Subject: Re: randomn problem
Posted by David Fanning on Sun, 11 Mar 2007 19:41:03 GMT
View Forum Message <> Reply to Message

askemer@gmail.com writes:

> I was playing around with randomn and noticed some weird behavior:
>
> IDL> print, stddev(randomn(seed, 1e7))
>
> I consistently get back numbers around ~0.992. I've tried it on a
> different computer, and the result was not exactly the same, but
> similar. If I change 1e7 to 1e8, the problem gets worse, and I get
> ~0.853. I've tried the syntax with floats, integers, and longs, and I
> still get the same answer. Does anyone know what could be going on?

IDL> print, !version
{ x86 Win32 Windows Microsoft Windows 6.3 Mar 23 2006 32 64}

IDL> print, stddev(randomn(seed, 1e7))
0.999961
IDL> print, stddev(randomn(seed, 1e7, /double))
1.0000060

Don't know. :-(

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: randomn problem
Posted by askemer on Sun, 11 Mar 2007 23:04:39 GMT
View Forum Message <> Reply to Message

Hmm, that's funny. I've tried it on Linux and on OSX. I wonder if
print, stddev(randomn(seed, 1e8)) works on your computer too? If so,
it could be a problem with my memory allocation.

-Andy

Subject: Re: randomn problem
Posted by David Fanning on Mon, 12 Mar 2007 01:10:08 GMT
View Forum Message <> Reply to Message

askemer@gmail.com writes:

> Hmm, that's funny. I've tried it on Linux and on OSX. I wonder if
> print, stddev(randomn(seed, 1e8)) works on your computer too? If so,
> it could be a problem with my memory allocation.

I can't allocate enough memory to make that big of an array
on my system.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: randomn problem
Posted by Ingo von Borstel on Mon, 12 Mar 2007 08:03:02 GMT
View Forum Message <> Reply to Message

Hi,

> I was playing around with randomn and noticed some weird behavior:
>
> IDL> print, stddev(randomn(seed, 1e7))
>
> I consistently get back numbers around ~0.992. I've tried it on a

IDL> print, !version
{ x86 linux unix linux 6.1 Jul 14 2004 32 64}
IDL> print, stddev(randomn(seed,1e7))
1.00015
IDL> print, stddev(randomn(seed,1e7))
1.00013
IDL> print, stddev(randomn(seed,1e7))
0.999803
IDL> print, stddev(randomn(seed,1e7))
0.999735
IDL> print, stddev(randomn(seed,1e7))
0.999860
IDL> print, stddev(randomn(seed,1e7))

```
     0.999887
IDL> print, stddev(randomn(seed,1e7))
      1.00021
IDL> print, stddev(randomn(seed,1e7))
     0.999798
IDL> print, stddev(randomn(seed,1e7))
      1.00019
```

I cannot test it reasonably with 10^8 elements, either.

Sorry, no idea.

Best regards,
Ingo

--
Ingo von Borstel <newsgroups@planetmaker.de>
Public Key: http://www.planetmaker.de/ingo.asc

If you need an urgent reply, replace newsgroups by vgap.

Subject: Re: randomn problem
Posted by Paolo Grigis on Mon, 12 Mar 2007 09:07:42 GMT
View Forum Message <> Reply to Message

The problem does not lie with randomn, but with
stddev. If you compute it using double precision
instead, the problem should solve itsef. Example:

```
a=fltarr(3e7)
a[0:3e7/2]=1.

print,stddev(a)
      0.528791
print,stddev(a,/double)
       0.50000000
```

Ciao,
Paolo

askemer@gmail.com wrote:
> Hi all,
>
> I was playing around with randomn and noticed some weird behavior:
>
> IDL> print, stddev(randomn(seed, 1e7))

>
> I consistently get back numbers around ~0.992. I've tried it on a
> different computer, and the result was not exactly the same, but
> similar. If I change 1e7 to 1e8, the problem gets worse, and I get
> ~0.853. I've tried the syntax with floats, integers, and longs, and I
> still get the same answer. Does anyone know what could be going on?
>
> -Andy
>

Subject: Re: randomn problem
Posted by Nigel Wade on Mon, 12 Mar 2007 16:07:29 GMT
View Forum Message <> Reply to Message

> askemer@gmail.com wrote:
>> Hi all,
>>
>> I was playing around with randomn and noticed some weird behavior:
>>
>> IDL> print, stddev(randomn(seed, 1e7))
>>
>> I consistently get back numbers around ~0.992. I've tried it on a
>> different computer, and the result was not exactly the same, but
>> similar. If I change 1e7 to 1e8, the problem gets worse, and I get
>> ~0.853. I've tried the syntax with floats, integers, and longs, and I
>> still get the same answer. Does anyone know what could be going on?
>>
>> -Andy
>>

Paolo Grigis wrote:

> The problem does not lie with randomn, but with
> stddev. If you compute it using double precision
> instead, the problem should solve itsef. Example:
>
> a=fltarr(3e7)
> a[0:3e7/2]=1.
>
> print,stddev(a)
> 0.528791
> print,stddev(a,/double)
> 0.50000000
>
> Ciao,
> Paolo
>

Something also changed between IDL 6.1 and IDL 6.2:

```
IDL Version 6.1 (linux x86 m32). (c) 2004, Research Systems, Inc.
IDL> print, stddev(randomn(seed,1e8))
% Compiled module: STDDEV.
% Compiled module: MOMENT.
      1.00003

IDL Version 6.2 (linux x86 m32). (c) 2005, Research Systems, Inc.
IDL> print, stddev(randomn(seed,1e8))
% Compiled module: STDDEV.
% Compiled module: MOMENT.
     0.970528
```

Maybe the algorithm has been changed to one which propogates more round-off error?

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555

Subject: Re: randomn problem
Posted by Ben Panter on Mon, 12 Mar 2007 16:17:53 GMT
View Forum Message <> Reply to Message

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

--
Ben Panter, Edinburgh, UK.
Email false, http://www.benpanter.co.uk
or you could try ben at ^^^^^^^^^^^^^^^

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

Subject: Re: randomn problem
Posted by JD Smith on Tue, 13 Mar 2007 16:25:34 GMT
View Forum Message <> Reply to Message

On Sun, 11 Mar 2007 16:04:39 -0700, askemer wrote:

> Hmm, that's funny. I've tried it on Linux and on OSX. I wonder if print,
> stddev(randomn(seed, 1e8)) works on your computer too? If so, it could be
> a problem with my memory allocation.
>
> -Andy

Clearly accumulating round-off:

IDL> print, stddev(randomn(seed, 1e8))
     0.927200

In fact stddev.pro uses MOMENT, which goes to some trouble to avoid
round-off, as follows:

    Resid = X - Mean
;   Var = TOTAL(Resid^2, Double = Double) / (nX-1.0);Simple formula

; Numerically-stable "two-pass" formula, which offers less

; round-off error. Page 613, Numerical Recipes in C.
    Var = (TOTAL(Resid^2, Double = Double) - $
          (TOTAL(Resid, Double = Double)^2)/nX)/(nX-1.0)

Though in this case it doesn't do much to help.

JD
