









# 1 ROMS-Tools: Reproducible Preprocessing and 2 Analysis for Regional Ocean Modeling with ROMS

3 **Nora Loose** <sup>1</sup>, **Tom Nicholas** <sup>2</sup>, **Sam Maticka**<sup>1</sup>, **Scott Eilerman**<sup>1</sup>,  
4 **Christopher McBride**<sup>1</sup>, **Dafydd Stephenson**<sup>1</sup>, **Ulla Heede**<sup>1</sup>, **Benjamin**  
5 **Saenz** <sup>1</sup>, **Kristen M. Thyng** <sup>1</sup>, **Scott Bachman**<sup>1</sup>, **Pierre Damien**<sup>3</sup>, **Alicia**  
6 **Karspeck**<sup>1</sup>, **Matthew C. Long**<sup>1</sup>, **M. Jeroen Molemaker**<sup>3</sup>, and **Abigale**  
7 **Wyatt** <sup>1</sup>

8 1 [C]Worthy LLC, Boulder, CO, United States 2 Earthmover PBC 3 University of California, Los  
9 Angeles, CA, United States

DOI: [10.xxxxx/draft](https://doi.org/10.xxxxx/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Published: unpublished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](#))

## 10 Summary

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

22 ROMS-Tools is a Python package that streamlines the **preparation and analysis of ROMS-**  
23 **MARBL simulations** by enabling users to generate regional grids, prepare model inputs  
24 efficiently, and analyze model outputs. A detailed overview of the package's functionality is  
25 available in the ROMS-Tools [documentation](#). By providing a modern, user-friendly interface,  
26 ROMS-Tools lowers technical barriers, improves reproducibility, and allows scientists to focus  
27 on research rather than data preparation. The package is installable via Conda or PyPI and  
28 can be run interactively in Jupyter notebooks.

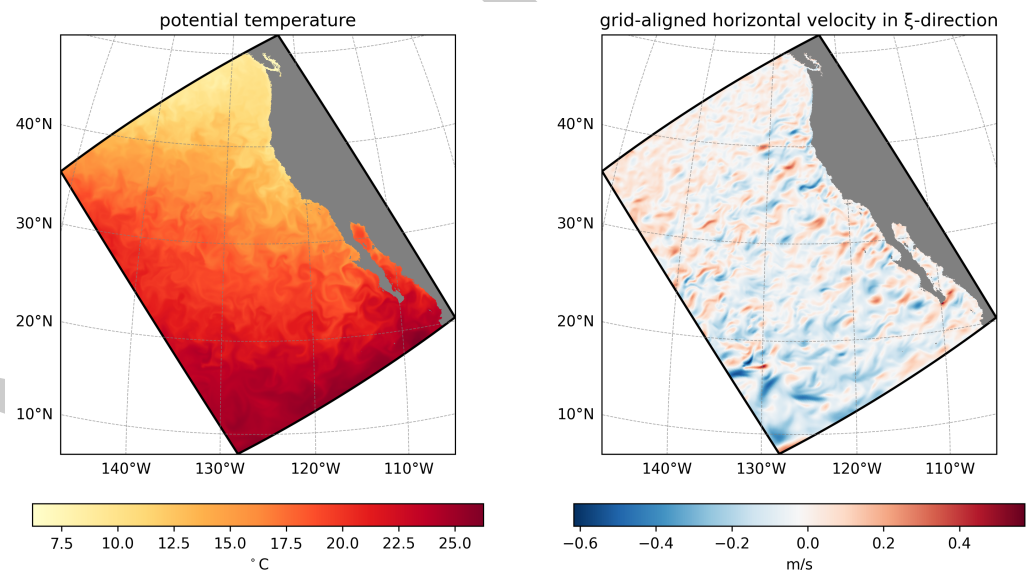
## 29 Statement of Need

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

39 Existing tools within the ocean modeling ecosystem do not fully address these challenges for  
40 ROMS-MARBL or ROMS users. While legacy MATLAB-based scripts developed at UCLA

41 (Molemaker, 2024) and Python packages such as pyroms (Hedstrom & contributors, 2023)  
42 provide critical functionality, both rely on low-level, manually coordinated steps that limit  
43 reproducibility, maintainability, and accessibility. Moreover, frameworks developed for other  
44 ocean models cannot be directly applied to ROMS due to fundamental differences in grid  
45 geometry, vertical coordinates, and model input requirements. As a result, users lack a modern,  
46 integrated framework for reproducible model setup and analysis that is specifically designed for  
47 ROMS and ROMS-MARBL.

