

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Published: unpublished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 efficiently, and analyze model outputs. 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 subsetted, 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-MARBL or ROMS users. While legacy MATLAB-based scripts developed at UCLA (Molemaker, 2024) and Python packages such as pyroms (Hedstrom & contributors, 2023) provide critical functionality, both rely on low-level, manually coordinated steps that limit

41 reproducibility, maintainability, and accessibility. Moreover, frameworks developed for other
42 ocean models cannot be directly applied to ROMS due to fundamental differences in grid
43 geometry, vertical coordinates, and model input requirements. As a result, users lack a modern,
44 integrated framework for reproducible model setup and analysis that is specifically designed for
45 ROMS and ROMS-MARBL.

46 ROMS-Tools was developed to fill this gap. It is an open-source Python framework designed for
47 researchers and practitioners who run ROMS or ROMS-MARBL regional ocean simulations,
48 including users in physical oceanography, marine biogeochemistry, and ocean-based CDR
49 applications. Current capabilities are fully compatible with UCLA-ROMS (Molemaker &
50 contributors, 2025a, 2025b), with potential support for other ROMS implementations, such as
51 Rutgers ROMS (Arango & contributors, 2024), in the future. ROMS-Tools supports large input
52 and output datasets via parallel computation with dask (Dask Development Team, 2016),
53 making workflows scalable from laptops to high-performance computing clusters. By lowering
54 technical barriers and improving transparency and reproducibility, ROMS-Tools enables more
55 efficient model development, facilitates scientific collaboration, and supports applications such
56 as verification of marine carbon removal strategies.

57 State of the Field

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

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

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

Overview of ROMS-Tools Functionality

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

Input Data and Preprocessing

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

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

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

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

- **Bathymetry processing:** The bathymetry is smoothed in two stages, first across the 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.

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

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

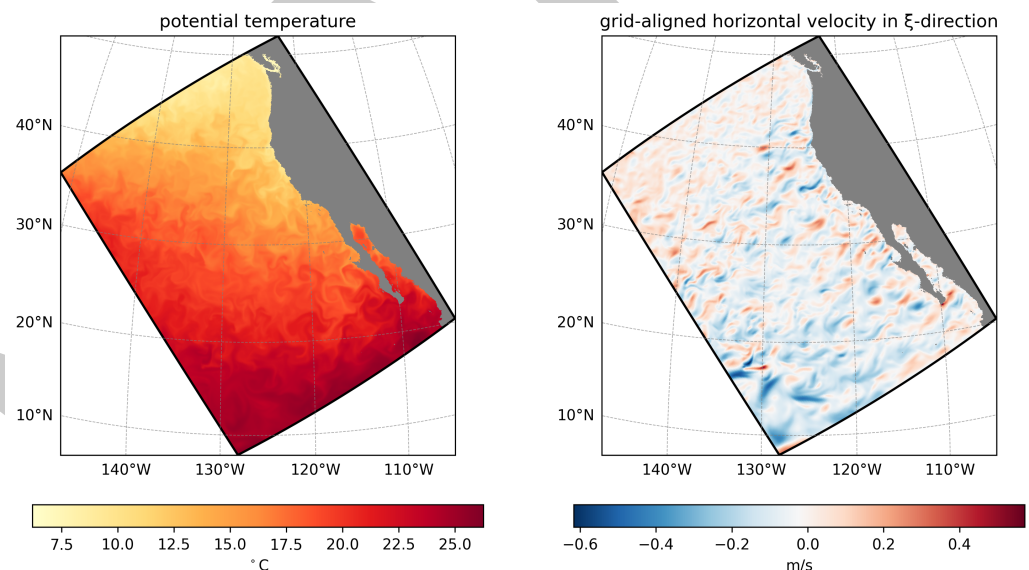


Figure 1: Surface initial conditions for the California Current System created with ROMS-Tools from GLORYS. Left: potential temperature. Right: grid-aligned horizontal velocity in ξ -direction. Shown for January 1, 2000.

162 Postprocessing and Analysis

163 ROMS-Tools supports postprocessing and analysis of ROMS-MARBL output, including
164 regridding from the native curvilinear, terrain-following grid to a standard latitude-longitude-
165 depth grid using xesmf (Zhuang et al., 2023), with built-in plotting for both grid types. The
166 analysis layer also includes specialized utilities for evaluating carbon dioxide removal (CDR)
167 interventions, such as generating carbon uptake and efficiency curves.

