

MATLAB's default generator is based on "Marsaglia's ziggurat algorithm" with a period of 2^64. The paper and example C code are here:

http://www.jstatsoft.org/v05/i08

Whatever generator you choose, you'll going to want to code it up as a DLM or be *very* patient. I think you'll run into a real bottleneck generating all those random numbers in IDL. Looking at the ziggurat code, it would be pretty easy to code it up as a DLM. I recommend Ronn Kling's "Calling C/C++ from IDL" www.kilvarock.com.

-Rick

john.copley@nist.gov wrote:
> Many thanks to those of you who responded to my posting entitled "
> Bizarre (?) behavior of randomu". Your comments have been very useful.
>
> I now have a different question. How good is Randomu? I am developing
> some code to calculate multiple neutron scattering intensities and
> typically in any given run I would expect to invoke randomu (or some
> other IDL procedure that generates uniformly distributed pseudo-random
> numbers between 0 and 1) several hundred to several thousand times,
> each time obtaining of order 1 million numbers, in other words
> generating 10^9 or more random numbers in any given run. Is randomu up
> to the task, or do I need something better?
>
> If I need something better what should I use? I have come across
> exotica such as the "MT19937 generator of Makoto Matsumoto and Takuji
> Nishimura [which] is a variant of the twisted generalized feedback
> shift-register algorithm" and "has a Mersenne prime period of 2^19937
> - 1 (about 10^6000) and is equi-distributed in 623 dimensions" but
> that sounds like overkill. On the other hand the so-called Wichmann-
> Hill algorithm looks interesting and it is supposedly very easy to
> code.
>
> Thoughts, anyone?
> Many thanks
> John

Subject: Re: How good is Randomu?
Posted by wita on Mon, 07 Jan 2008 10:32:52 GMT
View Forum Message <> Reply to Message

On Jan 4, 5:17 pm, john.cop...@nist.gov wrote:
> Many thanks to those of you who responded to my posting entitled "
> Bizarre (?) behavior of randomu". Your comments have been very useful.
>
> I now have a different question. How good is Randomu? I am developing
> some code to calculate multiple neutron scattering intensities and
> typically in any given run I would expect to invoke randomu (or some

> other IDL procedure that generates uniformly distributed pseudo-random
> numbers between 0 and 1) several hundred to several thousand times,
> each time obtaining of order 1 million numbers, in other words
> generating 10^9 or more random numbers in any given run. Is randomu up
> to the task, or do I need something better?
>
> If I need something better what should I use? I have come across
> exotica such as the "MT19937 generator of Makoto Matsumoto and Takuji
> Nishimura [which] is a variant of the twisted generalized feedback
> shift-register algorithm" and "has a Mersenne prime period of 2^19937
> - 1 (about 10^6000) and is equi-distributed in 623 dimensions" but
> that sounds like overkill. On the other hand the so-called Wichmann-
> Hill algorithm looks interesting and it is supposedly very easy to
> code.
>
> Thoughts, anyone?
> Many thanks
> John

Dear John,

I am also not a statistician and have no experience with periodicity for that large amount of random numbers. What may be of interest to you is that a fairly sophisticated random number generators was built in Numeric Python (www.numpy.org), I quote from the manual: "The fundamental random number generator is the Mersenne Twister based on code written by Makoto Matsumoto and Takuji Nishimura (and modified for Python by Raymond Hettinger)". Since NumPy is open source and written in C or Fortran, you might be able to take that code and modify it to suit your needs. That has the advantage that the code has already been fairly well tested by the python community.

with best regards,

Allard