Subject: gridding large amounts of data

Posted by dan on Wed, 05 Oct 1994 20:40:52 GMT

View Forum Message <> Reply to Message

I have a large table of data in the form

```
LATITUDE   LONGITUDE    VALUE
```

There can be a million rows of entries in this 3 colum table.
I would like to put the data into a 1x1 degree array (fltarr(360,180))
If there are multiple entries for a particular longitude and latitude
I would like to sum the values. I could do this in an ugly do loop that
looked like

```
data=fltarr(360,180)
for i=1,num do $
 data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
```

This works fine for small values of num, but for num>32,767 I get an error
saying my do loop index is too large. Can I write do loops with limits
greater than 32,767 ??   Could I rewrite this in vector syntax using the
where statement ??

```
--
 ************************************************************ ***
**  Dan Bergmann                      dbergmann@llnl.gov  **
**  Global Climate Research             fax   (510) 422-5844  **
**  Lawrence Livermore National Lab    human (510) 423-6765  **
```

Subject: Re: gridding large amounts of data

Posted by candey on Thu, 06 Oct 1994 00:06:12 GMT

View Forum Message <> Reply to Message

Several ideas. First, to get more than 32,767 values in the loop, declare the index variable as a long (for i=1L,num do $)

A second idea is to check the ROSAT IDL library and Astro library for suitable routines. As far as my understanding goes, ROSAT has large lists of hits with position info that are converted to a matrix like yours. See (from Astronomy lib's Other_sources.txt:

Gail Reichert at the GSFC High Energy Astrophysics Science Archive
Research Center (HEASARC) has been developing software to analyze
ROSAT data using IDL.   People interested in this software can contact her
at HEASRC::REICHERT (=15552::REICHERT) or reichert@heasrc.gsfc.nasa.gov

The files are available via Decnet on node HEASRC in the directory
IDL:[IDL...].   The procedures are found in the directory

heasrc::idl:[lib.rosat.experimental]

and an extensive LaTeX guide to the procedures can be found in the file

heasrc::idl:[lib.rosat.doc]idl_recipes.tex

The procedures are also available via anonymous FTP on the
machine legacy.gsfc.nasa.gov in the subdirectory rosat/software/idl.
The LaTeX guide is found on the same machine in the software/idl directory.

In article <36v30k$14n@danberg.llnl.gov>, dan@danberg.llnl.gov (Dan Bergmann) wrote:

```
>  I have a large table of data in the form
>
>  LATITUDE   LONGITUDE    VALUE
>
>  There can be a million rows of entries in this 3 colum table.
>  I would like to put the data into a 1x1 degree array (fltarr(360,180))
>  If there are multiple entries for a particular longitude and latitude
>  I would like to sum the values. I could do this in an ugly do loop that
>  looked like
>
>  data=fltarr(360,180)
>  for i=1,num do $
>   data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
>
>  This works fine for small values of num, but for num>32,767 I get an error
>  saying my do loop index is too large. Can I write do loops with limits
>  greater than 32,767 ??   Could I rewrite this in vector syntax using the
>  where statement ??
>  --
>   ************************************************************ ***
>  **  Dan Bergmann                          dbergmann@llnl.gov  **
>  **  Global Climate Research                  fax   (510) 422-5844  **
>  **  Lawrence Livermore National Lab    human (510) 423-6765  **
```

Subject: Re: gridding large amounts of data

Posted by landers on Thu, 06 Oct 1994 12:54:35 GMT

View Forum Message <> Reply to Message

In article <36v30k$14n@danberg.llnl.gov>, dan@danberg.llnl.gov (Dan Bergmann) writes:

[snip]

```
|> for i=1,num do $
|>  data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
```

|>
|> This works fine for small values of num, but for num>32,767 I get an error
|> saying my do loop index is too large. Can I write do loops with limits
|> greater than 32,767 ??

That's easy - use LONGS:

```
  for i = 0L, Num-1 do ...
           ^^
```

You should always use longs when you're addressing arrays - just in case
someday you have a bigger array - you won't always get errors....

Could I rewrite this in vector syntax using the
|> where statement ??

Well, for the loop you've written, you can use:

```
i = lindgen( num )
data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
```

If you're using PV-WAVE, there's a couple of good gridders (in what
used to be the ARL) called FAST_GRID3 and GRID_3D. Sounds like
FAST_GRID3 would work pretty good for your data.

I'm sure IDL has some kind of gridder, but I don't know what it is or
what it does.

;Dave

Subject: Re: gridding large amounts of data
Posted by landers on Thu, 06 Oct 1994 14:44:57 GMT

View Forum Message <> Reply to Message

In article <36v30k$14n@danberg.llnl.gov>, dan@danberg.llnl.gov (Dan Bergmann) writes:
 [snip]

|> for i=1,num do $
|> data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
|>
|> This works fine for small values of num, but for num>32,767 I get an error
|> saying my do loop index is too large. Can I write do loops with limits
|> greater than 32,767 ??

Use LONG integers for loop control:

```
  for i = 0L, Num-1 do ...
           ^^
```

In general, you should always use longs for array indexing operations to prevent just this problem.

|> .... Could I rewrite this in vector syntax using the
|> where statement ??

For the loop you have above, use something like:

```
   i = lindgen( Num )
   data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
```

On gridding - if you're using PV-WAVE, check out FAST_GRID3. I have had very good luck gridding data similar to what you're describing with this function.

If this won't do it for you, check out GRID_3D.

