JPL - Student Independent Research Intern (2023), 1–19

ARTICLE

Streamlining Documentation Process of ECCO version 4 revision 4 Datasets

Jose Juan Gonzalez Jr.[*,1] and Dr. Ian Fenty[*,2]

[1]Pasadena City College, Pasadena, United States
[2]Jet Propulsion Laboratory, La Cañada Flintridge, United States
[*]Corresponding author: jgonzalez281@go.pasadena.edu; ian.fenty@jpl.nasa.gov

(Completed 4 May 2023)

Abstract

In this project, we concentrate on streamlining the interpretation and understanding of satellite oceanographic and sea-ice datasets for the Estimating the Circulation and Climate of the Ocean (ECCO) program at the Jet Propulsion Laboratory (JPL). The main objective of the project is to streamline the efficiency of data interpretation to eventually support data retrieval and processing, thereby benefiting ECCO program's mission of generating estimations of ocean circulation and state estimates. Utilizing Latex, Python and its associated libraries such as xarray, matplotlib, cartopy, and ECCO's own library ecco_v4_py, we developed code that not only organizes metadata from JSON files and generates latex tables, but also parses data from netCDF files and creates plot figures and projections. Our user-friendly program allows for command line arguments and supports the generation of data specification documents, offering insights into netCDF granule data. The project's results have potential to enhance the ECCO team's workflow, promoting the development of ocean climate models and aiding decision-making. The evolution of this project holds significant potential for advancing our abilities to understand and model Earth's changing ocean climate dynamics.

Keywords: JPL; ecco; granule; metadata;

Abbreviations: JPL: Jet Propulsion Laboratory ECCO: Estimating the Circulation and Climate of the Ocean

1. Overview

1.1 Project Background

1.1.1 Oceanography Introduction

Oceanography, the study of the oceans which encompasses a wide arrange of variables such as ecosystem dynamics, ocean currents, fluxes of chemicals and more, is significant in increasing our understanding of the earth's total system and "natural and human-induced changes on the global environment." [1] Oceanographic research plays a huge role in understanding the world's climate and through collecting and analyzing long-term ocean data gathered from satellites, teams such as Estimating the Circulation and Climate of the Ocean (ECCO) are able to make sense of ocean data and understand the ocean in profoundly new ways.

The oceans act as repositories for heat and carbon dioxide and the oceans encompasses about 70% percent of the earth's surface, therefore they are very significant actors for all the heat and carbon dioxide on the earth. Throughout the lifetime of the earth, as climate change progresses, the ocean has been increasing the intake of heat and carbon dioxide from human impact. Studying these changes would be beneficial in understanding the full and complete picture behind the earth's constantly changing climate.

By investigating phenomena such as sea-level rise, heat absorption, changes in ocean circulation, and more, scientists can make accurate predictions and forecasting using more efficient strategies and models. Scientists can also figure out methods of mitigating and adapting to climate changes. Therefore, oceanography's contribution to the sciences is not only a matter of scientific exploration but a very important agent in the informed policy-making sustainable management of our planet's resources.

1.1.2 ECCO Program

Due to the growing concerns about climate change, in the 1980s development in technology made it conceivable to determine actual ocean datasets. The implementation of a global circulation model (GCM) seemed more possible than realistic. However, observational technologies were so inharmonious and there could only be a true synchronized estimate of the ocean through unification of observational technology into a GCM. [2]

The ECCO project team was created for the aspiration of formulating the best possible state estimates of the ocean circulation and it's role in our climate through the combination of observational data through different technologies unified into a GCM. [2] ECCO is continuously expanding the scope and fidelity of ocean state estimation. [3] The long term mission of ECCO now is to increase accessibility of global high-resolution coupled ocean/sea-ice/biochemical state estimates for the globally shared scientific advancement of oceanography.

1.1.3 ECCO role

Data insight is a core outcome of the ECCO project. The quality and precision of the model's output hinge upon the accurate gathering of oceanic data. The program utilizes a wide array of global ocean datasets which encompass a multitude of ocean observations. These datasets include variables such as temperature, salinity, sea surface height, and more.

However, the sole accumulation of data is only a portion of the equation. What sets ECCO apart is its emphasis on the statistical consistency between the collected data and the employed ocean general circulation models. This means that ECCO's solutions aren't just a match for most ocean observations, they are also in alignment with the models' predictions. The goal is to minimize the discrepancies between the model output and the observational data, leading to a best fit or statistically consistent solution.

By ensuring statistical consistency, ECCO produces reliable insights that contribute significantly to the scientific understanding of oceanic phenomena. The focus on statistical consistency also helps avoid misleading conclusions that might be produced from other model mismatches and the focus also enhances credibility of the results making them a valuable asset in oceanography and in the scientific effort to understand and address climate change.

