

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

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

<sup>7</sup> 1 [C]Worthy LLC, Boulder, CO, United States 2 Earthmover PBC 3 University of California, Los  
<sup>8</sup> 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Published: unpublished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](#))

## <sup>9</sup> Summary

<sup>10</sup> The ocean regulates Earth's climate and sustains marine ecosystems by circulating and storing  
<sup>11</sup> heat, carbon, oxygen, and nutrients, while exchanging heat and gases with the atmosphere.  
<sup>12</sup> Scientists study these processes using ocean models, which simulate the ocean on a grid.  
<sup>13</sup> **Regional ocean models** focus computational resources on a limited geographical area with fine  
<sup>14</sup> grid spacing, and can resolve fine-scale phenomena such as mesoscale and submesoscale features,  
<sup>15</sup> tidal dynamics, coastal currents, upwelling, and detailed biogeochemical (BGC) processes.  
<sup>16</sup> A widely used regional ocean model is the **Regional Ocean Modeling System (ROMS)**  
<sup>17</sup> ([Shchepetkin & McWilliams, 2005](#)). ROMS has been coupled to the Marine Biogeochemistry  
<sup>18</sup> Library (MARBL) ([Long et al., 2021](#); [Molemaker & contributors, 2025a](#)) to link physical and  
<sup>19</sup> BGC processes. ROMS-MARBL supports research on environmental management, fisheries,  
<sup>20</sup> regional climate impacts, and ocean-based carbon dioxide removal (CDR) strategies.

<sup>21</sup> Configuring a regional ocean model like ROMS-MARBL is technically challenging; it requires  
initialization and time-dependent forcing from oceanic and atmospheric data drawn from  
multiple external sources in diverse formats. These global datasets can reach several petabytes  
<sup>22</sup> and must be subsetted, processed, and mapped onto the target domain's geometry, yielding  
<sup>23</sup> input datasets of 10–100 terabytes for large regional models. Generating these input files is  
<sup>24</sup> time-consuming, error-prone, and hard to reproduce, creating a bottleneck for both new and  
<sup>25</sup> experienced users. The Python package ROMS-Tools addresses this challenge by providing  
<sup>26</sup> efficient, dask-backed ([Dask Development Team, 2016](#)), user-friendly tools that can be installed  
<sup>27</sup> via Conda or PyPI and run interactively from Jupyter notebooks. The package supports creating  
<sup>28</sup> regional grids, preprocessing all required model inputs, and postprocessing and analysis. Current  
<sup>29</sup> capabilities are fully compatible with UCLA-ROMS ([Molemaker & contributors, 2025a, 2025b](#)),  
<sup>30</sup> with potential support for other ROMS versions, such as Rutgers ROMS ([Arango & contributors,](#)  
<sup>31</sup> [2024](#)), in the future.

## <sup>34</sup> Input Data and Preprocessing

<sup>35</sup> ROMS-Tools generates the following input files for ROMS-MARBL:

- <sup>36</sup> 1. **Model Grid:** Customizable, curvilinear, and orthogonal grid designed to maintain a nearly  
<sup>37</sup> uniform horizontal resolution across the domain. The grid is rotatable to align with  
<sup>38</sup> coastlines and features a terrain-following vertical coordinate.
- <sup>39</sup> 2. **Bathymetry:** Derived from **SRTM15** ([Tozer et al., 2019](#)).
- <sup>40</sup> 3. **Land Mask:** Inferred from coastlines provided by **Natural Earth** or the Global Self-  
<sup>41</sup> consistent, Hierarchical, High-resolution Geography (**GSHHG**) Database ([Wessel &](#)

Smith, 1996).

