The Journal of Open Source Software

ROMS-Tools: A Python Package for Preparing and Analyzing ROMS Simulations

Nora Loose[1], **Tom Nicholas**[2], **Scott Eilerman**[1], **Christopher McBride**[1], **Sam Maticka**[1], **Dafydd Stephenson**[1], **Scott Bachman**[1], **Pierre Damien**[3], **Ulla Heede**[1], **Alicia Karspeck**[1], **Matthew C. Long**[1], **Jeroen Molemaker**[3], and **Abigale Wyatt**[1]

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

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The ocean shapes Earth's climate and sustains marine ecosystems by circulating and storing vast amounts of heat, oxygen, carbon, and nutrients, while exchanging heat and gases with the atmosphere. To understand these complex dynamics and processes, scientists rely on ocean models, powerful computer simulations of physical circulation and biogeochemical (BGC) dynamics. These models represent the ocean on a grid of cells, where finer grid spacing (more, smaller cells) provides higher fidelity and greater detail but requires significantly more computing power. While global ocean models simulate the entire ocean, **regional ocean models** focus computational resources on a specific area to achieve much finer grid spacing than is computationally feasible over the global domain. This finer grid spacing enables regional ocean models to explicitly resolve fine-scale phenomena, like mesoscale (10-100 km) and submesoscale (0.1-10 km) features, tidal dynamics, coastal currents, upwelling, and detailed BGC processes. Capturing these dynamics and processes at high fidelity is essential for applications in environmental management, fisheries, for assessing regional impacts of climate change, and for evaluating ocean-based carbon dioxide removal (CDR) strategies.

A widely used regional ocean model is the **Regional Ocean Modeling System (ROMS)** (Shchepetkin & McWilliams, 2005). To connect physical circulation with ecosystem dynamics and the ocean carbon cycle, ROMS has been coupled to a BGC model called the Marine Biogeochemistry Library (MARBL) (Long et al., 2021). This coupled framework allows researchers to explore a variety of scientific and practical questions. For example, it can be used to investigate the potential of ocean-based carbon removal strategies, such as adding alkaline materials to the ocean to sequester atmospheric carbon dioxide. It can also be used to study how physical processes drive ecosystem dynamics, such as how nutrient-rich waters from upwelling fuel the phytoplankton blooms that form the base of the marine food web (Gruber et al., 2006).

Input Data and Preprocessing

Whether for research or industrial-focused applications, configuring a regional ocean model like ROMS-MARBL remains a major technical challenge. Generating the required input files is time-consuming, error-prone, and difficult to reproduce, creating a bottleneck for both new and experienced model users. The Python package `ROMS-Tools` addresses this challenge by providing an efficient, user-friendly, and reproducible workflow to generate all required model input files. Its user interface and underlying data model are based on `xarray` (Hoyer & Hamman, 2017), enabling seamless handling of multidimensional datasets with rich metadata and optional parallelization via a `dask` (Dask Development Team, 2016) backend.

`ROMS-Tools` can automatically process commonly used datasets or incorporate custom user data and routines. Currently, it can generate the following inputs:

1. **Model Grid**: Customizable, curvilinear grid, rotatable to align with coastlines, with a terrain-following vertical coordinate.
2. **Bathymetry**: Derived from **SRTM15** (Tozer et al., 2019).
3. **Land Mask**: Inferred from **Natural Earth** coastlines.
4. **Physical Ocean Conditions**: Initial and open boundary conditions for sea surface height, temperature, salinity, and velocities derived from GLORYS (Lellouche et al., 2021).
5. **BGC Ocean Conditions**: Initial and open boundary conditions for dissolved inorganic carbon, alkalinity, and other biogeochemical tracers from CESM output (Yeager et al., 2022) or hybrid observational-model sources.
6. **Meteorological forcing**: Wind, radiation, precipitation, and air temperature/humidity processed from 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.
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, suitable for the simulation of field- or large-scale CDR experiments.

While the source datasets listed above (GLORYS, ERA5, SRTM15, TPXO, etc.) are the ones currently supported, the package's modular design makes it straightforward to add new data sources or custom routines in the future. To generate the model inputs listed above, 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 along the shelf, to ensure local steepness ratios are not exceeded and 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 regionmask (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 resolve discrepancies between source data and ROMS land masks that could otherwise produce artificial values in ocean 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). Optional sea surface height corrections can be applied, and velocities are rotated to align with the curvilinear ROMS grid.
- **Longitude conventions**: `ROMS-Tools` handles differences in longitude conventions, converting between [-180°, 180°] and [0°, 360°] 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.
- **Atmospheric data streaming**: ERA5 atmospheric data can be accessed directly from the cloud, removing the need for users to pre-download large datasets locally.

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 is shown in Figure 1.

