Subject: Re: IDL random number generator
Posted by James Kuyper on Mon, 12 May 2003 15:15:34 GMT
View Forum Message <> Reply to Message

krijger@astro.uu.nl wrote:
>
> James Kuyper <kuyper@saicmodis.com> wrote in message
news:<3EBBB786.9C52F5F3@saicmodis.com>...
>> krijger@astro.uu.nl wrote:
>>>
>>> Hi,
>>> I know that randomn is pseudo-random, how many numbers can you
>>> generate before the non-randomness kicks in?
>>>
>>> Thijs Krijger
>>
>> None. The non-randomness is there from the very beginning. You could
>> make a true random number generator by running it off of the radioactive
>> decay of atoms, or some similar hardware-based approach. However,
>> software random number generators are absolutely deterministic, once
>> you've set up the seed. You can set the seed form a clock setting, which
>> means that the precise sequence of random numbers generated depends upon
>> the precise time at which the program reads the clock. But even the very
>> first number can be absolutely predicted from the seed value.
>>
>> Every random number generator has a period, after which it starts
>> repeating the same exact sequence. How long that period is depends upon
>> the quality of the algorithm used. Commonly used algorithms have periods
>> in the range of 100,000 numbers or better. Very sophisticated generators
>> can have periods that are so long that your computer will become
>> obsolete before the sequence repeats.
>
> So, if in IDL I use the data=randomn(seed,N), then how big can N be
> (and I can make the claim that the numbers are still random (compared
> to each other))?

The numbers will never be truly random. They will always be
pseudo-random, no matter how big N is. There's no special point at which
the cease to be pseudo-random. That's a meaningless question, like
asking how many ducks are equivalent to one psuedonym. A meaningful
question to ask is how long will it be before the pseudo-random sequence
repeats. I don't know the answer to that one for this particular
generator. According to the online help:

"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."

If it's important to you, then you should probably track down those references and read them.

However, if you call randomu(seed), where 'seed' has not been defined, it will create the state array in a variable named 'seed'. That state array is a 36-element array of long integers. In principle, if their algorithm uses that array efficiently, the repeat period could be as long as 2**(36*32)= 3.2E798, which should be long enough for most purposes. :-) Let's put it this way. If you encoded the pseudo-random numbers on the energy levels of atoms, the entire visible universe isn't large enough (by many orders of magnitude) to record the entire sequence.