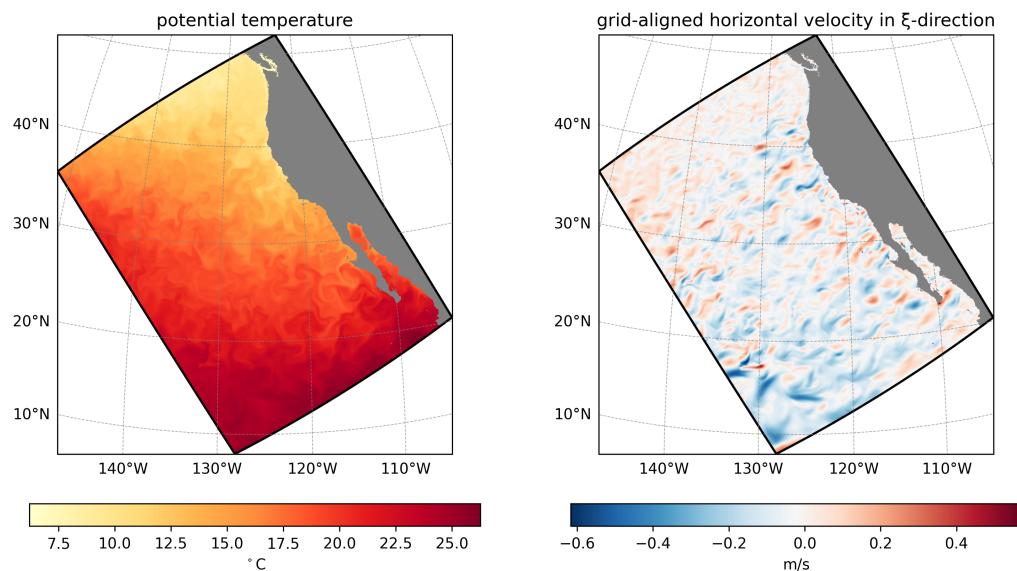
4. **Physical Ocean Conditions:** Initial and open boundary conditions for sea surface height, temperature, salinity, and velocities derived from the 1/12° Global Ocean Physics Reanalysis (**GLORYS**) ([Lellouche et al., 2021](#)).
  5. **BGC Ocean Conditions:** Initial and open boundary conditions for 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.
  - **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

95 span multiple cells can be configured by the user.  
 96 ▪ **Data streaming:** ERA5 atmospheric data can be accessed directly from the cloud,  
 97 removing the need for users to pre-download large datasets locally. Similar streaming  
 98 capabilities may be implemented for other datasets in the future.  
 99 Users can quickly design and visualize regional grids and inspect all input fields with built-in  
 100 plotting utilities. An example of surface initial conditions generated for a California Current  
 101 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: zonal velocity. Shown for January 1, 2000.

## 102 Postprocessing and Analysis

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

## 108 Statement of Need

109 Setting up a regional ocean model is a major technical undertaking. Traditionally, this work  
 110 has relied on a patchwork of custom scripts and lab-specific workflows, which can be time-  
 111 consuming, error-prone, and difficult to reproduce. These challenges slow down science, create  
 112 a steep barrier to entry for new researchers, and limit collaboration across groups.

113 Within the ROMS community, the preprocessing landscape has been shaped by tools like pyroms  
 114 ([Hedstrom & contributors, 2023](#)). While providing valuable low-level utilities, pyroms presents  
 115 challenges for new users: installation is cumbersome due to Python/Fortran dependencies, the  
 116 API is inconsistent, documentation is limited, and it lacks tests, CI, and support for modern  
 117 Python tools like xarray and dask. Since active development has largely ceased, its suitability  
 118 for new projects, such as CDR simulations, is limited.

119 Tools from other modeling communities cannot simply be adopted, since each ocean model  
 120 has distinct structural requirements. For example, the regional-mom6 package ([Barnes et al.,](#)

<sup>121</sup> 2024), developed for the Modular Ocean Model v6 (MOM6) (Adcroft et al., 2019), cannot be  
<sup>122</sup> used to generate ROMS inputs, because ROMS employs a terrain-following vertical coordinate  
<sup>123</sup> system that requires a specialized vertical regridding approach, whereas MOM6 accepts inputs  
<sup>124</sup> on arbitrary depth levels and does not require vertical regridding at all. Several other differences  
<sup>125</sup> further prevent cross-compatibility. Together, these limitations underscored the need for a  
<sup>126</sup> modern, maintainable, and reproducible tool designed specifically for ROMS.<sup>1</sup>

