The Journal of Open Source Software

xCDAT: A Python Package for Simple and Robust Analysis of Climate Data

Tom Vo [1], **Stephen Po-Chedley** [1], **Jason Boutte** [1], **Jiwoo Lee** [1], and **Chengzhu Zhang** [1]

1 Lawrence Livermore National Lab, Livermore, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @brian-rose
- @mgrover1

Submitted: 23 January 2024
Published: unpublished

Summary

xCDAT (Xarray Climate Data Analysis Tools) is an open-source Python package that extends Xarray (Hoyer & Hamman, 2017) for climate data analysis on structured grids. xCDAT streamlines analysis of climate data by exposing common climate analysis operations through a set of straightforward APIs. Some of xCDAT's key features include spatial averaging, temporal averaging, and regridding. These features are inspired by the Community Data Analysis Tools (CDAT) library (Dean N. Williams et al., 2009) (D. N. Williams, 2014) (Doutriaux et al., 2019) and leverage powerful packages in the Xarray ecosystem including xESMF (Zhuang et al., 2023), xgcm (Abernathey et al., 2022), and CF xarray (Cherian et al., 2023). To ensure general compatibility across various climate models, xCDAT operates on datasets that are compliant with the Climate and Forecast (CF) metadata conventions (Hassell et al., 2017).

Statement of Need

Analysis of climate data frequently requires a number of core operations, including reading and writing of netCDF files, horizontal and vertical regridding, and spatial and temporal averaging. While many individual software packages address these needs in a variety of computational languages, CDAT stands out because it provides these essential operations via open-source, interoperable Python packages. Since CDAT's inception, the volume of climate data has grown substantially as a result of both a larger pool of data products and increasing spatiotemporal resolution of model and observational data. Larger data stores are important for advancing geoscientific understanding, but also require increasingly performant software and hardware. These factors have sparked a need for new analysis software that offers core geospatial analysis functionalities capable of efficiently handling large datasets while using modern technologies and standardized software engineering principles.

xCDAT addresses this need by combining the power of Xarray with meticulously developed geospatial analysis features inspired by CDAT. Xarray is the foundation of xCDAT because of its widespread adoption, technological maturity, and ability to handle large datasets with parallel computing via Dask. Xarray is also interoperable with the scientific Python ecosystem (e.g., NumPy, pandas, Matplotlib), which greatly benefits users who need to use additional analysis tools. Since Xarray is designed as a general-purpose library, xCDAT fills in domain specific gaps by providing features to serve the climate science community *(refer to Key Features)*.

Performance is one fundamental driver in how xCDAT is designed, especially with large datasets. xCDAT conveniently inherits Xarray's support for parallel computing with Dask (Team, 2016). Parallel computing with Dask enables users to take advantage of compute resources through multithreading or multiprocessing. To use Dask's default multithreading scheduler, users only need to open and chunk datasets in Xarray before calling xCDAT APIs. xCDAT's seamless support for parallel computing enables users to run large-scale computations with minimal effort. If users require more resources, they can also configure and use a local Dask cluster to meet resource-intensive computational needs. Figure 1 shows xCDAT's significant performance advantage over CDAT for global spatial averaging on datasets of varying sizes.

