Subject: Question about projection for Google Earth
Posted by timothyja123 on Fri, 09 Aug 2013 02:11:55 GMT
View Forum Message <> Reply to Message

Hi Guys,

I'm trying to finish off some unfinished code that wasnt written by me. I'm fairly new to using projections, mapping, etc in IDL but from what I understand to map an image to Google Earth I need to use something like:

```
mapStruct = MAP_PROJ_INIT( 117 , /GCTP, limit=limit) ; Equirectangular projection
xy = Map_Proj_Forward(lon, lat, MAP_STRUCTURE=mapStruct)
lon = Reform(xy[0, *])
lat = Reform(xy[1, *])
```

The problem is that my lon/lat values contain a different number of values and Map_Proj_Forward() is complaining about this. Whay do they need to be the same? My overlay is a rectangular shape rather than a square why can this no be handled?

The code that I was left with does something like this but the image doesnt overlay correctly on Google Earth I assume this is because it uses the incorrect projection.

```
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,/cyl,limit=limit, /noborder, xmargin=0,ymargin=0

contour, values*mask,lon,lat, levels=levels,C_color=c_levels,/overplot,c_labels = 0,/cell_fill,
min_value = 2

hamax = 0 & bathy_mask = 0 & lat = 0 & lon = 0

most_image = tvrd()
Device, close = 1

; output image to a PNG file with transparency
name = 'test.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

Anyhelp or tips on this would be greatly appreciated. I have looked at some of David Fanning's guides on creating images for Google Earth but could find anything on this exact issue.

Subject: Re: Question about projection for Google Earth
Posted by David Fanning on Fri, 09 Aug 2013 02:26:08 GMT
View Forum Message <> Reply to Message

timothyja123@gmail.com writes:

>
> Hi Guys,
>
> I'm trying to finish off some unfinished code that wasnt written by me. I'm fairly new to using projections, mapping, etc in IDL but from what I understand to map an image to Google Earth I need to use something like:
>
> mapStruct = MAP_PROJ_INIT(117 , /GCTP, limit=limit) ; Equirectangular projection
> xy = Map_Proj_Forward(lon, lat, MAP_STRUCTURE=mapStruct)
> lon = Reform(xy[0, *])
> lat = Reform(xy[1, *])
>
> The problem is that my lon/lat values contain a different number of values and Map_Proj_Forward() is complaining about this. Whay do they need to be the same? My overlay is a rectangular shape rather than a square why can this no be handled?
>
>
> The code that I was left with does something like this but the image doesnt overlay correctly on Google Earth I assume this is because it uses the incorrect projection.
>
> 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,/cyl,limit=limit, /noborder, xmargin=0,ymargin=0
>
> contour, values*mask,lon,lat, levels=levels,C_color=c_levels,/overplot,c_labels = 0,/cell_fill, min_value = 2
>
> hamax = 0 & bathy_mask = 0 & lat = 0 & lon = 0
>
> most_image = tvrd()

> Device, close = 1
>
> ; output image to a PNG file with transparency
> name = 'test.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
>
>
> Anyhelp or tips on this would be greatly appreciated. I have looked at some of David Fanning's guides on creating images for Google Earth but could find anything on this exact issue.

All of these details are handled automatically for you in cgImage2KML.

http://www.idlcoyote.com/idldoc/cg/cgimage2kml.html

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: Question about projection for Google Earth
Posted by timothyja123 on Fri, 09 Aug 2013 04:50:05 GMT
View Forum Message <> Reply to Message

> All of these details are handled automatically for you in cgImage2KML.
>
> http://www.idlcoyote.com/idldoc/cg/cgimage2kml.html
>
> Cheers,
> David

Hi David,

Thanks for the reply. I have already tried to use your procedure however I ran into a problem I wasnt sure how to solve. Your procdure takes an image as an argument which is fine as I can use the image produced by my existing code.
However I then need to supply a map coordinate object (cgMap) from which map projection

information and map boundaries for the image overlay can be obtained.

The problem is that your cgMap function says: "Only GCTP projections are allowed. If you wish to use projections normally set up with Map_Set, use the comparable cgMap_Set command. "
But that command does not produce the cgMap object that I need for cgImage2KML.

I'm most likely missing something obvious (as I said I'm new to projections and mapping).

I tired using a map_set projection that had a matching GCTP projection MILLER_CYLINDRICAL and using the produced image with cgImage2KML but this didnt seem to work.