<sup>127</sup> ROMS-Tools was developed to meet this need. It draws on the legacy of the MATLAB  
<sup>128</sup> preprocessing scripts developed at UCLA (Molemaker, 2024), which encapsulate decades of  
<sup>129</sup> expertise in configuring regional ocean model inputs. While many of the core algorithms and  
<sup>130</sup> design principles are retained, ROMS-Tools provides an open-source Python implementation of  
<sup>131</sup> these MATLAB tools using an object-oriented programming paradigm. This implementation  
<sup>132</sup> supports a modern workflow with high-level API calls, improving reproducibility, minimizing  
<sup>133</sup> user errors, and allowing easy extension to new features, forcing datasets, and use cases. In  
<sup>134</sup> some cases, ROMS-Tools diverges from the MATLAB implementation to take advantage of  
<sup>135</sup> new methods or better integration with the modern Python ecosystem. By streamlining input  
<sup>136</sup> generation and analysis, ROMS-Tools reduces technical overhead, lowers the barrier to entry,  
<sup>137</sup> and enables scientists to focus on research rather than data preparation.

## <sup>138</sup> Software Design

<sup>139</sup> ROMS-Tools is designed to balance **ease of use, flexibility, reproducibility, and scalability** by  
<sup>140</sup> combining high-level user interfaces with a modular, extensible architecture.

### <sup>141</sup> Lessons from MATLAB Tools

<sup>142</sup> The legacy MATLAB preprocessing scripts were powerful but required users to edit source  
<sup>143</sup> code directly to configure simulations. This workflow led to frequent errors for new users,  
<sup>144</sup> made it difficult to track completed steps, and limited reproducibility. ROMS-Tools addresses  
<sup>145</sup> these issues with **high-level API calls**, automated error-prone steps, and explicit workflow state  
<sup>146</sup> management via YAML.

### <sup>147</sup> Design Trade-Offs

<sup>148</sup> A central design trade-off in ROMS-Tools is between **user control** and **automation**. Rather  
<sup>149</sup> than enforcing a fixed workflow, the package exposes key choices such as physical options  
<sup>150</sup> (e.g., corrections for radiation or wind), interpolation and fill methods, and computational  
<sup>151</sup> backends. This approach contrasts with opinionated frameworks that fix defaults and directory  
<sup>152</sup> structures to maximize automation. While users must make explicit decisions, some steps remain  
<sup>153</sup> automated to prevent errors. For example, bathymetry smoothing is applied automatically using  
<sup>154</sup> a fixed, non-tunable parameter, since insufficient or omitted smoothing can crash simulations  
<sup>155</sup> due to pressure gradient errors. This design choice directly addresses issues new users faced in  
<sup>156</sup> the MATLAB scripts, and balances **flexibility** and **safety**, enabling transparent experimentation  
<sup>157</sup> without exposing users to avoidable pitfalls.

<sup>158</sup> Another key design consideration is balancing **modular, incremental workflow steps**  
<sup>159</sup> with **reproducibility**. ROMS-Tools organizes tasks (such as creating InitialConditions,  
<sup>160</sup> BoundaryForcing, and SurfaceForcing) into small, composable components that can be  
<sup>161</sup> executed, saved, and revisited independently, rather than following a monolithic, fixed  
<sup>162</sup> workflow. All components depend on the Grid, but once it is created, the remaining objects  
<sup>163</sup> are independent. This modular approach avoids unnecessary recomputation when only some  
<sup>164</sup> inputs change but requires careful tracking of workflow state. To ensure reproducibility,  
<sup>165</sup> all configuration choices are stored in compact, text-based YAML files. These files are

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

166 version-controllable, easy to share, and eliminate the need to transfer large model input  
167 NetCDF datasets. By explicitly tracking workflow state, this design overcomes a key limitation  
168 of the MATLAB scripts and helps users manage experiments more reliably.

## 169 Architecture

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

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

## 187 Computational and Data Model Choices

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

## 195 Research Impact Statement

196 ROMS-Tools is used by two primary research communities. First, regional ocean modelers  
197 use it to generate reproducible input datasets for ROMS simulations; external users include  
198 researchers at **PNNL**, **WHOI**, and **UCLA**. Second, researchers in the ocean-based carbon  
199 dioxide removal (CDR) community use ROMS-Tools to configure reproducible ROMS-MARBL  
200 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  
201 or engaged with the project through GitHub or offline discussions. Several manuscripts from  
202 these communities are currently in preparation.

