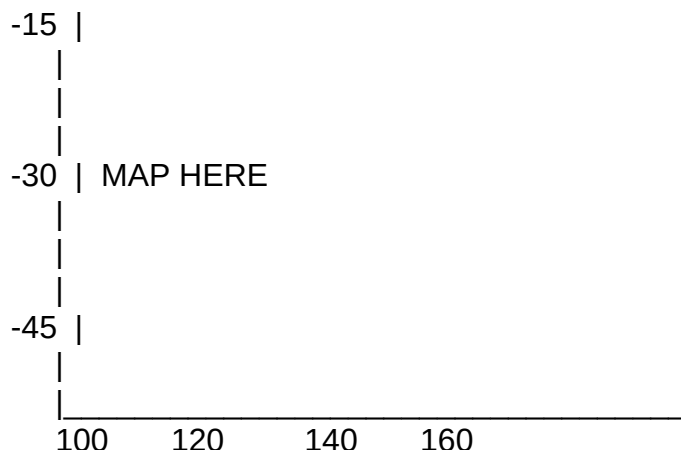

Subject: Q: Labeling axes when using MAP_SET
Posted by [morwenna](#) on Tue, 09 May 1995 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am new to this group and to IDL.

What I want is something like this:



I can draw the map with
`map_set,/mercator,/grid,/continents`
and add labels INSIDE the plot with
`map_set,/mercator,/grid,/continents,/label`
and move them around WITHIN the plot with `/latlab` and `/lonlab`
but I can't work out how to label the axes.

I have tried doing things like:

```
!y.ticks=4  
!y.tickv(0)=85.  
!y.tickv(1)=50.  
!y.tickv(2)=0.  
!y.tickv(3)=-50.  
!y.tickv(4)=-85.  
!y.tickname = ['85 ', '50', '0', '-50', '-85']  
map_set,/mercator,/grid,/continent
```

but I get it all squashed up along one edge :-(

If anyone can help me I would be most grateful.

While I am here, I know that there are problems with filling contours which are plotted over a map. Has anyone come up with a solution?

Morwenna

morwenna@cyclone.maths.monash.edu.au

Subject: Re: Q: Labeling axes when using MAP_SET

Posted by [afl](#) on Tue, 09 May 1995 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

