

ROMS-Tools: Reproducible Preprocessing and Analysis for Regional Ocean Modeling with ROMS

Nora Loose¹, Tom Nicholas², Scott Eilerman¹, Christopher McBride¹, Sam Maticka¹, Dafydd Stephenson¹, Scott Bachman¹, Pierre Damien³, Ulla Heede¹, Alicia Karspeck¹, Matthew C. Long¹, M. Jeroen Molemaker³, and Abigale Wyatt¹

¹ [C]Worthy LLC, Boulder, CO, United States ² Earthmover PBC ³ University of California, Los Angeles, CA, United States

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a

Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

Summary

The ocean regulates Earth's climate and sustains marine ecosystems by circulating and storing heat, carbon, oxygen, and nutrients, while exchanging heat and gases with the atmosphere. Scientists study these processes using ocean models, which simulate the ocean on a grid. **Regional ocean models** focus computational resources on a limited geographical area with fine grid spacing, and can resolve fine-scale phenomena such as mesoscale and submesoscale features, tidal dynamics, coastal currents, upwelling, and detailed biogeochemical (BGC) processes. A widely used regional ocean model is the **Regional Ocean Modeling System (ROMS)** (Shchepetkin & McWilliams, 2005). ROMS has been coupled to the Marine Biogeochemistry Library (MARBL) (Long et al., 2021; Molemaker & contributors, 2025a) to link physical and BGC processes. ROMS-MARBL supports research on environmental management, fisheries, regional climate impacts, and ocean-based carbon dioxide removal (CDR) strategies.

ROMS-Tools is a Python package that streamlines the **preparation and analysis of ROMS-MARBL simulations** by enabling users to generate regional grids, prepare model inputs, and analyze outputs efficiently. By providing a modern, user-friendly interface, ROMS-Tools lowers technical barriers, improves reproducibility, and allows scientists to focus on research rather than data preparation. The package is installable via Conda or PyPI and can be run interactively in Jupyter notebooks.

Statement of Need

Regional ocean models are essential tools for research in marine ecosystems, climate dynamics, and ocean-based CDR. However, configuring a regional ocean model like ROMS-MARBL is technically demanding. Model setup requires initialization and time-dependent forcing from oceanic and atmospheric datasets, drawn from multiple external sources in diverse formats. These global source datasets can span petabytes and must be subsetting, processed, and mapped onto the target model grid, producing 10–100 terabytes of input data for large regional domains. Generating these input files is time-consuming, error-prone, and difficult to reproduce. These challenges create a bottleneck for both new and experienced users, slow down science, and limit collaboration across groups.

Existing tools within the ocean modeling ecosystem do not fully address these challenges for ROMS or ROMS-MARBL. Legacy MATLAB-based scripts and packages such as pyroms (Hedstrom & contributors, 2023) provide critical functionality but rely on low-level, manually coordinated steps that limit reproducibility, maintainability, and accessibility. In contrast,

frameworks developed for other ocean models (e.g., MOM6) cannot be directly applied to ROMS due to fundamental differences in grid geometry, vertical coordinates, and model input data requirements. As a result, ROMS-MARBL users lack a modern, integrated framework for reproducible model setup and analysis that is specifically designed for ROMS and ROMS-MARBL.

ROMS-Tools was developed to fill this gap. It is an open-source Python framework designed for researchers and practitioners who run ROMS or ROMS-MARBL regional ocean simulations, including users in physical oceanography, marine biogeochemistry, and ocean-based CDR applications. Current capabilities are fully compatible with UCLA-ROMS (Molemaker & contributors, 2025a, 2025b), with potential support for other ROMS implementations, such as Rutgers ROMS (Arango & contributors, 2024), in the future. The package provides high-level APIs that automate and standardize preprocessing workflows and manage complex model configuration state via YAML, supporting reproducible simulation setup. ROMS-Tools supports large input and output datasets via parallel computation with dask (Dask Development Team, 2016), making workflows scalable from laptops to high-performance computing clusters. By lowering technical barriers and improving transparency and reproducibility, ROMS-Tools enables more efficient model development, facilitates scientific collaboration, and supports applications such as verification of marine carbon removal strategies and carbon credit assessment.