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

207 Additional evidence of community uptake comes from public usage metrics. At the time of  
208 writing, the GitHub repository shows **119 unique cloners in the past 14 days**, with stars  
209 from users at institutions including the University of Waikato, NCAR, University of Maryland,  
210 National Oceanography Centre, McGill University, UC Santa Cruz, and others. Distribution  
211 statistics indicate **over 3,100 conda-forge downloads in the past six months**, including **68  
212 downloads of the most recent release (v3.3.0)**, and **more than 48,000 total PyPI downloads**.

<sup>213</sup> PyPI counts include automated continuous integration (CI) usage by ROMS-Tools, in addition  
<sup>214</sup> to direct user installations. In contrast, conda-forge downloads of v3.3.0 reflect exclusively  
<sup>215</sup> human-initiated installs, as C-Star's CI workflows currently pin pre-v3.3.0 releases of ROMS-  
<sup>216</sup> Tools.

## <sup>217</sup> AI Usage Disclosure

<sup>218</sup> Generative AI tools were used to help write docstrings, develop tests, and improve the clarity  
<sup>219</sup> and readability of both the ROMS-Tools documentation and manuscript text. All AI-assisted  
<sup>220</sup> content was reviewed and verified by the authors for technical accuracy and correctness.

## <sup>221</sup> Acknowledgements

<sup>222</sup> Development of ROMS-Tools has been supported by ARPA-E (DE-AR0001838) and  
<sup>223</sup> philanthropic donations to [C]Worthy from the Grantham Foundation for the Environment,  
<sup>224</sup> the Chan Zuckerberg Initiative, Founders Pledge, and the Ocean Resilience Climate Alliance.

## <sup>225</sup> References

- <sup>226</sup> Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., Dunne, J.  
<sup>227</sup> P., Griffies, S. M., Hallberg, R., Harrison, M. J., Held, I. M., Jansen, M. F., John, J. G.,  
<sup>228</sup> Krasting, J. P., Langenhorst, A. R., Legg, S., Liang, Z., McHugh, C., Radhakrishnan,  
<sup>229</sup> A., ... Zhang, R. (2019). The GFDL Global Ocean and Sea Ice Model OM4.0: Model  
<sup>230</sup> Description and Simulation Features. *Journal of Advances in Modeling Earth Systems*,  
<sup>231</sup> 11(10), 3167–3211. <https://doi.org/10.1029/2019MS001726>
- <sup>232</sup> Arango, H., & contributors. (2024). Rutgers ROMS. In *GitHub repository*. GitHub. <https://github.com/myroms/roms>
- <sup>234</sup> Barnes, A. J., Constantinou, N. C., Gibson, A. H., Kiss, A. E., Chapman, C., Reilly, J.,  
<sup>235</sup> Bhagtani, D., & Yang, L. (2024). Regional-mom6: A Python package for automatic  
<sup>236</sup> 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>
- <sup>238</sup> Bell, N., Olson, L. N., Schroder, J., & Southworth, B. (2023). PyAMG: Algebraic multigrid  
<sup>239</sup> solvers in python. *Journal of Open Source Software*, 8(87), 5495. <https://doi.org/10.21105/joss.05495>
- <sup>241</sup> Busecke, J., & contributors. (2025). Xgcm. In *GitHub repository*. GitHub. <https://github.com/xgcm/xgcm>
- <sup>243</sup> 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](https://journals.ametsoc.org/view/journals/hydr/3/6/1525-7541_2002_003_0660_eofdfc_2_0_co_2.xml)
- <sup>246</sup> Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. <http://dask.pydata.org>
- <sup>248</sup> Egbert, G. D., & Erofeeva, S. Y. (2002). Efficient Inverse Modeling of Barotropic Ocean  
<sup>249</sup> 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)
- <sup>251</sup> Garcia, H. E., Boyer, T. P., Baranova, O. K., Locarnini, R. A., Mishonov, A. V., Grodsky,  
<sup>252</sup> A., Paver, C. R., Weathers, K. W., Smolyar, I. V., Reagan, J. R., Seidov, D., & Zweng,  
<sup>253</sup> M. M. (2019). *World ocean atlas 2018: Product documentation* (A. Mishonov, Ed.).  
<sup>254</sup> NOAA/NCEI.