48 ROMS-Tools was developed to fill this gap. It is an open-source Python framework designed for  
49 researchers and practitioners who run ROMS or ROMS-MARBL regional ocean simulations,  
50 including users in physical oceanography, marine biogeochemistry, and ocean-based CDR  
51 applications. Current capabilities are fully compatible with UCLA-ROMS (Molemaker &  
52 contributors, 2025a, 2025b), with potential support for other ROMS implementations, such as  
53 Rutgers ROMS (Arango & contributors, 2024), in the future. The package handles large input  
54 and output datasets via parallel computation with dask (Dask Development Team, 2016),  
55 making workflows scalable from laptops to high-performance computing clusters. Built-in  
56 visualization tools enable quick inspection of regional grids as well as model input and output  
57 fields. For example, Figure 1 shows surface initial conditions for a California Current System  
58 simulation at 5 km horizontal resolution, generated and visualized directly using ROMS-Tools.  
59 By lowering technical barriers and improving transparency and reproducibility, ROMS-Tools  
60 enables more efficient model development, facilitates scientific collaboration, and supports  
61 applications such as verification of marine carbon removal strategies.



**Figure 1:** Surface initial conditions for the California Current System created and visualized with ROMS-Tools. Left: potential temperature. Right: grid-aligned horizontal velocity in  $\xi$ -direction. Shown for January 1, 2000.

## 62 State of the Field

63 Historically, setting up a regional ocean model required a patchwork of custom scripts and  
64 lab-specific workflows, resulting in error-prone and difficult-to-reproduce processes. Within  
65 the ROMS community, tools like pyroms (Hedstrom & contributors, 2023) addressed some  
66 of these issues by providing low-level Python utilities for preprocessing ROMS model inputs.  
67 However, pyroms has several limitations: installation is cumbersome due to Python/Fortran  
68 dependencies, the API is inconsistent, and documentation and tests are missing. The package  
69 does not support modern tools such as xarray (Hoyer & Hamman, 2017), nor reproducible

70 workflows. Active development has ceased, and maintenance (including compatibility with  
71 newer Python versions) is no longer provided. Together, these limitations make it very difficult  
72 to add new features, such as support for BGC and CDR applications, and improvements to  
73 user-friendliness.

74 Tools from other modeling communities cannot be directly applied to ROMS because each  
75 model has distinct structural requirements and input conventions. For example, the regional-  
76 mom6 package (Barnes et al., 2024), developed for regional configurations of the Modular  
77 Ocean Model v6 (MOM6) (Adcroft et al., 2019), cannot generate ROMS inputs. ROMS  
78 uses a terrain-following vertical coordinate system that requires specialized vertical regridding,  
79 whereas MOM6 accepts inputs on arbitrary depth levels and does not require vertical regridding  
80 at all. While ROMS and MOM6 differ in fundamental ways, regional-mom6 represents  
81 the closest comparable tool to ROMS-Tools in the wider modeling ecosystem. Notably, the  
82 main development cycles of regional-mom6 and ROMS-Tools overlapped (regional-mom6:  
83 2023–2024; ROMS-Tools: 2024–2025, based on public GitHub commits). Had the developers  
84 been aware of each other, a shared framework could potentially have been created, with  
85 model-specific adaptations layered on top. Adapting one framework to the other now would  
86 require extensive architectural changes.

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

## 96 Software Design

97 ROMS-Tools emphasizes ease of use, flexibility, reproducibility, and scalability through a modular  
98 architecture and high-level user interfaces.

### 99 Design Trade-Offs

100 A central design trade-off in ROMS-Tools is between **user control** and **automation**. Rather than  
101 enforcing a fixed workflow, the package exposes key choices such as physical options (e.g.,  
102 corrections for radiation or wind), interpolation and fill methods, and computational backends.  
103 This approach contrasts with opinionated frameworks that fix defaults and directory structures to  
104 maximize automation. While users must make explicit decisions, some steps remain automated  
105 to prevent errors. For example, bathymetry smoothing is applied automatically using a fixed,  
106 non-tunable parameter, since insufficient or omitted smoothing can crash simulations due to  
107 pressure gradient errors. This design choice addresses issues encountered by new users of the  
108 original UCLA MATLAB scripts and balances flexibility with safety, enabling experimentation  
109 while avoiding common pitfalls.

110 Another key design consideration is balancing **modular, incremental workflow steps**  
111 with **reproducibility**. ROMS-Tools organizes tasks (such as creating InitialConditions,  
112 BoundaryForcing, and SurfaceForcing) into small, composable components that can be  
113 executed, saved, and revisited independently, rather than following a monolithic, fixed  
114 workflow. All components depend on the Grid, but once it is created, the remaining objects  
115 are independent. This modular approach avoids unnecessary recomputation when only some  
116 inputs change but requires careful tracking of workflow state. To ensure reproducibility,  
117 all configuration choices are stored in compact, text-based YAML files. Since Grid is the  
118 only shared dependency, serializing any object (e.g., InitialConditions) also captures the

119 Grid configuration. A single YAML file therefore contains everything needed to reproduce a  
120 workflow step. These files are version-controllable, easy to share, and eliminate the need to  
121 transfer large model input NetCDF datasets.

