Subject: Re: Randomu seed initialization
Posted by Jeremy Bailin on Thu, 12 Nov 2009 19:31:04 GMT
View Forum Message <> Reply to Message

On Nov 12, 8:59 am, Conor <cmanc...@gmail.com> wrote:
> On Nov 11, 3:22 pm, pp <pp.pente...@gmail.com> wrote:
>
>
>
>
>
>> On Nov 11, 5:54 pm, Conor <cmanc...@gmail.com> wrote:
>
>>> So I'm really wondering if I've hit some sort of strange IDL bug
>>> (although everytime I thought that so far I've been wrong). Still,
>>> thought I'd post about it.
>
>>> I've got a routine I'm running to do some simulations. It calls a
>>> function (generate_population) which generates 1000 variables
>>> populated along a certain distribution. In my routine I later
>>> generate another set of 1000 variables to select some of the generated
>>> population. In both of these cases IDL has to initialize the random
>>> seed generator because (of course) the generate_population routine is
>>> a separate function and so the main routine can't use the 'seed'
>>> variable generated in generate_population.
>
>>> However, I noticed that the cut being made was very strange and not at
>>> all what it should have been. After some examination, I discovered
>>> that the random variables being created by generate_population were
>>> the exact same random variables being used to make the cuts - two
>>> different calls to randomu() were returning exactly the same random
>>> variables. Which means that they were using the same seed. Except I
>>> wasn't giving either one the seed - it was being generated
>>> automatically by IDL. Weird... I've tried reproducing this by making
>>> simple routines that do the same sort of thing, but I don't have the
>>> same problem. Which would make me think the problem is with something
>>> I'm doing, but at the same time I just don't see any problems in my
>>> code - the part dealing with the random number generators is simple
>>> enough. I'm hoping someone here might have some insights into how IDL
>>> initializes the random number generator that might help me figure out
>>> what's going on. For now I've fixed the problem because I pass the
>>> seed back and forth between the two routines (so it only gets
>>> initialized once), but I'd really like to know what's happening. It's
>>> hard for me to post the code because it's part of a routine that relys
>>> on some large data files, but if people think it might help I might be
>>> able to parse it down to something post-able.
>
>> What exactly do you mean by "wasn't giving either one the seed"?

>> Randomu cannot be called without a seed argument, so what were you
>> using?
>
>> Did you see randomu's help for the explanation of how the seed
>> variable works, in all the different ways it can assume?
>
> Sorry. By "didn't give it a seed" I mean, "passed it an undefined
> variable", under which circumstances it initializes the seed itself.
> I've read through randomu's help on how the seed works a number of
> times, including yesterday when I had this problem. This is the first
> time I've ever had any problems using randomu.

If I had to speculate, I'd say that the variable that you're passing
as the seed isn't undefined like you think it is. Have you tried doing
a "help, seed" right before the randomu call?

-Jeremy.