State of the Field

Historically, setting up a regional ocean model required a patchwork of custom scripts and lab-specific workflows, resulting in error-prone and difficult-to-reproduce processes. Within the ROMS community, tools like pyroms (Hedstrom & contributors, 2023) addressed some of these issues by providing low-level Python utilities for preprocessing ROMS model inputs. However, pyroms has several limitations: installation is cumbersome due to Python/Fortran dependencies, the API is inconsistent, and documentation and tests are missing. The package does not support modern tools such as xarray (Hoyer & Hamman, 2017), nor reproducible workflows. Active development has ceased, and maintenance (including compatibility with newer Python versions) is no longer provided, making it very difficult to add new features, such as support for BGC tracers and CDR applications.

Tools from other modeling communities cannot be directly applied to ROMS because each model has distinct structural requirements and input conventions. For example, the regional-mom6 package (Barnes et al., 2024), developed for regional configurations of the Modular Ocean Model v6 (MOM6) (Adcroft et al., 2019), cannot generate ROMS inputs. ROMS uses a terrain-following vertical coordinate system that requires specialized vertical regridding, whereas MOM6 accepts inputs on arbitrary depth levels and does not require vertical regridding at all. While ROMS and MOM6 differ in fundamental ways, regional-mom6 represents the closest comparable tool to ROMS-Tools in the wider modeling ecosystem. Notably, the development cycles of regional-mom6 and ROMS-Tools overlapped (regional-mom6: 2023–2024; ROMS-Tools: 2024–2025, based on public GitHub commits). Had the developers been aware of each other, a shared framework could potentially have been created, with model-specific adaptations layered on top. Adapting one framework to the other now would require extensive architectural changes.

Legacy MATLAB preprocessing scripts developed at UCLA (Molemaker, 2024) encapsulate decades of expertise in configuring regional ocean models, but require users to edit source code directly, making workflows error-prone, difficult to reproduce, and challenging to extend to new datasets or applications. ROMS-Tools provides a modern, open-source Python implementation of these scripts, retaining core algorithms while offering high-level APIs, automated intermediate steps, and explicit workflow state management via YAML. This object-oriented design improves reproducibility, reduces user errors, and supports extensibility, while leveraging modern Python tools such as xarray and dask. In some cases, ROMS-Tools diverges from the original MATLAB

91 implementation to incorporate improved methods or better integrate with the Python ecosystem,
92 creating a maintainable, scalable workflow for ROMS-MARBL simulations.

93 Overview of ROMS-Tools Functionality

94 ROMS-Tools provides a comprehensive workflow for generating, processing, and analyzing
95 ROMS-MARBL model inputs and outputs, as detailed below.

96 Input Data and Preprocessing

97 ROMS-Tools generates the following input files for ROMS-MARBL:

- 98 1. **Model Grid:** Customizable, curvilinear, and orthogonal grid designed to maintain a nearly
99 uniform horizontal resolution across the domain. The grid is rotatable to align with
100 coastlines and features a terrain-following vertical coordinate.
- 101 2. **Bathymetry:** Derived from **SRTM15** (Tozer et al., 2019).
- 102 3. **Land Mask:** Inferred from coastlines provided by **Natural Earth** or the Global Self-
103 consistent, Hierarchical, High-resolution Geography (**GSHHG**) Database (Wessel &
104 Smith, 1996).
- 105 4. **Physical Ocean Conditions:** Initial and open boundary conditions for sea surface height,
106 temperature, salinity, and velocities derived from the 1/12° Global Ocean Physics
107 Reanalysis (**GLORYS**) (Lellouche et al., 2021).
- 108 5. **BGC Ocean Conditions:** Initial and open boundary conditions for dissolved inorganic
109 carbon, alkalinity, and other biogeochemical tracers from Community Earth System
110 Model (**CESM**) output (Yeager et al., 2022) or hybrid observational-model sources
111 (Garcia et al., 2019; Huang et al., 2022; Lauvset et al., 2016; Yang et al., 2020; Yeager
112 et al., 2022).
- 113 6. **Meteorological forcing:** Wind, radiation, precipitation, and air temperature/humidity
114 processed from the global 1/4° ECMWF Reanalysis v5 (**ERA5**) (Hersbach et al., 2020)
115 with optional corrections for radiation bias and coastal wind.
- 116 7. **BGC surface forcing:** Partial pressure of carbon dioxide, as well as iron, dust, and
117 nitrogen deposition from **CESM** output (Yeager et al., 2022) or hybrid observational-
118 model sources (Hamilton et al., 2022; Kok et al., 2021; Landschützer et al., 2016; Yeager
119 et al., 2022).
- 120 8. **Tidal Forcing:** Tidal potential, elevation, and velocities derived from **TPXO** (Egbert &
121 Erofeeva, 2002) including self-attraction and loading (SAL) corrections.
- 122 9. **River Forcing:** Freshwater runoff derived from Dai & Trenberth (Dai & Trenberth, 2002)
123 or user-provided custom files.
- 124 10. **CDR Forcing:** User-defined interventions that inject BGC tracers at point sources
125 or as larger-scale Gaussian perturbations to simulate CDR interventions. The CDR
126 forcing is prescribed as volume and tracer fluxes (e.g., alkalinity for ocean alkalinity
127 enhancement, iron for iron fertilization, or other BGC constituents). Users can control
128 the magnitude, spatial footprint, and temporal evolution, allowing flexible representation
129 of CDR interventions.

