Subject: Re: on reading NCDF files
Posted by Don Woodraska on Tue, 20 Jan 2004 23:50:20 GMT
View Forum Message <> Reply to Message

On 1/20/04 1:01 AM, in article
3ee6ff80.0401192300.534dc5b1@posting.google.com, "Sangwoo"
<leesw@astro.snu.ac.kr> wrote:

> Hi everyone!
>
> I'm gonna read a NCDF file. It may contain several variables within
> itself. When I extract a variable named "image", the procedure is as
> follows:
>
> cid=ncdf_open('test.nc')
> vid=ncdf_varid(cid,'image')
> ncdf_varget,cid,vid,image
>
> But suppose I don't know the name of each variable unfortunately. Is
> there any way to figure out the details of the included
> variables(name, dimension, etc.)? The second command fails when I put
> a wrong variable. And is there any way to read the included variables
> all at once? (something like /all keyword)
>
> I wonder if there's any way to figure out all details of the included
> variables from an NCDF file directly.

I'm the science data processing manager for a NASA instrument called the Solar EUV Experiment. For what it's worth, we use a (semi-generic) routine that my supervisor wrote called read_netcdf.pro and a companion routine write_netcdf.pro for all levels of our daily routine science data processing. We routinely read and write netcdf files containing structures and/or arrays of structures up to a nested depth of 4.

These procedures are available at
http://lasp.colorado.edu/see/see_software.html

There are examples in the ftp directory where the code resides, too. It's worked on every netcdf file I've come across, even those created by the other 3 instruments on our spacecraft. Read_netcdf.pro returns an anonymous structure (or array of structures) and an optional attributes string array.

These IDL routines insulate the end user from all the gory details of reading and writing NetCDF files. They accept almost any IDL structure.

It's not perfect, and some of it may be darn ugly. The price for hiding the user from the details is performance. It tends to be slow for files that are larger than 30 MB or so. Also, you'll be much better off reading the data

file from a local fast drive, than a network drive. We routinely read and write structures that are nested 4 structures deep, so that's one limit (depth of 4). Also, objects and pointers are not allowed, but that's really a file format restriction.

I hope these routines will save at least one person the trouble of writing custom read and write routine.

Cheers,
Don Woodraska
