
Subject: Re: PVWAVE and NCAR mapping
Posted by [keller](#) on Tue, 24 Sep 1991 15:41:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <LEGLER.91Sep23113234@masig2.masig1.ocean.fsu.edu>
legler@masig2.masig1.ocean.fsu.edu (David M. Legler) writes:

>
> I need some help drawing some geographical outlines on a PVWave plot.
> Something like what NCAR graphics can do. For example, lets say a
> sample plot of the US outline in black with an overlay of colors
> indicating temperatures for a specific month. NCAR has a very nice
> geographical map drawing facility and our sales rep says he thinks
> others have already written code to use their data base for map
> drawing. Is that true, please speak up if so.
>
> Thanks,
>
> Mr. David M. Legler ||(904)644-3797
> Mesoscale Air-Sea Interaction ||Arpa legler@masig1.ocean.fsu.edu(128.186.3.1)
> Group ||SPAN SCRI1::"legler@masig1.ocean.fsu.edu"
> MS B-174 Love-012 ||Bitnet legler%masig1.ocean.fsu.edu@fsuavm
> Florida State University ||
> Tallahassee, FL 32306-3041 || "An Apple II a day, keeps the PC blues away"
>

BELOW is an IDL map program that I use. It draws a 'pseudo mercator'
map: lats/lons are parallel and equidistant.
I've included a subroutine that converts lats/lons to Lambert Conformal.
In DOMAP are two commented out lines that apply the conversion.

The algorithm in words: Find a map data base (I'm assuming you can
acquire one).

This will probably be divided into 'line segments'...a series of
lat/lon pairs that are to be connected 'pen down'. There is also
likely to be associated with each line segment the max and min
lat/lon for each line segment. This is used to determine whether
or not you wish to even consider drawing the line. This data
may come separately from the line segments, or may be listed
within the segment as if it were the first two points:

43.00 -89.00 41.00 -91.00 42.12 -89.34 41.56 -89.78.....
^^^^ ^^^^^ ^^^^^ ^^^^^

MaxlatMinlonMinlat Maxlon

These are your data structures to work with.

The DOMAP code looks intimidating because 1) I make the user key in
POSITIVE west longitudes, and I jump thru hoops to convert to negative.
2) I set/preserve a bunch of system variables.

Look for 'MEAT OF THE PROGRAM'

Note your 'screen' will be in lat/lon coordinates, with negative longitudes for the U.S. I leave it that way, so that the user can follow up and plot something on it.

Following the 'meat' is 'LAST LINE SEGMENT', which is the same code, with concessions to the problem of the last subscript. This is the 'last line segment' of the entire data base, so, it's probably not drawn on any given map anyway.

To overlay something on top: This is tricky only in that you have to think. (If someone knows better, tell us all!).

I'll assume that you have a grid that is one gridpoint at every lat/lon square. Big assumption! What you need to do is to reset your coordinate system to something that the grid knows: 'grid coordinates'! Example follows:

```
avetemps = fltarr(41,26)
;CODE TO FILL THIS ARRAY HERE
domap,25,50,65,105      ;MY FAVORITE MAP...NOTE THAT IT MATCHES THE
                        ;DIMENSIONS OF THE GRID
set_xy,0,40,0,25        ;SET COORDINATE SYSTEM FOR GRID
!noeras = 1             ;SO CONTOUR WILL NOT ERASE YOUR MAP
contour,AVETEMPS        ;
set_xy,-105,-65,25,50    ;RESTORE COORDINATE SYSTEM TO MATCH THE MAP
                        ;(PSEUDO MERCATOR), not necessary for this
                        ;example...just being 'general'
```

I found the NCAR Graphics manual to be helpful on map projections other than pseudo mercator. There is a 10 page or so tutorial on the equations that cover all the conic projections, it's under EZMAP in one of the NCAR manuals. That is where I got the TOLAMBERT code.

Below is code that transforms the lats and lons to 'Lambert Conformal': Note that map these get transformed to a U,V system, which is (as you would expect) a cartesian sytem, with a min and max of +/-1. I forget which directions are positive. Experiment with TOLAMBERT and you'll find out.

Below that is the MAP program.

I think I've covered the basics. Play with the routines in the usual interactive way, and you'll be able to go far.

```
;=====
pro TOLAMBERT, lat, lon, u, v, stdlat1, stdlat2, stdlon
;INPUT: lat,lon
```

```

;OUTPUT: u,v, that ought to be in a plane, converted
;          to look like Lambert Conformal
IF n_params(0) lt 7 THEN BEGIN
  QHELP:
  print," Keyin:
  ;;;;;;;;;NOTE REVID (REVERSE VIDEO) IS IN MY PRIVATE LIBRARY
  print," " ,revid(1),"IDL> TOLAMBERT,lat,lon,u,v,stdlat1,stdlat2,stdlon",revid(0)
  print," Where:
  print,"   OUTPUTS are variables U,V (u-v coordinate system, ranges from -1 to +1
  print,"   INPUTS  are lat,lon
  print,"           stdlat1, stdlat2 are southern, northern standard latitudes,
  print,"           stdlon is standard longitude.
  print,"   Reference: NCAR Graphics GKS manual, under EZMAP
  print," Example:
  print,"   IDL> TOLAMBERT,35,98,u,v,30,45,90
  print,"   IDL> xyouts,u,v,"Chickasha, Oklahoma"
  GOTO,THE_END
ENDIF
IF is_help(lat) THEN GOTO,QHELP      ;IS_HELP() IS IN 'MY' LIBRARY
a=0.785398163
;pi=3.141592654
rlat1=1.*stdlat1/!radeg
rlat2=1.*stdlat2/!radeg
clon=1.*stdlon/!radeg
cone=(alog10(cos(rlat1))-alog10(cos(rlat2)))/$
      (alog10(tan(a-(rlat1/2.)))-alog10(tan(a-(rlat2/2.))))
latrad = lat/!radeg
lonrad = lon/!radeg
r = (tan(a-(latrad/2.)))^cone
u = r*sin(cone*(lonrad-clon))
v = r*cos(cone*(lonrad-clon)) ;make negative if need be
THE_END:
END

;=====
;MAP.PRO
; D. KELLER SEPT 1988 CIMMS @ NOAA/NSSL
;MOD JULY 23, 90, CHANGED NPTS TO NPTSEG

pro MAPHELP,fake
print," "
print," Keyin:
print,"IDL>DOMAP              (entire USA)
print,"IDL>DOMAP,minlat,maxlat,minlon,maxlon,linetype
print,"              (use '+' numbers for west longitude)
print," "
END

