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], **Matt 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Published: unpublished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 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 higher resolution (more, smaller cells) provides 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 higher resolution and can therefore resolve fine-scale processes like mesoscale (10-100 km) and submesoscale (0.1-10 km) features, tidal dynamics, coastal currents, upwelling, and detailed BGC cycles.

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 can be coupled to a BGC model, for example the Marine Biogeochemistry Library (MARBL) (Long et al., 2021). This coupled framework allows researchers to explore 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).

Yet configuring a regional ocean model like ROMS-MARBL remains a major challenge. Generating the required input files is time-consuming, error-prone, and difficult to reproduce, creating a bottleneck for both new and experienced researchers. The Python package `ROMS-Tools` addresses this challenge by providing an efficient, user-friendly, and reproducible workflow to generate all required inputs, including:

- **Model Grid**: A customizable, curvilinear grid that can be rotated to align with coastlines and with a terrain-following vertical coordinate.
- **Bathymetry and Land Mask**: High-resolution seafloor depth from **SRTM15** (Tozer et al., 2019) and a corresponding land-sea mask from **Natural Earth**.
- **Initial Conditions** and **Boundary Forcing**: Physical state variables (temperature, velocities, etc.) from **GLORYS** (Lellouche et al., 2021) and BGC fields (e.g., alkalinity) from hybrid observational-model sources.
- **Atmospheric Forcing:** Meteorological drivers (wind, radiation, precipitation) from ERA5 (Hersbach et al., 2020) and surface BGC forcing (pCO_2, dust, nitrogen deposition) from hybrid observational-model sources.
- **Tidal Forcing:** Tidal potential, elevation, and velocities derived from **TPXO** (Egbert & Erofeeva, 2002) including corrections for self-attraction and loading (SAL).

- **River Forcing:** Freshwater runoff from **Dai & Trenberth** (Dai & Trenberth, 2002) or custom user-provided files.
- **CDR Forcing**: Flexible user-defined tracers for Carbon Dioxide Removal (CDR) or other interventions.

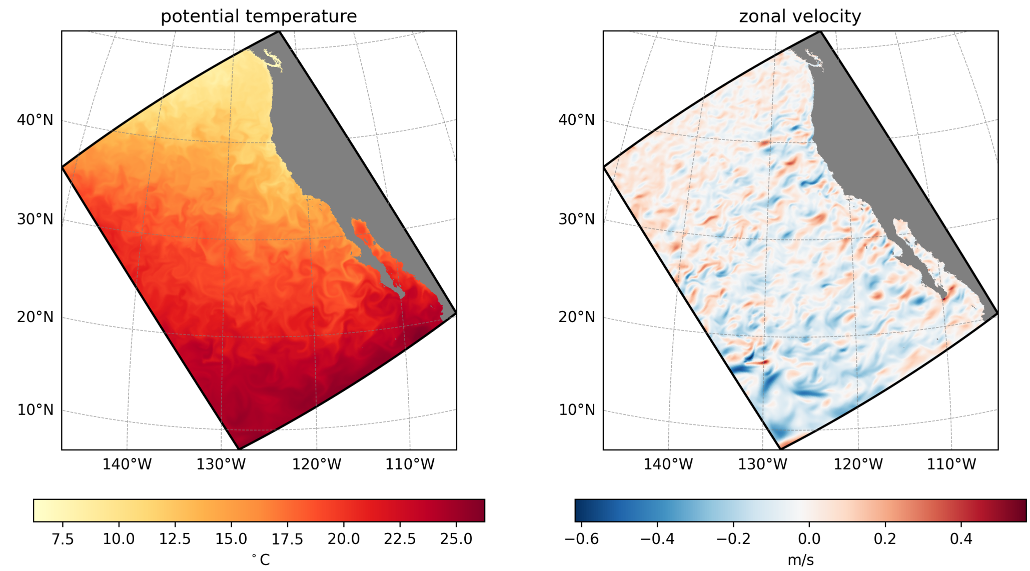

Figure 1: Surface initial conditions for the California Current System created with `ROMS-Tools` from GLORYS. Left: potential temperature. Right: zonal velocity. Shown for January 1, 2000.

An example of the generated inputs is shown in Figure Figure 1, which illustrates surface initial conditions for the California Current System created with `ROMS-Tools`.

While generating input files, `ROMS-Tools` automates several complex intermediate processing steps. It efficiently fills land values in the input data via an algebraic multigrid method using `pyamg` (Bell et al., 2023). It then performs horizontal and vertical regridding from standard lat-lon-depth grids to the model's curvilinear grid with terrain-following vertical coordinate using libraries like `xarray` (Hoyer & Hamman, 2017) and `xgcm` (Busecke & contributors, 2025). The workflow also applies bathymetry smoothing to reduce pressure-gradient errors, handles longitude conversions (between -180° to 180° and 0° to 360°), and formats all NetCDF outputs with the metadata expected by ROMS-MARBL. Currently, `ROMS-Tools` fully supports UCLA-ROMS (Molemaker & contributors, 2025); support for other versions, such as Rutgers ROMS (Arango & contributors, 2024), may be added in the future with community contributions. A notable feature is the ability to stream ERA5 data directly from the cloud, so users do not need to download the data beforehand. `ROMS-Tools` offers functionality for partitioning input files, a requirement for parallelized ROMS simulations across multiple nodes. Additionally, it can recombine the partitioned model output files. For analysis, the package includes postprocessing utilities for generating plots, performing specialized tasks useful for CDR monitoring, reporting, and verification, and regridding model output from the native ROMS grid to a standard lat-lon-depth grid using xesmf (Zhuang et al., 2023).

`ROMS-Tools` is designed to support modern, reproducible workflows. It is easily installable via Conda and 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. For performance, the package is integrated with dask (Dask Development Team, 2016) to enable efficient, out-of-core computations on large datasets. Finally, to guarantee reliability, the software is rigorously tested with continuous integration (CI) and supported by comprehensive documentation.

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, a fragmented approach that is 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 API and limited documentation make it harder to learn. The package was not designed with reproducible workflows in mind, and it lacks tests, continuous integration, and support for modern Python tools such as `xarray` and `dask`. Since development of `pyroms` has largely ceased, its suitability for new projects is increasingly limited. Importantly, tools from other modeling communities cannot simply be adapted, 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.

`ROMS-Tools` was developed to meet this need. It draws on the legacy of the UCLA MATLAB preprocessing scripts (Molemaker, 2024), which encapsulate decades of community expertise in configuring regional ocean model inputs. While many of the core algorithms and design principles are retained, `ROMS-Tools` reimplements them in Python, where it modernizes the workflow, adopts object-oriented programming, and improves reproducibility. In some cases, it 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. While designed for ocean modelers developing new domains, the package is also gaining traction in the Carbon Dioxide Removal (CDR) community, where it enables testing of different climate intervention scenarios within existing, well-validated model setups.

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

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

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

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

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