Subject: Re: Help setting up an array
Posted by Craig Markwardt on Wed, 28 Mar 2001 22:49:08 GMT
View Forum Message <> Reply to Message

Peter Thorne <peter.thorne@uea.ac.uk> writes:

> Thanks to everyone who has replied. At nearly 11pm I'm not sure whether
> its exactly what I'm looking for, I shall investigate further tomorrow.
> It seems like as suggested I have been looking at it for too long so
> have tried to explain it in far too much difficulty, sorry. So, I'll
> give a hopefully better example:
>
> 3-D (to keep everyone happy, theoretically could be expected to be 2 to
> 5 dimensional)
>
> Locations array (points within a 3-D ellipsoid)
> x y z (coordinates)
> (1.5,3.4,2.0) point 0
> (3.,-0.5,6.3) point 1
> (1.3,2,.-4.5) point 2
> (-0.1,1.7,0.1) point 3
> .
> .
> .
> .
> (3.1,9.2,-1.4) point npoint
>
> npoint is of order (10,000)
>
> From this I wish to create a say 50x50x50 grid which covers all
> plausible values (found by min and max in each column of the locations
> array).
>
> Then I need to rebin each of these points into the 3-D grid-space, so
> each grid-box has a value which is the number of these original points
> which fall within that grid-box. Other considerations are peripheral,
> the problem arises in this transformation from the locations array to a
> finite difference grid in which the values can be rebinned and how they
> are rebinned.
>
> This may have been covered already, but as my IDL license is at work and
> not home I can't check :(
>
> Thanks again for all the pointers and comments

Yeah, this is indeed a job for HISTOGRAM. What you need to do is check out how HIST_2D is implemented. This can generalize to many dimensions. The one thing that HIST_2D does *not* do is to pass along

the REVERSE_INDICES array, but it is trivial to add this if you need
it (say, if you want to "invert" the histogram and find out which
points fall in a particular bin).

The end result will be something like this (for 3d, but not tested):

```
i = floor((x-xmin)/xstep)
j = floor((y-ymin)/ystep)
k = floor((z-zmin)/zstep)

ijk = i + nx*(j + ny*k)

h = histogram(ijk)
h = reform(h, nx, ny, nz, /overwrite)
```

If you are clever you can generalize this to multi-d. Doing
multi-dimensional histograms, with weighting, is something I've been
intending to do in a program called CMHISTOGRAM. Alas it is only half
written.

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------