Subject: Re: Irregular grid -> 2D binned
Posted by will[1] on Mon, 24 Jul 2006 00:54:32 GMT
View Forum Message <> Reply to Message

Wow, this actually worked much, much faster than griddata (and is more understandable than the extremely verbose, but not very informative, griddata documentation). Thank you!

Ed Hyer wrote:
> also note that JD smith has written a routine called HIST_ND (get it
> from David Fanning's website), which will do histograms in up to eight
> dimensions, and provide reverse indices.
>
> Your example is rather interesting, in that your initial data set is
> itself a histogram...
>
> Here's how to get with HIST_ND where you're going:
>
> h2=hist_nd(transpose([[data1],[data2]]),[bin1,bin2],min=[min
1,min2],max=[max1,max2],reverse_indices=ri)
> totals2=h2*0; initialize totals
> for i=0l,n_elements(h2)-1 do if(h2[i] ne 0) then
> totals2[i]=total(value[ri[ri[i]:(ri[i+1]-1)]])
>
> I've been doing this so much it's becoming muscle memory.
>
>
>
> will wrote:
>> Excellent, thank you! (and here I was going to do it all so manually).
>>
>> will
>>
>> jgc wrote:
>>> have a look at GRIDDATA and GRID_INPUT functions, this last with
>>> DUPLICATES=Avg
>>>
>>> J.
>>>
>>> will wrote:
>>>> I've sucked it up, I think I need a push in the right direction.
>>>> Imagine that I have a long (~6 million points) list of data that
>>>> includes three fields: latitude, longitude, abundance. The lats and
>>>> lons are all over the place (i.e. no regular grid) and I'd like to bin
>>>> them into an image. Additionally I'd like to average abundance of each
>>>> bin to be the color for the image.
>>>>

>>>> I can use the reverse indices keword in histogram. I can even use mean
>>>> pretty well. I can get a binned histogram using histo_2D. But I'm
>>>> having a hard time thinking about how to go from the reverse indices of
>>>> histogram to the histo_2D case which doesn't offer the same keyword.
>>>> The only way I can think of to do this is to:
>>>>
>>>> a) do a histogram of the latitudes (using RI)
>>>> b) do a histogram of the longitudes (using RI)
>>>> c) find the intersection of indices for each bin that I want
>>>> d) "flatten" the lat, lon, indices/abund cube with matrix
>>>> multiplication
>>>>
>>>> It's the "c" part I'm sketched out on, my brain's can't think around
>>>> anything but a very evil Loop.
>>>>
>>>> Has this really easy or been answered here before? (Array decimation
>>>> was the closest thing I found).
>>>>
>>>> Thanks in advance!