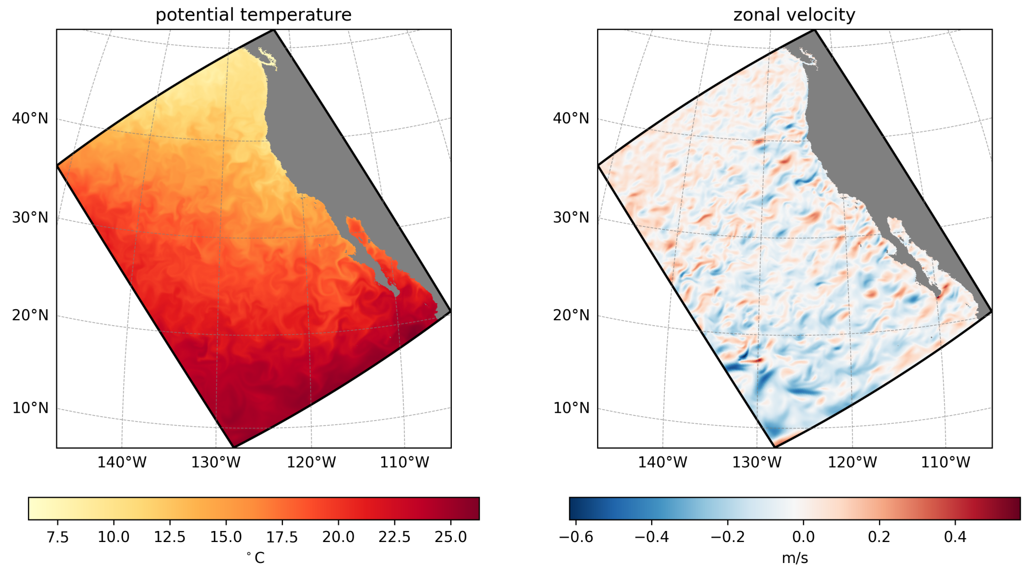

Figure 1: Surface initial conditions for the California Current System created with `ROMS-Tools` from GLORYS. Left: potential temperature. Right: zonal velocity. Shown for January 1, 2000.

`ROMS-Tools` also includes features that facilitate simulation management. It supports partitioning input files to enable parallelized ROMS simulations across multiple nodes, and writes NetCDF outputs with metadata fully compatible with ROMS-MARBL. Currently, UCLA-ROMS (Molemaker & contributors, 2025) is fully supported, with the potential to add other ROMS versions, such as Rutgers ROMS (Arango & contributors, 2024), in the future.

Postprocessing and Analysis

`ROMS-Tools` includes analysis tools for postprocessing ROMS-MARBL output. It first provides a joining tool (the counterpart to the input file partitioning utility described earlier) that merges ROMS output files produced as tiles from multi-node simulations. Beyond file management, there are `ROMS-Tools` analysis utilities for general-purpose tasks, such as loading model output directly into an Xarray dataset with additional useful metadata, enabling seamless use of the Pangeo scientific Python ecosystem for further analysis and visualization. The analysis layer also supports regridding from the native curvilinear ROMS grid with terrain-following coordinate to a standard latitude-longitude-depth grid using xesmf (Zhuang et al., 2023). Beyond these general-purpose features, the `ROMS-Tools` analysis layer offers a suite of targeted tools for evaluating CDR interventions. These include utilities for generating standard plots, such as CDR efficiency curves, and performing specialized tasks essential for CDR monitoring, reporting, and verification.

Workflow, Reproducibility, and Performance

`ROMS-Tools` is designed to support modern, reproducible workflows. It is easily installable via Conda or PyPI and can be run interactively from Jupyter Notebooks. To ensure reproducibility and facilitate collaboration, each workflow is defined in a simple YAML configuration file. These compact, text-based YAML files can be version-controlled and easily shared, eliminating the need to transfer large NetCDF files between researchers, as source data like GLORYS and ERA5 are accessible in the cloud. For performance, the package is integrated with `dask` (Dask Development Team, 2016) to enable efficient, out-of-core computations on large datasets. Finally, to ensure reliability, the software is rigorously tested with continuous integration (CI) and supported by comprehensive documentation.

The Journal of Open Source Software

Statement of Need

Setting up a regional ocean model is a major undertaking. It requires generating a wide range of complex input files, including the model grid, initial and boundary conditions, and forcing from the atmosphere, tides, and rivers. Traditionally, this work has depended on a patchwork of custom scripts and lab-specific workflows, which can be time-consuming, error-prone, and difficult to reproduce. These challenges slow down science, create a steep barrier to entry for new researchers, and limit collaboration across groups.

