Subject: on reading NCDF files
Posted by leesw on Tue, 20 Jan 2004 08:01:06 GMT
View Forum Message <> Reply to Message

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.

Subject: Re: on reading NCDF files
Posted by Liam Gumley on Tue, 20 Jan 2004 16:12:32 GMT
View Forum Message <> Reply to Message

"Sangwoo" <leesw@astro.snu.ac.kr> wrote in message
news:3ee6ff80.0401192300.534dc5b1@posting.google.com...
> 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.

You may wish to try the following sample programs from my book:

NCDF_VARDIR: returns a list of variable names in a netCDF file
NCDF_ATTDIR: returns a list of attribute names associated with a given variable

The programs are available at

http://www.gumley.com/PIP/About_Book.html

in the "Sample Programs" archive files. Reading and writing netCDF files in IDL is explained in detail in Chapter 4.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: on reading NCDF files
Posted by K. Bowman on Tue, 20 Jan 2004 17:38:32 GMT

View Forum Message <> Reply to Message

In article <3ee6ff80.0401192300.534dc5b1@posting.google.com>,
 leesw@astro.snu.ac.kr (Sangwoo) wrote:

Several readers have suggested using the NCDF inquire functions built-in to IDL. I often find it quicker to run

```
ncdump -h myfile.ncd
```

at the command line.

You do have to install the netCDF library, which is available from

 http://my.unidata.ucar.edu/content/software/netcdf/index.htm l

In my opinion, it is worth installing just to get ncdump.

Ken Bowman

Subject: Re: on reading NCDF files
Posted by Paul Van Delst[1] on Tue, 20 Jan 2004 18:07:32 GMT

View Forum Message <> Reply to Message

Sangwoo 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.

Yes, there is. For me at least, that's the point of netCDF.

I have a simple-minded read_ncdf() function that may do what you want. You can read all the variable data by default:

```
IDL> error_status=read_ncdf('sndr_g12.EmisCoeff.nc',x,/quiet)
% Compiled module: READ_NCDF.
% Compiled module: IS_NCDF.
% Loaded DLM: NCDF.
IDL> help, x, /struct
** Structure <82417b4>, 10 tags, length=37856, data length=37856, refs=1:
   RELEASE         LONG                 2
   VERSION         LONG                 2
   THETA_OFFSET    DOUBLE         0.10000000
   THETA_MAX       DOUBLE          65.100000
   NCEP_SENSOR_ID  LONG      Array[18]
   WMO_SATELLITE_ID
                   LONG      Array[18]
   WMO_SENSOR_ID   LONG      Array[18]
   SENSOR_CHANNEL  LONG      Array[18]
   WIND_SPEED      DOUBLE    Array[13]
   EMIS_COEFFICIENTS
                   DOUBLE    Array[4, 5, 18, 13]
```

Or just the data you want:

```
IDL>  error_status=read_ncdf('sndr_g12.EmisCoeff.nc',x,variable_li st=['Wind_Speed'],/quiet)
IDL> help, x, /struct
** Structure <82468d4>, 1 tags, length=104, data length=104, refs=1:
   WIND_SPEED      DOUBLE    Array[13]
```

Or just the global attributes:

```
IDL>  error_status=read_ncdf('sndr_g12.EmisCoeff.nc',x,/global_att ributes,/quiet)
IDL> help, x, /struct
** Structure <8240f34>, 7 tags, length=84, data length=84, refs=1:
   WRITE_MODULE_HISTORY
                  STRING    '$Id: write_emiscoeff_netcdf.pro,v 2.0 2003/06'...
   CREATION_DATE_AND_TIME
                  STRING    'Tue Sep 16 16:53:38 2003'
   TITLE          STRING    'Emissivity fit coefficients for GOES-12 SOUND'...
   HISTORY        STRING    '$Id: compute_emissivity_coefficients.pro,v 2.'...
   SENSOR_NAME    STRING    'SOUNDER'
   PLATFORM_NAME  STRING    'GOES-12'
   COMMENT        STRING    'Sensor emissivity created by Convolution:Spec'...
```

or all the variables and their attributes:

```
IDL>  error_status=read_ncdf('sndr_g12.EmisCoeff.nc',x,/variable_a ttributes,/quiet)
IDL> help, x, /struct
** Structure <8241f2c>, 10 tags, length=38024, data length=38024, refs=1:
   RELEASE        STRUCT    -> <Anonymous> Array[1]
   VERSION        STRUCT    -> <Anonymous> Array[1]
   THETA_OFFSET   STRUCT    -> <Anonymous> Array[1]
   THETA_MAX      STRUCT    -> <Anonymous> Array[1]
   NCEP_SENSOR_ID STRUCT    -> <Anonymous> Array[1]
   WMO_SATELLITE_ID
                  STRUCT    -> <Anonymous> Array[1]
   WMO_SENSOR_ID  STRUCT    -> <Anonymous> Array[1]
   SENSOR_CHANNEL STRUCT    -> <Anonymous> Array[1]
   WIND_SPEED     STRUCT    -> <Anonymous> Array[1]
   EMIS_COEFFICIENTS
                  STRUCT    -> <Anonymous> Array[1]
IDL> help, x.emis_coefficients, /struct
** Structure <8240f3c>, 3 tags, length=37464, data length=37464, refs=2:
   DATA           DOUBLE    Array[4, 5, 18, 13]
   LONG_NAME      STRING    'Emissivity model fit coefficients.'
   UNITS          STRING    'None.'
```

And you can use the tag_names routine to act on just the variables.

cheers,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC

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