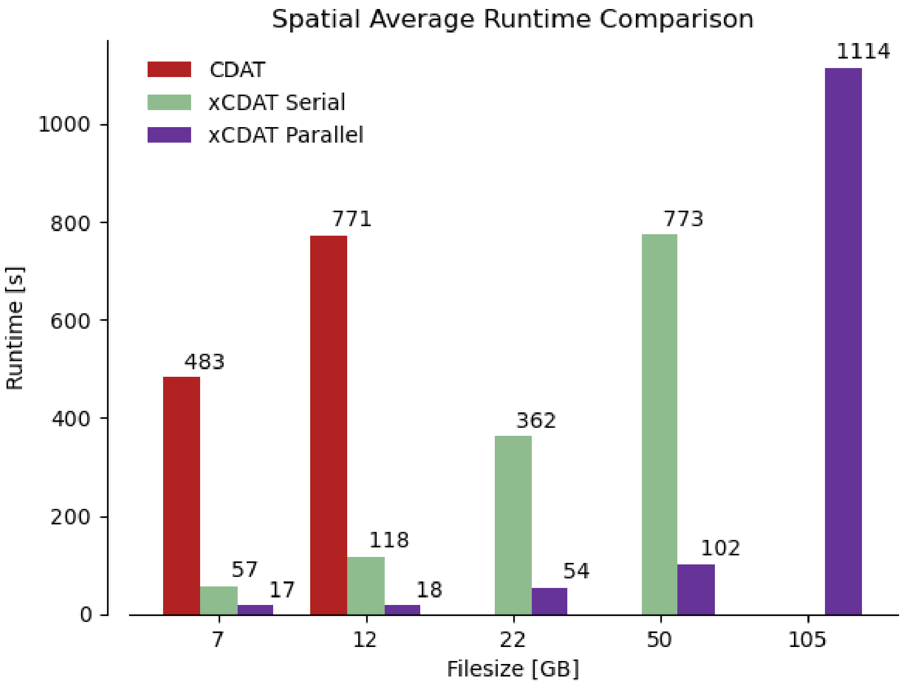
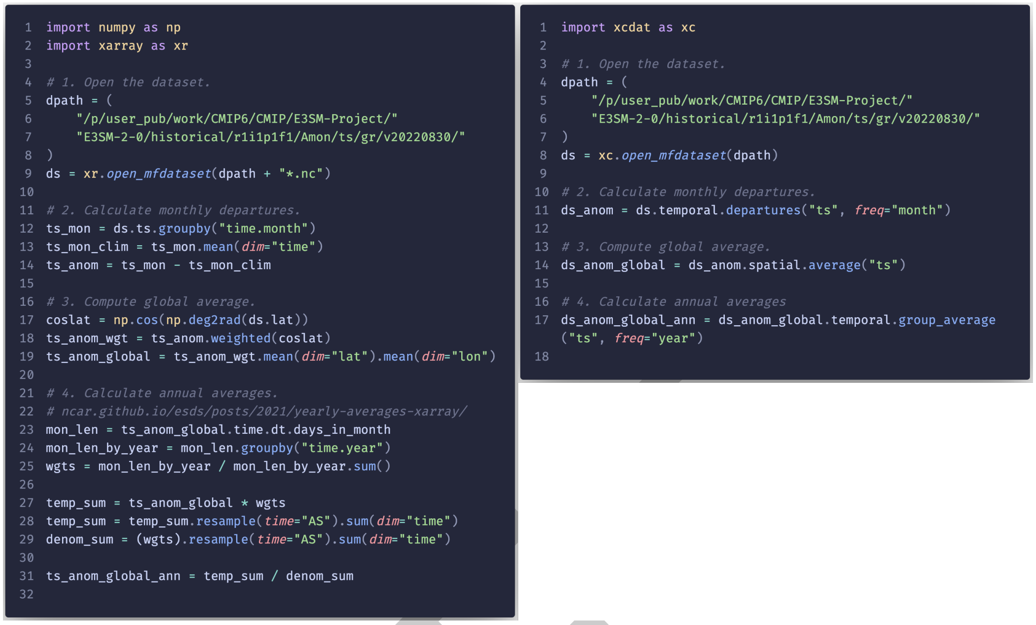


Figure 1: A performance benchmark for global spatial averaging computations using CDAT (serial only) and xCDAT (serial and parallel with Dask distributed scheduler). xCDAT outperforms CDAT by a wide margin for the 7 GB and 12 GB datasets. Runtimes could not be captured for CDAT with datasets >= 22 GB and xCDAT serial for the 105 GB dataset due to memory allocation errors. The performance benchmark setup and scripts are available in the `xcdat-validation` repo. *Disclaimer: Performance will vary depending on hardware, dataset shapes/sizes, and how Dask and chunking schemes are configured. There are also some cases where selecting a regional averaging domain (e.g., Niño 3.4) can lead to CDAT outperforming xCDAT.*

