

---

Subject: Re: Day/Night Terminator on Map Projection  
Posted by [kBob](#) on Sat, 08 Apr 2006 06:23:04 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Try the code below. It uses routines from the Johns Hopkins University/Applied Physics Laboratory.

Example:

```
IDL> TestSL, Longitude, Latitude
```

Kelly Dean  
Fort Collins, Colorado

```
-----  
FUNCTION InitColorID  
;  
; Load discrete color table.  
;  
tek_color  
;  
; Match color indices to colors we want to use  
;  
IF ( !D.N_COLORS GT 256 ) THEN BEGIN  
  TVLCT, RedTable, GreenTable, BlueTable, /GET  
  black = ( 256L * BlueTable(0) + GreenTable(0) ) * 256L +  
  RedTable(0)  
  white = ( 256L * BlueTable(1) + GreenTable(1) ) * 256L +  
  RedTable(1)  
  red   = ( 256L * BlueTable(2) + GreenTable(2) ) * 256L +  
  RedTable(2)  
  green = ( 256L * BlueTable(3) + GreenTable(3) ) * 256L +  
  RedTable(3)  
  dk_blue = ( 256L * BlueTable(4) + GreenTable(4) ) * 256L +  
  RedTable(4)  
  lt_blue = ( 256L * BlueTable(5) + GreenTable(5) ) * 256L +  
  RedTable(5)  
  six    = ( 256L * BlueTable(6) + GreenTable(6) ) * 256L +  
  RedTable(6)  
  yellow = ( 256L * BlueTable(7) + GreenTable(7) ) * 256L +  
  RedTable(7)  
ENDIF ELSE BEGIN  
  black=0 & white=1 & red=2 & green=3 & dk_blue=4 & lt_blue=5 & six=6 &  
  yellow=7  
ENDELSE  
  
ColorID = { ColorID,      $ ; Color Index
```

```
Black:black, $
White:white, $
Red:red, $
Green:green, $
DkBlue:dk_blue, $
LtBlue:lt_blue, $
Six:six, $
Yellow:Yellow }
```

```
RETURN, ColorID
END
```

```
;-----
```

```
PRO FOV2, longitude, latitude, SatRadius, ColorNum
```

```
;
; Procedure FOV2 will determine the maximum field of view
; and draw a circle around the subsatellite point.
```

```
;
; Input:
;   latitude : Earth latitude of subsatellite point
;   longitude : Earth longitude of subsatellite point
;   SatRadius : Satellite radius from earth center
;   ColorNum : Color number (default : 2)
```

```
IF ( N_ELEMENTS(ColorNum) EQ 0 ) THEN ColorNum = 2
```

```
EarthRadius = 6.371009D6 ; Mean Earth Radius
```

```
ScanAngle = ASIN( EarthRadius / SatRadius)
```

```
Arc_Dist = !DPI - 1.57080D - ScanAngle
```

```
LonLat0 = [longitude, latitude] ; Initial point specified in
radians
```

```
lon = FLTarr(37)
```

```
lat = FLTarr(37)
```

```
FOR lcv = 0, 36 DO BEGIN
```

```
  Az = lcv * 10.0
```

```
  Results = LL_ARC_DISTANCE(LonLat0, Arc_Dist, Az, /Degrees)
```

```
  lon(lcv) = results(0)
```

```
  lat(lcv) = results(1)
```

```
ENDFOR
```

```
oPLOT, lon, lat, COLOR=ColorNum
```

```
END
```

```
;+
```

```
; NAME:
```

```
PRO SunPlot, Init=Init
```

```
;
```

```
; PURPOSE:
```

```
; Procedure SunPlot will plot the location of the sun on a map.
```

```
;
```

```

; INPUTS:
;   None
;
; KEYWORD PARAMETERS:
;   Init - Create sun symbol to be plotted
;
; SIDE EFFECTS:
;   Requires public domain IDL routines from JHU
;   Assumes that MAP_SET and plotting window are already set.
;
;
; MODIFICATION HISTORY:
;   Created by Kelly Dean : December 1997
;-
;
;   Load discrete color table.
;
ColorID = InitColorID()
;
IF ( KEYWORD_SET(Init)) THEN BEGIN
;----- Sun Symbol -----
a = maken(0,360,17)
r = maken(1,1,17)+(1-findgen(17) mod 2)
polrec,r,a,x,y,/deg
usersym,x,y,color=ColorID.yellow,/fill
SunTLE = -1
ENDIF

;----- Handle time -----
dt0 = systime()      ; Current time is def.
dt = dt0            ; Copy date/time string.
off = gmt_offsec()  ; Find correction from local to UT.
dt_tm_inc, dt, off  ; Convert to UT.
dt_tm_brk, dt, dd, tt ; Break input into date and time.
date2ymd, dd, y, m, d ; Break date into y,m,d.
jd = ymd2jd(y,m,d)  ; Find Julian Day number.
ut = secstr(tt)/3600. ; Convert time to UT in hours.
;----- Solar RA/Dec -----
sun, y, m, d, ut, app_ra=ra, app_dec=dec, dist=dist
;----- GMST (Greenwich Mean Sidereal Time) -----
st = lmst(jd,ut/24.,0)*24
;----- Subsolar point -----
lat = dec
lng = 15.0*(ra-st)
;
; Plot sun's location and terminator
;
plots, lng, lat, psym=8

```

```
FOV2, lng, lat, (dist * 1.4956D11), ColorID.yellow
END
```

```
;-----
```

```
PRO LACMap, lon, lat
;
; Load discrete color table.
;
ColorID = InitColorID()
;
; Set up an orthographic projection centered over the north Atlantic.
; Fill the hemisphere with dark blue.
; Specify black gridlines
;
MAP_SET, /ORTHO, lat, lon, 0, $
    /ISOTROPIC, $
    /HORIZON, $
    E_HORIZON={FILL:1, COLOR:ColorID.dkblue}, $
    /NoBorder, $
    COLOR=lt_blue
; Fill the continent boundaries with solid white.
MAP_CONTINENTS, /FILL_CONTINENTS, COLOR=ColorID.white
; Overplot coastline data.
MAP_CONTINENTS, /COASTS, COLOR=ColorID.black
; Add rivers, in light blue.
MAP_GRID, LatDel = 15, LonDel = 30, Color = ColorID.black
END
```

```
;-----
```

```
PRO TestSL, Lon, Lat

; Hint: Color works best when decompose=1
; Hint: Default map location Colorado, USA

IF ( N_ELEMENTS( Lon ) EQ 0 ) THEN Lon = -105.0
IF ( N_ELEMENTS( Lat ) EQ 0 ) THEN Lat = 45.0

WINDOW, 0, XSize=512, YSize=512, TITLE='Sun Location'
LACMAP, Lon, Lat
SUNPLOT, /Init

END
```