




Seaduck: A python package for Eulerian and Lagrangian interpolation on ocean datasets

Wenrui Jiang ¹, Thomas W. N. Haine ¹, and Mattia Almansi ²

¹ Department of Earth and Planetary Sciences, The Johns Hopkins University ² B-Open Solutions s.r.l, Italy

DOI: [10.21105/joss.05967](https://doi.org/10.21105/joss.05967)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Kristen Thyng](#) 

Reviewers:

- [@pnsaevik](#)
- [@syedhamidali](#)

Submitted: 02 August 2023

Published: 19 December 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Numerical simulations of the Earth's oceans are becoming more realistic and sophisticated. Their complex layout and sheer volume make it difficult for researchers to access and understand these data, however. Additionally, most ocean models, mostly finite-volume models, compute and calculate spatially-integrated properties, such as grid-cell averaged temperature or wall-integrated mass flux. On the other hand, in-situ oceanographic observations are effectively collected at points in space and time. This fundamental difference makes the comparison between observations and results from numerical simulation difficult.

In this work, we present seaduck, a Python package that can perform both Eulerian interpolation and Lagrangian particle simulation on generic ocean datasets with good performance and scalability. This package accesses numerical datasets from the perspective of space-time points. It automatically navigates complex dataset layouts (grid topologies) and transforms discrete information into continuous fields. The values and derivatives of those fields can be accessed at any point in the 3+1 dimensional space-time domain defined by the user. For example, users can compute oceanographic properties as Eulerian timeseries (at fixed points in space), like from moored oceanographic current meters, or as Lagrangian timeseries (drifting with the current), like from oceanographic floats

Statement of need

The seaduck package is different from other ocean analytical tools (e.g., oceanspy ([Almansi et al., 2019](#))) because it accesses the circulation model data from the perspective of an arbitrary space-time point. Users define the points of interest using longitude, latitude, depth, and time. The package then reads necessary information from nearby model grid points and constructs the continuous (scalar or vector) field around the points. The index lookup and space-time interpolation involved in this process is done efficiently with `scipy.spatial.cKDtree` ([Virtanen et al., 2020](#)) and numba ([Lam et al., 2015](#)) compiled code, respectively. As the points can be defined arbitrarily in the model domain, accessing discrete numerical output feels to the user like retrieving values from a continuous field, despite the complex model grid.

The points can be stationary (fixed in space, or Eulerian) or be advected by a vector velocity field (Lagrangian). Most Lagrangian particle packages (e.g., Forget ([2021](#))) compute particle trajectories by solving the initial value problem numerically. Instead, seaduck uses efficient, accurate, mass-conserving analytic formulae, which assumes a step-wise steady velocity field similar to that used by TRACMASS ([Aldama-Campino et al., 2020](#)). The Lagrangian advection code is largely numba compiled, and the total amount of computation is less than solving the problem numerically. The Lagrangian particle functionality is based on the above-mentioned interpolation utilities, thus, it automatically navigates the complex topology of numerical ocean models.

Seaduck's interpolation functionality is similar to other interpolation tools (e.g., `scipy` (Virtanen et al., 2020) and `xESMF` (Zhuang et al., 2023)). Seaduck only supports interpolation using neighboring grid points, but users control the properties of a hierarchy of interpolation kernels; which is an advantage. These properties include: (1) The space-time shape of the interpolation kernel(s), which defines which neighboring points are used, and therefore how the continuous field is estimated. (2) The kernel weight function, which allows users to calculate generic linear operations on the data beyond simple interpolation, such as differentiation and smoothing. (3) The hierarchy of kernels near land-masked points.

With the above functionality, seaduck can accomplish many common tasks in ocean model data analysis, including interpolation, regridding, and Lagrangian particle simulation. Less common tasks are also possible, such as interpolation in Lagrangian label space, and analysis of tracer budgets along Lagrangian trajectories. We also strive to make seaduck an accessible education tool by creating a very simple high-level default interface, which is intended for people with little programming background, and for people who want to quickly try the tool.

Usage Examples

While some usage examples are presented here, many more can be found in the documentation for seaduck (<https://macekuailv.github.io/seaduck/>). The notebooks of the following examples run on SciServer (Taghizadeh-Popp et al., 2020), an openly available cloud compute resource for scientific data analysis. A supplementary GitHub repository (https://github.com/MaceKuailv/seaduck_sciserver_notebook) holds all SciServer notebooks, and is being continuously maintained.