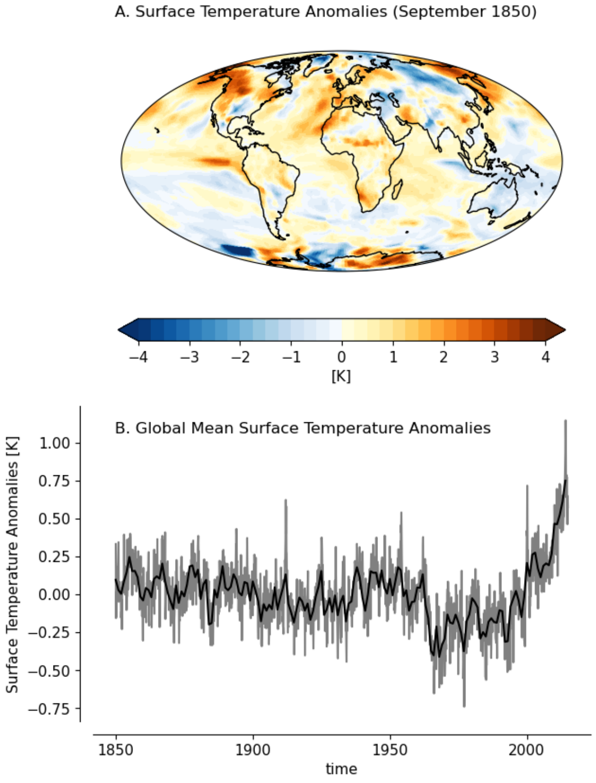
xCDAT's intentional design emphasizes software sustainability and reproducible science. It aims to make analysis code reusable, readable, and less-error prone by abstracting common Xarray boilerplate logic into simple and configurable APIs. xCDAT extends Xarray by using accessor classes that operate directly on Xarray Dataset objects. xCDAT is rigorously tested using real-world datasets and maintains 100% unit test coverage (at the time this paper was written). To demonstrate the value in xCDAT's API design, Figure 2 compares code to calculate annual averages for global climatological anomalies using Xarray against xCDAT. xCDAT requires fewer lines of code and supports further user options (e.g., regional or seasonal averages, not shown). Figure 2 shows the plots for the results produced by xCDAT.



```python
import numpy as np
import xarray as xr

# 1. Open the dataset.
dpath = (
    "/p/user_pub/work/CMIP6/CMIP/E3SM-Project/"
    "E3SM-2-0/historical/r1i1p1f1/Amon/ts/gr/v20220830/"
)
ds = xr.open_mfdataset(dpath + "*.nc")

# 2. Calculate monthly departures.
ts_mon = ds.ts.groupby("time.month")
ts_mon_clim = ts_mon.mean(dim="time")
ts_anom = ts_mon - ts_mon_clim

# 3. Compute global average.
coslat = np.cos(np.deg2rad(ds.lat))
ts_anom_wgt = ts_anom.weighted(coslat)
ts_anom_global = ts_anom_wgt.mean(dim="lat").mean(dim="lon")

# 4. Calculate annual averages.
# ncar.github.io/esds/posts/2021/yearly-averages-xarray/
mon_len = ts_anom_global.time.dt.days_in_month
mon_len_by_year = mon_len.groupby("time.year")
wgts = mon_len_by_year / mon_len_by_year.sum()

temp_sum = ts_anom_global * wgts
temp_sum = temp_sum.resample(time="AS").sum(dim="time")
denom_sum = (wgts).resample(time="AS").sum(dim="time")

ts_anom_global_ann = temp_sum / denom_sum