This procedure performs labeling of lat/lon points around the edge of a map. It is pretty specific to the way I prefer things, but it may help you with your problem as well. Good luck with it. It is a bit verbose, but seems to do the job with a variety of map projections...

```
; Procedure to label the lat/lon values around the boundary of a map.
```

```
;
```

```
; Originator: Andrew F. Loughe (2/22/95)
```

```
pro map_label, color=color, charsize=charsize, $  
    lhs=lhs, rhs=rhs, $  
    latdel=latdel, londel=londel, $  
    nudgex=nudgex, nudgey=nudgey, help=help
```

```
; Print a help message to the IDLTERM.
```

```
if (n_elements(help) gt 0) then begin
```

```
    message, ' map_label, color=color, charsize=charsize,' + $
```

```
        ' lhs=lhs, rhs=rhs, latdel=latdel, londel=londel,' + $
```

```
        ' nudgex=nudgex, nudgey=nudgey, help=help'
```

```
endif
```

```
; Set some defaults
```

```
if (n_elements(color) eq 0) then color = !p.color
```

```
if (n_elements(charsize) eq 0) then charsize = 0.8
```

```
lhs = keyword_set (lhs)
```

```
rhs = keyword_set (rhs)
```

```
if (lhs eq 0 and rhs eq 0) then lhs = 1
```

```
if (n_elements(latdel) eq 0) then latdel = 30
```

```
if (n_elements(londel) eq 0) then londel = 30
```

```
if (n_elements(nudgex) eq 0) then nudgex=0
```

```
if (n_elements(nudgey) eq 0) then nudgey=0
```

```
on_error, 2
```

```
; Get lat/lon boundaries of the map.
```

```
lonmin = !map.out(2)
```

```
lonmax = !map.out(3)
```

```
latmin = !map.out(4)
```

```
latmax = !map.out(5)
```

```
if (lonmin eq lonmax) then lonmax = lonmin + 360.
```

```
DX = abs (lonmax - lonmin)
```

```
; Determine character height and width. Apply charsize.
```

```
char_ht = convert_coord([0,!d.y_ch_size],/device,/to_norm)
```

```
char_ht = char_ht(1) * 1.0
```

```
if (!d.name ne 'X' and charsize gt 1.) then $
```

```
    char_ht = char_ht * charsize
```

```
char_wd = convert_coord([0,!d.x_ch_size],/device,/to_norm)
```

```
char_wd = char_wd(1)
```

```
; Nudging factor (convert from data to normalized)
```

```
y_avg = .5*(latmin + latmax)
```

```
nudgex1 = convert_coord( [nudgex,y_avg], /data, /to_norm )
```

```
nudgex2 = convert_coord( [nudgex+nudgex,y_avg], /data, /to_norm )
```

```
nudgex = nudgex2 - nudgex1
```

```
nudgex = nudgex(0)
```

```
x_avg = .5*(lonmin + lonmax)
```

```
nudgey1 = convert_coord( [x_avg,nudgey], /data, /to_norm )
```

```
nudgey2 = convert_coord( [x_avg,nudgey+nudgey], /data, /to_norm )
```

```
nudgey = nudgey2 - nudgey1
```

```
nudgey = nudgey(1)
```

```
; Test to see how close the lower longitude points are.
```

```
; If they are too close, then place longitude labels along EQ.
```

```
xypos1 = convert_coord([lonmin, latmin], /data, /to_norm)
```

```
xypos2 = convert_coord([lonmin+DX/2., latmin], /data, /to_norm)
```

```
bottom = 'yes'
```

```
if ( abs(xypos1(0) - xypos2(0)) lt .2 ) then bottom='no'
```

```
; Plot longitude labels along BOTTOM boundary of the map.
```

```
for i = fix(lonmin), fix(lonmax) do begin
```

```
    if ( (i mod londel) eq 0 ) then begin
```

```
        ii = fix(abs(i))
```

```
        if (i gt 180) then ii = abs(360 - ii)
```

```
; Determine text of longitude label.
```

```
    append = "
```

```
    if (i gt 0 and i lt 180) then append='E'
```

```
    if (i lt 0 or i gt 180) then append='W'
```

```
    if (i gt 360 and i lt 540) then append='E'
```

```
    if (i gt 540 and i lt 720) then append='W'
```

```
    if ( (abs(i) mod 180) eq 0 ) then append=""
```

```
    if ( (abs(i) mod 360) eq 0 ) then ii='0'
```

```
    label = strcompress(string(ii), /rem) + append
```

```

; Determine where to place longitude label.
  xypos = convert_coord([i, latmin], /data, /to_norm)
  x = xypos(0)
  y = !y.window(0) - (char_ht)*1.2 + nudgey

; Some projections have longitude labels at EQ.
  if ( bottom eq 'no' ) then begin
    xypos = convert_coord([i, .5], /data, /to_norm)
    x = xypos(0)
    y = xypos(1)
    if ( x ge !x.window(0)+char_wd*2. and $
        x le !x.window(1)-char_wd*2. and $
        y ge !y.window(0)+char_ht and $
        y le !y.window(1)-char_ht ) then $
      xyouts,x,y,label,charsize=charsize,color=color,align=.5,/nor m

    endif else begin

; Plot longitude labels at bottom of the map.
    if ( x ge !x.window(0)+char_wd*2. and $
        x le !x.window(1)-char_wd*2. ) then $
      xyouts,x,y,label,charsize=charsize,color=color,align=.5,/nor m
    endelse
  endif

endfor ; i

; Plot latitude labels along LEFT or RIGHT boundary of the map.

for i = latmin, latmax do begin
  if ( (i mod latdel eq 0) ) then begin
    ii = fix(abs(i))

; Determine text of latitude label.
    append=""
    if (i lt 0) then append='S'
    if (i gt 0) then append='N'
    if (i eq 0) then ii='EQ'
    label = strcompress(string(ii), /rem) + append

; Determine where to place latitude label.

; Work in from the far left to find the y position.
    if (lhs eq 1) then begin
      for x1 = lonmin-10., lonmax, .1 do begin

```

```

        xypos = convert_coord( [x1, i], /data, /to_norm)
        xypos2 = xypos
        xypos2(0) = xypos2(0) - char_wd
        xypos2(1) = xypos2(1) + char_ht
        if (xypos2(0) ge !x.window(0) and xypos2(1) ge $
            !y.window(0) and xypos2(1) le !y.window(1)) $
            then goto, jump2
    endfor

; Work in from the far right to find the y position.
    endif else begin
        for x1 = lonmax+10., lonmin, -.1 do begin
            xypos = convert_coord( [x1, i], /data, /to_norm)
            xypos2 = xypos
            xypos2(0) = xypos2(0) + char_wd
            xypos2(1) = xypos2(1) + char_ht
            if (xypos2(0) le !x.window(1) and xypos2(1) ge $
                !y.window(0) and xypos2(1) le !y.window(1)) $
                then goto, jump2
        endfor
    endelse
jump2: y = xypos(1) - (char_ht*.25)

; Move end latitudes around a bit.
    if (i eq latmax) then y = xypos(1) - (char_ht * .2)
    if (i eq latmin) then y = xypos(1)

; Find lefthand or righthand side of the plot boundary.
    if (lhs eq 1) then x = !x.window(0) - (char_wd*.5) + nudgex
    if (rhs eq 1) then x = !x.window(1) + (char_wd) + nudgex

; Plot latitude label.
    xyouts, x, y, label, charsize=charsize, color=color, $
        align=(lhs eq 1), /norm

    endif
endfor ; i

end

--

```

Andrew F. Loughe
 University of Colorado, CIRES
 Campus Box 449

email: afl@cdc.noaa.gov
 voice: (303) 492-0707
 fax: (303) 497-7013

Boulder, CO 80309-0449 USA
