Subject: Re: Faster way to "shift" array?
Posted by Yngvar Larsen on Tue, 11 Jun 2013 21:39:07 GMT
View Forum Message <> Reply to Message

On Tuesday, 11 June 2013 14:50:43 UTC+2, rj...@le.ac.uk wrote:
> I have a longitude array which ranges from 0 to 360 but I want it to range from -180 to 180.
>
> Currently I'm doing this:
>
> lon=shift(lon, n_elements(lon)/2.)
> lon[where(lon GT 180)]=lon[where(lon GT 180)]-360.
> The lon array is 3600 elements and the shift command is taking around 1 second.

Huh? That cannot be correct. On my machine:

```
IDL> lon = 360*lindgen(3600)/3599
IDL> t=systime(/sec)& lon=shift(lon, n_elements(lon)/2.) & lon[where(lon GT 180)]=lon[where(lon GT 180)]-360. & print, systime(/sec)-t
   0.00010800362
```

The full operation took 0.1 millisecond.

```
IDL> lon = 360*lindgen(3600)/3599
IDL> t=systime(/sec)&for n=0,9999 do lon=shift(lon, n_elements(lon)/2.)&print, systime(/sec)-t
     0.044115782
```

So the shift operation alone took 0.04 seconds for 10000 iterations, i.e. 4 microseconds per iteration.

> When multiplied by the thousands of files I need to handle this becomes quite a considerable time component. Is there a faster way to do this?

Most likely.

```
N = n_elements(lon)/2
lon = shift(lon, N)
lon[0:N-1] -= 360.
```

But I'm 100% sure that this is not your bottleneck. I assume you also have to shift your 2D data array, not just the 1D longitude array? In that case, I would look there first.

--
Yngvar