```

```python
import xcdat as xc

# 1. Open the dataset.
dpath = (
    "/p/user_pub/work/CMIP6/CMIP/E3SM-Project/"
    "E3SM-2-0/historical/r1i1p1f1/Amon/ts/gr/v20220830/"
)
ds = xc.open_mfdataset(dpath)

# 2. Calculate monthly departures.
ds_anom = ds.temporal.departures("ts", freq="month")

# 3. Compute global average.
ds_anom_global = ds_anom.spatial.average("ts")

# 4. Calculate annual averages
ds_anom_global_ann = ds_anom_global.temporal.group_average
("ts", freq="year")

```

Figure 2: A comparison of the code to calculate annual averages for global climatological anomalies in A) Xarray and B) xCDAT. xCDAT abstracts most of the Xarray boilerplate logic for calculating weights and grouping data by specific time frequencies, leading to code that is more readable, maintainable, and flexible. The results from both sets of code are within machine precision.



A. Surface Temperature Anomalies (September 1850)

B. Global Mean Surface Temperature Anomalies

Figure 3: A) Monthly surface skin temperature anomalies for September 1850. B) Monthly (gray) and annual (black) global mean surface skin temperature anomaly values. Temperature data is from an E3SMv2 climate model (Golaz et al., 2022) simulation over the historical period (1850 – 2014).

xCDAT's mission is to provide a maintainable and extensible package that serves the needs of the climate community in the long-term. xCDAT is a community-driven project and the development team encourages all who are interested to get involved through the GitHub repository.

Key Features

Extension of `xarray.open_dataset()` and `xarray.open_mfdataset()` with post-processing options

xCDAT extends `xarray.open_dataset()` and `xarray.open_mfdataset()` with additional post-processing operations to support climate data analysis. These APIs can generate missing coordinate bounds for the X, Y, T, and/or Z axes and lazily decode time coordinates represented by `cftime` (more info). Other functionality includes re-centering time coordinates between time bounds and converting the longitudinal axis orientation between [0, 360) and [-180, 180).

Robust interpretation of CF metadata

xCDAT uses CF xarray to interpret CF metadata present in datasets, enabling xCDAT to operate generally across model and observational datasets that are CF-compliant. This feature enables xCDAT to generate missing coordinate bounds, recognize the coordinates and coordinate bounds to use for computational operations, and lazily decode time coordinates based on the CF calendar attribute.

Temporal averaging

xCDAT's temporal averaging API utilizes Xarray and Pandas. It includes features for calculating time series averages (single-snapshot), grouped time series averages (e.g., seasonal or annual averages), climatologies, and departures. Averages can be weighted (default) or unweighted. There are optional configurations for seasonal grouping including how to group the month of December (DJF or JFD) and defining custom seasons to group by.

Geospatial weighted averaging

xCDAT's geospatial weighted averaging supports rectilinear grids with an option to compute averages over a regional domain (e.g., tropical region, Niño 3.4 region).

Horizontal structured regridding

xCDAT makes use of xESMF for horizontal regridding capabilities. It simplifies and extends the xESMF horizontal regridding API by generating missing bounds and ensuring bounds and metadata are preserved in the output dataset. xCDAT also offers a Python implementation of regrid2 for handling cartesian latitude by longitude grids.

Vertical structured regridding

xCDAT makes use of xgcm for vertical regridding capabilities. It simplifies and extends the xgcm vertical regridding API by automatically attempting to determine the grid point position relative to the bounds, transposing the output data to match the dimensional order of the input data, and ensuring bounds and metadata are preserved in the output dataset.

Documentation & Case Studies

The xCDAT documentation includes the public API list and a Jupyter Notebook Gallery that demonstrates real-world applications of xCDAT:

- A Gentle Introduction to xCDAT (Xarray Climate Data Analysis Tools)
- General Dataset Utilities
- Calculate Geospatial Weighted Averages from Monthly Time Series
- Calculate Time Averages from Time Series Data
- Calculating Climatology and Departures from Time Series Data
- Horizontal Regridding
- Vertical Regridding

Distribution

xCDAT is available for Linux, MacOS, and Windows via the conda-forge channel on Anaconda. The GitHub Repository is where we host all development activity. xCDAT is released under the Apache 2-0 license.

