

Folks,

I have discovered another map projection problem this morning.
It seems Map_Proj_Init is calculating incorrect UV ranges
(they are wrong by 10 degrees) when the center latitude is
not 0.0. Here is a short test program:

```
FOR j=0,10 DO BEGIN
  center_lon = 0 + j*36.0d < 360.0
  mapStruct = Map_Proj_Init(117, CENTER_LONGITUDE=center_lon)
  print, center_lon, mapStruct.uv_box[[0,2]]

  ll = Map_Proj_Inverse(mapStruct.uv_box[[0,2]], $
    mapStruct.uv_box[[1,3]], MAP_STRUCTURE=mapStruct)
  print, center_lon, ll[0,0], ll[0,1]
  print, ""
ENDFOR
END
```

Here are the results. The numbers should be symmetric,
but you can see they are not.

0.000000	-20015077.	20015077.
0.000000	180.00000	-180.00000
36.0000	-19570298.	19347908.
36.0000	-140.00000	-150.00000
72.0000	-19125518.	19792688.
72.0000	-100.00000	-110.00000
108.000	-19792688.	19125518.
108.000	-70.000000	-80.000000
144.000	-19347908.	19570298.
144.000	-30.000000	-40.000000
180.000	-20015077.	18903129.
180.000	0.00000000	-10.000000
216.000	-19570298.	19347908.
216.000	40.000000	30.000000
252.000	-19125518.	19792688.

252.000	80.000000	70.000000
288.000	-19792688.	19125518.
288.000	110.00000	100.00000
324.000	-19347908.	19570298.
324.000	150.00000	140.00000
360.000	-20015077.	20015066.
360.000	180.00000	179.99990

I couldn't figure out why my grid lines were always getting screwed up when I calculated them using the UV range! :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Another Map Projection Problem with Map_Proj_Init

Posted by [David Fanning](#) on Tue, 06 Dec 2011 00:43:12 GMT

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

David Fanning writes:

```
> I have discovered another map projection problem this morning.
> It seems Map_Proj_Init is calculating incorrect UV ranges
> (they are wrong by 10 degrees) when the center latitude is
> not 0.0. Here is a short test program:
>
> FOR j=0,10 DO BEGIN
>   center_lon = 0 > j*36.0d < 360.0
>   mapStruct = Map_Proj_Init(117, CENTER_LONGITUDE=center_lon)
>   print, center_lon, mapStruct.uv_box[[0,2]]
>
>   ll = Map_Proj_Inverse(mapStruct.uv_box[[0,2]], $
>     mapStruct.uv_box[[1,3]], MAP_STRUCTURE=mapStruct)
>   print, center_lon, ll[0,0], ll[0,1]
>   print, ""
> ENDFOR
> END
>
```

```

> Here are the results. The numbers should be symmetric,
> but you can see they are not.
>
>    0.000000    -20015077.    20015077.
>    0.000000     180.00000    -180.00000
>
>    36.0000    -19570298.    19347908.
>    36.0000    -140.00000    -150.00000
>
>    72.0000    -19125518.    19792688.
>    72.0000    -100.00000    -110.00000
>
>    108.000    -19792688.    19125518.
>    108.000    -70.000000    -80.000000
>
>    144.000    -19347908.    19570298.
>    144.000    -30.000000    -40.000000
>
>    180.000    -20015077.    18903129.
>    180.000     0.00000000    -10.000000
>
>    216.000    -19570298.    19347908.
>    216.000     40.000000     30.000000
>
>    252.000    -19125518.    19792688.
>    252.000     80.000000     70.000000
>
>    288.000    -19792688.    19125518.
>    288.000    110.00000     100.00000
>
>    324.000    -19347908.    19570298.
>    324.000    150.00000     140.00000
>
>    360.000    -20015077.    20015066.
>    360.000     180.00000     179.99990
>
> I couldn't figure out why my grid lines were always getting
> screwed up when I calculated them using the UV range! :-(

```

It turns out, according to the folks in technical support, that to get correct UV_BOX values from Map_Proj_Init you MUST use the LIMIT keyword, even (well, *especially*) if you don't plan to limit the map projection in any way.