1.2 Nature and Scope of the Problem

1.2.1 Challenges in Interpreting and Utilizing Acquired Data

Interpreting and utilizing acquired data poses numerous challenges, especially in the field of oceanographic research. Once data has been collected, researchers must navigate through complex datasets, decipher metadata, and understand the meaning that the data expresses. This process is time-consuming and requires a high level of expertise and concentration.

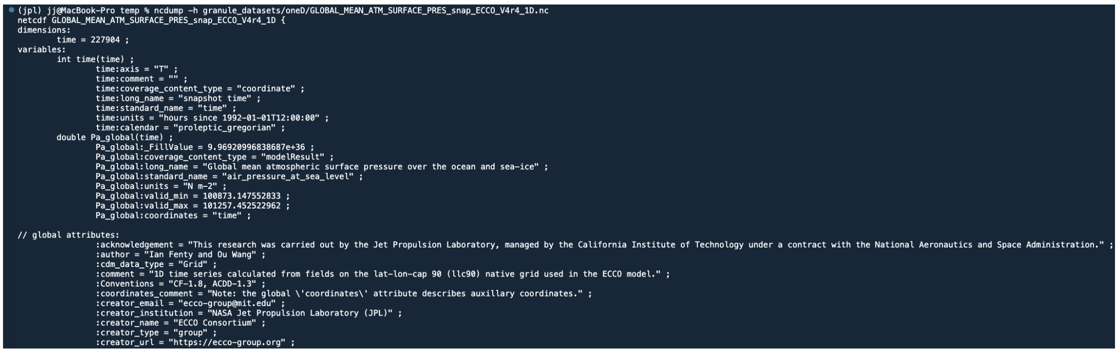
There is also the probability of errors in interpretation which leads to faulty conclusions negatively affecting research outcomes. Given the complexity of oceanographic data, ensuring accuracy in interpretation of the data is an important task. These challenges emphasize the need for an effective means of simplifying data interpretation and utilization, thereby enhancing the efficiency and accuracy of oceanographic research.

1.2.2 Managing and Understanding Large, Complex Datasets

Large and complex datasets, which is the main output of ECCO's model and oceanographic research in general, present difficulties in interpretation and understanding. The datasets encompass different types and sizes of data which can reach such large amounts. While the size and complexity grows, so does the challenge of navigating and understanding it.

Navigating large datasets requires significant computational resources and techniques that can be compiled using advanced computational libraries on the Python programming language. Understanding the data requires a foundational understanding of the data in general. The absence of effective data navigation and understanding strategies serves as a barrier to other scientists. Therefore, the absence could lead to slow or negative research progress, highlighting the need for improved data navigation and understanding solutions.

1.2.3 The Importance of Efficient Parsing and Documentation of netCDF Files and Metadata



```
(jpl) jj@MacBook-Pro temp % ncdump -h granule_datasets/oneD/GLOBAL_MEAN_ATM_SURFACE_PRES_snap_ECCO_V4r4_1D.nc
netcdf GLOBAL_MEAN_ATM_SURFACE_PRES_snap_ECCO_V4r4_1D {
dimensions:
        time = 227904 ;
variables:
        int time(time) ;
                time:axis = "T" ;
                time:comment = "" ;
                time:coverage_content_type = "coordinate" ;
                time:long_name = "snapshot time" ;
                time:standard_name = "time" ;
                time:units = "hours since 1992-01-01T12:00:00" ;
                time:calendar = "proleptic_gregorian" ;
        double Pa_global(time) ;
                Pa_global:_FillValue = 9.96920996838687e+36 ;
                Pa_global:coverage_content_type = "modelResult" ;
                Pa_global:long_name = "Global mean atmospheric surface pressure over the ocean and sea-ice" ;
                Pa_global:standard_name = "air_pressure_at_sea_level" ;
                Pa_global:units = "N m-2" ;
                Pa_global:valid_min = 100873.147552833 ;
                Pa_global:valid_max = 101257.452522962 ;
                Pa_global:coordinates = "time" ;

// global attributes:
                :acknowledgement = "This research was carried out by the Jet Propulsion Laboratory, managed by the California Institute of Technology under a contract with the National Aeronautics and Space Administration." ;
                :author = "Ian Fenty and Ou Wang" ;
                :cdm_data_type = "Grid" ;
                :comment = "1D time series calculated from fields on the lat-lon-cap 90 (llc90) native grid used in the ECCO model." ;
                :Conventions = "CF-1.8, ACDD-1.3" ;
                :coordinates_comment = "Note: the global \'coordinates\' attribute describes auxillary coordinates." ;
                :creator_email = "ecco-group@mit.edu" ;
                :creator_institution = "NASA Jet Propulsion Laboratory (JPL)" ;
                :creator_name = "ECCO Consortium" ;
                :creator_type = "group" ;
                :creator_url = "https://ecco-group.org" ;
```

Figure 1. Example of the contents of an ECCO 1-dimensional dataset using terminal command ncdump

Network Common Data Form (netCDF) files are a common data type in oceanography because of their purpose for storing multidimensional scientific data. Efficiently parsing these files to understand their contents is vital for efficient data analysis and contributions to oceanography. However, this task can be burdensome to scientists who are not acclimated to the metadata conventions used in the netCDF datasets.

