
Subject: Re: Map Projections and Contour Plots.
Posted by [Martin Schultz](#) on Wed, 16 Jun 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Grady Daub wrote:

>
> Martin Schultz wrote:
>

> Your convert_lon program was not attached.

sorry! this time it should be ...

>
> Would your program make my non-monotonic lat,lon,data arrays in a form expected by CONTOUR ?

no. it just converts lon value by lon value to fall into the range of -180 to +180 or 0 to 360. You would need to use the SORT function to get monotonic values first.

>
> I've tried TRIANGULATE and TRIGRID and SPH_SCAT (the latter exactly as shown on [dfanning.com](#), except with my own data)
> and the results are weird. Is SPH_SCAT, when applied to data with max/min of around 200/-200, supposed to produce
> output past 50000? That's the weird part.

I haven't used triangulate and the likes for a long time, and I must have deleted the code where I last did it. But I remember it had helped me a great deal to actually plot the triangles returned by triangulate in order to believe what IDL (and I) were doing...

(The CELL_FILL question was already answered by wmc)

Regards,
Martin

--

```
|||||-----//|  
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,  
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551  
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/  
; $Id: convert_lon.pro,v 1.11 1999/05/20 00:58:48 mgs Exp $
```

```
-----  
;+  
; NAME:  
; CONVERT_LON  
;  
; PURPOSE:
```

```

; Convert longitudes from -180..180 to 0..360 or vice
; versa.
;
;
; CATEGORY:
; Tools
;
; CALLING SEQUENCE:
; CONVERT_LON,data,names,Pacific=Pacific,Atlantic=Atlantic, $
; minval=minval
;
; INPUTS:
; DATA -> A data array (lines,vars) or vector containing
; longitude data. If DATA is a 2D array, the NAMES
; parameter must be given to identify the LONGitude variable.
;
; NAMES -> A string list of variable names. The longitude data
; must be labeled 'LON', unless specified with the LONNAME
; keyword. The NAMES parameter is not needed, if a data
; vector is passed.
;
; KEYWORD PARAMETERS:
; PACIFIC -> Convert longitudes from -180..180 to 0..360
;
; ATLANTIC -> Convert from 0..360 to -180..180
;
; LONNAME -> Name of the longitude variable if a name other
; than 'LON' is used.
;
; OUTPUTS:
; The longitude column in the data array will be changed.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;
; EXAMPLE:
; londat = [ -180.,-179.,-0.1,0.1,179.,180.,270.,359.]
; CONVERT_LON,londat,/Pacific
; print,londat
;
; CONVERT_LON,londat,/Atlantic
; print,londat
;
; MODIFICATION HISTORY:
; mgs, 25 Aug 1998: VERSION 1.00
; mgs, 19 May 1999: - now makes sure that longitude range does

```

```

;           not exceed -180..180 or 0..360
;
;
;-
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine convert_lon"
;----- --

pro convert_lon,data,names,pacific=pacific,atlantic=atlantic, $
    lonname=lonname

    minval = -180.0001

    if (n_elements(lonname) eq 0) then lonname = 'LON'

    if (n_elements(data) lt 2) then return

; get size information of data and find LON column
s = size(data)
if (s[0] eq 1) then ind = 0 $ ; data is vector
else begin
    ; Find LON variable
    ind = where(strucase(names) eq lonname)
    if (ind[0] lt 0) then begin
        print, '*** CONVERT_LON: Cannot find ',lonname,' in data set!'
        return
    endif
endelse

; Atlantic: Convert longitudes greater 180 by subtracting 360
; also add N*360 to longitude values less than -180
if (keyword_set(Atlantic)) then begin
    repeat begin
        lon = data[* ,ind[0]]
        index = where(lon gt 180.,count)
        if (index[0] ge 0) then data[index,ind[0]] = lon[index]-360.
    endrep until(count eq 0)
    repeat begin

```

```
lon = data[* , ind[0]]
index = where(lon lt -180., count)
if (index[0] ge 0) then data[index, ind[0]] = lon[index]+360.
endrep until(count eq 0)
endif
```

```
; Pacific: convert negative longitudes by adding 360
; also subtract N*360 for longitude values greater than 360
if (keyword_set(Pacific)) then begin
  repeat begin
    lon = data[* , ind[0]]
;   index = where(lon gt minval AND lon lt 0.)
    index = where(lon lt 0., count)
    if (index[0] ge 0) then data[index, ind[0]] = lon[index]+360.
  endrep until(count eq 0)
  repeat begin
    lon = data[* , ind[0]]
    index = where(lon gt 360., count)
    if (index[0] ge 0) then data[index, ind[0]] = lon[index]-360.
  endrep until(count eq 0)
endif
```

```
return
end
```

File Attachments

1) [convert_lon.pro](#), downloaded 128 times