Within the ROMS community, the preprocessing landscape has been shaped by tools like `pyroms` (Hedstrom & contributors, 2023). While `pyroms` has long provided valuable low-level utilities, it also presents challenges for new users. Installation can be cumbersome due to its Python and Fortran dependencies, and its inconsistent Application Programming Interface (API) and limited documentation make it hard to learn. The package was not designed with reproducible workflows in mind, and it lacks tests, CI, and support for modern Python tools such as `xarray` and `dask`. Since development of `pyroms` has largely ceased, its suitability for new projects, such as CDR simulations, is increasingly limited. Importantly, tools from other modeling communities cannot simply be adopted, since each ocean model has distinct structural requirements. For example, the new `regional-mom6` package (Barnes et al., 2024), developed for MOM6 (Adcroft et al., 2019), cannot be used to generate ROMS inputs, because ROMS employs a terrain-following vertical coordinate system that requires a fundamentally different regridding approach, whereas MOM6 accepts inputs on arbitrary depth levels. Several other differences further prevent cross-compatibility. Together, these limitations underscored the need for a modern, maintainable, and reproducible tool designed specifically for ROMS.[1]

`ROMS-Tools` was developed to meet this need. It draws on the legacy of the MATLAB preprocessing scripts developed at UCLA (Molemaker, 2024), which encapsulate decades of expertise in configuring regional ocean model inputs. While many of the core algorithms and design principles are retained, `ROMS-Tools` provides an open-source Python implementation of these MATLAB tools using an object-oriented programming paradigm. This implementation enables a modernized workflow driven by high-level user API calls, enhancing reproducibility, reducing the potential for user errors, and supporting extensibility for additional features, forcing datasets, and use cases. In some cases, `ROMS-Tools` diverges from the MATLAB implementation to take advantage of new methods or better integration with the modern Python ecosystem. By streamlining input generation and analysis, `ROMS-Tools` reduces technical overhead, lowers the barrier to entry, and enables scientists to focus on research rather than data preparation. The primary users of the package include ocean modelers developing new domains and researchers in the CDR community, who use `ROMS-Tools` to test climate intervention scenarios.

Acknowledgements

Acknowledgement of any financial support.

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

[1] In the future, packages like ROMS-Tools and regional-mom6 could share a common backbone, with model-specific adaptations layered on top.

The Journal of Open Source Software

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

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

Gruber, N., Frenzel, H., Doney, S. C., Marchesiello, P., McWilliams, J. C., Moisan, J. R., Oram, J. J., Plattner, G.-K., & Stolzenbach, K. D. (2006). Eddy-resolving simulation of plankton ecosystem dynamics in the California Current System. *Deep Sea Research Part I: Oceanographic Research Papers*, *53*(9), 1483–1516. https://doi.org/10.1016/j.dsr.2006.06.005

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/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

Lellouche, E., Jean-Michel Greiner, Bourdallé-Badie, R., Garric, G., Melet, A., Drévillon, M., Bricaud, C., Hamon, M., Le Galloudec, O., Regnier, C., Candela, T., Testut, C.-E., Gasparin, F., Ruggiero, G., Benkiran, M., Drillet, Y., & Le Traon, P.-Y. (2021). The Copernicus Global 1/12° Oceanic and Sea Ice GLORYS12 Reanalysis. *Frontiers in Earth Science*, *9*. https://doi.org/10.3389/feart.2021.698876

Long, M. C., Moore, J. K., Lindsay, K., Levy, M., Doney, S. C., Luo, J. Y., Krumhardt, K. M., Letscher, R. T., Grover, M., & Sylvester, Z. T. (2021). *Simulations With the Marine Biogeochemistry Library (MARBL)*. https://doi.org/10.1029/2021MS002647

Molemaker, J. (2024). UCLA MATLAB tools. In *GitHub repository*. GitHub. https://github.com/nmolem/ucla-tools

Molemaker, J., & contributors. (2025). UCLA-ROMS. In *GitHub repository*. GitHub. https://github.com/CESR-lab/ucla-roms

Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system

(ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, *9*(4), 347–404. https://doi.org/10.1016/j.ocemod.2004.08.002

Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., & Wessel, P. (2019). Global Bathymetry and Topography at 15 Arc Sec: SRTM15+. *Earth and Space Science*, *6*(10), 1847–1864. https://doi.org/10.1029/2019EA000658

Yeager, S. G., Rosenbloom, N., Glanville, A. A., Wu, X., Simpson, I., Li, H., Molina, M. J., Krumhardt, K., Mogen, S., Lindsay, K., Lombardozzi, D., Wieder, W., Kim, W. M., Richter, J. H., Long, M., Danabasoglu, G., Bailey, D., Holland, M., Lovenduski, N., … King, T. (2022). The Seasonal-to-Multiyear Large Ensemble (SMYLE) prediction system using the Community Earth System Model version 2. *Geoscientific Model Development*, *15*(16), 6451–6493. https://doi.org/10.5194/gmd-15-6451-2022

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., Scott, R., RondeauG, Rasp, S., Smith, T. J., Stachelek, J., Plough, M., & Li, X. (2023). xESMF: Universal regridder for geospatial data. In *GitHub repository*. Zenodo. https://doi.org/10.5281/zenodo.4294774