
Subject: Re: Histogram indeterminate results
Posted by [thompson](#) on Thu, 12 Dec 2002 20:11:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth:

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

```
n = nx*ny*LONG((z0 - z_min)/dz) + $
      nx*LONG((y0 - y_min)/dy) + $
      LONG((x0 - x_min)/dx)
if (MIN(n) lt 0) OR (MAX(n) GE nbins) THEN $
MESSAGE, 'Some data out of range'
hh = HISTOGRAM(n, MIN = 0, BINSIZE = 1, NBINS = nbins)
```

(If you lengthen NBINS by one, then change the "GE" to "GT".) That way, you can isolate whether the problem is within HISTOGRAM or in the data that you're passing to it.

William Thompson

P.S. I would also check that there isn't some problem with the way the keyword SINE_LAT is handled.

Kenneth Bowman <k-bowman@null.tamu.edu> writes:

> Since HISTOGRAM is undoubtedly the most important procedure in all of
> IDL, I thought many folks would be interested in this problem. I'll be
> happy if someone tells me that the fault lies with my code rather than
> IDL. I am sending a problem report to RSI.

> Ken Bowman

> This is IDL 5.6 under Mac OS X 10.2.

> My program reads a large number of data files containing the positions
> of points in a finite 3-D volume (x,y,z). It uses histogram to find the
> number of particles in discrete boxes (bins) within the 3-D volume. To
> check the calculations, after each file I compare the number of
> particles with the cumulative total of the histogram. If they do not

> match, the program stops and issues an error message. This is a check
> to make sure the input points are all in the volume.

> Repeatedly running the program on the same input files results in the
> program stopping at different points in the execution (that is, while
> processing different input files).

> Here is the basic code:

```
> h = LONARR(nx, ny, nz)
> nbins = nx*ny*nz
> FOR ifile = 0, nfiles-1 DO BEGIN
>   iid = NCDF_OPEN(infile[ifile])
>   NCDF_VARGET, iid, x_name, x0
>   NCDF_VARGET, iid, y_name, y0
>   NCDF_VARGET, iid, z_name, z0
>   NCDF_CLOSE, iid
>   np = N_ELEMENTS(x0)

>   IF KEYWORD_SET(sine_lat) THEN y0 = SIN(!DTOR*y0)

>   hh = HISTOGRAM(nx*ny*LONG((z0 - z_min)/dz) + $
>         nx*LONG((y0 - y_min)/dy) + $
>         LONG((x0 - x_min)/dx), $
>   MIN = 0, BINSIZE = 1, NBINS = nbins)

>   IF (ROUND(TOTAL(hh, /DOUBLE)) NE np) THEN $
>     MESSAGE, 'Some particles not counted in histogram.'

>   h = h + hh
> ENDFOR
```

> Here is the output and some diagnostics. The program stops at the
> MESSAGE statement in the FOR loop above.

> % MEAN_HIST_XYZ: Some particles not counted in histogram.

```
> IDL> print, ROUND(TOTAL(hh, /DOUBLE)), np
>   399999   400000
```

> The cumulative histogram is less than the number of input points.

> First I check to make sure that the input values are in the correct
> range, $x = \{0, 360\}$, $y = \{-1, 1\}$, and $z = \{0, 1000\}$.

> IDL> print, min(x0), max(x0), min(y0), max(y0), min(z0), max(z0)
> 0.00134277 359.999 -0.999998 0.999993 3.78302e-06
> 1000.00
>
> The input values look OK.

> Next I check to be sure that I am computing the bin indices correctly:

> 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
>

> These are also in the correct range, {0, 144000}.

> Finally, I re-compute the histogram. (Sorry, I should have saved the
> old one for comparison.)

> IDL> hh = HISTOGRAM(nx*ny*LONG((z0 - z_min)/dz) + \$
> IDL> nx*LONG((y0 - y_min)/dy) + \$
> IDL> LONG((x0 - x_min)/dx), \$
> IDL> MIN = 0, BINSIZE = 1, NBINS = nbins)
> IDL> print, ROUND(TOTAL(hh, /DOUBLE)), np
>
> 400000 400000

> This time it (apparently) counts the particles correctly.

> It does not appear to be an I/O problem (no I/O since the error
> occurred). It looks like histogram is producing indeterminate results.

Subject: Re: Histogram indeterminate results
Posted by [K. Bowman](#) on Thu, 12 Dec 2002 23:12:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <ataqhb\$pr\$1@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

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 confidence 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=modload&name=News&file=article&sid=11&mode=thread&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 <ataqhb$pr$1@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
