Subject: Random Number Replication
Posted by dsheerin on Fri, 06 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hi
I have tried to use the following code to generate a sequence of
random numbers:

```
FOR k = 1,  4 DO BEGIN
 p = RANDOMU (S)
 PRINT, p
ENDFOR
```

Why, when I repeat the operation, does the code (nearly) give me the
same list of random numbers shifted by 1? I thought that each time
RANDOMU was called it took a new value for the SEED due to S being
undefined. How do I 'undefine' S after it has been called?

Cheers

David Sheerin

Subject: Re: Random Number Replication
Posted by landsman on Sat, 07 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <37aaad97.5297206@trog.dera.gov.uk>,
  dsheerin@dera.gov.uk (David Sheerin) wrote:
> Hi
> I have tried to use the following code to generate a sequence of
> random numbers:
>
> FOR k = 1, 4 DO BEGIN
> p = RANDOMU (S)
> PRINT, p
> ENDFOR
>
> Why, when I repeat the operation, does the code (nearly) give me the
> same list of random numbers shifted by 1? I thought that each time
> RANDOMU was called it took a new value for the SEED due to S being
> undefined. How do I 'undefine' S after it has been called?

David,

    This is due to a bug in RSI's implementation of the RANDOMU function
in V5.1.1 and V5.2. I believe it has been fixed in V5.2.1 (see below)
but I haven't been able to check V5.2.1 out yet. The bug is that only

during the *first* call to RANDOMU is a seed initialized inside a program. Thus, if a program test.pro contains multiple RANDOMU calls, then on a second call to test.pro, SEED will take the value it had after the first call to RANDOMU during the first call to test.pro. The net effect is that you see a shift by 1, in the output of the random numbers.

Another problem in V5.1 and V5.2 -- I'm not sure if it is related -- is that the seed variable is initialized to the same value at the start of each session rather than the system clock. The bug fix notes in V5.2.1 specifically say that this problem is now fixed:
****

RANDOMN and RANDOMU Function Initialization:

The RANDOMN and RANDOMU functions are now properly initialized with the system time the first time they are used.
****

but I am not sure if your problem is also fixed.

--Wayne Landsman landsman@mpb.gsfc.nasa.gov