Subject: Re: Indices ?
Posted by David Fanning on Wed, 01 Dec 2004 12:58:44 GMT
View Forum Message <> Reply to Message

Pepe writes:

```
> On this very newsgroup Matt McCabe once wrote: " I always get nervous
> posting to this site for fear of asking a particularly stupid
> question....well, here goes."
>
> Well, let me say that I feel the very same way...
```

Oh, well, fear. We'd probably all be better off never getting
out of bed. :-)

```
> I have the following situation:
> a = fltarr(x, y)   ;say satellite sea surface temperature data
> b = fltarr(x)      ;say the longitude points of the grid
> c = fltarr(y)      ;say the latitude points of the grid
>
> data_valid = where(a GT 0.0) ;search for the valid sst values
>
> I need to know the longitude "b(lon_valid)" and latitude
> "c(lat_valid)" points of the valid sst values "a(data_valid)".
> How can I get to them ?
```

It is not totally clear to me exactly why you think
you need them. There will certainly be points on an
X grid (for example) that contain both valid and
invalid points. So, I can't see that having a(data_valid)
would be very helpful in any particular application
that I can envision.

But that said, you can turn your 1-D subscripts
into, for example, column and row subscripts by using
any of the routines mentioned on this newsgroup over
the years, for example:

   http://www.dfanning.com/tips/where_to_2d.html

Or, just use the built-in IDL routine ARRAY_INDICES:

```
   r = Array_Indices(data_valid)
   col_indices = Reform(r[0,*])
   row_indices = Reform(r[1,*])
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Indices ?
Posted by George N. White III on Wed, 01 Dec 2004 17:06:14 GMT
View Forum Message <> Reply to Message

On Wed, 1 Dec 2004, Pepe wrote:

```
> On this very newsgroup Matt McCabe once wrote: " I always get nervous
> posting to this site for fear of asking a particularly stupid
> question....well, here goes."
>
> Well, let me say that I feel the very same way...
>
> I have the following situation:
> a = fltarr(x, y)   ;say satellite sea surface temperature data
> b = fltarr(x)      ;say the longitude points of the grid
> c = fltarr(y)      ;say the latitude points of the grid
>
> data_valid = where(a GT 0.0) ;search for the valid sst values
```

On a cold Canadian December day it would be nice to thnk that somewhere
in the world you can assume SST>0.

> I need to know the longitude "b(lon_valid)" and latitude
> "c(lat_valid)" points of the valid sst values "a(data_valid)".
> How can I get to them ?

David gave one approach. I sometimes find it helpful, e.g,
to apply criteria such as "lon > 0.5*lat", to create
arrays with all rows(lon)/columns(lat) the same:

