Subject: Re: Help setting up an array
Posted by Paul van Delst on Wed, 28 Mar 2001 17:51:56 GMT
View Forum Message <> Reply to Message

As Pavel said, I assume you mean FLTARR(2, 53, 6, 4, 6) when you talk of 5-dimensional space.

To create n-dimensional arrays with n unknown until run time, use MAKE_ARRAY.

To bin up all the values, it sounds like a coordinate conversion might be the easiest? If I understand correctly, and using a 3-D example, what you want to do is bin up values so that all points in a 3-D box (defined by the size of the division per dimension) would be lumped into a (weighted) single element of the re-binned space? Sorta like a multidimensional HISTOGRAM? The coordinate conversion would use the actual array indices along with the division value you pick for each dimension. Maybe you could even use HISTOGRAM? The help doesn't put a limit on the dimensions allowed in the input.... other out there would know better. The HISTOGRAM King, JD, might have an opinion.

If the above is a total load of bollocks compared to what you wnat to do then I will say I found your explanation of the problem difficult to follow - a sure sign you've been staring at it for too long. :o) A simpler example for us regular old 3-d creatures may provide more insight. For yourself as well as others.

paulv

Peter Thorne wrote:
>
> Apologies if this is trivial, but I have been working for about 8 hours
> on it and can't even begin to see how to code it in IDL. Any help very
> gratefully received. I am setting up a function which receives the
> location of points in n-dimensional space for m fields, as well as their
> weights. On call the function does not know the size of any of these
> dimensions.
>
> Simplifying to m=1 (this m dimension should be trivial) the input is:
>
> an array of size (n x npoints) locations of each point in the
> n-dimensional space
>
> a vector of size (npoints) the weights.
>
> Now, what I want to be able to do is re-bin these points into an
> n-dimensional discretized space array which encloses all points. I can
> work out the limits of this space by simply finding min and max in each
> of the n dimensions of the location array. I then need to split this
> grid into nbox n-dimensional discrete boxes (say 50 divisions per
> dimension, boxes need not be of equal size in each dimension) and place

> the respective points in their boxes in this finite representation,
> weighted by weight (trivial).
>
> At present I am having two major problems:
>
> 1. How to declare this discretized array and the limits in n-dimensional
> space of each box into which I bin my values given that I know n and
> nbox.
>
> and doubtless much more difficult to overcome:
>
> 2. How to sensibly code selection criteria to ascertain whether a
> particular value belongs to a particular grid box.
>
> Perhaps it is not possible to code without an a priori knowledge of n -
> the dimensionality of the problem to know how many for loops to use? I
> can't see how where statements could be used.
>
> Any help or useful pointers very gratefully received.
>
> Thanks
>
> Peter

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.