```
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


contour, values*mask,lon,lat, levels=levels,C_color=c_levels,/overplot,c_labels = 0,/cell_fill,
min_value = 2


most_image = tvrd()
Device, close = 1


name = 'MOST_max_amplitude.kml'
outputOverlayFile = filepath(name, root_dir=sourcepath(), subdir=['output'])

googleMapCoord = Obj_New('cgMap', 'Miller Cylindrical', limit=limit)
cgImage2KML, most_image, googleMapCoord, filename=outputOverlayFile,
LATLONBOX=LATLONBOX
```

Subject: Re: Question about projection for Google Earth
Posted by David Fanning on Fri, 09 Aug 2013 10:51:27 GMT
View Forum Message <> Reply to Message

timothyja123@gmail.com writes:

> Thanks for the reply. I have already tried to use your procedure however I ran into a problem I wasnt sure how to solve. Your procdure takes an image as an argument which is fine as I can use the image produced by my existing code.
> However I then need to supply a map coordinate object (cgMap) from which map projection information and map boundaries for the image overlay can be obtained.

>
> The problem is that your cgMap function says: "Only GCTP projections are allowed. If you wish to use projections normally set up with Map_Set, use the comparable cgMap_Set command. "
> But that command does not produce the cgMap object that I need for cgImage2KML.
>
> I'm most likely missing something obvious (as I said I'm new to projections and mapping).
>
> I tired using a map_set projection that had a matching GCTP projection MILLER_CYLINDRICAL and using the produced image with cgImage2KML but this didnt seem to work.

The map coordinate object cgImage2KML needs is the one that describes
the image you are trying to display. This image will then be warped into
the map projection required for display on Google Earth. From your
previous post, I assume you would make the map coordinate object like
this:

```
;;;mapStruct = MAP_PROJ_INIT( 117 , /GCTP, limit=limit)
mapCoord = cgMap(117, /GCTP, LIMIT=limit)
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: Question about projection for Google Earth
Posted by timothyja123 on Mon, 19 Aug 2013 05:04:30 GMT
View Forum Message <> Reply to Message

> The map coordinate object cgImage2KML needs is the one that describes
> the image you are trying to display. This image will then be warped into
> the map projection required for display on Google Earth. From your
> previous post, I assume you would make the map coordinate object like
> this:
>
> ;;;mapStruct = MAP_PROJ_INIT(117 , /GCTP, limit=limit)
> mapCoord = cgMap(117, /GCTP, LIMIT=limit)

Hi David,

Again thanks for your reply but I think you are missing the part I'm having trouble with. I understand how your functions work. The problem is I can not create the image in the 117 GCTP format to begin with.

From my first post:
"The problem is that my lon/lat values contain a different number of values and Map_Proj_Forward() is complaining about this. Whay do they need to be the same? My overlay is a rectangular shape rather than a square why can this not be handled? "

The code in my first post is what I WANT to be able to do but can't. Maybe I should have tried to be more clear.

Subject: Re: Question about projection for Google Earth
Posted by Fabzi on Mon, 19 Aug 2013 07:27:35 GMT
View Forum Message <> Reply to Message

On 08/19/2013 07:04 AM, timothyja123@gmail.com wrote:
> "The problem is that my lon/lat values contain a different number of values and
>Map_Proj_Forward() is complaining about this. Whay do they need to be the same?
>My overlay is a rectangular shape rather than a square why can this not be handled?"

I don't really understand what you mean with "rather than a square" but Map_Proj_Forward() converts point coordinates into other points coordinates. As far as I know, points always have (at least) two coordinates so if you want to transform N points you should have N lats, and N lons. The real question is: why do you have a different number of lat and lon values?

The most typical case would be that your data is in equirectangular projection. For example:
IDL> nx = 360
IDL> ny = 180
IDL> lon = FINDGEN(nx) - 179.5
IDL> lat = FINDGEN(ny) - 89.5

if you want to pass them to map_proj you need "rectangles" as you say:
IDL> lon = lon # (LONARR(ny) + 1)
IDL> lat = lat ## (LONARR(nx) + 1)
IDL> help, lon, lat
LON FLOAT = Array[360, 180]
LAT FLOAT = Array[360, 180]

not sure if that helps but I am quite sure that Map_Proj_Forward() is not going to accept different vectors of lat/lon in the close future.

Cheers

Subject: Re: Question about projection for Google Earth
Posted by David Fanning on Mon, 19 Aug 2013 12:26:11 GMT
View Forum Message <> Reply to Message

timothyja123@gmail.com writes:

> "The problem is that my lon/lat values contain a different number of values and Map_Proj_Forward() is complaining about this. Whay do they need to be the same? My overlay is a rectangular shape rather than a square why can this not be handled? "

In my experience, if you want things to work the way you expect them to work, you would be better off using Coyote Library routines than IDL routines. :-)

But, yes, make rectangles out of your Lat/Lon vectors as Fabien suggests, and you will be fine.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

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 modifed 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 (AABBGGRR format)
  printf, kml_lun, '    <color>ffffffff</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
```

Subject: Re: Question about projection for Google Earth
Posted by David Fanning on Tue, 03 Sep 2013 13:03:23 GMT
View Forum Message <> Reply to Message

timothyja123@gmail.com writes:

> 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.

I think I would just make rectangular arrays out of your latitude and
longitude vectors to match your data, as Fabien suggested.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")
