The Journal of Open Source Software

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.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 02 August 2023
Published: unpublished

Summary

Numerical simulations of the Earth's oceans are becoming more realistic and sophisticated. Their complex layout and shear 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 and Lagrangian interpolation 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 to continuous fields. The values and derivatives of those fields can be accessed at any point in the domain defined by the user. Similar to fixed and drifting observational oceanographic instrument platforms, the points can be either stationary (Eulerian) or advected by the flow (Lagrangian).

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 [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 provides highly-customizable interpolation methods for both Eulerian and Lagrangian points. Users can control all the properties of a hierarchy of kernels, including: (1) The shape of the interpolation kernel(s) in both spatial and temporal dimensions, which defines which neighboring points are used, and therefore how the continuous field is estimated. (2) The interpolation weight function, which allows users to calculate generic linear operations on the data, such as differentiation and smoothing, in all four dimensions. The hierarchy of kernels controls behaviour near land-masked points. Specifically, the hierarchy consists of successively more compact kernels that are used depending on the proximity of land 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). This is an MITgcm (Marshall et al., 1997) simulation 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, η, to a uniform grid spacing in the southern part of the domain. In Fig. 1, the coherent patch between 66.5 N and 67 N is a very dense scatter plot of the interpolated value where neighboring points are connected together. The rest of this plot is scatter plot of model output at center grid points. The close agreement between the interpolated and output value can be clearly seen in Fig. 1. The interpolation also remains smooth near strong gradient and land boundaries.

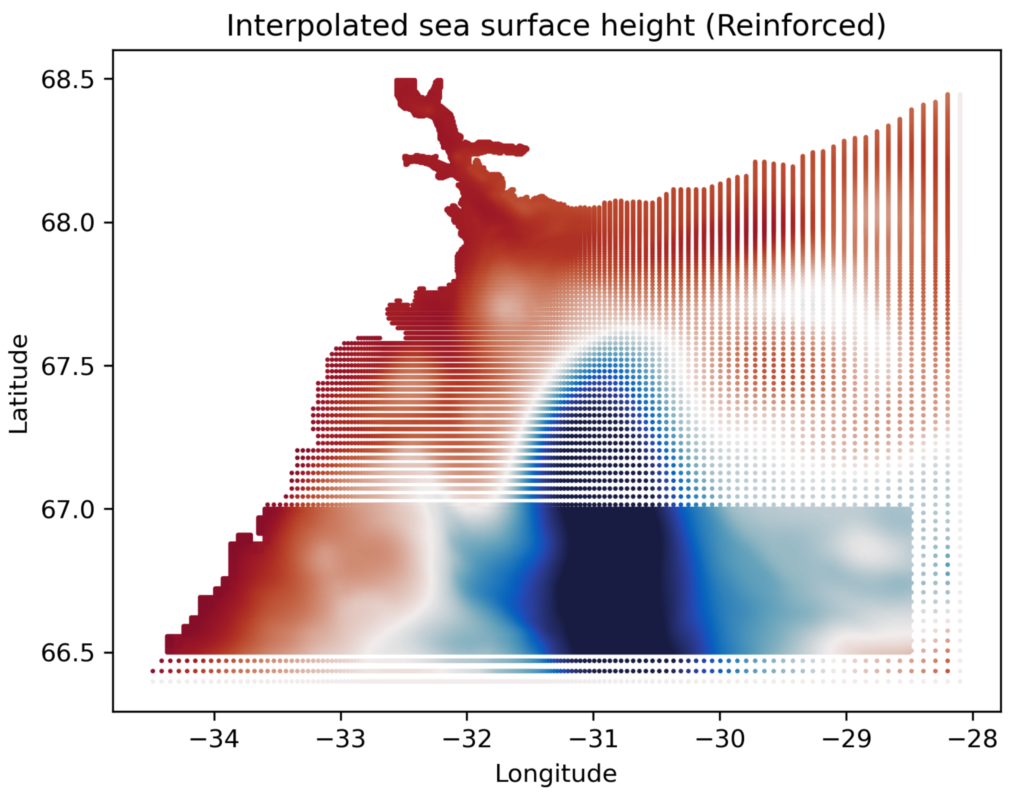

Figure 1: Fig.1 Scatterplot with colors showing the sea surface height value near Kangerdlugssuaq Fjord defined in the model and interpolated by seaduck.

