Subject: Re: randomn problem
Posted by Paolo Grigis on Tue, 13 Mar 2007 11:03:15 GMT
View Forum Message <> Reply to Message

Ben Panter wrote:
> Nigel Wade wrote:
>
>> Something also changed between IDL 6.1 and IDL 6.2:
>> Maybe the algorithm has been changed to one which propogates more
>> round-off
>> error?
>
> It would seem that it didn't get better in 6.3:
>
> IDL> print, !version
> { x86 linux unix linux 6.3 Mar 23 2006 32 64}
>
> IDL> print, stddev(randomn(seed, 1e8))
> % Compiled module: STDDEV.
> 0.852648
>
> IDL> print, stddev(randomn(seed, 1e8, /double))
> 1.0000419
>
> Ben
>

Ok, let's try to understand what's going on. stddev calls
moment which uses the built-in function total. The moment
algorithm did not change in recent versions. We actually
expect total to fail to sum 1d8 small floats accurately
in single precision, whereas this should be no problem
in double precision. But it looks like in some versions of
IDL double precision is used even when the input is float
and no /double keyword is set. To test this, we can use
the following commands:

```
n=10L^7
a=replicate(0.1,n)
print,'TOTAL A:(SINGLE PREC.):',total(a)
print,'TOTAL A (DOUBLE PREC.): ',total(a,/double)
```

The output is (the exact value will depend on the hardware)

```
TOTAL A:(SINGLE PREC.):  1.08794e+06
TOTAL A (DOUBLE PREC.):       1000000.0
```

in 5.4,5.5, 6.2 and 6.3 and

TOTAL A:(SINGLE PREC.): 1.00000e+06
TOTAL A (DOUBLE PREC.):   1000000.0

in 5.6 and 6.0.

So, I would guess that in these two versions, the total is internally computed in double precision even when /double is not set. Or may this be due to a different way in threading the "total" operation?

Ciao,
Paolo