Interpolation / regridding

As an example of seaduck's interpolation/regridding functionality, consider a realistic simulation of the Kangerdlugssuaq Fjord, which is in east Greenland (Fraser et al., 2018). The simulation uses the MITgcm (Marshall et al., 1997) with uneven grid spacing such that grid cells within the fjord are much more densely packed than elsewhere. The goal is to interpolate, and hence regrid, the sea surface height field, to a uniform grid spacing in the southern part of the domain. In Figure. 1, the coherent patch between 66.5 N and 67 N is a very dense scatter plot of the interpolated field. The close agreement between the interpolated and output value can be clearly seen in Figure. 1. The interpolated field also remains smooth near strong gradients and land boundaries.

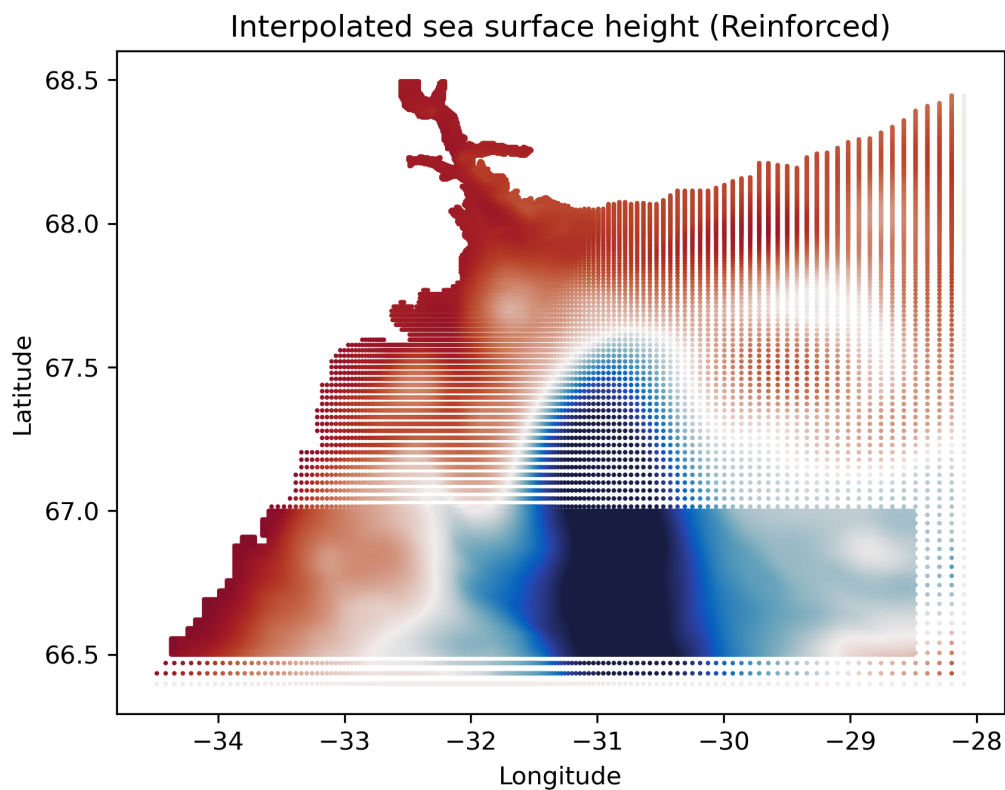


Figure 1: Scatterplot with colors showing the sea surface height value near Kangerdlugssuaq Fjord defined in the model and interpolated by seaduck.

Global particle simulation on LLC4320

In this example, a stationary, surface slice of the LLC4320 (Rocha et al., 2016) simulation is used. LLC4320 is a kilometer-scale model of the global ocean circulation with complex topology. 150,000 Lagrangian particles are released randomly and evenly on the globe, and seaduck computes their trajectories for 30 days. Figure. 2 shows the particle trajectories for the northern hemisphere, which contains around 10^8 velocity points. The colors denote the current speed. This simulation takes about an hour to run on SciServer (Taghizadeh-Popp et al., 2020).

