
Subject: Spherical Contour Plotting

Posted by [Adam Bayliss](#) on Wed, 09 Jan 2002 16:22:53 GMT

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

Hi,

I'd like take a 2d array (containing a surface component of a vector) and make a contour plot on a sphere. So basically I'd like to take the results of the "contour" procedure, and project this graph onto a sphere. Any ideas would be appreciated.

Adam

Subject: Re: Spherical Contour Plotting

Posted by [Craig Markwardt](#) on Wed, 09 Jan 2002 23:57:12 GMT

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

Adam Bayliss <rabayliss@students.wisc.edu> writes:

> Hi,
> I'd like take a 2d array (containing a surface component of a vector)
> and make a contour plot on a sphere. So basically I'd like to take the
> results of the "contour" procedure, and project this graph onto a
> sphere. Any ideas would be appreciated.

Are you looking for the ORTHOGRAPHIC mapping projection? See MAP_SET, and try this to see if it is what you want,

```
map_set, /continents, name='orthographic'
```

You should be able to /OVERPLOT your contours directly, after you selection the viewpoint of course.

Good luck,
Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Spherical Contour Plotting

Posted by [k-bowman](#) on Thu, 10 Jan 2002 17:03:06 GMT

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

Sorry, my newsreader wrapped my last posting.

Ken

PRO SPHERICAL_PLOT

COMPILE_OPT IDL2

cr = "

nx = 65 ;Grid resolution in longitude

ny = 33 ;Grid resolution in latitude

x = (360.0/(nx-1))*FINDGEN(nx) ;Longitude grid

y = -90 + (180.0/(ny-1))*FINDGEN(ny) ;Latitude grid

xx = x # REPLICATE(1.0, ny) ;2-D longitude grid

yy = REPLICATE(1.0, nx) # y ;2-D latitude grid

z = SIN(!DTOR*xx) * COS(!DTOR*yy) ;Test function to contour

;Standard contour plot on satellite map projection

MAP_SET, /SATELLITE, /CONT, /ISOTROPIC

CONTOUR, z, x, y, LEVELS = -0.95 + 0.1*FINDGEN(20), /OVERPLOT ;Contour z

PRINT, 'Enter <cr> to continue.'

READ, cr

;Get contour info

CONTOUR, z, x, y, PATH_INFO = path_info, PATH_XY = path_xy, \$;Contour z, save
contour info

/PATH_DATA_COORDS, CLOSED = 0, LEVELS = -0.95 + 0.1*FINDGEN(20)

;2-D plot using contour info

PLOT, [0, 0], [1, 1], /NODATA, \$

XTITLE = 'Longitude', \$

XSTYLE = 1, \$

XRANGE = [0.0, 360.0], \$

XTICKS = 4, \$

YTITLE = 'Latitude', \$

YSTYLE = 1, \$

YRANGE = [-90., 90.0], \$

YTICKS = 6

FOR k = 0, N_ELEMENTS(path_info)-1 DO BEGIN

i0 = path_info[k].offset

;First element of the k'th contour

i1 = i0 + path_info[k].n - 1

;Last element of the k'th contour

; PRINT, k, path_info[k].type, i0, i1

xc = REFORM(path_xy[0,i0:i1])

;Extract x-coords of k'th contour

yc = REFORM(path_xy[1,i0:i1])

;Extract y-coords of k'th contour

IF (path_info[k].type EQ 1) THEN BEGIN

;Close contours, if needed

xc = [xc, path_xy[0,i0]]

```

    yc = [yc, path_xy[1,i0]]
ENDIF

PLOTS, xc, yc                                ;Plot contours
ENDFOR
PRINT, 'Enter <cr> to continue.'
READ, cr

;3-D plot using contour info
PLOT_3DBOX, [0,0], [0,0], [0,0], /NODATA, $
  XTITLE = 'X', $
  XSTYLE = 1, $
  X RANGE = [-1.0, 1.0], $
  XTICKS = 4, $
  YTITLE = 'Y', $
  YSTYLE = 1, $
  Y RANGE = [-1., 1.0], $
  YTICKS = 4, $
  ZTITLE = 'Z', $
  ZSTYLE = 1, $
  Z RANGE = [-1.0, 1.0], $
  ZTICKS = 4

r = 0.9
FOR k = 0, N_ELEMENTS(path_info)-1 DO BEGIN
  i0 = path_info[k].offset                    ;First element of the k'th contour
  i1 = i0 + path_info[k].n - 1                ;Last element of the k'th contour
; PRINT, k, path_info[k].type, i0, i1
  xc = REFORM(path_xy[0,i0:i1])                ;Extract x-coords of k'th contour
  yc = REFORM(path_xy[1,i0:i1])                ;Extract y-coords of k'th contour
  IF (path_info[k].type EQ 1) THEN BEGIN
    xc = [xc, path_xy[0,i0]]                    ;Close contours, if needed
    yc = [yc, path_xy[1,i0]]
  ENDIF

  x3 = r * COS(!DTOR*yc) * COS(!DTOR*xc)
  y3 = r * COS(!DTOR*yc) * SIN(!DTOR*xc)
  z3 = r * SIN(!DTOR*yc)
  PLOTS, x3, y3, z3, /T3D
ENDFOR

END

```
