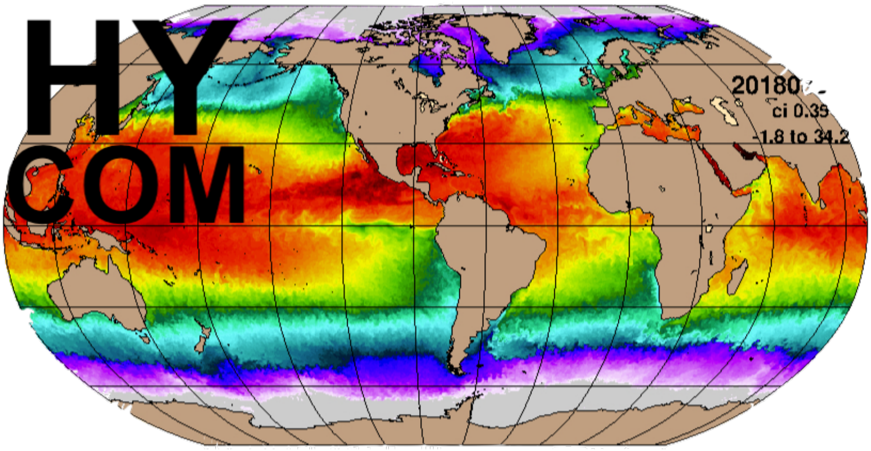
C++ NetCDF Interface for the HYCOM Global Ocean Model

User Manual

Blake Cole, Massachusetts Institute of Technology

August 22nd, 2018



Contents

1 Introduction

HYbrid Coordinate Ocean Model (HYCOM)

The HYCOM consortium is a multi-institutional effort sponsored by the National Ocean Partnership Program (NOPP), as part of the U. S. Global Ocean Data Assimilation Experiment (GODAE), to develop and evaluate a data-assimilative hybrid isopycnal-sigma-pressure (generalized) coordinate ocean model (called HYbrid Coordinate Ocean Model or HYCOM). The GODAE objectives of three-dimensional depiction of the ocean state at fine resolution in real time, provision of boundary conditions for coastal and regional models, and provision of oceanic boundary conditions for a global coupled ocean-atmosphere prediction model, are being addressed by a partnership of institutions that represent a broad spectrum of the oceanographic community. For more information, please visit the HYCOM website: `https://hycom.org/`.

Accessing HYCOM NetCDF Data

According to UCAR, netCDF (Network Common Data Form) is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. However, while netCDF effectively enables the efficient storage exceedingly large datasets, accessing these datasets – or small pieces of them – is not always an intuitive exercise.

The ***netcdf_hycom*** application is designed to extract a user-specified subset of spatio-temporal data from the HYCOM Global Ocean Forecast System *(GOFS 3.1)*, via a user-friendly C++ interface. The GOFS 3.1 model contains four 4D fields, including:

- ocean temperature [°C]
- salinity [psu]
- current x-velocity [m/s]
- current y-velocity [m/s]

It also includes five 3D fields, which are defined at only one depth, including:

- surface level [m]
- benthic ocean temperature [°C]
- benthic salinity [psu]
- benthic current x-velocity [m/s]
- benthic current y-velocity [m/s]

As of August 22nd, 2018, ***netcdf_hycom*** is only capable of extracting two of the available nine fields: *ocean temperature* and *salinity.* However, it could be expanded to read the other fields with truly minimal effort. Once the data is extracted from the HYCOM database, it is stored in 4D *c-style* arrays. If desired, the user can request that a new netCDF file be created, representing the desired subset of interest. This functionality could be particularly useful when conducting field work in areas without reliable internet access.

To ensure that the correct data has been written, the user may find it useful to inspect the new netCDF file using ***netcdf_hycom_readonly***. This auxiliary application looks for the netCDF file created by *netcdf_hycom*, in a fixed directory `[netcdf_hycom/data/salt_temp_4D.nc]`, inspects it, and prints its contents.

