Subject: Re: [Q] map_grid and rotated maps
Posted by marq on Thu, 20 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Hello,

after looking through the source code of RSI's map_set.pro, I finally found a solution working for me. The problem (from my point of view) in the map_grid part seems to be that the minimum/maximum longitudes and latitudes which are used to determine the meridians and parallels to be drawn are taken from the !map.out variable. That is certainly correct if the entire area can be seen with the current projection. If parts of the earth's surface are not visible, however, the !map.out elements used seem to describe something like the 'visible area'.

I've extracted the map_grid part from map_set.pro and modified it a bit; when calling the new routine with the same arguments as map_grid, the result should be same. I've added keywords that allow the specification of a latitude and a longitude which are forced to be part of the grid, plus the possibility to specify a region of the globe where a grid is to be drawn.

For those interested, I'll append my stuff to this posting. I would be happy to receive informations about bugs, suggestions, improvements,...

Regards,

Chris Marquardt (marq@strat01.met.fu-berlin.de)

---------- <snip here> ----------

```
pro map_grd, equator = equator, flat = flat, greenwich = greenwich, flon = flon,    $
          limit = limit, no_extend = no_extend, latdel = latdel, londel = londel, $
          glinestyle = glinestyle, glinethick = glinethick,   $
          label = label, lonlab = lonlab, latlab = latlab, lonalign = lonalign,$
          latalign = latalign, charsize = charsize, color = color, t3d=t3d, $
          no_grid=no_grid, zvalue=zvalue,      $
          whole_map=whole_map
;+
; NAME:
;    MAP_GRD
;
; PURPOSE:
;    Put a grid on a previously established map projection.
;
; CATEGORY:
;    Mapping.
```

```
;
; CALLING SEQUENCE:
;   MAP_GRD, ...
;
; INPUTS:
;   MAP_GRD accepts all inputs and parameters which are accepted by
;   map_grid. In addition, the following inputs and keywords are allowed:
;
; OPTIONAL INPUTS:
;   FLAT:      Forces FLAT to be one of the grid's latitudes.
;   FLON:      Forces FLON to be one of the grid's longitudes.
;   LIMIT:     A four or eight element vector.
;              If a four element vector, [Latmin, Lonmin, Latmax, Lonmax]
;               specifying the boundaries of the grid to be drawn
;               (Latmin, Lonmin) and (Latmax, Lonmax) are the latitudes
;               and longitudes of two diagonal points on the boundary
;               with Latmin < Latmax and Lonmin < Lonmax.
;              If an eight element vector: [ lat0, lon0, lat1, lon1, lat2,
;               lon2, lat3, lon3] specify four points on the map which
;               give, respectively, the location of a point on the left edge,
;               top edge, right edge, and bottom edge of the grid extent.
;              Note: LIMIT overwrites FLON and FLAT.
;
; KEYWORD PARAMETERS:
;   /EQUATOR:   Draw the equator (same as FLAT = 0.).
;   /GRENNWICH: Draw the Greenwich meridian (same as FLON = 0.).
;   /NO_EXTEND: Don't let meridians extend the range where latitudes
;               are drawn. LIMIT implies /NO_EXTEND.
;
; OUTPUTS:
;   None.
;
; OPTIONAL OUTPUTS:
;   None.
;
; COMMON BLOCKS:
;   None.
;
; SIDE EFFECTS:
;   None.
;
; RESTRICTIONS:
;   None.
;
; PROCEDURE:
;   The routine map_grid was extracted from RSI's map_set.pro and
;   somewhat modified. You might want to know their copyright:
;
```

```
;    Copyright (c) 1993-1994, Research Systems, Inc.  All rights reserved.
;         Unauthorized reproduction prohibited.
;
; EXAMPLE:
;    To get the same as with map_grid, <options>:
;         map_grd, <options>
;
;    To draw a grid onto an orthographic projection centered above
;    Boulder, CO, try
;         map_set, 40.02, -105.16, /iso, /orthographic, /continents
;         map_grd, /equator, /greenwich
;    Overlay a high resolution grid covering central US with
;         map_grd, limit = [35.,-115.,50,-55.], latdel=5., londel=5.
;
;    To draw a polar stereographic map of the southern hemisphere with
;    latitudes in a distance of 20 degrees, Africa on top but without
;    letting the meridians converge at the south pole, try
;         map_set, -90., 30., /iso, /stereographic, /continents
;         map_grd, latdel=20., /equator, /greenwich, /no_extend
;
; MODIFICATION HISTORY:
;
;       Thu Jun 20 12:19:41 1996, Christian Marquardt <marq@kassandra>
;
;  Created.
;
;-
;
if (!x.type NE 2) THEN $  ; make sure we have mapping coordinates
   message, 'map_grd---Current ploting device must have mapping coordinates'

; no grid? - in case someone wants just to put labels
if not keyword_set(no_grid) then no_grid = 0

; T3D support
if n_elements(t3d) le 0 then t3d = 0
if n_elements(zvalue) eq 0 then zvalue = 0

; Get lat/lon ranges from !MAP (to get nice grid spacings)