The purpose of metadata in netCDF files is to provide context. Essential context includes information about the data's origin, structure, and meaning. The documentation of this information in an articulate manner is an essential task that can streamline the total data understanding and analysis process. The creation of an accessible document that describes the contents of the netCDF dataset fields can save time and reduce potential errors in researchers' data analysis process. Therefore, this highlights the important task of efficient extraction, parsing and documentation of netCDF files and metadata in oceanographic research.

1.3 Rationale for the Project

Streamlining the data interpretation process will significantly improve the research workflow, reducing the time required to interpret and utilize the data. Consequently, leading to faster generation of results that are more accurate and insightful to the field of oceanography. For ECCO, this means greater accessibility in understanding their ocean climate model and output contributing to a better understanding of climate change which could potentially influence policy and decision-making in climate science. In the future, as our ECCO model increases in complexity, so too much our tools for data interpretation. By streamlining that process, the project can be setup for further future advancement.

2. Results

2.1 Implementation of Python Libraries

Python is a suitable platform for the project due to its versatility, efficiency, and extensive ecosystem of computational and scientific libraries that it offers. Libraries such as xarray, which simplifies working with multi-dimensional arrays, and matplotlib, a powerful visualizations library, formed the foundational workflow for handling and visualizing complex datasets. Cartopy, a library used for geospatial data processing, was vital in refined map generation and geographical computations. Lastly, in conjunction with ECCO's self developed Python library, ecco_v4_py, it streamlined interaction with ECCO model dataset outputs. Together, these libraries provided a comprehensive toolkit for understanding the contents of the netCDF datasets' fields.

The JSON python library was pivotal in parsing through json files and extracting information on global and variable attributes. The purpose and output would be used for creating tables that displayed their definitions and details in a document. The versatility and portability of JavaScript Object Notation (JSON) made it ideal for defining the metadata of netCDF files thereby increasing the efficiency of the document generator project.

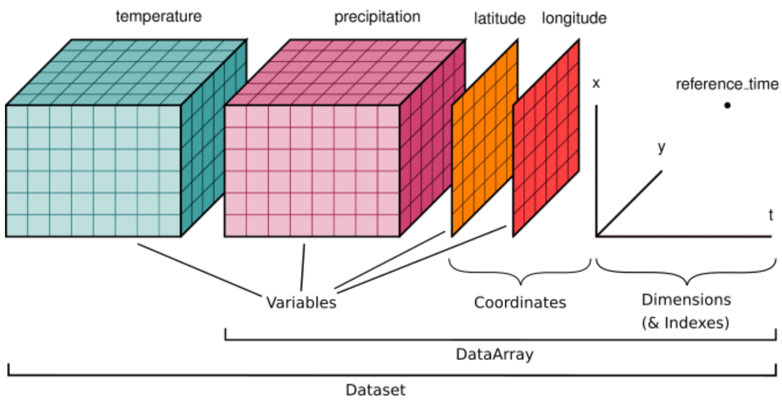


Figure 2. Example of an xarray DataSet with DataArrays

The xarray library was a core library in the documentation process, with its dataset object mirroring the netCDF design structure. Therefore, it was an excellent container and medium to navigate through the netCDF files, document its contents, and extract visualizations.

Matplotlib was another valuable library that harnessed the power of python to create figures representing the data on the netCDF datasets. There was certain modifications made to the field DataArrays, such as setting the depth value to 0 or 1 to craft plots suited for an example representation. The visual representation was essential in understanding the ECCO model's datasets because a visual representation can convey complex data more effectively and faster than words.

