Subject: wrong results...
Posted by elias on Mon, 02 May 2005 13:49:00 GMT
View Forum Message <> Reply to Message

Hi,

The following problem probably has something to do with definitions of
floating, integer etc, but I don't really know to solve it (although i
am sure the solution will be simple in the end...).

I have a data file that, amongst others, contains some longitude
columns. The longitude data is defined from -180 to 180 deg in this
data file. For some consistency with some other analysis I did before,
I want to convert it to 0-->360 deg.

So simply, I ask when it reads a longitude <0 to do: lon=360+lon

However, I am getting in the new data file created, values of longitude
more than 600 sometimes.... I am not sure why this happens

Below I post this simple code.

~~~~~~ ~~~~~~~~~~~~~~

```
PRO restruct, fil, output

;reads data file of moon l-shell
;crossings and restructures its data
;so that:
;Cassini and Moon longitude>0
;(Cassini-Moon) longitude>0


filetemplate=ASCII_Template(fil)
data=Read_ASCII(fil,Template=filetemplate)

crossings=n_elements(data.field1)

openw, lunit, /get_lun, output

for i=0L,crossings-1 do begin

   if data.field3[i] lt 0 then begin
     data.field3[i]=360+data.field3[i]

   endif
```
~~~~~~

```
    if data.field6[i] lt 0 then begin
      data.field6[i]=360+data.field6[i]

    endif

    dlon=data.field6[i]-data.field3[i]

    if dlon lt 0 then begin
      data.field3[i]=360+data.field3[i]

    endif

    thisformat='(f14, 7f14)'
    printf, lunit, data.field1[i], data.field2[i],$
          data.field3[i], data.field4[i],$
          data.field5[i], data.field6[i],$
          data.field7[i], dlon, format=thisformat

endfor

free_lun, lunit

end
```

Subject: Re: wrong results...
Posted by Ken Mankoff on Tue, 03 May 2005 22:56:45 GMT
View Forum Message <> Reply to Message

On Tue, 3 May 2005, yp wrote:

> Kenneth Bowman wrote:
>>
>> lon = (lon + 360.0) MOD 360.0
>>
>
> Thats brilliant!
>
>> By the way, the reverse transformation is
>>
>> lon = lon - (LONG(lon)/180)*360.0
>>
>
> This is great too; but how to parse the last element? Using the
> above transformation, 180.0 becomes -180.0 at both ends. This does
> transform the coordinate exactlty for all elements except +180.0.
> Of course, zonally 180 = -180 from a geographic perspective.

The IDL astronomy library has a CIRRANGE procedure (circle range) that forces things to 0->360. It is pretty easy to modify it to a RANGECIR procedure that does the -180 to 180 range. Works for radians too via a keyword. Check it out, it might be a good starting place.

-k.

--
http://spacebit.dyndns.org/