Subject: Re: Histogram indeterminate results
Posted by Wolf Schweitzer on Sat, 14 Dec 2002 15:29:38 GMT
View Forum Message <> Reply to Message

I can't offer help, but I can offer another view on this which relates to similar errors of IDL.

A while ago, I was intrigued about not finding the correct solution for a couple of trigonometric operations. My results were OFF! I used textbooks. I used Mathematica. I used paper and pencil. I used IDL break points and error visualization. And lo and behold, I found the problem.

Now know that mathematical results can be *simply* numerically unstable if calculated in the wrong data type on IDL.

My solution is to test code items in critical ranges and go down to that level of numerical stability / instability, and have errors or confindence levels, typed out and included in your operations where applicable (some calls to 'minimum' functions, for example, should add for slack of about 3* machine epsilon), and use appropriate data types if necessary. It may mean to alter routines or functions called by your code.

I visualized a crass example here:
http://www.swisswuff.ch/pnphoenix721/html/modules.php?op=mod load&name=News&file=article&sid=11&mode=thre ad&order=0&thold=0

This example is nothing else but trying to find the center of a circle in 3 3d-points, by cutting the planes through the midpoints of the triangle sides (result: centerline) with the triangle plane.

What you see here is the error between floating point ('wrong') and double precision ('correct') solutions. The closer two triangle points of the initial triangle are, the larger the error in getting the centerpoint using floating point variables. Of course, double precision variables also, eventually, will reach a similar limit.

Does this, in any abstract way, relate to your problem?

Wolf Schweitzer
http:/www.swisswuff.ch

Kenneth Bowman wrote:
> In article <ataqhbprr1@skates.gsfc.nasa.gov>,
> thompson@orpheus.nascom.nasa.gov (William Thompson) wrote:
>
>

>> I think the problem is that NBINS should be
>>
>> nbins = nx*ny*nz + 1
>>
>> to accommodate points which fall on exactly the maxima, i.e. points with
>> x=360, y=1, and z=1000, or so close that round-off error makes it look that
>> way. I would change your code to read
>
>
> That is plausible, but the actual values of the input to histogram at
> the time the error occurred are clearly not out of the range [0, nbins]
>
>
>>> IDL> n = nx*ny*LONG((z0 - z_min)/dz) + $
>>> IDL>   nx*LONG((y0 - y_min)/dy) + $
>>> IDL>   LONG((x0 - x_min)/dx)
>>> IDL> print, min(n), max(n), nbins
>>>
>>>
>>> 30   143991   144000
>>
>
>> P.S. I would also check that there isn't some problem with the way the
>> keyword
>> SINE_LAT is handled.
>
>
> I was worried about round-off error, in that calculation in particular,
> but as the diagnosis shows, 0 is LE n and LT nbins.
>
> The really disturbing thing is that running the same program repeatedly
> on the same data generates errors in different places.
>
> Ken