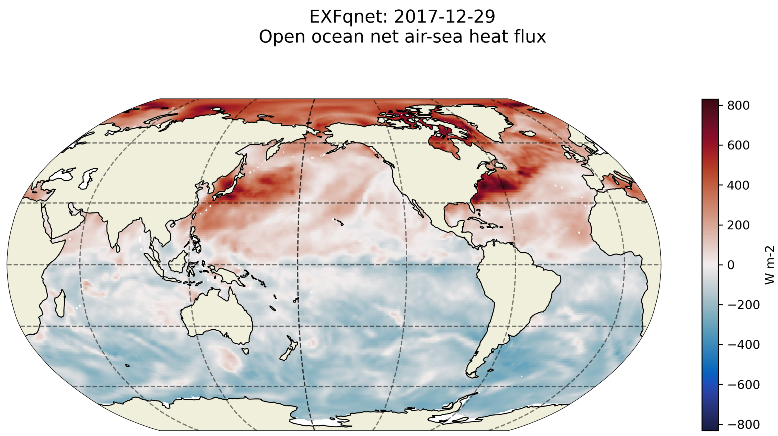


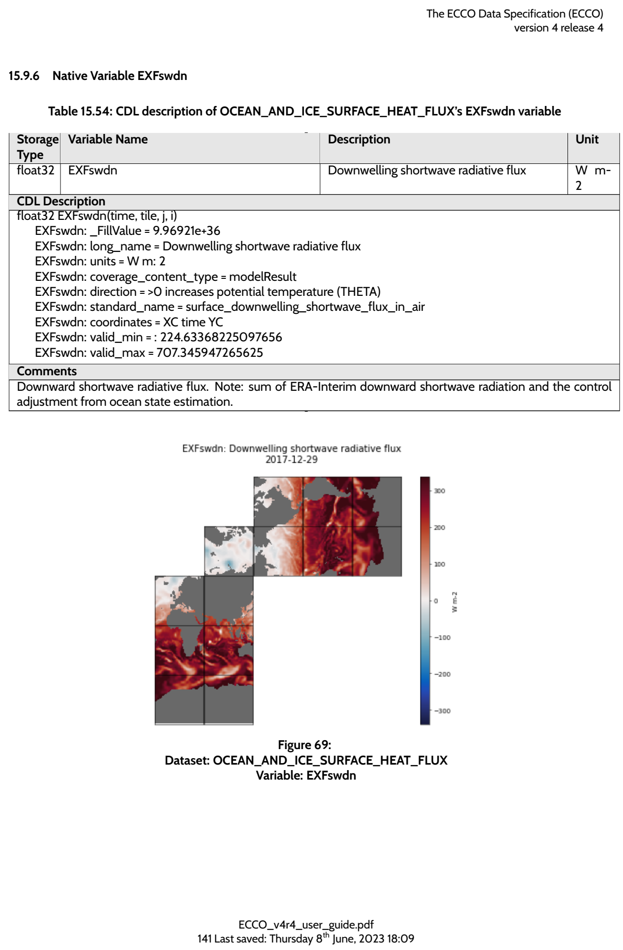
Figure 3. Plot of latlon dataset: OCEAN_AND_ICE_SURFACE_HEAT_FLUX field: EXFqnet

Plotting the projections would not have been possible without cartopy because cartopy was able to facilitate the designation of the colormaps and determine the minimum and maximum for the fields. The foundation for all plotting was built on top of ecco_v4_py which already had functions for certain projections. However, my task was to further streamline process of viewing the data and set a default viewing experience for all fields output from the ECCO project.

2.2 Overview of the Created Document

Through the power of LaTeX and inspiration from the Group for High Resolution Sea Surface Temperature (GHRSST), a documentation template was developed to use as a skeleton for the ECCO version 4 revision 4 netCDF documentation. The GHRSST Data Specification (GDS) Version 2.0 [4] provided a well structured technical specification of the products and metadata underlying their model SST outputs which happen to be netCDF files as well. Their careful detailing of global attributes, variable attributes, and the description of dimensions, variables, and attributes of all dataset fields correlated with the objectives of my project.

The ECCO documentation provides insight into the structure and definitions of various components of the netCDF datasets. For each dataset field, a developed Python program was ran to create a description and visualization. This approach applied across every single field for each dataset output by the ECCO model, ensuring a detailed overview of the data. The document now serves as a guide to understanding and interpreting the vast information contained in the ECCO model's outputs.



The ECCO Data Specification (ECCO)
version 4 release 4

15.9.6 Native Variable EXFswdn

Table 15.54: CDL description of OCEAN_AND_ICE_SURFACE_HEAT_FLUX's EXFswdn variable

Storage Type	Variable Name	Description	Unit
float32	EXFswdn	Downwelling shortwave radiative flux	W m-2

CDL Description

float32 EXFswdn(time, tile, j, i)
EXFswdn: _FillValue = 9.96921e+36
EXFswdn: long_name = Downwelling shortwave radiative flux
EXFswdn: units = W m: 2
EXFswdn: coverage_content_type = modelResult
EXFswdn: direction = >0 increases potential temperature (THETA)
EXFswdn: standard_name = surface_downwelling_shortwave_flux_in_air
EXFswdn: coordinates = XC time YC
EXFswdn: valid_min = : 224.63368225097656
EXFswdn: valid_max = 707.345947265625

Comments

Downward shortwave radiative flux. Note: sum of ERA-Interim downward shortwave radiation and the control adjustment from ocean state estimation.