Another way to say this is that if you plan to use Map_Proj_Init and set the CENTER_LONGITUDE to anything other than 0, you MUST also use the LIMIT keyword to produce correct results. This is, as you might expect,

not documented. :-)

I've updated my cgMap object to handle the problem correctly.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Another Map Projection Problem with Map_Proj_Init
Posted by [Fabzou](#) on Tue, 06 Dec 2011 10:32:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks David.

Just to be sure: if I never used UV_box values but a LOT of the projections I am using have a non-zero center longitude, this bug won't affect me in any sense, will it?

Fab

On 12/06/2011 01:43 AM, David Fanning wrote:

> David Fanning writes:

>

>> I have discovered another map projection problem this morning.

>> It seems Map_Proj_Init is calculating incorrect UV ranges

>> (they are wrong by 10 degrees) when the center latitude is

>> not 0.0. Here is a short test program:

>>

>> FOR j=0,10 DO BEGIN

>> center_lon = 0> j*36.0d< 360.0

>> mapStruct = Map_Proj_Init(117, CENTER_LONGITUDE=center_lon)

>> print, center_lon, mapStruct.uv_box[[0,2]]

>>

>> ll = Map_Proj_Inverse(mapStruct.uv_box[[0,2]], \$

>> mapStruct.uv_box[[1,3]], MAP_STRUCTURE=mapStruct)

>> print, center_lon, ll[0,0], ll[0,1]

>> print, ""

>> ENDFOR

>> END

```

>>
>> Here are the results. The numbers should be symmetric,
>> but you can see they are not.
>>
>>    0.000000    -20015077.    20015077.
>>    0.000000     180.00000    -180.00000
>>
>>    36.0000    -19570298.    19347908.
>>    36.0000    -140.00000    -150.00000
>>
>>    72.0000    -19125518.    19792688.
>>    72.0000    -100.00000    -110.00000
>>
>>    108.000    -19792688.    19125518.
>>    108.000    -70.000000    -80.000000
>>
>>    144.000    -19347908.    19570298.
>>    144.000    -30.000000    -40.000000
>>
>>    180.000    -20015077.    18903129.
>>    180.000     0.00000000    -10.000000
>>
>>    216.000    -19570298.    19347908.
>>    216.000     40.000000     30.000000
>>
>>    252.000    -19125518.    19792688.
>>    252.000     80.000000     70.000000
>>
>>    288.000    -19792688.    19125518.
>>    288.000    110.00000     100.00000
>>
>>    324.000    -19347908.    19570298.
>>    324.000    150.00000     140.00000
>>
>>    360.000    -20015077.    20015066.
>>    360.000     180.00000     179.99990
>>
>> I couldn't figure out why my grid lines were always getting
>> screwed up when I calculated them using the UV range! :-(
>
> It turns out, according to the folks in technical support,
> that to get correct UV_BOX values from Map_Proj_Init you
> MUST use the LIMIT keyword, even (well, *especially*) if
> you don't plan to limit the map projection in any way.
>
> Another way to say this is that if you plan to use
> Map_Proj_Init and set the CENTER_LONGITUDE to anything
> other than 0, you MUST also use the LIMIT keyword to

```

> produce correct results. This is, as you might expect,
> not documented. :-)
>
> I've updated my cgMap object to handle the problem
> correctly.
>
> Cheers,
>
> David
>
>

Subject: Re: Another Map Projection Problem with Map_Proj_Init
Posted by [David Fanning](#) on Tue, 06 Dec 2011 13:35:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Fabzou writes:

> Just to be sure: if I never used UV_box values but a LOT of the
> projections I am using have a non-zero center longitude, this bug won't
> affect me in any sense, will it?

Well, I don't know. Probably not. I was using the information to label my maps, and it was certainly affecting me there. These kinds of errors make me a little nervous, because I have the impression map projections are not really understood at Excelis. This impression is nearly always reinforced by the explanations I get for the error. Although, to be fair, the explanation is almost always passing through a third party, who I don't really expect to understand map projections, and that may be adding noise to the explanation.

In any case, I tend to use my own version of these map routines, which fix the numerous problems I have encountered while I've been looking into it. (Not to say my own programs are completely bug free. But I do work hard to make sure they are right.)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.

