Subject: Histogram indeterminate results
Posted by K. Bowman on Wed, 11 Dec 2002 17:29:49 GMT
View Forum Message <> Reply to Message

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/0 since the error occurred). It looks like histogram is producing indeterminate results.

Subject: Re: Histogram indeterminate results
Posted by James Kuyper on Sat, 14 Dec 2002 17:03:16 GMT
View Forum Message <> Reply to Message

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

Keep in mind that there's nothing magical about double precision vs. single precision. For a sufficiently nasty situation, even double precision is inadequate; even quad precision can fail. Sometimes the best approach is not to use higher precision, but to find an entirely different approach that's not as much of a strain on the available precision. Unfortunately, it takes a lot of mathematical sophistication to identify such a technique.