Figure 69:
Dataset: OCEAN_AND_ICE_SURFACE_HEAT_FLUX
Variable: EXFswdn

ECCO_v4r4_user_guide.pdf
141 Last saved: Thursday 8th June, 2023 18:09

Figure 4. Example of a page for a field being described

2.3 Effects on Data Interpretation Process

This project increases the data interpretation and understanding process of the ECCO project by introducing a streamlined approach to extracting the netCDF dataset fields and generating visualizations. These visualizations aided in creating insight and understanding towards the intricacies of the field data.

Furthermore, another notable outcome and byproduct of the project was the Python script designed for plotting dataset fields. The tool served as a efficient solution for quickly visualizing the netCDF fields. It has the flexibility of plotting specific fields, all fields, or even coordinate-based fields when provided the required command line arguments. This facilitated more efficient data interpretation and offered personalized insights based on user's instructions.

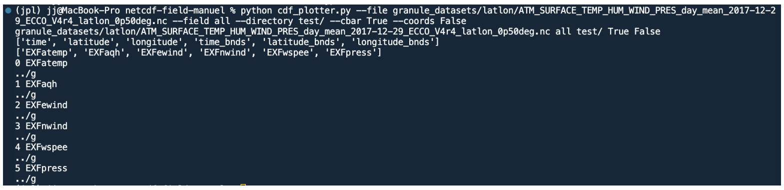


Figure 5. Example of Python script for plotting dataset fields

3. Conclusion

While the original project's goal was to streamline the data acquisition process of the ECCO model. After discussions with my mentor, a more suitable use of time was to focus on streamlining the understanding process of the datasets. Thereby, ensuring that other interested researchers in ECCO's datasets can access and start analyzing the data in a more efficient manner.

This project successfully achieved the new main goal: to streamline the data understanding and interpretation process within the ECCO program. Utilizing the Python libraries, it has generated comprehensive documentation for the ECCO netCDF datasets.

The enhancements in data interpretation and acquisition introduced by the project have potential to faciliated oceanographic research, contributing towards a greater understanding of climate change patterns. Therefore, by making ECCO datasets more accessible, the project supports the broader scientific mission of fostering better informed climate modeling and forecasting.

Future work can be built upon this project by refining organization and setup for a more data-efficient process in the extraction of the netCDF files, while keeping the comprehensive contents of the document the same.

In reflection, the project is a significant step forward in handling and interpreting ECCO's climate data. It demonstrates the power of Python in scientific computing and highlights the importance of accessibility in data acquisition and interpretation.

4. Method

The methodology of this project involved several steps and utilized various technologies, including Python libraries, the LaTeX document preparation system, the wget command-line utility, the Conda package manager, and the Visual Studio Code integrated development environment (IDE).

4.1 Setting up the Environment

The project was developed in a Conda environment to ensure consistency and reproducibility. The environment was created using the command `conda create –name jpl python=3.8`. After activating the environment with `conda activate jpl`, several Python libraries were installed, including xarray, matplotlib, and ECCO's ecco_v4_py library. These libraries were installed using the `conda install` command.

4.2 Data Acquisition

The netCDF datasets from the ECCO project were acquired using the wget utility. The following command line arguments was used to download the datasets:

```
mkdir -p data wget –no-verbose –no-clobber –continue -i 5237392644-download.txt
-P data/
```

In this command, `5237392644-download.txt` is a file containing the URLs of the datasets to download, which were obtained from EarthData Search, a search and discovery application for NASA's Earth Observing System Data and Information System (EOSDIS) [5]. The datasets were downloaded into several directories named under the same parent directory granule_datasets.

4.3 Document Preparation and Data Parsing

A LaTeX document was prepared as a template for the final documentation. This document contained several placeholders sections and others that were populated with data parsed from the netCDF datasets. Python's JSON library was used to parse data and insert it into the LaTeX document. The document was organized into sections, with each section corresponding to a particular aspect of the datasets, such as global attributes, variable attributes, and field contents.

4.4 Data Visualization

Data visualizations were created using a combination of the xarray, matplotlib, and ecco_v4_py libraries. These visualizations provided a visual representation of the data in each dataset. The `data_var_plot` function was a key part of this process, which is one of the project's Python function that creates and saves a plot of a given data variable from a dataset. The function uses the matplotlib library to create the plot and saves the plot as a PNG file in a specified directory. The resulting PNG file is then included in the LaTeX document using the `\includegraphics` command.