```

```

function ONSCREEN,i,mLAT,xLAT,mLON,xLON
;INPUT : i(th) line segment within mlat,xlat,mLon,xlon
;OUTPUT: 0=no, 1=yes
;NOTE : mlon, xlon should be negative west
common EXTENTS,mLATS,xLATS,mLONS,xLONS,bEGS,NPTSEG
;try to make the segment to be off of the screen
ONSCREEN = 1
IF mLATS(i) GT XLAT THEN GOTO,ZERO
IF XLATS(i) LT MLAT THEN GOTO,ZERO
IF mLONS(i) GT XLON THEN GOTO,ZERO
IF XLONS(i) LT MLON THEN GOTO,ZERO
GOTO,QEND
ZERO:
ONSCREEN = 0
QEND:
RETURN,onscreen
END

pro DOMAP,mLAT,xLAT,mLON,xLON,linetype,restore
;INPUT : mlat,xlat,mLon,xlon: Min/Max LATitude, LONGitude. optional LINETYPE
;OUTPUT: Pseudo-Mercator map
common NSEGMENTS,nmaplines
common MAPLALO,maplats,maplons
common EXTENTS,mLATS,xLATS,mLONS,xLONS,bEGS,NPTSEG

;VARIABLE STRATEGY:
; MLAT,XLAT,MLON,XLON ARE PRESERVED:
; MINLAT,MINLON,MAXLAT,MAXLON ARE MUCKED WITH within this routine.

np = n_params(0)
lsave = !linetype
!linetype = 0
CASE np OF
0 : BEGIN minlat=25 & maxlat=50 & minlon=-130 & maxlon=-65 & END
1 : BEGIN MAPHELP & GOTO,THE_END & END
2 : BEGIN MAPHELP & GOTO,THE_END & END
3 : BEGIN MAPHELP & GOTO,THE_END & END
4 : BEGIN
    q=xlon & maxlon=-mlon & minlon=-q & maxlat=xlat & minlat=mlat
    END
5 : BEGIN q=xlon & maxlon=-mlon & minlon=-q & minlat=mlat & maxlat=xlat
    !linetype=linetype
    END
6 : BEGIN q=xlon & maxlon=-mlon & minlon=-q & minlat=mlat & maxlat=xlat
    !linetype=linetype
    END
ELSE : BEGIN MAPHELP & GOTO,THE_END & END

```

```

ENDCASE
xmins=!xmin & xmaxs=!xmax & ymins=!ymin & ymaxs=!ymax
cxm = !cxmin & cxx = !cxmax & cym = !cymin & cyx = !cymax
xs = !xs & ys = !ys
;
;MEAT OF THE PROGRAM
set_xy,minlon,maxlon,minlat,maxlat
FOR i=0,nmaplines-2 DO BEGIN
    flag = ONSCREEN(i,minlat,maxlat,minlon,maxlon)
    IF flag eq 1 THEN BEGIN
        b = begs(i)
        e = begs(i+1) - 1
        ;LAMBERT EXAMPLE    tolambert,maplats(b:e),-maplons(b:e),x,y,stdlat1,stdlat2,std lon
        ;LAMBERT EXAMPLE    plots,x,y
        plots,maplons(b:e),maplats(b:e)
    ENDIF
ENDFOR
;LAST LINE SEGMENT
flag = ONSCREEN(i,minlat,maxlat,minlon,maxlon)
IF flag eq 1 THEN BEGIN
    b = begs(nmaplines-1)
    e = b + 10 - 1                ;LAST SEGMENT LENGTH
    plots,maplons(b:e),maplats(b:e)
ENDIF
!!linetype = lsave
IF n_params(0) eq 6 THEN BEGIN
    !xmin=xmins & !ymin=ymins & !xmax=xmaxs & !ymax=ymaxs
    !cxmin = cxm & !cxmax = cxx & !cymin = cym & !cymax = cyx
    !xs = xs & !ys = ys
ENDIF
THE_END:
END

;***** MAIN *****
common MAPLALO,maplats,maplons
common EXTENTS,mlats,xlats,mlons,xlons,begs,NPTSEG
common NSEGMENTS,nmaplines

dlk = 'disk$scratch:[publicscr.keller]map.stuff' ;THIS IS MY DATABASE
IF n_elements(maplats) eq 0 THEN RESTORE,dlk
MAPMENU

END

```