If you have IDL, well, I'm sure they have some sort of gridding stuff, but I can't speak to that...
;Dave

Subject: Re: gridding large amounts of data
Posted by agraps on Fri, 07 Oct 1994 05:44:00 GMT

View Forum Message <> Reply to Message

dan@danberg.llnl.gov (Dan Bergmann) writes:

[..]
> ... I could do this in an ugly do loop that
> looked like

> data=fltarr(360,180)
> for i=1,num do $
> data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)

> This works fine for small values of num, but for num>32,767 I get an error
> saying my do loop index is too large.

This is a kind of idl "gotcha". I wrestled with this on a late evening, a few years ago, when I similarly had to loop for greater than 32,767 times.

The secret is to make i a long integer:

```
for iL=1,num do $
   ^^^
```

And that will solve your problem.

Amara

** **********
Amara Graps   email: agraps@netcom.com
Computational Physicist   vita: finger graps@clio.arc.nasa.gov
Intergalactic Reality   bio: finger -lm agraps@netcom.com
** **********
"Picture a massless particle." --A Koan of Modern Physics

--

** **********
Amara Graps   email: agraps@netcom.com
Computational Physicist   vita: finger graps@clio.arc.nasa.gov
Intergalactic Reality   bio: finger -lm agraps@netcom.com
** **********
"Picture a massless particle." --A Koan of Modern Physics

Subject: Re: gridding large amounts of data
Posted by agraps on Fri, 07 Oct 1994 15:29:28 GMT

View Forum Message <> Reply to Message

I said:
> The secret is to make i a long integer:
> for iL=1,num do $
>   ^^^

Oops!! This one is still getting me! I meant:

```
for i=1L,num
      ^^
```

Amara

--

** **********
Amara Graps   email: agraps@netcom.com
Computational Physicist   vita: finger graps@clio.arc.nasa.gov
Intergalactic Reality   bio: finger -lm agraps@netcom.com

** **********

"Picture a massless particle." --A Koan of Modern Physics

Subject: Re: gridding large amounts of data
Posted by lmudge on Mon, 10 Oct 1994 00:47:20 GMT
View Forum Message <> Reply to Message

In article 0510942006120001@peace.gsfc.nasa.gov, candey@nssdca.gsfc.nasa.gov (Robert M. Candey) writes:
> Several ideas. First, to get more than 32,767 values in the loop, declare the index variable as a long (for i=1L,num do $)
>
> A second idea is to check the ROSAT IDL library and Astro library for suitable routines. As far as my understanding goes, ROSAT has large lists of hits with position info that are converted to a matrix like yours. See (from Astronomy lib's Other_sources.txt:
>
> Gail Reichert at the GSFC High Energy Astrophysics Science Archive
> Research Center (HEASARC) has been developing software to analyze
> ROSAT data using IDL.   People interested in this software can contact her
> at HEASRC::REICHERT (=15552::REICHERT) or reichert@heasrc.gsfc.nasa.gov
>
> The files are available via Decnet on node HEASRC in the directory
> IDL:[IDL...].   The procedures are found in the directory
>
> heasrc::idl:[lib.rosat.experimental]
>
> and an extensive LaTeX guide to the procedures can be found in the file
>
> heasrc::idl:[lib.rosat.doc]idl_recipes.tex
>
> The procedures are also available via anonymous FTP on the
> machine legacy.gsfc.nasa.gov in the subdirectory rosat/software/idl.
> The LaTeX guide is found on the same machine in the software/idl directory.
> _____________
>
> In article <36v30k$14n@danberg.llnl.gov>, dan@danberg.llnl.gov (Dan Bergmann) wrote:
>
>> I have a large table of data in the form
>>
>> LATITUDE   LONGITUDE   VALUE
>>
>> There can be a million rows of entries in this 3 colum table.
>> I would like to put the data into a 1x1 degree array (fltarr(360,180))
>> If there are multiple entries for a particular longitude and latitude
>> I would like to sum the values. I could do this in an ugly do loop that
>> looked like
>>

>> data=fltarr(360,180)
>> for i=1,num do $
>> data(longitude(i),latitude(i)) = data(longitude(i),latitude(i)) + value(i)
>>
>> This works fine for small values of num, but for num>32,767 I get an error
>> saying my do loop index is too large. Can I write do loops with limits
>> greater than 32,767 ??   Could I rewrite this in vector syntax using the
>> where statement ??
>> --
>> ** ***
>> ** Dan Bergmann   dbergmann@llnl.gov **
>> ** Global Climate Research   fax   (510) 422-5844 **
>> ** Lawrence Livermore National Lab   human (510) 423-6765 **

The limit for 16 bit signed integers is 32,767 as stated in the IDL manuals.
I have met this problem of exceeding this limit in a FOR loop and have been
able to get around it by using longword integers. This will give you a limit
of 2,147,483,647 (2^31 - 1) according to the manual.

Leith Mudge

End Of Message

Subject: Re: gridding large amounts of data
Posted by bowman on Sat, 15 Oct 1994 18:59:03 GMT
View Forum Message <> Reply to Message

> The secret is to make i a long integer:
> for iL=1,num do $

This is kind of off the subject, but would it break a lot of things to
make the default integer type in IDL be long rather than short? I have
largely gotten in the habit of writing *all* integers nnL to avoid this
problem. What will happen when we have 64-bit integer support in IDL?

Just some musings on backward compatibility ...

Ken Bowman

--
Dr. Kenneth P. Bowman   409-862-4060
Associate Professor   409-862-4132 fax
Climate System Research Program   bowman@csrp.tamu.edu
Department of Meteorology   PP-Glider
Texas A&M University

College Station, TX   77843-3150