Subject: Re: Indices ?
Posted by Chris Lee on Thu, 02 Dec 2004 20:44:09 GMT
View Forum Message <> Reply to Message

In article <1102003397.924936.135630@c13g2000cwb.googlegroups.com>,
"Unknown" <sdj@tiscali.it> wrote:
> sensor1 is 4320 lon points by 2160 lat points
> sensor2 is 4096 lon points by 2048 lat points

> My problem lies in finding the correct indices for the 1d arrays. i.e
> the indices: 'valid_lat_s2' ; 'valid_lon_s2' ; 'valid_lat_s1' ;
> 'valid_lon_s1'
> valid = where(data_sensor1 NE land)

lat1 and lon1 will be valid wherever data1 is valid, and the same for
lat2, lon2. So the only question is whether you have the lat and lon as
1d or 2d arrays.

If you have 2d arrays, (then contouring lat1 or lon1 will give you a
series of straight lines of longitude or latitude), then simply use the
result from the WHERE to index the lat and lon arrays, so

valid2=where(data2 is valid) ;pseudo code...
valid2=where(data2 is valid) ;pseudo code...
lon=lon2[valid2]
lat=lat2[valid2]

data=[lon1[valid1], lat1[valid1], data1[valid1]] ;might need a transpose

If your lat and lon arrays are 1d vectors, then you can either use
George's original method of constructing a 2d array, i.e.
>>> lon=b#transpose(1+lonarr(y))
>>> lat=(1+lonarr(x))#transpose(c)

Once you have these, use them as above, valid data points should have
valid lat and longitude points. If you want an alternative method, then
you can use (something like)

lon=lon2[valid2 mod n_lon] ; n_lon = number of longitude points
lat=lat2[long(valid2 / n_lat)]

which may, or may not, work, the same for lon1 and lat1.

You might also want to try TRIANGULATE and TRIGRID, or SPH_SCAT instead
of the INTERP_SPHERE. I dare say they are similar in method and results,
but with the blistering speed (and eccentricities) of TRIANGULATE...

for triangulate, trigrid..

triangulate , lon1[valid1], lat1[valid1], triangles,sphere=s,/degrees
;degrees only if your lat, lon are in degrees...
result = trigrid(lon1[valid1], lat1[valid1], data1[valid1],
triangles,sphere=s, xgrid=lon2[valid2], ygrid=lat2[valid2])

This will interpolate data1 onto data2's grid. If your data2 grid is
regular. Then you can use some of the other keywords in trigrid if you
need to. Or for more fun, forget the WHERE command, and use the MISSING
and MIN_VALUE, MAX_VALUE keywords in trigrid, vis. (assuming the `SST'
value of the land is -1e30 say)

triangulate, lon1, lat1, triangles,sphere=s,/degrees
result = trigrid(lon1, lat1, data1,triangles, xout=lon2, yout=lat2,
missing=LAND_VALUE, MIN_VALUE=LAND_VALUE*0.9)

I forgot my cellphone PIN this morning, but TRIGRID keywords are no
problem.... why ?

Chris.