4.5 Automated Processes

Finally, the document generation process was automated by a Python script. This script executed the data parsing and visualization processes, and then wrote the results into the LaTeX document. The document was compiled using the `pdflatex` command.

In conclusion, this project demonstrated how Python and LaTeX can be used together to automate the generation of documentation for complex datasets. The combination of data parsing, visualization, and automated document generation resulted in a comprehensive and understandable documentation of the ECCO project's netCDF datasets.

References

[1] NASA. *Oceanography - Science Mission Directorate*. Accessed: 2023-06-07. n.d. URL: https://science.nasa.gov/earth-science/focus-areas/oceanography.

[2] Carl Wunsch and Patrick Heimbach. "Estimated decadal changes in the North Atlantic meridional overturning circulation and heat flux 1993–2004". In: *Journal of Physical Oceanography* 36.11 (2007), pp. 2012–2024.

[3] NASA ECCO. *ECCO History*. 2023. URL: https://ecco-group.org/about.htm (visited on 06/07/2023).

[4] GHRSST Science Team. *The Recommended GHRSST Data Specification (GDS) 2.0, document revision 5*. Available from the GHRSST International Project Office, 2012. 2010. URL: https://doi.org/10.5281/zenodo.4700466.

[5] *Earthdata Search*. Accessed: 2023-06-07. 2023. URL: https://www.earthdata.nasa.gov/learn/earthdata-search.

5. Acknowledgments

I would like to express my gratitude towards my mentor Ian Fenty, he has given me a wonderful introduction into the field and workflow of science, specifically the relationship between oceanography and computational programming. His exceptional guidance and contributions has been instrumental to the success of this project.

I owe a great deal of thanks to Ou Wang, who guided me through the process of uploading the project to the ECCO-GROUP/ECCO-Dataset-Production repository.

I would like to extend my gratitude to the Education Office of JPL for this incredible opportunity, and to Pasadena City College. The college has not only made me aware of this opportunity but also facilitated my success by providing me with access to the necessary resources.

Appendix 1. Code

Total project is uploaded to ECCO-GROUP/ECCO-Dataset-Production repository: https://github.com/ECCO-GROUP/ECCO-Dataset-Production/tree/main/document_generator

Appendix 2. Document Pages

The ECCO Data Specification (ECCO)
version 4 release 4

3 Table of Contents

Contents

13.2 GDS 2.0 netCDF Global Attributes

Table 8-1 below summarizes the global attributes that are mandatory for every GDS 2.0 netCDF data file. More details on the CF-mandated attributes (as indicated in the Source column) are available at: `http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4/cf-conventions.html#attribute-appendix` and information on the ACDD recommendations is available at `http://www.unidata.ucar.edu/software/netcdf-java/formats/DataDiscoveryAttConvention.html`.

Table 13.1: Mandatory global attributes for GDS 2.0 netCDF data files

Global Attribute Name	Type	Description	Source
acknowledgement	string	A place to acknowledge various types of support for the project that produced this data.	ACDD
cdm_data_type	string	The data type, as derived from Unidata's Common Data Model Scientific Data types and understood by THREDDS. (This is a THREDDS "dataType", and is different from the CF NetCDF attribute 'featureType', which indicates a Discrete Sampling Geometry file in CF.)	ACDD
comment	string	Miscellaneous information about the data, not captured elsewhere. This attribute is defined in the CF Conventions.	CF, ACDD
conventions	string	A text string identifying the netCDF conventions followed (e.g., CF-1.4, ACDD 1-3).	
creator_email	string	The email address of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	ACDD
creator_name	string	The name of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	ACDD
creator_url	string	The URL of the of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.	ACDD
date_created	string	The date on which this version of the data was created.	ACDD
easternmost_longitude	float	Decimal degrees east, range -180 to +180. This is equivalent to ACDD geospatial_lon_max.	podaac
geospatial_lat_resolution	float	Latitude Resolution in units matching geospatial_lat_units.	ACDD
geospatial_lat_units	string	Units of the latitudinal resolution. Typically "degrees_north"	ACDD
geospatial_lon_resolution	float	Longitude Resolution in units matching geospatial_lon_resolution	ACDD
geospatial_lon_units	string	Units of the longitudinal resolution. Typically "degrees_east"	ACDD
history	string	The name of the institution principally responsible for originating this data. This attribute is recommended by the CF convention.	CF, ACDD