168 Software Design

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

171 Lessons from MATLAB Tools

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

177 Design Trade-Offs

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

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

199 Architecture

200 At the user-facing level, ROMS-Tools provides high-level objects such as Grid, InitialConditions,
201 and BoundaryForcing. Each object exposes a consistent interface (`.ds`, `.plot()`, `.save()`,
202 `.to_yaml()`), allowing users to call the same methods in sequence and inspect attributes that
203 are always present. This design reduces cognitive overhead and makes workflows predictable.

204 Internally, ROMS-Tools follows a **layered, modular architecture**. Low-level classes
205 (`LatLonDataset`, `ROMSDataset`) handle data ingestion and preprocessing, including common
206 operations such as subdomain selection and lateral land filling. Source-specific datasets
207 (e.g., `ERA5Dataset`, `GLORYSDataset`, `SRTMDataset`) inherit from these base classes and
208 encode dataset-specific conventions like variable names, coordinates, and masking. Adding
209 support for a new data source typically requires only a small subclass to define variable
210 mappings while reusing existing logic, minimizing changes to the core code. High-level classes
211 (`Grid`, `InitialConditions`, `BoundaryForcing`) build on these low-level datasets to produce
212 ready-to-use modeling inputs, performing tasks such as regridding and final assembly. This
213 layered design enhances **extensibility and maintainability**, avoiding the pitfalls of the monolithic
214 MATLAB scripts.

215 Computational and Data Model Choices

216 ROMS-Tools is built on xarray, which provides a clear, consistent interface for exploring and
217 inspecting labeled, multi-dimensional geophysical datasets. Users can take advantage of
218 xarray's intuitive indexing, plotting, and metadata handling. Optional dask enables parallel
219 and out-of-core computation for very large input and output datasets.

220 Research Impact Statement

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

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

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

243 AI Usage Disclosure

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

247 Acknowledgements

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

251 References

252 Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., Dunne, J.
253 P., Griffies, S. M., Hallberg, R., Harrison, M. J., Held, I. M., Jansen, M. F., John, J. G.,
254 Krasting, J. P., Langenhorst, A. R., Legg, S., Liang, Z., McHugh, C., Radhakrishnan,
255 A., ... Zhang, R. (2019). The GFDL Global Ocean and Sea Ice Model OM4.0: Model
256 Description and Simulation Features. *Journal of Advances in Modeling Earth Systems*,
257 *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>
- Huang, Y., Tagliabue, A., & Cassar, N. (2022). Data-Driven Modeling of Dissolved Iron in the Global Ocean. *Frontiers in Marine Science*, 9. <https://doi.org/10.3389/fmars.2022.837183>
- Kok, J. F., Adebisi, A. A., Albani, S., Balkanski, Y., Checa-Garcia, R., Chin, M., Colarco, P. R., Hamilton, D. S., Huang, Y., Ito, A., Klose, M., Leung, D. M., Li, L., Mahowald, N. M., Miller, R. L., Obiso, V., Pérez García-Pando, C., Rocha-Lima, A., Wan, J. S., & Whicker, C. A. (2021). Improved representation of the global dust cycle using observational constraints on dust properties and abundance. *Atmospheric Chemistry and Physics*, 21(10), 8127–8167. <https://doi.org/10.5194/acp-21-8127-2021>