Projects using xCDAT

xCDAT is actively being integrated as a core component of the Program for Climate Model Diagnosis and Intercomparison (PCMDI) Metrics Package (Jiwoo Lee et al., 2023) (J. Lee et al., 2023) and the Energy Exascale Earth System Model (E3SM) Diagnostics Package (C. Zhang et al., 2022) (J. C. Zhang et al., 2023). xCDAT is also included in the E3SM Unified Anaconda Environment (Asay-Davis, 2023) deployed on numerous U.S Department of Energy supercomputers to run E3SM software tools. Members of the development team are also active users of xCDAT and apply the software to advance their own climate research (Po-Chedley et al., 2022).

Acknowledgements

xCDAT is jointly developed by scientists and developers at Lawrence Livermore National Laboratory (LLNL) from the Energy Exascale Earth System Model (E3SM) Project and Program for Climate Model Diagnosis and Intercomparison (PCMDI). The work is performed for the E3SM project, which is sponsored by Earth System Model Development (ESMD) program, and the Simplifying ESM Analysis Through Standards (SEATS) project, which is sponsored by the Regional and Global Model Analysis (RGMA) program. ESMD and RGMA are programs for the Earth and Environmental Systems Sciences Division (EESSD) in the Office of Biological and Environmental Research (BER) within the Department of Energy's Office of Science. This work is performed under the auspices of the U.S. Department of Energy by LLNL under Contract No. DE-AC52-07NA27344.

Thank you to all of the xCDAT contributors and users including Rob Jacob, Ana Ordonez, Mark Zelinka, Christopher Terai, Min-Seop Ahn, Celine Bonfils, Jean-Yves Peterschmitt, Olivier Marti, Andrew Manaster, and Andrew Friedman. We also give a special thanks to Karl Taylor, Peter Gleckler, Paul Durack, and Chris Golaz who all have provided valuable knowledge and guidance throughout the course of this project.

References

Abernathey, R. P., Busecke, J. J. M., Smith, T. A., Deauna, J. D., Banihirwe, A., Nicholas, T., Fernandes, F., James, B., Dussin, R., Cherian, D. A., Caneill, R., Sinha, A., Uieda, L., Rath, W., Balwada, D., Constantinou, N. C., Ponte, A., Zhou, Y., Uchida, T., & Thielen, J. (2022). *Xgcm* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.7348619

Asay-Davis, X. (2023). *E3SM-unified: A metapackage for a unified anaconda environment for analyzing results from the energy exascale earth system model (E3SM)* (Version v1.9.1). GitHub. https://github.com/E3SM-Project/e3sm-unified

Cherian, D., Almansi, M., Bourgault, P., Thyng, K., Thielen, J., Magin, J., Aoun, A., Buntemeyer, L., Caneill, R., Davis, L., Fernandes, F., Hauser, M., Heerdegen, A., Kent, J., Mankoff, K., Müller, S., Schupfner, M., Vo, T., & Haëck, C. (2023). *Cf_xarray* (Version v0.8.5). Zenodo. https://doi.org/10.5281/zenodo.10038784

Doutriaux, C., Nadeau, D., Wittenburg, S., Lipsa, D., Muryanto, L., Chaudhary, A., & Williams, D. N. (2019). *CDAT/cdat: CDAT 8.1* (Version v8.1). Zenodo. https://doi.org/10.5281/zenodo.2586088

Golaz, J.-C., Van Roekel, L. P., Zheng, X., Roberts, A. F., Wolfe, J. D., Lin, W., Bradley, A. M., Tang, Q., Maltrud, M. E., Forsyth, R. M., Zhang, C., Zhou, T., Zhang, K., Zender, C. S., Wu, M., Wang, H., Turner, A. K., Singh, B., Richter, J. H., … Bader, D. C. (2022). The DOE E3SM model version 2: Overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, *14*(12), e2022MS003156. https://doi.org/https://doi.org/10.1029/2022MS003156