130 Some source datasets are accessed automatically by ROMS-Tools, including Natural Earth, Dai
131 & Trenberth runoff, and ERA5 meteorology, while users must manually download SRTM15,
132 GSHHG, GLORYS, the BGC datasets, and TPXO tidal files. Although these are the datasets
133 currently supported, the modular design of ROMS-Tools makes it straightforward to add new
134 source datasets in the future.

135 To generate the model inputs, ROMS-Tools automates several intermediate processing steps,
136 including:

- 137 ■ **Bathymetry processing:** The bathymetry is smoothed in two stages, first across the
138 entire model domain and then locally in areas with steep slopes, to ensure local steepness

ratios do not exceed a prescribed threshold in order to reduce pressure-gradient errors. A minimum depth is enforced to prevent water levels from becoming negative during large tidal excursions.

- **Mask definition:** The land-sea mask is generated by comparing the ROMS grid's horizontal coordinates with a coastline dataset using the regionmask package (Hauser et al., 2024). Enclosed basins are subsequently filled with land.
- **Land value handling:** Land values are filled via an algebraic multigrid method using pyamg (Bell et al., 2023) prior to horizontal regridding. This extends ocean values into land areas to reconcile discrepancies between source data and ROMS land masks, ensuring that no NaNs or land-originating values contaminate ocean grid cells.
- **Regridding:** Ocean and atmospheric fields are horizontally and vertically regridded from standard latitude-longitude-depth grids to the model's curvilinear grid with a terrain-following vertical coordinate using xarray (Hoyer & Hamman, 2017) and xgcm (Busecke & contributors, 2025). Velocities are rotated to align with the curvilinear ROMS grid.
- **Longitude conventions:** ROMS-Tools handles differences in longitude conventions, converting between $[-180^\circ, 180^\circ]$ and $[0^\circ, 360^\circ]$ as needed.
- **River locations:** Rivers that fall within the model domain are automatically identified and relocated to the nearest coastal grid cell. Rivers that need to be shifted manually or span multiple cells can be configured by the user.
- **Data streaming:** ERA5 atmospheric data can be accessed directly from the cloud, removing the need for users to pre-download large datasets locally. Similar streaming capabilities may be implemented for other datasets in the future.

Users can quickly design and visualize regional grids and inspect all input fields with built-in plotting utilities. An example of surface initial conditions generated for a California Current System simulation at 5 km horizontal grid spacing is shown in Figure 1.

