Subject: Faster way to "shift" array?
Posted by rjp23 on Tue, 11 Jun 2013 12:50:43 GMT
View Forum Message <> Reply to Message

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. 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?

Cheers

Rob

Subject: Re: Faster way to "shift" array?
Posted by rjp23 on Tue, 11 Jun 2013 12:52:45 GMT
View Forum Message <> Reply to Message

I forgot to add, I then have a corresponding data array which needs to be shifted in the same way.

Subject: Re: Faster way to "shift" array?
Posted by David Fanning on Tue, 11 Jun 2013 13:04:57 GMT
View Forum Message <> Reply to Message

rjp23@le.ac.uk writes:

> 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. 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?

I think it is the Where function that is slowing you down. I've always

used the formulas on this page to do the conversion. I've never noticed
them being slow. :-)

  http://www.idlcoyote.com/map_tips/lonconvert.html

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: Faster way to "shift" array?
Posted by rjp23 on Tue, 11 Jun 2013 13:23:05 GMT
View Forum Message <> Reply to Message

On Tuesday, June 11, 2013 2:04:57 PM UTC+1, David Fanning wrote:
> I think it is the Where function that is slowing you down. I've always
>
> used the formulas on this page to do the conversion. I've never noticed
>
> them being slow. :-)
>
>
>
>   http://www.idlcoyote.com/map_tips/lonconvert.html
>
>
>
> Cheers,
>
>
>
> David
>

Hi David,

I put some timestamps in and it seems that the WHERE is near instantaneous (thousandths of a second) but it is the SHIFT that's slow.

The problem with doing it by a conversion method is that I have a data array that corresponds to the longitude array and things like contour don't like the discontinuities that introduces (e.g. it'd go 179, 180, -180, -179, etc) so I think I need to actually manipulate the positions in the array, rather than the values... (happy to be corrected!)

Cheers

Rob

Subject: Re: Faster way to "shift" array?
Posted by David Fanning on Tue, 11 Jun 2013 13:28:50 GMT
View Forum Message <> Reply to Message

rjp23@le.ac.uk writes:

> The problem with doing it by a conversion method is that I have a data array that corresponds to the longitude array and things like contour don't like the discontinuities that introduces (e.g. it'd go 179, 180, -180, -179, etc) so I think I need to actually manipulate the positions in the array, rather than the values... (happy to be corrected!)

Well, I don't know. I've worked with map projections for a long time and
have never felt a need to shift any data. I'm probably doing it wrong.
;-)

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: Faster way to "shift" array?
Posted by David Fanning on Tue, 11 Jun 2013 13:35:44 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>
> rjp23@le.ac.uk writes:
>

>> The problem with doing it by a conversion method is that I have a data array that corresponds to the longitude array and things like contour don't like the discontinuities that introduces (e.g. it'd go 179, 180, -180, -179, etc) so I think I need to actually manipulate the positions in the array, rather than the values... (happy to be corrected!)
>
> Well, I don't know. I've worked with map projections for a long time and
> have never felt a need to shift any data. I'm probably doing it wrong.
> ;-)

I do occasionally, not often, have to add a column to a longitude and data array to eliminate a gap in a filled contour plot. But, shifting the data is not going to make any difference when you put the data on a map projection. If it did, you would never be able to move the center of a map projection without shifting the data. And, clearly, you can shift the center of a map projection without changing anything at all about the data.

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: Faster way to "shift" array?
Posted by rjp23 on Tue, 11 Jun 2013 13:44:01 GMT
View Forum Message <> Reply to Message

On Tuesday, June 11, 2013 2:28:50 PM UTC+1, David Fanning wrote:
> I'm probably doing it wrong.
>
> ;-)
>
> Cheers,
>
> David
>

I suspect not :p

I think my issue might be how I then subset a geographic region if it crosses the date line.

Thanks for pointing me in the right direction :-)

Subject: Re: Faster way to "shift" array?
Posted by Fabzi on Tue, 11 Jun 2013 16:33:07 GMT
View Forum Message <> Reply to Message

On 06/11/2013 03:44 PM, rjp23@le.ac.uk wrote:> On Tuesday, June 11, 2013 2:28:50 PM UTC+1, David Fanning wrote:
>> I'm probably doing it wrong.
> I suspect not :p
>
> I think my issue might be how I then subset a geographic
> region if it crosses the date line.
>
> Thanks for pointing me in the right direction :-)

I also agree that sometimes it is impossible to overcome a subset problem without having to shift the data. Let's take the example of netcdf files, which have the nice capability to be subset _without_ reading the full databox in the memory. If your subset goes something like: [-40, 40] in longitude and your data is in [0, 360] then you can't efficiently use the netcdf COUNT and OFFSET keywords.
Data organized in [0, 360] has the bad property of cutting Europe and Africa in two, while [-180, 180] mostly cuts oceans. It depends on how often you use the data...

I tried this on my laptop:

```
pro test
  t_lon = findgen(7200) / 20.
  data =  (LONARR(3601)+1) ## t_lon
  ; Convert longs
  lon = ((t_lon + 180) MOD 360) - 180
  nl = n_elements(lon)/2.
  print, 'Shift'
  tic
  new_data1 = shift(data, nl)
  toc
  print, 'Concatenate'
  tic
  new_data2 = [data[nl:*, *], data[0:nl-1,*]]
  toc
end
```

```
Shift
% Time elapsed: 0.098950863 seconds.
Concatenate
% Time elapsed: 0.20306301 seconds.
```

It seems that shift is twice as fast as the "array concatenation" solution.

Subject: Re: Faster way to "shift" array?
Posted by David Fanning on Tue, 11 Jun 2013 17:23:34 GMT
View Forum Message <> Reply to Message

Fabien writes:

> I also agree that sometimes it is impossible to overcome a subset
> problem without having to shift the data. Let's take the example of
> netcdf files, which have the nice capability to be subset _without_
> reading the full databox in the memory. If your subset goes something
> like: [-40, 40] in longitude and your data is in [0, 360] then you can't
> efficiently use the netcdf COUNT and OFFSET keywords.
> Data organized in [0, 360] has the bad property of cutting Europe and
> Africa in two, while [-180, 180] mostly cuts oceans. It depends on how
> often you use the data...

I'm certainly not arguing that data doesn't occasionally need to be reorganized, and I've certainly used the Shift function to do so. Rather, I'm arguing that I have rarely, if ever, used the shift function for the sole purpose of drawing a contour plot on a map projection. In fact, in this situation it rarely matters if your longitude vector goes from 0 to 360 or -180 to 180. The data should be drawn (God help us!) in the same location regardless. :-)

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: 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