Hassell, D., Gregory, J., Blower, J., Lawrence, B. N., & Taylor, K. E. (2017). A data model of the climate and forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1). *Geoscientific Model Development*, *10*(12), 4619–4646. https://doi.org/10.5194/gmd-10-4619-2017

Hoyer, S., & Hamman, J. J. (2017). Xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, *5*, 10. https://doi.org/10.5334/jors.148

Lee, J., Gleckler, P. J., Ahn, M.-S., Ordonez, A., Ullrich, P. A., Sperber, K. R., Taylor, K. E., Planton, Y. Y., Guilyardi, E., Durack, P., Bonfils, C., Zelinka, M. D., Chao, L.-W., Dong, B., Doutriaux, C., Zhang, C., Vo, T., Boutte, J., Wehner, M. F., … Krasting, J. (2023). Objective evaluation of earth system models: PCMDI metrics package (PMP) version 3. *EGUsphere*, *2023*, 1–48. https://doi.org/10.5194/egusphere-2023-2720

Lee, Jiwoo, Gleckler, P., Ordonez, A., Ahn, M.-S., Ullrich, P., Vo, T., Boutte, J., Doutriaux, C., Durack, P., Shaheen, Z., Muryanto, L., Painter, J., & Krasting, J. (2023). *PCMDI/pcmdi_metrics: PMP version 3.1.2* (Version v3.1.2). Zenodo. https://doi.org/10.5281/zenodo.10236521

Po-Chedley, S., Fasullo, J. T., Siler, N., Labe, Z. M., Barnes, E. A., Bonfils, C. J. W., & Santer, B. D. (2022). Internal variability and forcing influence model–satellite differences in the rate of tropical tropospheric warming. *Proceedings of the National Academy of Sciences*, *119*(47), e2209431119. https://doi.org/10.1073/pnas.2209431119

Team, D. D. (2016). *Dask: Library for dynamic task scheduling*. https://dask.org

Williams, D. N. (2014). Visualization and analysis tools for ultrascale climate data. *Advanced Earth and Space Sciences*, *95*(42), 377–378. https://doi.org/10.1002/2014EO420002

Williams, Dean N., Doutriaux, C. M., Drach, R. S., & Mccoy, R. B. (2009). *The flexible climate data analysis tools (CDAT) for multi-model climate simulation data*. 254–261. https://doi.org/10.1109/ICDMW.2009.64

Zhang, C., Golaz, J.-C., Forsyth, R., Vo, T., Xie, S., Shaheen, Z., Potter, G. L., Asay-Davis, X. S., Zender, C. S., Lin, W., Chen, C.-C., Terai, C. R., Mahajan, S., Zhou, T., Balaguru, K., Tang, Q., Tao, C., Zhang, Y., Emmenegger, T., … Ullrich, P. A. (2022). The E3SM diagnostics package (E3SM diags v2.7): A python-based diagnostics package for earth system model evaluation. *Geoscientific Model Development*, *15*(24), 9031–9056. https://doi.org/10.5194/gmd-15-9031-2022

Zhang, J. C., Shaheen, Z., Vo, T., Forsyth, R., Golaz, Asay-Davis, X., Mahfouz, N., Bradley, A. M., & Doutriaux, C. (2023). *E3SM-project/e3sm_diags: v2.9.0* (Version v2.9.0). Zenodo. https://doi.org/10.5281/zenodo.8339034

Zhuang, J., Dussin, R., Huard, D., Bourgault, P., Banihirwe, A., Raynaud, S., Malevich, B., Schupfner, M., Filipe, Levang, S., Gauthier, C., Jüling, A., Almansi, M., Richardscottoz, Rondeaug, Rasp, S., Smith, T. J., Stachelek, J., Plough, M., … Li, X. (2023). *Pangeo-data/xESMF: v0.8.2* (Version v0.8.2). Zenodo. https://doi.org/10.5281/zenodo.8356796