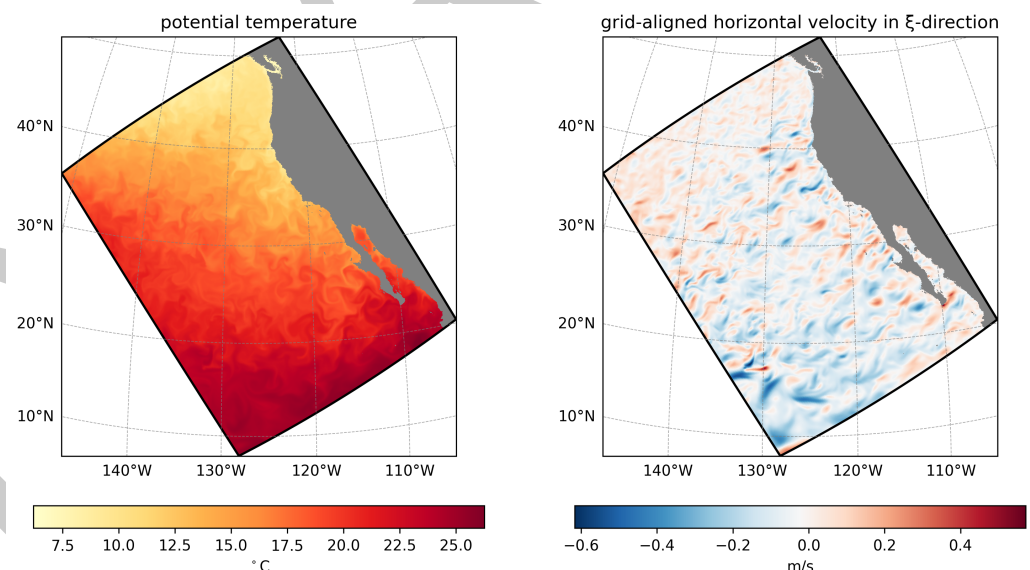


Figure 1: Surface initial conditions for the California Current System created with ROMS-Tools from GLORYS. Left: potential temperature. Right: grid-aligned horizontal velocity in ξ -direction. Shown for January 1, 2000.

Postprocessing and Analysis

ROMS-Tools supports postprocessing and analysis of ROMS-MARBL output, including regridding from the native curvilinear, terrain-following grid to a standard latitude-longitude-depth grid using xesmf (Zhuang et al., 2023), with built-in plotting for both grid types. The

analysis layer also includes specialized utilities for evaluating carbon dioxide removal (CDR) interventions, such as generating carbon uptake and efficiency curves.

Software Design

ROMS-Tools is designed to balance **ease of use, flexibility, reproducibility, and scalability** by combining high-level user interfaces with a modular, extensible architecture.

Lessons from MATLAB Tools

The legacy MATLAB preprocessing scripts were powerful but required users to edit source code directly to configure simulations. This workflow led to frequent errors for new users, made it difficult to track completed steps, and limited reproducibility. ROMS-Tools addresses these issues with **high-level API calls**, automated error-prone steps, and explicit workflow state management via YAML.

Design Trade-Offs

A central design trade-off in ROMS-Tools is between **user control** and **automation**. Rather than enforcing a fixed workflow, the package exposes key choices such as physical options (e.g., corrections for radiation or wind), interpolation and fill methods, and computational backends. This approach contrasts with opinionated frameworks that fix defaults and directory structures to maximize automation. While users must make explicit decisions, some steps remain automated to prevent errors. For example, bathymetry smoothing is applied automatically using a fixed, non-tunable parameter, since insufficient or omitted smoothing can crash simulations due to pressure gradient errors. This design choice directly addresses issues new users faced in the MATLAB scripts, and balances **flexibility** and **safety**, enabling transparent experimentation without exposing users to avoidable pitfalls.

Another key design consideration is balancing **modular, incremental workflow steps** with **reproducibility**. ROMS-Tools organizes tasks (such as creating InitialConditions, BoundaryForcing, and SurfaceForcing) into small, composable components that can be executed, saved, and revisited independently, rather than following a monolithic, fixed workflow. All components depend on the Grid, but once it is created, the remaining objects are independent. This modular approach avoids unnecessary recomputation when only some inputs change but requires careful tracking of workflow state. To ensure reproducibility, all configuration choices are stored in compact, text-based YAML files. These files are version-controllable, easy to share, and eliminate the need to transfer large model input NetCDF datasets. By explicitly tracking workflow state, this design overcomes a key limitation of the MATLAB scripts and helps users manage experiments more reliably.

Architecture

At the user-facing level, ROMS-Tools provides high-level objects such as Grid, InitialConditions, and BoundaryForcing. Each object exposes a consistent interface (`.ds`, `.plot()`, `.save()`, `.to_yaml()`), allowing users to call the same methods in sequence and inspect attributes that are always present. This design reduces cognitive overhead, makes workflows predictable, and removes the need for new users to edit raw scripts or manually track intermediate files, as was required with the MATLAB tools.