- 255 Hamilton, D. S., Perron, M. M. G., Bond, T. C., Bowie, A. R., Buchholz, R. R., Guieu, C., Ito,  
 256 A., Maenhaut, W., Myriokefalitakis, S., Olgun, N., Rathod, S. D., Schepanski, K., Tagliabue,  
 257 A., Wagner, R., & Mahowald, N. M. (2022). Earth, Wind, Fire, and Pollution: Aerosol  
 258 Nutrient Sources and Impacts on Ocean Biogeochemistry. *Annual Review of Marine Science*,  
 259 14(Volume 14, 2022), 303–330. <https://doi.org/10.1146/annurev-marine-031921-013612>
- 260 Hauser, M., Spring, A., Busecke, J., Driel, M. van, Lorenz, R., & readthedocs-assistant.  
 261 (2024). *Regionmask/regionmask: Version 0.12.1*. Zenodo. <https://doi.org/10.5281/zenodo.10849860>
- 263 Hedstrom, K., & contributors. (2023). Pyroms. In *GitHub repository*. GitHub. <https://github.com/ESMG/pyroms>
- 265 Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas,  
 266 J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellán,  
 267 X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., ... Thépaut, J.-N.  
 268 (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*,  
 269 146(730), 1999–2049. [https://doi.org/https://doi.org/10.1002/qj.3803](https://doi.org/10.1002/qj.3803)
- 270 Hoyer, S., & Hamman, J. (2017). Xarray: ND labeled arrays and datasets in python. *Journal  
 271 of Open Research Software*, 5(1). <https://doi.org/10.5334/jors.148>
- 272 Huang, Y., Tagliabue, A., & Cassar, N. (2022). Data-Driven Modeling of Dissolved Iron in the  
 273 Global Ocean. *Frontiers in Marine Science*, 9. <https://doi.org/10.3389/fmars.2022.837183>
- 274 Kok, J. F., Adebiyi, A. A., Albani, S., Balkanski, Y., Checa-Garcia, R., Chin, M., Colarco,  
 275 P. R., Hamilton, D. S., Huang, Y., Ito, A., Klose, M., Leung, D. M., Li, L., Mahowald,  
 276 N. M., Miller, R. L., Obiso, V., Pérez García-Pando, C., Rocha-Lima, A., Wan, J. S., &  
 277 Whicker, C. A. (2021). Improved representation of the global dust cycle using observational  
 278 constraints on dust properties and abundance. *Atmospheric Chemistry and Physics*, 21(10),  
 279 8127–8167. <https://doi.org/10.5194/acp-21-8127-2021>
- 280 Landschützer, P., Gruber, N., & Bakker, D. C. E. (2016). Decadal variations and trends  
 281 of the global ocean carbon sink. *Global Biogeochemical Cycles*, 30(10), 1396–1417.  
 282 <https://doi.org/10.1002/2015GB005359>
- 283 Lauvset, S. K., Key, R. M., Olsen, A., Heuven, S. van, Velo, A., Lin, X., Schirnick, C., Kozyr,  
 284 A., Tanhua, T., Hoppema, M., Jutterström, S., Steinfeldt, R., Jeansson, E., Ishii, M.,  
 285 Perez, F. F., Suzuki, T., & Watelet, S. (2016). A new global interior ocean mapped  
 286 climatology: The  $1^\circ \times 1^\circ$  GLODAP version 2. *Earth System Science Data*, 8(2), 325–340.  
 287 <https://doi.org/https://doi.org/10.5194/essd-8-325-2016>
- 288 Lellouche, E., Jean-Michel Greiner, Bourdallé-Badie, R., Garric, G., Melet, A., Drévillon,  
 289 M., Bricaud, C., Hamon, M., Le Galloudec, O., Regnier, C., Candela, T., Testut, C.-E.,  
 290 Gasparin, F., Ruggiero, G., Benkiran, M., Drillet, Y., & Le Traon, P.-Y. (2021). The  
 291 Copernicus Global  $1/12^\circ$  Oceanic and Sea Ice GLORYS12 Reanalysis. *Frontiers in Earth  
 292 Science*, 9. <https://doi.org/10.3389/feart.2021.698876>
- 293 Long, M. C., Moore, J. K., Lindsay, K., Levy, M., Doney, S. C., Luo, J. Y., Krumhardt, K.  
 294 M., Letscher, R. T., Grover, M., & Sylvester, Z. T. (2021). *Simulations With the Marine  
 295 Biogeochemistry Library (MARBL)*. <https://doi.org/10.1029/2021MS002647>
- 296 Molemaker, J. (2024). UCLA MATLAB tools. In *GitHub repository*. GitHub. <https://github.com/nmolem/ucla-tools>
- 298 Molemaker, J., & contributors. (2025a). UCLA-ROMS. In *GitHub repository*. GitHub.  
 299 <https://github.com/CESR-lab/ucla-roms>
- 300 Molemaker, J., & contributors. (2025b). UCLA-ROMS. In *GitHub repository*. GitHub.  
 301 <https://github.com/CWorthy-ocean/ucla-roms>
- 302 Shchepetkin, A. F., & McWilliams, J. C. (2005). The regional oceanic modeling system