lonmin = !map.out(2) & lonmax1 = !map.out(3)
if lonmax1 le lonmin then lonmax1 = lonmax1 + 360.
latmin = !map.out(4) & latmax = !map.out(5)

;Default grid spacings...
if n_elements(latdel) eq 0 then latd = 1 else latd = latdel
if n_elements(londel) eq 0 then lond = 1 else lond = londel

if n_elements(latdel) eq 0 then latdel = map_grid_incr(latmax-latmin)
```

```
if n_elements(londel) eq 0 then londel = map_grid_incr(lonmax1-lonmin)

; Force some latitudes/longitudes to be contained in the grid
if keyword_set(equator) then flat = 0.       ; Draw equator
if keyword_set(greenwich) then flon = 0.     ; Draw Greenwich meridian

; Get lat/lon ranges if /equator and/or /greenwich are set
if n_elements(flat) eq 1 then begin
   lats = [reverse(flat - latdel*findgen(90./latdel + 1)),  $
                flat + latdel*(1. + findgen(90./latdel + 1))]
   lats = lats(where(lats ge -90. and lats le 90.))
   latmin = min(lats, max = latmax)
endif

if n_elements(flon) eq 1 then begin
   lons = [reverse(flon - londel*findgen(360./londel + 1)),  $
                flon + londel*(1. + findgen(360./londel + 1))]
   lons = lons(where(lons ge 0. and lons le 360.))
   lonmin = min(lons, max = lonmax) & lonmax1 = lonmax
endif

; Get lat/lon ranges if limits are specified
if n_elements(limit) eq 4 then begin
   latmin = limit(0) & latmax = limit(2)
   lonmin = limit(1) & lonmax = limit(3) & lonmax1 = lonmax
   no_extend = 1
endif else if n_elements(limit) eq 8 then begin
   latmin = min(limit([0, 2, 4, 6]), max = latmax)
   lonmin = min(limit([1, 3, 5, 7]), max = lonmax) & lonmax1 = lonmax
   no_extend = 1
endif

; if WHOLE_MAP specified, or the deltas are < 1,
; do not convert the limits into integers
if (not keyword_set(whole_map)) then begin
   if abs(latmax - latmin) gt 5. and latd ge 1 then begin ;Make range integers
 latmin = float(floor(latmin))
 latmax = ceil(latmax)
 endif
   if abs(lonmax1 - lonmin) gt 5 and lond ge 1 then begin
 lonmin = float(floor(lonmin))
 lonmax1 = ceil(lonmax1)
 endif
endif

if N_Elements(glinestyle) EQ 0 THEN glinestyle =1
if N_Elements(glinethick) EQ 0 THEN glinethick =1
```

```
if n_elements(color) le 0 then begin ;Default color?
 if (!d.flags and 512) ne 0 then color = 0 else color = !d.n_colors-1
 endif

if N_Elements(label) NE 0 OR (N_ELEMENTS(Latlab) ne 0) $
 OR (N_Elements(LonLab) NE 0) THEN BEGIN
 printno = 1
 printno2 = 1
 if N_Elements(Latlab) eq 0 THEN Latlab = (lonmin + lonmax1)/2
 if N_ELements(LonLab) eq 0 THEN LonLab = (latmin +latmax)/2
endif ELSE BEGIN
 printno = -1
 printno2 = -1
ENDELSE
                                ; of grid numbers
if n_elements(latalign) eq 0 THEN latalign = .5 ;Text alignment of lat labels
if n_elements(lonalign) eq 0 THEN lonalign = .5 ;Text alignment of lon labels
if n_elements(charsize) eq 0 THEN charsize =  1

step = 4 < (latmax - latmin)/10.
len = long((latmax-latmin) / step + 1)

if keyword_set(no_extend) then begin
   lati = (float(latmax-latmin) / (len-1.)) * findgen(len) + latmin ;lats
endif else begin
   lati = (float(180.) / (len-1.)) * findgen(len) - 90. ;lats
endelse


First = 1

for lon = lonmin, lonmax1, londel do begin
    if (lon lt -180) then lon2 =lon +360  $
     else if (lon gt 180) then lon2 = -360 +lon $
     else lon2 = lon
    pres = convert_coord(lon, latmin,/to_norm)
    pres = !map.out(0:1)
    pres1 = convert_coord(lon, latmax,/to_norm)
    pres1 = !map.out(0:1)
    lon1 = lon
    if First eq 1  THEN First = 0 else $
      if abs(pres(0) - past(0)) GE !map.out(6)   OR  $
         abs(pres(1) - past(1)) GE !map.out(7)   OR  $
         abs(pres1(0) - past1(0)) GE !map.out(6) OR  $
         abs(pres1(1) - past1(1)) GE !map.out(7)     $
       THEN BEGIN
        if(lon ge 0) then dd = .0001 else dd = -.0001
        lon1 = lon - dd
```

```
      ENDIF
    past = pres
    past1 = pres1

    if (not no_grid) then plots, lon1, lati, zvalue, $
 color = color, t3d=t3d, NOCLIP=0,linestyle=glinestyle,thick=glinethick

    if lon2 ne long(lon2) then fmt = '(f7.2)' else fmt = '(i4)'
    if (printno eq 1) and $  ;Dont repeat -180....
 ((lonmin ne -180) or (lonmax1 ne 180) or (lon ne -180)) then $
 xyouts,lon, LonLab, z=zvalue, ali=lonalign, t3d=t3d, color=color,$
          strtrim(string(lon2,format=fmt),2), charsize = charsize
    printno  = 1 - printno
 endfor

step = 4 < (lonmax1 - lonmin)/10.
len = (lonmax1-lonmin)/step + 1
loni = findgen(len)*step + lonmin

if (loni(len-1) NE lonmax1) THEN  BEGIN
    loni = [loni, lonmax1]
    len = len + 1
ENDIF

for lat = float(latmin), latmax, latdel do begin
   if lat ne long(lat) then fmt = '(f7.2)' else fmt = '(i4)'
   if printno2 eq 1 then xyouts,latlab,lat, z=zvalue, ali=latalign, t3d=t3d, $
 strtrim(string(lat, format=fmt),2), charsize = charsize, color=color
   printno2 = 1 - printno2
   if (not no_grid) then plots,loni, lat, zvalue, $
 NOCLIP=0,linestyle=glinestyle,color = color, thick=glinethick, t3d=t3d
 endfor

end
```