Internally, ROMS-Tools follows a **layered, modular architecture**. Low-level classes (LatLonDataset, ROMSDataset) handle data ingestion and preprocessing, including common operations such as subdomain selection and lateral land filling. Source-specific datasets (e.g., ERA5Dataset, GLORYSDataset, SRTMDataset) inherit from these base classes and encode dataset-specific conventions like variable names, coordinates, and masking. Adding

support for a new data source typically requires only a small subclass to define variable mappings while reusing existing logic, minimizing changes to the core code. High-level classes (Grid, InitialConditions, BoundaryForcing) build on these low-level datasets to produce ready-to-use modeling inputs, performing tasks such as regridding and final assembly. This layered design enhances **extensibility and maintainability**, avoiding the pitfalls of the monolithic MATLAB scripts.

Computational and Data Model Choices

ROMS-Tools is built on xarray, which provides a clear, consistent interface for exploring and inspecting labeled, multi-dimensional geophysical datasets. Users can take advantage of xarray's intuitive indexing, plotting, and metadata handling. Optional dask support allows workflows to scale from laptops to HPC systems, enabling parallel and out-of-core computation for very large input and output datasets. By combining modern Python tools with a user-friendly interface, ROMS-Tools addresses the usability challenges that hampered new users in the MATLAB-based workflow.

Research Impact Statement

ROMS-Tools is used by two primary research communities. First, regional ocean modelers use it to generate reproducible input datasets for ROMS simulations; external users include researchers at **PNNL**, **WHOI**, and **UCLA**. Second, researchers in the ocean-based carbon dioxide removal (CDR) community use ROMS-Tools to configure reproducible ROMS-MARBL simulations of climate intervention scenarios, with adopters including **[C]Worthy**, **Carbon to Sea**, **Ebb Carbon**, and **SCCWRP**. All of these groups have contacted the developers directly or engaged with the project through GitHub or offline discussions. Several manuscripts from these communities are currently in preparation.

Beyond standalone use, ROMS-Tools is integrated into broader scientific workflows, including C-Star ([Stephenson & contributors, 2025](#)), an open-source platform that provides scientifically credible monitoring, reporting, and verification (MRV) for the emerging marine carbon market.

Additional evidence of community uptake comes from public usage metrics. At the time of writing, the GitHub repository shows **119 unique cloners in the past 14 days**, with stars from users at institutions including the University of Waikato, NCAR, University of Maryland, National Oceanography Centre, McGill University, UC Santa Cruz, and others. Distribution statistics indicate **over 3,100 conda-forge downloads in the past six months**, including **68 downloads of the most recent release (v3.3.0)**, and **more than 48,000 total PyPI downloads**. PyPI counts include automated continuous integration (CI) usage by ROMS-Tools, in addition to direct user installations. In contrast, conda-forge downloads of v3.3.0 reflect exclusively human-initiated installs, as C-Star's CI workflows currently pin pre-v3.3.0 releases of ROMS-Tools.

AI Usage Disclosure

Generative AI tools were used to help write docstrings, develop tests, and improve the clarity and readability of both the ROMS-Tools documentation and manuscript text. All AI-assisted content was reviewed and verified by the authors for technical accuracy and correctness.

Acknowledgements

Development of ROMS-Tools has been supported by ARPA-E (DE-AR0001838) and philanthropic donations to [C]Worthy from the Grantham Foundation for the Environment, the Chan Zuckerberg Initiative, Founders Pledge, and the Ocean Resilience Climate Alliance.

