Geosci. Model Dev. Discuss., 8, C269–C272, 2015
www.geosci-model-dev-discuss.net/8/C269/2015/

Geoscientific Model Development
Discussions
Open Access

Interactive comment on "NCIO 1.0: a simple Fortran NetCDF interface" *by* A. Robinson and M. Perrette

Anonymous Referee #1

Received and published: 26 March 2015

General comments

It's a good idea to provide a library that will make netcdf IO in fortran easier, even at the cost of performance (at least for prototyping).

The paper is fairly easy to read. It was very useful to be able to read the fortran example provided on github (test_ncio.f90), so it would probably be a good idea mention the github repository at the beginning of the paper and also include a "Download" section on github page to help novice users download the code

eg git clone https://github.com/alex-robinson/ncio.git

Other comments

line 25: the paper should mention the CMOR (Climate Model Output Rewriter) library

http://www2-pcmdi.llnl.gov/cmor

and the CF (Climate and Forecast) conventions

http://cfconventions.org/

Explicitly mentioning CF will also make it clearer for some people why you are using the long_name/standard_name/... attributes later. Are you speaking about CF on page 304 when you mention "convention attributes"?

line 23: there should be some more explanation about why and when some people have to use nc_write_map. Is it something you need when using dimensions defined in km?

There should be a note about the use of the extended dimension and how, for instance, the time axis can be extended with repeated calls to nc_write. test_ncio.f90 uses the unlimited=TRUE parameter in nc_write_dim, but the documentation of nc_write_dim on page 315 does NOT mention this useful parameter

The paper does not mention if the library will work with netcdf3 or netcdf4

The paper does not mention at all how the errors are handled.

What happens if you try to nc_create on an existing file? Do you get an error or will the existing file be erased?

It is a limitation (an acceptable limitation) to be able to only read and write string attributes. It could be a good idea to use a more explicit name for the functions

nc_read_attr => nc_read_str_attr

nc_write_attr => nc_write_str_attr

The tables describing the function calls mix input and output parameters and do not explicitely list the allowed types. This should be improved e.g, p 311, nc_read_attr

value (OUT) value of the STRING attribute to be read

Typos =====

page 302, line 4, cUmbersome, "module NCIO" -> "NCIO module"

line 11, revolutionized?

line 23, common TASKS?

page 303, line 5, NCIO module

line 9, hopes->hope?

line 20, analOgous?

line 21, "the subroutine will read all the variable"?

page 304, line 13, "like with"?

page 305, line 12, "be prepared" => "be ready"?

line 17, exCeption

page 306, line 15, to A file

line 27, use the correct "ff" in "affect"

page 307 and 308, it would be nice to rewrite a bit the "Conclusion" (revert a bit the order of some sentences?)

p 310, specify in another way that nc_size returns an integer.

p 313, varname, "name of the variable the attribute should be associated to"

p 315, add the "unlimited" parameter

p 316, remove "fortran data type of" for the 'dat' parameter