```
lon=b#transpose(1+lonarr(y))
lat=(1+lonarr(x))#transpose(c)
```

--
George N. White III <aa056@chebucto.ns.ca>
Head of St. Margarets Bay, Nova Scotia, Canada

Subject: Re: Indices ?

Posted by David Fanning on Wed, 01 Dec 2004 17:22:05 GMT
View Forum Message <> Reply to Message

George N. White III writes:

> David gave one approach. I sometimes find it helpful, e.g,
> to apply criteria such as "lon > 0.5*lat", to create
> arrays with all rows(lon)/columns(lat) the same:
>
> lon=b#transpose(1+lonarr(y))
> lat=(1+lonarr(x))#transpose(c)

Oh, that's a good idea. Although I'm still not clear
what you *do* with the information. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Indices ?
Posted by sdj on Thu, 02 Dec 2004 16:03:17 GMT
View Forum Message <> Reply to Message

OK, so let me explain what I need to do and why I think that the
aforementioned information is necessary. Please excuse me if this
message is a tad long...

I have two satellite sensors giving me SST values on two different
grids:
sensor1 is 4320 lon points by 2160 lat points
sensor2 is 4096 lon points by 2048 lat points

I need to interpolate the sensor1 data onto the sensor2 grid. To do
this I have found via this newsgroup a useful routine written way back
in 1994 by Dan Bergmann: interp_sphere.pro

The interp_sphere.pro routine is called in the following way:
IDL> grid = INTERP_SPHERE(lat,lon,data)
where
lat: The latitudes on the grid where interpolated
values are desired (in degrees)
lon: The longitudes on the grid where interpolated

values are desired (in degrees)
data: An array (3,ndata) where ndata is the number of
data points, and can be any number larger than N.
each row of data should contain a longitude, a
latitude, and a value to be interpolated.

Therefore in my case:
lat => (lat_sensor2[valid_lat_s2])
lon => (lon_sensor2[valid_lon_s2])
data =>
 (lat_sensor1[valid_lat_s1],lon_sensor1[valid_lon_s1],data_se nsor1[valid])

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'

The valid index for the data_sensor1 array is easy enough to find:
valid = where(data_sensor1 NE land)

If have a grid_sensor2 array which acts as a land/sea mask, how can I
get the
'valid_lat_s2' and 'valid_lon_s2' indices ?

And finally returning to my original post, how do I relate the 'valid'
indices of the data_sensor1 array to the 'valid_lat_s1' and
'valid_lon_s1' of the lat_sensor1 and lon_sensor1 arrays ?

I hope I have managed not to confuse you too much. I realize that I
might be just complicating my life, but I would rather hope that I
might be almost there...

Again thanks very much for your help.

Best Regards,
Pepe

David Fanning wrote:
> George N. White III writes:
>
>> David gave one approach. I sometimes find it helpful, e.g,
>> to apply criteria such as "lon > 0.5*lat", to create
>> arrays with all rows(lon)/columns(lat) the same:
>>
>> lon=b#transpose(1+lonarr(y))
>> lat=(1+lonarr(x))#transpose(c)
>
> Oh, that's a good idea. Although I'm still not clear

> what you *do* with the information. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Indices ?
Posted by David Fanning on Thu, 02 Dec 2004 16:31:08 GMT
View Forum Message <> Reply to Message

sdj@tiscali.it writes:

> OK, so let me explain what I need to do and why I think that the
> aforementioned information is necessary. Please excuse me if this
> message is a tad long...
>
> I have two satellite sensors giving me SST values on two different
> grids:
> sensor1 is 4320 lon points by 2160 lat points
> sensor2 is 4096 lon points by 2048 lat points
>
> I need to interpolate the sensor1 data onto the sensor2 grid. To do
> this I have found via this newsgroup a useful routine written way back
> in 1994 by Dan Bergmann: interp_sphere.pro
>
> The interp_sphere.pro routine is called in the following way:
> IDL> grid = INTERP_SPHERE(lat,lon,data)
> where
> lat: The latitudes on the grid where interpolated
> values are desired (in degrees)
> lon: The longitudes on the grid where interpolated
> values are desired (in degrees)
> data: An array (3,ndata) where ndata is the number of
> data points, and can be any number larger than N.
> each row of data should contain a longitude, a
> latitude, and a value to be interpolated.
>
> Therefore in my case:
> lat => (lat_sensor2[valid_lat_s2])
> lon => (lon_sensor2[valid_lon_s2])
> data =>
> (lat_sensor1[valid_lat_s1],lon_sensor1[valid_lon_s1],data_se nsor1[valid])

I don't think so. I think maybe you are confusing
the grid with the data that goes *on* the grid.
It seems to me that lat and lon are just
lat_sensor2 and lon_sensor2.

> I hope I have managed not to confuse you too much.

Well, I'm always confused, but I'm not ready to think
I'm wrong, yet. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Indices ?
Posted by sdj on Thu, 02 Dec 2004 18:01:23 GMT
View Forum Message <> Reply to Message

OK, so let me explain what I need to do and why I think that the
aforementioned information is necessary. Please excuse me if this
message is a tad long...

I have two satellite sensors giving me SST values on two different
grids:
sensor1 is 4320 lon points by 2160 lat points