14.2 Native coordinates NetCDF GRID_GEOMETRY_ECCO

Table 14.1: Variables in the dataset GRID_GEOMETRY_ECCO

Dataset:	GRID_GEOMETRY_ECCO
Field:	XC
Field:	YC
Field:	XG
Field:	YG
Field:	CS
Field:	SN
Field:	rA
Field:	dxG
Field:	dyG
Field:	Depth
Field:	rAz
Field:	dxC
Field:	dyC
Field:	rAw
Field:	rAs
Field:	hFacC
Field:	hFacW
Field:	hFacS
Field:	maskC
Field:	maskW
Field:	maskS

14.2.5 Native coordinates Variable CS

Table 14.6: CDL description of GRID_GEOMETRY_ECCO's CS variable

Storage Type	Variable Name	Description	Unit
float32	CS	cosine of tracer grid cell orientation vs geographical north	1
CDL Description			
float32 CS(tile, j, i) CS: _FillValue = 9.96921e+36 CS: long_name = cosine of tracer grid cell orientation vs geographical north CS: units = 1 CS: coordinate = YC XC CS: coverage_content_type = modelResult CS: coordinates = YC XC			
Comments			
CS and SN are required to calculate the geographic (meridional, zonal) components of vectors on the curvilinear model grid. Note: for vector R with components R_x and R_y: R_{east} = CS R_x - SN R_y. R_{north} = SN R_x + CS R_y			

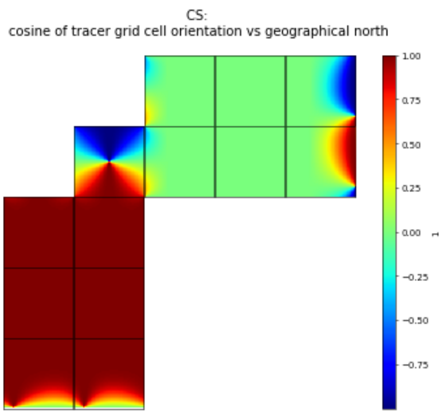


Figure 7:
Dataset: GRID_GEOMETRY_ECCO
Variable: CS

15.21.2 Native Variable SIvice

Table 15.110: CDL description of SEA_ICE_VELOCITY's SIvice variable

Storage Type	Variable Name	Description	Unit
float32	SIvice	Sea-ice velocity in the model +y direction	m s-1
CDL Description			
float32 SIvice(time, tile, j_g, i) SIvice: _FillValue = 9.96921e+36 SIvice: long_name = Sea: ice velocity in the model +y direction SIvice: units = m s: 1 SIvice: mate = SIuice SIvice: coverage_content_type = modelResult SIvice: standard_name = sea_ice_y_velocity SIvice: coordinates = time SIvice: valid_min = : 0.4000000059604645 SIvice: valid_max = 0.4000000059604645			
Comments			
Horizontal sea-ice velocity in the +y direction at the 'v' face of the tracer cell on the native model grid. Note: in the Arakawa-C grid, horizontal velocities are staggered relative to the tracer cells with indexing such that +SIvice(i,j_g) corresponds to +y fluxes through the 'v' face of the tracer cell at (i,j,k=0). Also, the model +y direction does not necessarily correspond to the geographical north-south direction because the x and y axes of the model's curvilinear lat-lon-cap (llc) grid have arbitrary orientations which vary within and across tiles.			

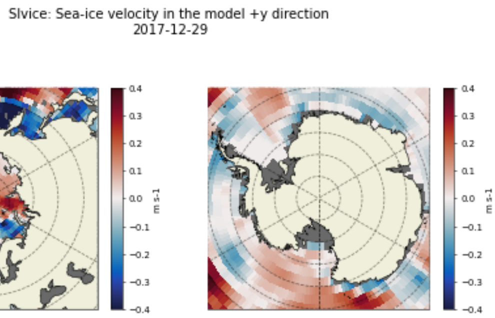


Figure 113:
Dataset: SEA_ICE_VELOCITY
Variable: SIvice

15.2.6 Native Variable EXFwspee

Table 15.7: CDL description of ATM_SURFACE_TEMP_HUM_WIND_PRES's EXFwspee variable

Storage Type	Variable Name	Description	Unit
float32	EXFwspee	Wind speed	m s-1
CDL Description			
float32 EXFwspee(time, tile, j, i) EXFwspee: _FillValue = 9.96921e+36 EXFwspee: long_name = Wind speed EXFwspee: units = m s: 1 EXFwspee: coverage_content_type = modelResult EXFwspee: standard_name = wind_speed EXFwspee: coordinates = time XC YC EXFwspee: valid_min = 0.27271032333374023 EXFwspee: valid_max = 45.87086486816406			
Comments			
10-m wind speed magnitude (>= 0) over open water. Only used for the calculation of air-sea fluxes using bulk formulae. Note: not adjusted by the ocean state estimation and not necesarily consistent with EXFuwind and EXFvwind because EXFuwind and EXFvwind are calculated from EXFtaux and EXFtauy using bulk formulae. EXFwspee != sqrt(EXFuwind**2 + EXFvwind**2.			