## 122 Architecture

123 At the user-facing level, ROMS-Tools provides high-level objects such as `Grid`, `InitialConditions`,  
124 and `BoundaryForcing`. Each object exposes a consistent interface (`.ds`, `.plot()`, `.save()`,  
125 `.to_yaml()`), allowing users to apply the same methods across workflow steps and inspect  
126 standardized attributes that are always present. This consistency reduces cognitive overhead  
127 and makes workflows predictable.

128 Internally, ROMS-Tools follows a layered, modular architecture. Low-level dataset classes  
129 (`LatLonDataset`, `ROMSDataset`) handle data ingestion and preprocessing tasks such as  
130 subdomain selection and lateral land filling. Source-specific datasets (e.g., `ERA5Dataset`,  
131 `GLORYSDataset`, `SRTMDataset`) inherit from these base classes and encode dataset-specific  
132 conventions like variable names, coordinates, and masking. Supporting a new data source  
133 typically requires only a small subclass defining these mappings while reusing existing  
134 preprocessing logic, minimizing changes to the core code.

135 High-level classes (`Grid`, `InitialConditions`, `BoundaryForcing`) build on these low-level  
136 dataset abstractions to generate ready-to-use modeling inputs through operations such as  
137 regridding and final assembly. This layered design improves **extensibility and maintainability**.

## 138 Computational and Data Model Choices

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

## 143 Research Impact Statement

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

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

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

## 165 AI Usage Disclosure

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

## 169 Acknowledgements

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

## 173 References

- 174 Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., Dunne, J.  
175 P., Griffies, S. M., Hallberg, R., Harrison, M. J., Held, I. M., Jansen, M. F., John, J. G.,  
176 Krasting, J. P., Langenhorst, A. R., Legg, S., Liang, Z., McHugh, C., Radhakrishnan,  
177 A., ... Zhang, R. (2019). The GFDL Global Ocean and Sea Ice Model OM4.0: Model  
178 Description and Simulation Features. *Journal of Advances in Modeling Earth Systems*,  
179 *11*(10), 3167–3211. <https://doi.org/10.1029/2019MS001726>
- 180 Arango, H., & contributors. (2024). Rutgers ROMS. In *GitHub repository*. GitHub. <https://github.com/myroms/roms>
- 182 Barnes, A. J., Constantinou, N. C., Gibson, A. H., Kiss, A. E., Chapman, C., Reilly, J.,  
183 Bhagtani, D., & Yang, L. (2024). Regional-mom6: A Python package for automatic  
184 generation of regional configurations for the Modular Ocean Model 6. *Journal of Open*  
185 *Source Software*, *9*(100), 6857. <https://doi.org/10.21105/joss.06857>
- 186 Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. <http://dask.pydata.org>
- 188 Hedstrom, K., & contributors. (2023). Pyroms. In *GitHub repository*. GitHub. <https://github.com/ESMG/pyroms>
- 190 Hoyer, S., & Hamman, J. (2017). Xarray: ND labeled arrays and datasets in python. *Journal*  
191 *of Open Research Software*, *5*(1). <https://doi.org/10.5334/jors.148>
- 192 Long, M. C., Moore, J. K., Lindsay, K., Levy, M., Doney, S. C., Luo, J. Y., Krumhardt, K.  
193 M., Letscher, R. T., Grover, M., & Sylvester, Z. T. (2021). *Simulations With the Marine*  
194 *Biogeochemistry Library (MARBL)*. <https://doi.org/10.1029/2021MS002647>
- 195 Molemaker, J. (2024). UCLA MATLAB tools. In *GitHub repository*. GitHub. <https://github.com/nmolem/ucla-tools>
- 197 Molemaker, J., & contributors. (2025a). UCLA-ROMS. In *GitHub repository*. GitHub.  
198 <https://github.com/CESR-lab/ucla-roms>
- 199 Molemaker, J., & contributors. (2025b). UCLA-ROMS. In *GitHub repository*. GitHub.  
200 <https://github.com/CWorthy-ocean/ucla-roms>
- 201 Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system  
202 (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model.  
203 *Ocean Modelling*, *9*(4), 347–404. <https://doi.org/10.1016/j.ocemod.2004.08.002>
- 204 Stephenson, D., & contributors. (2025). *C-star*. [C]Worthy. <https://github.com/CWorthy-ocean/C-Star>
- 205