2 System Requirements

1. ***Linux Ubuntu 18.04.01 LTS***
   1. (a) Required:
      - i. GNU Compiler (`g++`)
      - ii. NetCDF Library for C++ (`libnetcdf-c++4-dev`)
        Dependencies:
        - `hdf5-helpers`
        - `libaec-dev`
        - `libcurl4-gnutls-dev`
        - `libhdf5-cpp-100`
        - `libhdf5-dev`
        - `libnetcdf-dev`
   2. (b) Recommended:
      - i. NetCDF Viewer (`nvciew`)
      - ii. NetCDF Operators (`nco`)
      - iii. Climate Data Operators (`cdo`)
      - iv. Optional netCDF C++ Libraries
        - `libnetcdf-c++4` (legacy)
        - `libnetcdf-c++4-1`
        - `libnetcdf13`

2. ***macOS High Sierra 10.13.5***
   1. (a) Required:
      - i. GNU Compiler (`g++`)
      - ii. NetCDF Library for C++ (`netcdf-cxx4`)
        Dependencies:
        - `hdf5`
        - `netcdf`
   2. (b) Recommended:
      - i. NetCDF Viewer (`nvciew`)
      - ii. NetCDF Operators (`nco`)
      - iii. Climate Data Operators (`cdo`)

3 Download via GitHub

This program is available free of charge, under a Creative Commons License, via GitHub.

`https://github.mit.edu/blerk/netcdf_hycom.git`

4 Compiling

The netCDF C++ libraries downloaded via *MacPorts* or *apt-get* will be located in non-default directories; accordingly, a few command line switches are required to compile correctly. The examples below are intended to be instructional; while they will work "out-of-the-box" for many users, they are provided for guidance only. It is important to remember that the relevant libraries and header files may exist in other non-default locations, depending on the user's operating system and build structure.

Linux Ubuntu 18.04.01 LTS

```
$ g++ -o netcdf_hycom netcdf_hycom.cpp -Wall -I/usr/include
-L/usr/lib/x86_64-linux-gnu -lnetcdf_c++4
```

macOS High Sierra 10.13.5

```
$ g++ -o netcdf_hycom netcdf_hycom.cpp -Wall -I/opt/local/include
-L/opt/local/lib -lnetcdf_c++4
```

Description of Command Line Switches:

1. Show (almost) all compiler warnings.
   `-Wall`
2. Augment the g++ "include" path (where the .h header file lives).
   `-I/..`
3. Augment the g++ "link" path (where the .a static library file lives).
   `-L/..`
4. Specify which library needs to be included (libnetcdf_c++4).
   `-lnetcdf_c++4`

5 Usage

Provided you have managed to successfully compile the program on your machine, you are ready to extract a chunk of HYCOM model data, at your leisure! Thankfully, ***netcdf_hycom*** interfaces directly with the HYCOM database via OPeNDAP, so you will not need to visit the HYCOM website. All you need is an internet connection!

5.1 Specifying the Bounding Box

The program allows users to specify a spatio-temporal bounding box via one of two methods:

1. **Command Line Switches**

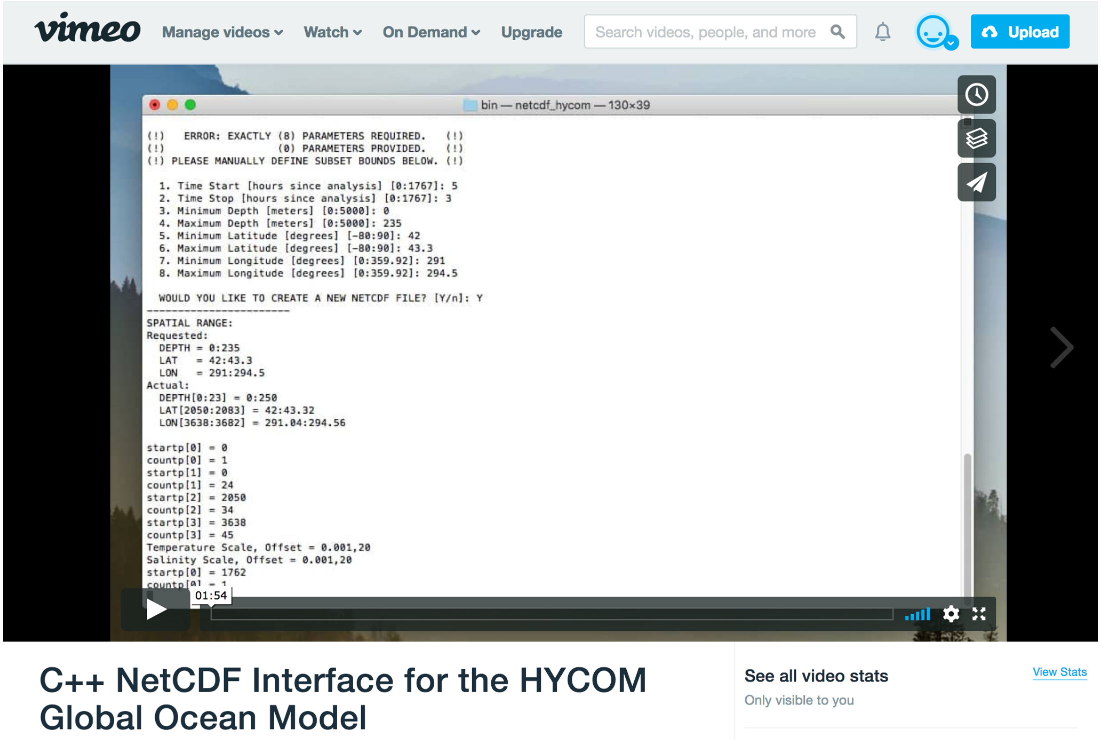
`-time=current`	:	time: most recent data record only
`--hours=[INT]`	:	time: block of hours to now
`--days=[INT]`	:	time: block of days to now
`--weeks=[INT]`	:	time: block of weeks to now
`--tmin=[INT]`	:	time: start time, hours from now
`--tmax=[INT]`	:	time: end time, hours from now
`--depthmin=[FLOAT]`	:	depth: shallowest depth
`--depthmax=[FLOAT]`	:	depth: deepest depth
`--latmin=[FLOAT]`	:	latitude: southern edge
`--latmax=[FLOAT]`	:	latitude: northern edge
`--lonmin=[FLOAT]`	:	longitude: western edge
`--lonmax=[FLOAT]`	:	longitude: eastern edge
`--newfile=true`	:	write new netCDF file

2. **User Input During Runtime**

The program will first look for a total of (8) parameters provided on the command line, to fully define a spatio-temporal bounding box: minimum and maximum values for time, depth, latitude, and longitude. In addition, the user can request that a new netCDF file be written by using the **--newfile=true** command line switch. Use the **--help** command for a full list of acceptable command line inputs.

If the user prefers to avoid using the command line, the program will query the user during runtime; at this point, the user can specify the bounding box, and whether or not they would like to create a new netCDF file.

For demonstration video, please visit: `https://vimeo.com/287917189`



```
(!)   ERROR: EXACTLY (8) PARAMETERS REQUIRED.   (!)
(!)                  (0) PARAMETERS PROVIDED.   (!)
(!) PLEASE MANUALLY DEFINE SUBSET BOUNDS BELOW. (!)

  1. Time Start [hours since analysis] [0:1767]: 5
  2. Time Stop [hours since analysis] [0:1767]: 3
  3. Minimum Depth [meters] [0:5000]: 0
  4. Maximum Depth [meters] [0:5000]: 235
  5. Minimum Latitude [degrees] [-80:90]: 42
  6. Maximum Latitude [degrees] [-80:90]: 43.3
  7. Minimum Longitude [degrees] [0:359.92]: 291
  8. Maximum Longitude [degrees] [0:359.92]: 294.5

  WOULD YOU LIKE TO CREATE A NEW NETCDF FILE? [Y/n]: Y
-----------------------
SPATIAL RANGE:
Requested:
  DEPTH = 0:235
  LAT   = 42:43.3
  LON   = 291:294.5
Actual:
  DEPTH[0:23] = 0:250
  LAT[2050:2083] = 42:43.32
  LON[3638:3682] = 291.04:294.56

startp[0] = 0
countp[0] = 1
startp[1] = 0
countp[1] = 24
startp[2] = 2050
countp[2] = 34
startp[3] = 3638
countp[3] = 45
Temperature Scale, Offset = 0.001,20
Salinity Scale, Offset = 0.001,20
startp[0] = 1762
countp[0] = 1
```

C++ NetCDF Interface for the HYCOM Global Ocean Model

*Figure 1: Screenshot from **netcdf_hycom** demonstration video.*

6 Contact

For assistance with this program, please contact:

- **Blake Cole**
  `blerk@mit.edu`
- **Henrik Schmidt**
  `henrik@mit.edu`