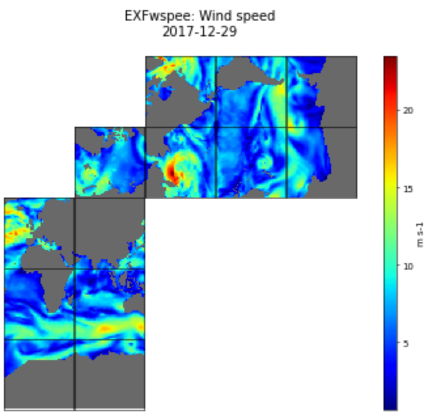


Figure 29:
Dataset: ATM_SURFACE_TEMP_HUM_WIND_PRES
Variable: EXFwspee

17.4.5 Latlon Variable EXFqnet

Table 17.25: CDL description of OCEAN_AND_ICE_SURFACE_HEAT_FLUX's EXFqnet variable

Storage Type	Variable Name	Description	Unit
float32	EXFqnet	Open ocean net air-sea heat flux	W m-2
CDL Description			
float32 EXFqnet(time, latitude, longitude) EXFqnet: _FillValue = 9.96921e+36 EXFqnet: coverage_content_type = modelResult EXFqnet: direction = >0 increases potential temperature (THETA) EXFqnet: long_name = Open ocean net air: sea heat flux EXFqnet: units = W m: 2 EXFqnet: coordinates = time EXFqnet: valid_min = : 687.8736572265625 EXFqnet: valid_max = 3408.977783203125			
Comments			
Net air-sea heat flux (turbulent and radiative) per unit area of open water (not covered by sea-ice). Note: net upward heat flux over open water, calculated as EXFlwnet+EXFswnet-EXFlh-EXFhs.			

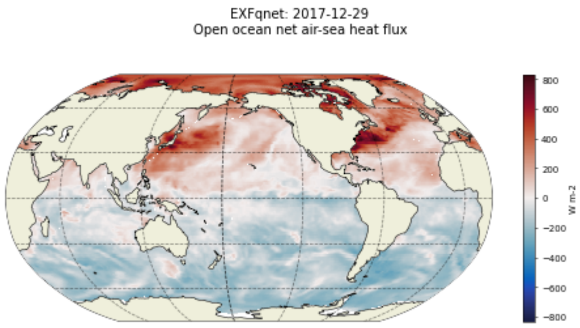


Figure 141:
Dataset: OCEAN_AND_ICE_SURFACE_HEAT_FLUX
Variable: EXFqnet

17.13.1 Latlon Variable SIeice

Table 17.64: CDL description of SEA_ICE_VELOCITY's SIeice variable

Storage Type	Variable Name	Description	Unit
float32	SIeice	Zonal (east-west) sea-ice velocity	m s-1
CDL Description			
float32 SIeice(time, latitude, longitude) SIeice: _FillValue = 9.96921e+36 SIeice: coverage_content_type = modelResult SIeice: long_name = Zonal (east: west) sea: ice velocity SIeice: standard_name = eastward_sea_ice_velocity SIeice: units = m s: 1 SIeice: coordinates = time SIeice: valid_min = : 0.5656854510307312 SIeice: valid_max = 0.5656854510307312			
Comments			
Zonal (east-west) componet of sea-ice velocity. Note: mask with SIarea to remove nonzero values where ice is absent. SIeice is calculated by interpolating the model's x and y components of sea-ice velocity (SIuice and SIvice) to tracer cell centers and then finding the zonal component of the interpolated vectors. It is NOT recommended to use SIuice and SIvice for sea-ice volume budget calculations because interpolating SIuice and SIvice from the model grid to the lat-lon grid introduces errors. Perform sea-ice mass budget calculations with ADVxHEFF, ADVyHEFF, DFxHEFF, and DFyHEFF on the native model grid.			

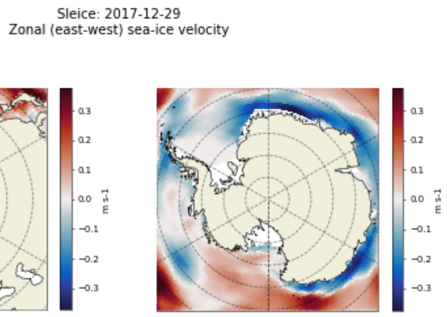


Figure 171:
Dataset: SEA_ICE_VELOCITY
Variable: SIeice