Subject: Alternatives to griddata()
Posted by Andrew[3] on Thu, 24 May 2007 21:25:28 GMT
View Forum Message <> Reply to Message

Hi everyone,

I'm currently using IDL to process remote sensing data of arctic sea ice. Recently, I've been modifying an existing set of IDL programs to handle some new high-resolution data. The main program does work with the new data set, but it runs much more slowly. i.e. from under one minute to about 4 minutes per image. (the images have gone from 400x340 to 1000x850)

I've gone through that program eliminating for loops wherever possible, but it's still quite slow. Yesterday, I got the bright idea to use the systime() function and figure out which part of the program is taking so long. It turns out that the griddata() function is the culprit. For one image, that one function takes up 80-90% of the total program run time! (According to the time outputs anyway.)

I am wondering if there is an alternative method to take irregularly spaced data stored as 3 vectors (xindex, yindex, and data), plot it on a 2D grid and interpolate it that is a bit faster. I'd like to keep using the inverse distance method of interpolation or something with similar behaviour to compare the high-res and low-res data. I've tried some IDL programming of my own, but nothing that I come up with is as fast as griddata.

Any ideas?

Andrew

Subject: Re: Alternatives to griddata()
Posted by Andrew[3] on Fri, 25 May 2007 18:26:42 GMT
View Forum Message <> Reply to Message

On May 24, 5:25 pm, Andrew <Andrew.B.La...@gmail.com> wrote:
> Hi everyone,
>
> I'm currently using IDL to process remote sensing data of arctic sea
> ice. Recently, I've been modifying an existing set of IDL programs to
> handle some new high-resolution data. The main program does work with
> the new data set, but it runs much more slowly. i.e. from under one
> minute to about 4 minutes per image. (the images have gone from
> 400x340 to 1000x850)
>
> I've gone through that program eliminating for loops wherever

> possible, but it's still quite slow. Yesterday, I got the bright idea
> to use the systime() function and figure out which part of the program
> is taking so long. It turns out that the griddata() function is the
> culprit. For one image, that one function takes up 80-90% of the total
> program run time! (According to the time outputs anyway.)
>
> I am wondering if there is an alternative method to take irregularly
> spaced data stored as 3 vectors (xindex, yindex, and data), plot it on
> a 2D grid and interpolate it that is a bit faster. I'd like to keep
> using the inverse distance method of interpolation or something with
> similar behaviour to compare the high-res and low-res data. I've tried
> some IDL programming of my own, but nothing that I come up with is as
> fast as griddata.
>
> Any ideas?
>
> Andrew

Ok, turns out I answered my own question. I just have to make use of the SEARCH_ELLIPSE keyword, and it gives me the result I'm looking for. The images are virtually identical and produced in less than a quarter of the time. Leave it to IDL to provide some extremely handy non-intuitive functionality :)