References

- Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., Dunne, J. P., Griffies, S. M., Hallberg, R., Harrison, M. J., Held, I. M., Jansen, M. F., John, J. G., Krasting, J. P., Langenhorst, A. R., Legg, S., Liang, Z., McHugh, C., Radhakrishnan, A., ... Zhang, R. (2019). The GFDL Global Ocean and Sea Ice Model OM4.0: Model Description and Simulation Features. *Journal of Advances in Modeling Earth Systems*, 11(10), 3167–3211. <https://doi.org/10.1029/2019MS001726>
- Arango, H., & contributors. (2024). Rutgers ROMS. In *GitHub repository*. GitHub. <https://github.com/myroms/roms>
- Barnes, A. J., Constantinou, N. C., Gibson, A. H., Kiss, A. E., Chapman, C., Reilly, J., Bhagtani, D., & Yang, L. (2024). Regional-mom6: A Python package for automatic generation of regional configurations for the Modular Ocean Model 6. *Journal of Open Source Software*, 9(100), 6857. <https://doi.org/10.21105/joss.06857>
- Bell, N., Olson, L. N., Schroder, J., & Southworth, B. (2023). PyAMG: Algebraic multigrid solvers in python. *Journal of Open Source Software*, 8(87), 5495. <https://doi.org/10.21105/joss.05495>
- Busecke, J., & contributors. (2025). Xgcm. In *GitHub repository*. GitHub. <https://github.com/xgcm/xgcm>
- Dai, A., & Trenberth, K. E. (2002). *Estimates of Freshwater Discharge from Continents: Latitudinal and Seasonal Variations*. https://journals.ametsoc.org/view/journals/hydr/3/6/1525-7541_2002_003_0660_eofdfc_2_0_co_2.xml
- Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. <http://dask.pydata.org>
- Egbert, G. D., & Erofeeva, S. Y. (2002). Efficient Inverse Modeling of Barotropic Ocean Tides. *Journal of Atmospheric and Oceanic Technology*, 19(2), 183–204. [https://doi.org/10.1175/1520-0426\(2002\)019%3C0183:EIMOBO%3E2.0.CO;2](https://doi.org/10.1175/1520-0426(2002)019%3C0183:EIMOBO%3E2.0.CO;2)
- Garcia, H. E., Boyer, T. P., Baranova, O. K., Locarnini, R. A., Mishonov, A. V., Grodsky, A., Paver, C. R., Weathers, K. W., Smolyar, I. V., Reagan, J. R., Seidov, D., & Zweng, M. M. (2019). *World ocean atlas 2018: Product documentation* (A. Mishonov, Ed.). NOAA/NCEI.
- Hamilton, D. S., Perron, M. M. G., Bond, T. C., Bowie, A. R., Buchholz, R. R., Guieu, C., Ito, A., Maenhaut, W., Myriokefalitakis, S., Olgun, N., Rathod, S. D., Schepanski, K., Tagliabue, A., Wagner, R., & Mahowald, N. M. (2022). Earth, Wind, Fire, and Pollution: Aerosol Nutrient Sources and Impacts on Ocean Biogeochemistry. *Annual Review of Marine Science*, 14(Volume 14, 2022), 303–330. <https://doi.org/10.1146/annurev-marine-031921-013612>
- Hauser, M., Spring, A., Busecke, J., Driel, M. van, Lorenz, R., & readthedocs-assistant. (2024). *Regionmask/regionmask: Version 0.12.1*. Zenodo. <https://doi.org/10.5281/zenodo.10849860>
- Hedstrom, K., & contributors. (2023). Pyroms. In *GitHub repository*. GitHub. <https://github.com/ESMG/pyroms>
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., ... Thépaut, J.-N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730), 1999–2049. <https://doi.org/10.1002/qj.3803>
- Hoyer, S., & Hamman, J. (2017). Xarray: ND labeled arrays and datasets in python. *Journal of Open Research Software*, 5(1). <https://doi.org/10.5334/jors.148>