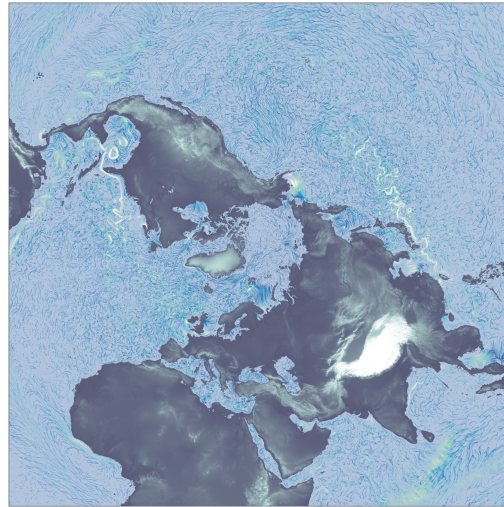


Figure 2: Particle streaklines of particle advected by stationary 2D slice of the LLC4320 simulation. Colors denote the current speed.

Acknowledgments

The authors thank Erik van Sebille and Alex Szlay for enlightening discussions during the development of the package.

References

- Aldama-Campino, A., Döös, K., Kjellsson, J., & Jönsson, B. (2020). *TRACMASS: Formal release of version 7.0 (v7.0-beta)*. Zenodo. <https://doi.org/10.5281/zenodo.4337926>
- Almansi, M., Gelderloos, R., Haine, T. W. N., Saberi, A., & Siddiqui, A. H. (2019). OceanSpy: A Python package to facilitate ocean model data analysis and visualization. *Journal of Open Source Software*, 4, 1506. <https://doi.org/10.21105/JOSS.01506>
- Delandmeter, P., & Sebille, E. V. (2019). The Parcels v2.0 Lagrangian framework: New field interpolation schemes. *Geoscientific Model Development*, 12, 3571–3584. <https://doi.org/10.5194/GMD-12-3571-2019>
- Forget, G. (2021). IndividualDisplacements.jl: A Julia package to simulate and study particle displacements within the climate system. *Journal of Open Source Software*, 6, 2813. <https://doi.org/10.21105/JOSS.02813>
- Fraser, N. J., Inall, M. E., Magaldi, M. G., Haine, T. W. N., & Jones, S. C. (2018). Wintertime Fjord-Shelf Interaction and Ice Sheet Melting in Southeast Greenland. *Journal of Geophysical Research: Oceans*, 123, 9156–9177. <https://doi.org/10.1029/2018JC014435>
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based Python JIT Compiler. *Proceedings of LLVM-HPC 2015: 2nd Workshop on the LLVM Compiler Infrastructure in HPC - Held in Conjunction with SC 2015: The International Conference for High Performance Computing, Networking, Storage and Analysis, 2015-January*. <https://doi.org/10.1145/2833157.2833162>
- Marshall, J., Adcroft, A., Hill, C., Perelman, L., & Heisey, C. (1997). A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102, 5753–5766. <https://doi.org/10.1029/96JC02775>

- Rocha, C. B., Chereskin, T. K., Gille, S. T., & Menemenlis, D. (2016). Mesoscale to Submesoscale Wavenumber Spectra in Drake Passage. *Journal of Physical Oceanography*, 46, 601–620. <https://doi.org/10.1175/JPO-D-15-0087.1>
- Taghizadeh-Popp, M., Kim, J. W., Lemson, G., Medvedev, D., Raddick, M. J., Szalay, A. S., Thakar, A. R., Booker, J., Chhetri, C., Dobos, L., & Rippin, M. (2020). SciServer: A Science Platform for Astronomy and Beyond. *Astronomy and Computing*, 33. <https://doi.org/10.1016/j.ascom.2020.100412>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods* 2020 17:3, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Zhuang, J., dussin, raphael, Huard, D., Bourgault, P., Banihirwe, A., Raynaud, S., Malevich, B., Schupfner, M., Filipe, Levang, S., Gauthier, C., Jüling, A., Almansi, M., RichardScottOZ, RondeauG, Rasp, S., Smith, T. J., Stachelek, J., Plough, M., ... Li, X. (2023). *Pangeo-data/xESMF: v0.8.2* (Version v0.8.2). Zenodo. <https://doi.org/10.5281/zenodo.8356796>