
Subject: Re: Question about projection for Google Earth
Posted by [timothyja123](#) on Mon, 02 Sep 2013 05:15:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, I think I need to start over as I dont think I'm getting across the trouble I'm having. Rather than ask questions about what I *THINK* needs to be done I will describe the data I have and what I want to doo with it and maybe someone can give me some suggestions.

ok I have three arrays.

```
LAT      DOUBLE  = Array[2032]
LON      DOUBLE  = Array[4051]
HAMAX    INT     = Array[4051, 2032]
```

Lat, Lon are in degrees.

All I want to do is create an image with a contour (created with the hamax values) and overlay that image onto Google Earth.

Here is my (very unsuccessful) test program to attempt to get this working:

It creates two different versions of KML one using the Coyote function and one more manually.

PRO Create_kml

```
COMPILE_OPT IDL2, LOGICAL_PREDICATE
```

```
CATCH, errorNum
IF (errorNum NE 0) THEN BEGIN
  HELP, /Last_Message, Output = lastError
  v = DIALOG_MESSAGE(lastError)
  Error = LastError[0]
  RETURN
ENDIF
```

```
widget_control, hourglass = 1
```

```
current_win = !d.window
```

```
restore, filename='c:\tmp\google_earth.sav'
```

```
; use an absolute color scale to highlight the lower values
levels = [2,5,10,15,20,25,30,35,40,50,60,75,100,150,200,300]
c_levels = bytarr(N_Elements(levels))
for j = 0, N_Elements(levels)-1 do begin
  c_levels[j] = 240 + byte(j)
endfor
```

```
xsize=4000
```

```
ratio = float(N_Elements(lat))/float(N_Elements(lon))
ysize = fix(ratio*xsize)
```

```
SET_PLOT, 'Z'
Device, z_buffering = 0, set_pixel_depth=8
Device, set_resolution = [xsize,ysize]
```

```
; define colors for HAMAX
loadct, 12, /silent
tvlct,r2,g2,b2,/get
```

```
;use Brewer Table, line 999 in XLS file modified for color 6 and 7
r = [39,77,127,184,230,224,186,241,222,197,142]
g = [100,146,188,225,245,224,186,182,119,27,1]
b = [25,33,65,134,208,224,186,218,174,125,82]
```

```
For i = 0,10 do begin
```

```
    r2[245+i] = r[i]
    g2[245+i] = g[i]
    b2[245+i] = b[i]
```

```
Endfor
```

```
tvlct, r2,g2,b2
tvlct,0,0,0,0
!p.background = 0
erase, 0
```

```
limit = dblarr(4)
limit[0] = min(lat)
limit[1] = min(lon)
limit[2] = max(lat)
limit[3] = max(lon)
```

```
polon = limit[1]+0.5*(limit[3]-limit[1])
map_set, 0,polon,0,/MILLER_CYLINDRICAL,limit=limit, /noborder, xmargin=0,ymargin=0
```

```
help, lat
help, lon
help, hamax
```

```
contour, hamax*bathy_mask,lon,lat, levels=levels,C_color=c_levels,/overplot,c_labels =
0,/cell_fill, min_value = 2
```

```
most_image = tvrd()
Device, close = 1
```

```

; output image to a PNG file with transparency
name = 'MOST_max_amplitude1.png'
outputOverlayFile = filepath(name, root_dir=sourcepath(), subdir=['output'])

; set transparency
idx = where(most_image LT 240, trans_count)
if trans_count GT 0 then begin
    write_png,outputOverlayFile ,most_image, r2,g2,b2, transparent=idx
endif

most_image = READ_PNG(outputOverlayFile, r2,g2,b2, transparent=idx)

; output image to a PNG file with transparency
name = 'MOST_max_amplitude.kml'
outputOverlayFile2 = filepath(name, root_dir=sourcepath(), subdir=['output'])

googleMapCoord = Obj_New('cgMap', 'Miller Cylindrical',
CENTER_LONGITUDE=polon);,limit=limit);, , center_latitude=0)
;LATLONBOX=[limit[0], limit[2], limit[1], limit[3]]
cgImage2KML, most_image, googleMapCoord, filename=outputOverlayFile2

.*****
;
; write kml file
.*****
name = 'MOST_max_amplitude1.kml'
outputKMLFile = filepath(name, root_dir=sourcepath(), subdir=['output'])

openw, kml_lun, outputKMLFile, /get_lun

slat = -38.1661
slon = 177.945

; description
descrip = 'MOST tsunami model: maximum wave amplitude for scenario '
descrip = descrip +'centred at lat '+strtrim(mean(slat),2) +' lon '+ strtrim(mean(slon),2)

; header info
printf, kml_lun, '<?xml version="1.0" encoding="UTF-8"?>'
printf, kml_lun, '<kml xmlns="http://earth.google.com/kml/2.0">'
printf, kml_lun, ' <GroundOverlay>'
printf, kml_lun, ' <description>'+descrip+'</description>'
printf, kml_lun, ' <name>' + file_basename(outputKMLFile, '.kml') + '</name>'

; </LookAt> refer to kml guide http://code.google.com/apis/kml/documentation/kmlreference.h
tml#lookat
printf, kml_lun, ' <LookAt>'
printf, kml_lun, ' <longitude>'+strtrim(mean(slon),2)+'</longitude>'
printf, kml_lun, ' <latitude>'+strtrim(mean(slat),2)+'</latitude>'

```

```
printf, kml_lun, '    <altitudeMode>absolute</altitudeMode>'
printf, kml_lun, '    <altitude>2000000</altitude>'
printf, kml_lun, '    <tilt>0.0</tilt>'
printf, kml_lun, '    <heading>0.0</heading>'
printf, kml_lun, '    </LookAt>'
```

; Add the icon tag (the actual image), for example:

```
printf, kml_lun, '    <Icon>'
printf, kml_lun, '        <href>' + outputOverlayFile + '</href>'
printf, kml_lun, '    </Icon>'
```

; Not sure what this does. Make it red for now (AABBGRR format)

```
printf, kml_lun, '    <color>ffffff</color>'
```

; Add the corners to the kml file

```
printf, kml_lun, '    <LatLonBox id="khLatLonBox565">'
printf, kml_lun, '        <north>' + strtrim(limit[2], 2) + '</north>'
printf, kml_lun, '        <south>' + strtrim(limit[0], 2) + '</south>'
printf, kml_lun, '        <east>' + strtrim(limit[3], 2) + '</east>'
printf, kml_lun, '        <west>' + strtrim(limit[1], 2) + '</west>'
printf, kml_lun, '        <rotation>0</rotation>'
printf, kml_lun, '    </LatLonBox>'
```

```
printf, kml_lun, ' </GroundOverlay>'
```

```
printf, kml_lun, '</kml>'
```

```
close, kml_lun
free_lun, kml_lun
```

```
widget_control, hourglass = 0
```

```
SET_PLOT, 'WIN'
```

```
wset, current_win
```

```
END
```