- 303 (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model.  
304 *Ocean Modelling*, 9(4), 347–404. <https://doi.org/10.1016/j.ocemod.2004.08.002>
- 305 Stephenson, D., & contributors. (2025). *C-star*. [C]Worthy. <https://github.com/CWorthy-ocean/C-Star>
- 307 Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., & Wessel, P. (2019).  
308 Global Bathymetry and Topography at 15 Arc Sec: SRTM15+. *Earth and Space Science*,  
309 6(10), 1847–1864. <https://doi.org/10.1029/2019EA000658>
- 310 Wessel, P., & Smith, W. H. F. (1996). A global, self-consistent, hierarchical, high-resolution  
311 shoreline database. *Journal of Geophysical Research: Solid Earth*, 101(B4), 8741–8743.  
312 <https://doi.org/10.1029/96JB00104>
- 313 Yang, S., Chang, B. X., Warner, M. J., Weber, T. S., Bourbonnais, A. M., Santoro, A. E.,  
314 Kock, A., Sonnerup, R. E., Bullister, J. L., Wilson, S. T., & Bianchi, D. (2020). Global  
315 reconstruction reduces the uncertainty of oceanic nitrous oxide emissions and reveals a  
316 vigorous seasonal cycle. *Proceedings of the National Academy of Sciences of the United  
317 States of America*, 117(22), 11954–11960. <https://doi.org/10.1073/pnas.1921914117>
- 318 Yeager, S. G., Rosenbloom, N., Glanville, A. A., Wu, X., Simpson, I., Li, H., Molina, M.  
319 J., Krumhardt, K., Mogen, S., Lindsay, K., Lombardozzi, D., Wieder, W., Kim, W. M.,  
320 Richter, J. H., Long, M., Danabasoglu, G., Bailey, D., Holland, M., Lovenduski, N., ...  
321 King, T. (2022). The Seasonal-to-Multiyear Large Ensemble (SMYLE) prediction system  
322 using the Community Earth System Model version 2. *Geoscientific Model Development*,  
323 15(16), 6451–6493. <https://doi.org/10.5194/gmd-15-6451-2022>
- 324 Zhuang, J., Dussin, R., Huard, D., Bourgault, P., Banhirwe, A., Raynaud, S., Malevich,  
325 B., Schupfner, M., Filipe, Levang, S., Gauthier, C., Jüling, A., Almansi, M., Scott,  
326 R., RondeauG, Rasp, S., Smith, T. J., Stachelek, J., Plough, M., & Li, X. (2023).  
327 xESMF: Universal regridder for geospatial data. In *GitHub repository*. Zenodo. <https://doi.org/10.5281/zenodo.4294774>