- 304 Huang, Y., Tagliabue, A., & Cassar, N. (2022). Data-Driven Modeling of Dissolved Iron in the
305 Global Ocean. *Frontiers in Marine Science*, 9. <https://doi.org/10.3389/fmars.2022.837183>
- 306 Kok, J. F., Adebisi, A. A., Albani, S., Balkanski, Y., Checa-Garcia, R., Chin, M., Colarco,
307 P. R., Hamilton, D. S., Huang, Y., Ito, A., Klose, M., Leung, D. M., Li, L., Mahowald,
308 N. M., Miller, R. L., Obiso, V., Pérez García-Pando, C., Rocha-Lima, A., Wan, J. S., &
309 Whicker, C. A. (2021). Improved representation of the global dust cycle using observational
310 constraints on dust properties and abundance. *Atmospheric Chemistry and Physics*, 21(10),
311 8127–8167. <https://doi.org/10.5194/acp-21-8127-2021>
- 312 Landschützer, P., Gruber, N., & Bakker, D. C. E. (2016). Decadal variations and trends
313 of the global ocean carbon sink. *Global Biogeochemical Cycles*, 30(10), 1396–1417.
314 <https://doi.org/10.1002/2015GB005359>
- 315 Lauvset, S. K., Key, R. M., Olsen, A., Heuven, S. van, Velo, A., Lin, X., Schirnack, C., Kozyr,
316 A., Tanhua, T., Hoppema, M., Jutterström, S., Steinfeldt, R., Jeansson, E., Ishii, M.,
317 Perez, F. F., Suzuki, T., & Watelet, S. (2016). A new global interior ocean mapped
318 climatology: The 1° × 1° GLODAP version 2. *Earth System Science Data*, 8(2), 325–340.
319 <https://doi.org/https://doi.org/10.5194/essd-8-325-2016>
- 320 Lellouche, E., Jean-Michel Greiner, Bourdallé-Badie, R., Garric, G., Melet, A., Drévillon,
321 M., Bricaud, C., Hamon, M., Le Galloudec, O., Regnier, C., Candela, T., Testut, C.-E.,
322 Gasparin, F., Ruggiero, G., Benkiran, M., Drillet, Y., & Le Traon, P.-Y. (2021). The
323 Copernicus Global 1/12° Oceanic and Sea Ice GLORYS12 Reanalysis. *Frontiers in Earth*
324 *Science*, 9. <https://doi.org/10.3389/feart.2021.698876>
- 325 Long, M. C., Moore, J. K., Lindsay, K., Levy, M., Doney, S. C., Luo, J. Y., Krumhardt, K.
326 M., Letscher, R. T., Grover, M., & Sylvester, Z. T. (2021). *Simulations With the Marine*
327 *Biogeochemistry Library (MARBL)*. <https://doi.org/10.1029/2021MS002647>
- 328 Molemaker, J. (2024). UCLA MATLAB tools. In *GitHub repository*. GitHub. <https://github.com/nmolem/ucla-tools>
- 329 <https://github.com/nmolem/ucla-tools>
- 330 Molemaker, J., & contributors. (2025a). UCLA-ROMS. In *GitHub repository*. GitHub.
331 <https://github.com/CESR-lab/ucla-roms>
- 332 Molemaker, J., & contributors. (2025b). UCLA-ROMS. In *GitHub repository*. GitHub.
333 <https://github.com/CWorthy-ocean/ucla-roms>
- 334 Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system
335 (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model.
336 *Ocean Modelling*, 9(4), 347–404. <https://doi.org/10.1016/j.ocemod.2004.08.002>
- 337 Stephenson, D., & contributors. (2025). *C-star*. [C]Worthy. <https://github.com/CWorthy-ocean/C-Star>
- 338 <https://github.com/CWorthy-ocean/C-Star>
- 339 Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., & Wessel, P. (2019).
340 Global Bathymetry and Topography at 15 Arc Sec: SRTM15+. *Earth and Space Science*,
341 6(10), 1847–1864. <https://doi.org/10.1029/2019EA000658>
- 342 Wessel, P., & Smith, W. H. F. (1996). A global, self-consistent, hierarchical, high-resolution
343 shoreline database. *Journal of Geophysical Research: Solid Earth*, 101(B4), 8741–8743.
344 <https://doi.org/10.1029/96JB00104>
- 345 Yang, S., Chang, B. X., Warner, M. J., Weber, T. S., Bourbonnais, A. M., Santoro, A. E.,
346 Kock, A., Sonnerup, R. E., Bullister, J. L., Wilson, S. T., & Bianchi, D. (2020). Global
347 reconstruction reduces the uncertainty of oceanic nitrous oxide emissions and reveals a
348 vigorous seasonal cycle. *Proceedings of the National Academy of Sciences of the United*
349 *States of America*, 117(22), 11954–11960. <https://doi.org/10.1073/pnas.1921914117>
- 350 Yeager, S. G., Rosenbloom, N., Glanville, A. A., Wu, X., Simpson, I., Li, H., Molina, M.
351 J., Krumhardt, K., Mogen, S., Lindsay, K., Lombardozzi, D., Wieder, W., Kim, W. M.,