sensor2 is 4096 lon points by 2048 lat points

I need to interpolate the sensor1 data onto the sensor2 grid. To do
this I have found via this newsgroup a useful routine written way back
in 1994 by Dan Bergmann: interp_sphere.pro

The interp_sphere.pro routine is called in the following way:
IDL> grid = INTERP_SPHERE(lat,lon,data)
where
lat: The latitudes on the grid where interpolated
values are desired (in degrees)
lon: The longitudes on the grid where interpolated
values are desired (in degrees)
data: An array (3,ndata) where ndata is the number of
data points, and can be any number larger than N.
each row of data should contain a longitude, a

latitude, and a value to be interpolated.

Therefore in my case:
lat => (lat_sensor2[valid_lat_s2])
lon => (lon_sensor2[valid_lon_s2])
data =>
 (lat_sensor1[valid_lat_s1],lon_sensor1[valid_lon_s1],data_se nsor1[valid])

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'

The valid index for the data_sensor1 array is easy enough to find:
valid = where(data_sensor1 NE land)

If have a grid_sensor2 array which acts as a land/sea mask, how can I
get the
'valid_lat_s2' and 'valid_lon_s2' indices ?

And finally returning to my original post, how do I relate the 'valid'
indices of the data_sensor1 array to the 'valid_lat_s1' and
'valid_lon_s1' of the lat_sensor1 and lon_sensor1 arrays ?

I hope I have managed not to confuse you too much. I realize that I
might be just complicating my life, but I would rather hope that I
might be almost there...

Again thanks very much for your help.

Best Regards,
Pepe

David Fanning wrote:
> George N. White III writes:
>
>> David gave one approach. I sometimes find it helpful, e.g,
>> to apply criteria such as "lon > 0.5*lat", to create
>> arrays with all rows(lon)/columns(lat) the same:
>>
>> lon=b#transpose(1+lonarr(y))
>> lat=(1+lonarr(x))#transpose(c)
>
> Oh, that's a good idea. Although I'm still not clear
> what you *do* with the information. :-)
>
> Cheers,
>

> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Indices ?
Posted by tam on Thu, 02 Dec 2004 18:54:04 GMT
View Forum Message <> Reply to Message

sdj@tiscali.it wrote:

> OK, so let me explain what I need to do and why I think that the
> aforementioned information is necessary. Please excuse me if this
> message is a tad long...
>
> I have two satellite sensors giving me SST values on two different
> grids:
> sensor1 is 4320 lon points by 2160 lat points
> sensor2 is 4096 lon points by 2048 lat points
>
> I need to interpolate the sensor1 data onto the sensor2 grid. To do
> this I have found via this newsgroup a useful routine written way back
> in 1994 by Dan Bergmann: interp_sphere.pro
>
> The interp_sphere.pro routine is called in the following way:
> IDL> grid = INTERP_SPHERE(lat,lon,data)
> where
> lat: The latitudes on the grid where interpolated
> values are desired (in degrees)
> lon: The longitudes on the grid where interpolated
> values are desired (in degrees)
> data: An array (3,ndata) where ndata is the number of
> data points, and can be any number larger than N.
> each row of data should contain a longitude, a
> latitude, and a value to be interpolated.
>
> Therefore in my case:
> lat => (lat_sensor2[valid_lat_s2])
> lon => (lon_sensor2[valid_lon_s2])
> data =>
> (lat_sensor1[valid_lat_s1],lon_sensor1[valid_lon_s1],data_se nsor1[valid])
>
> 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'

>
> The valid index for the data_sensor1 array is easy enough to find:
> valid = where(data_sensor1 NE land)
>
> If have a grid_sensor2 array which acts as a land/sea mask, how can I
> get the
> 'valid_lat_s2' and 'valid_lon_s2' indices ?
>
> And finally returning to my original post, how do I relate the 'valid'
> indices of the data_sensor1 array to the 'valid_lat_s1' and
> 'valid_lon_s1' of the lat_sensor1 and lon_sensor1 arrays ?
>
> I hope I have managed not to confuse you too much. I realize that I
> might be just complicating my life, but I would rather hope that I
> might be almost there...
>
> Again thanks very much for your help.
>
> Best Regards,
> Pepe
>
>

So you have something like (with presumably higher resolution!)

```
   lon0 = findgen(360)
   lat0 = findgen(181)-90
   nx   = 360

   ; Your dimensions are different... Let's make the number of lons, nx.

   data = fltarr(360,181)

   w = where(... data is valid ...)

   lon1 = a set of longitudes you want to interpolate to
   lat1 = a set of latitudes you want to interpolate to

   ox = n_elements(lon1)
   oy = n_elements(lat1)

   a = fltarr(oy)+1
   olon1 = lon1 # a ;  Creates a 2D array of longitudes for each output point
   a = fltarr(ox)+1
   olat1 = a # lat1 ;  Ditto for lats

   ; Don't know if interp_sphere will handle 2-D output coordinates.
   ; if not then...
```

```
        olon1 = reform(olon1, ox*oy)
        olat1 = reform(olat1, ox*oy)

    odata = interp_sphere(olon1, olat1, [lon0[w mod nx], lat0[w /nx], data[w]])

        odata = reform(odata, ox, oy) ; if needed to get back to a 2-d array
```

Done...

Regards,
Tom McGlynn

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.