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. Fig. 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).

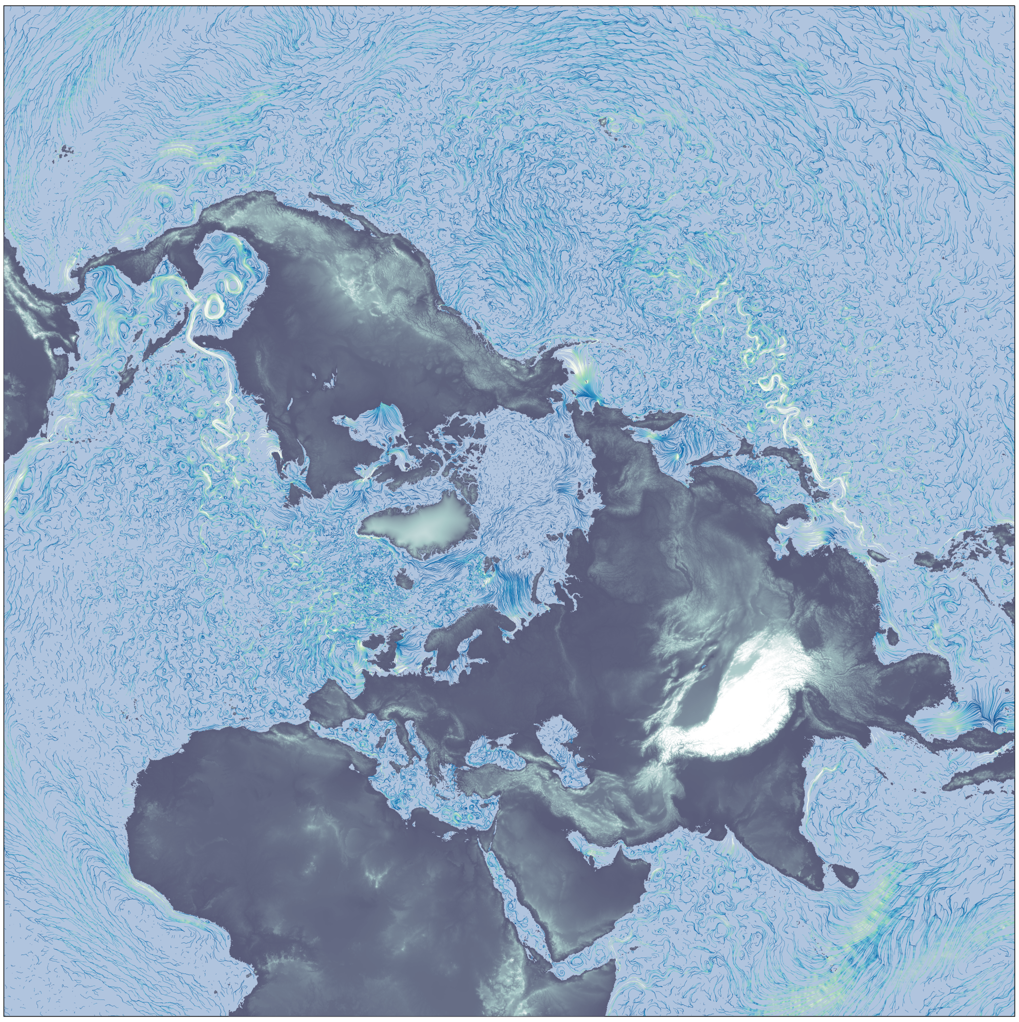

Figure 2: Fig 2. Streaklines of particle advected by stationary 2D slice of the LLC4320 simulation. Colors denote the current speed.

Acknowledgements

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

References

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

Campino, A. A., Döös, K., Berglund, S., Dey, D., Kjellsson, J., Jonsson, B., Campino, A. A., Döös, K., Berglund, S., Dey, D., Kjellsson, J., & Jonsson, B. (2020). TRACMASS - a mass conserving trajectory code for ocean and atmosphere general circulation models. *EGUGA*, 6984. https://doi.org/10.5194/EGUSPHERE-EGU2020-6984

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