- 306 Landschützer, P., Gruber, N., & Bakker, D. C. E. (2016). Decadal variations and trends
307 of the global ocean carbon sink. *Global Biogeochemical Cycles*, 30(10), 1396–1417.
308 <https://doi.org/10.1002/2015GB005359>
- 309 Lauvset, S. K., Key, R. M., Olsen, A., Heuven, S. van, Velo, A., Lin, X., Schirnack, C., Kozyr,
310 A., Tanhua, T., Hoppema, M., Jutterström, S., Steinfeldt, R., Jeansson, E., Ishii, M.,
311 Perez, F. F., Suzuki, T., & Watelet, S. (2016). A new global interior ocean mapped
312 climatology: The 1° × 1° GLODAP version 2. *Earth System Science Data*, 8(2), 325–340.
313 <https://doi.org/https://doi.org/10.5194/essd-8-325-2016>
- 314 Lellouche, E., Jean-Michel Greiner, Bourdallé-Badie, R., Garric, G., Melet, A., Drévilion,
315 M., Bricaud, C., Hamon, M., Le Galloudec, O., Regnier, C., Candela, T., Testut, C.-E.,
316 Gasparin, F., Ruggiero, G., Benkiran, M., Drillet, Y., & Le Traon, P.-Y. (2021). The
317 Copernicus Global 1/12° Oceanic and Sea Ice GLORYS12 Reanalysis. *Frontiers in Earth*
318 *Science*, 9. <https://doi.org/10.3389/feart.2021.698876>
- 319 Long, M. C., Moore, J. K., Lindsay, K., Levy, M., Doney, S. C., Luo, J. Y., Krumhardt, K.
320 M., Letscher, R. T., Grover, M., & Sylvester, Z. T. (2021). *Simulations With the Marine*
321 *Biogeochemistry Library (MARBL)*. <https://doi.org/10.1029/2021MS002647>
- 322 Molemaker, J. (2024). UCLA MATLAB tools. In *GitHub repository*. GitHub. <https://github.com/nmolem/ucla-tools>
- 323
- 324 Molemaker, J., & contributors. (2025a). UCLA-ROMS. In *GitHub repository*. GitHub.
325 <https://github.com/CESR-lab/ucla-roms>
- 326 Molemaker, J., & contributors. (2025b). UCLA-ROMS. In *GitHub repository*. GitHub.
327 <https://github.com/CWorthy-ocean/ucla-roms>
- 328 Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system
329 (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model.
330 *Ocean Modelling*, 9(4), 347–404. <https://doi.org/10.1016/j.ocemod.2004.08.002>
- 331 Stephenson, D., & contributors. (2025). *C-star*. [C]Worthy. [https://github.com/CWorthy-](https://github.com/CWorthy-ocean/C-Star)
332 [ocean/C-Star](https://github.com/CWorthy-ocean/C-Star)
- 333 Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., & Wessel, P. (2019).
334 Global Bathymetry and Topography at 15 Arc Sec: SRTM15+. *Earth and Space Science*,
335 6(10), 1847–1864. <https://doi.org/10.1029/2019EA000658>
- 336 Wessel, P., & Smith, W. H. F. (1996). A global, self-consistent, hierarchical, high-resolution
337 shoreline database. *Journal of Geophysical Research: Solid Earth*, 101(B4), 8741–8743.
338 <https://doi.org/10.1029/96JB00104>
- 339 Yang, S., Chang, B. X., Warner, M. J., Weber, T. S., Bourbonnais, A. M., Santoro, A. E.,
340 Kock, A., Sonnerup, R. E., Bullister, J. L., Wilson, S. T., & Bianchi, D. (2020). Global
341 reconstruction reduces the uncertainty of oceanic nitrous oxide emissions and reveals a
342 vigorous seasonal cycle. *Proceedings of the National Academy of Sciences of the United*
343 *States of America*, 117(22), 11954–11960. <https://doi.org/10.1073/pnas.1921914117>
- 344 Yeager, S. G., Rosenbloom, N., Glanville, A. A., Wu, X., Simpson, I., Li, H., Molina, M.
345 J., Krumhardt, K., Mogen, S., Lindsay, K., Lombardozzi, D., Wieder, W., Kim, W. M.,
346 Richter, J. H., Long, M., Danabasoglu, G., Bailey, D., Holland, M., Lovenduski, N., ...
347 King, T. (2022). The Seasonal-to-Multiyear Large Ensemble (SMYLE) prediction system
348 using the Community Earth System Model version 2. *Geoscientific Model Development*,
349 15(16), 6451–6493. <https://doi.org/10.5194/gmd-15-6451-2022>
- 350 Zhuang, J., Dussin, R., Huard, D., Bourgault, P., Banihirwe, A., Raynaud, S., Malevich,
351 B., Schupfner, M., Filipe, Levang, S., Gauthier, C., Jüling, A., Almansi, M., Scott,
352 R., RondeauG, Rasp, S., Smith, T. J., Stachelek, J., Plough, M., & Li, X. (2023).
353 xESMF: Universal regridding for geospatial data. In *GitHub repository*. Zenodo. <https://doi.org/10.5281/zenodo.7688888>

DRAFT