Overview:

This manuscript introduces PyESPERv1.01.01, a Python-based implementation of empirical seawater property estimation routines (ESPERs), previously developed and made available only in MATLAB by author Carter. These routines estimate core seawater biogeochemical properties —such as total alkalinity, dissolved inorganic carbon, total pH, nitrate, phosphate, silicate, and oxygen—using inputs like geographic coordinates, depth, salinity, and up to four additional predictors (e.g., temperature and biogeochemical information). Two statistical algorithms, a locally interpolated regression (LIR) and a neural network (NN) estimation are averaged to produce a best estimate.

By transitioning ESPERs to Python, the authors enhance accessibility for the scientific community, as Python is an open-source language widely used in oceanographic research. The study also documents modifications made to reduce discrepancies between the Python and MATLAB implementations and evaluates the disagreements between the methods. The implementation also updates underlying datasets using Global Ocean Data Analysis Project (GLODAPv2.2022) dataset and addresses a couple minor issues with the original code.

The work submitted here will be a valuable resource to the community and required a large amount of detailed assessment and validation. I recommend publication after consideration and edits based on the range of suggestions from reviewers.

We thank you for the constructive feedback.

General Feedback:

This work will have substantial impact on the field of ocean biogeochemistry and carbon cycling, as well as serve as an important resource for characterizing baseline inorganic carbon chemistry in the context of marine carbon dioxide removal (mCDR) activities. While the concepts and ideas are not new, and build on the original ESPER, transitioning this tool to Python will broaden accessibility and encourage further scientific inquiry and discovery.

We concur and appreciate the feedback

The calculations/algorithms used are described in precise and comprehensive detail. Care is taken to evaluate uncertainty, as well as assess internal consistency within the inorganic carbon system.

Thank you for the feedback.

I commend the authors for making the code available on GitHub through a Jupyter Notebook example. However, two improvements would make this much more accessible to the community: (1) I am very surprised the performance was so much worse with python relative to Matlab. Profiling the code to see where the slowdown is likely could lead to massive performance improvements with some refactoring. (2) providing the code in a pip or conda installable package would make it much more reproducible and less error prone.

We appreciate the suggestions for improvements. (1) We have indeed profiled the code and found the slowdown to be during interpolations. This was greatly improved by packaging it, which we are near completion of. (2) We are nearing completion of the pip installable package also, which should be ready by the time of formal publication. Please check the GitHub page for the package.

The overall presentation is clear, although somewhat dense. I appreciate the detailed documentation of methodology though.

Thank you for the comments.

Minor Feedback:

Do you have insight into why DIC and pH seems to have considerably larger python-Matlab differences?

Yes, this is because the current methods for estimation of contributions of anthropogenic carbon (C_{ant}) to DIC and pH involves interpolations, which did not match well between Python and MATLAB versions. Other estimated properties (e.g., TA, nitrate, phosphate, silicate, and oxygen) do not require estimates of C_{ant}. Please see the following modified explanation to help clarify this.

L. 347-349. The largest relative disagreements were found for DIC and pH_T, though these disagreements remained small relative to measurement uncertainties. These minor offsets are attributed to the programming language differences in the interpolation of the C_{ant} adjustment, which is only applied to these two properties.

L145: For clarification – NN functions were translated from scratch? Was this compared to using something 'out of the box' like pytorch? It would be interesting to compare both reproducibility and performance.

We did translate the neural networks from scratch because we wanted an exact replica (to the best of our ability). The translation (PyESPER_NN) indeed did replicate ESPER_NN results to within machine precision. We feel that it is unlikely that independently trained neural networks would provide as similar results as our present method, but do not rule out the possibility of providing a "python-trained" option in future ESPER updates.

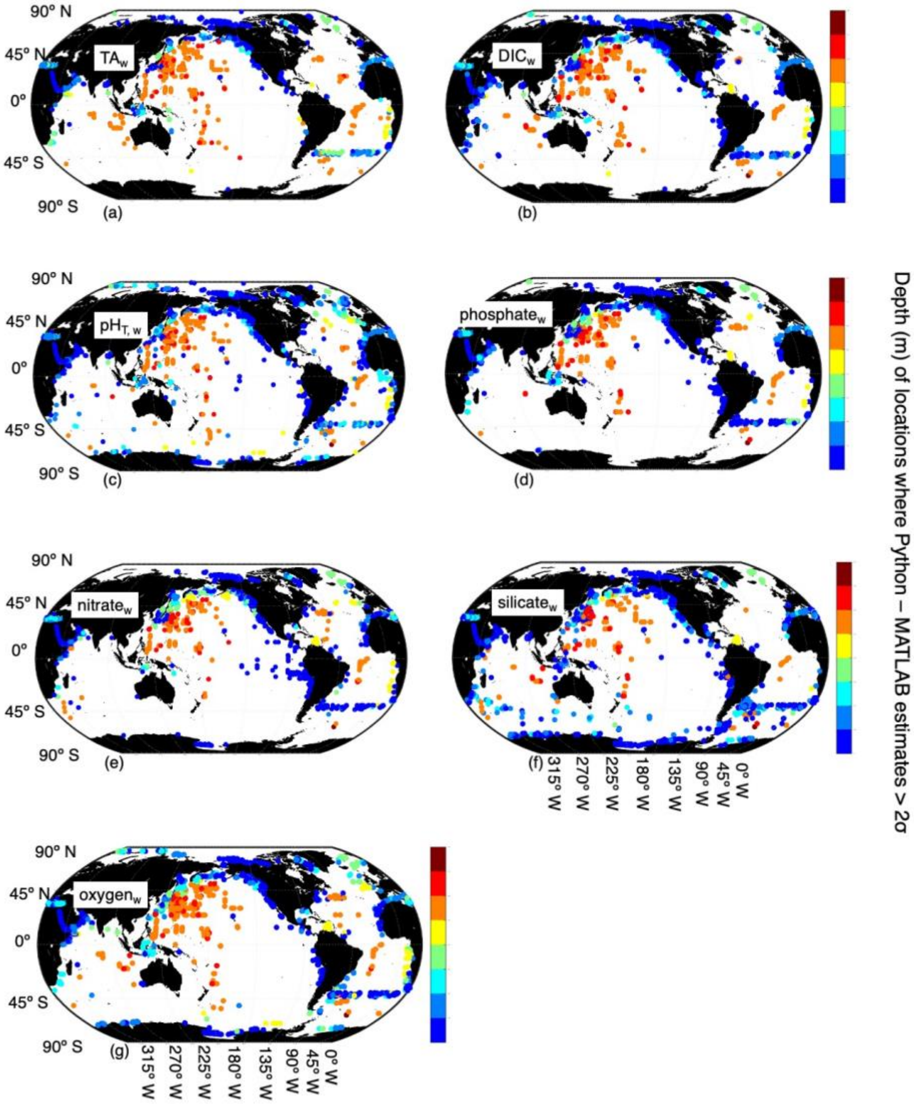
Figure B2: There seems to be structure in the large mismatches – For example in the North Pacific along margins, and perhaps on an A10 GO-SHIP line. Could you add discussion on this? Does this point towards potentially a data problem with one cruise?

This is true that there are areas where the mismatch are greater (although not for one particular cruise). These areas align with places where the "edges" of our interpolated grid occur. This is caused by differences in interpolation and extrapolation between the two coding languages, where interpolating between previously extrapolated areas (in MATLAB) is not a very good reproduction of the MATLAB mathematical method. We have modified the text as follows, to aid with this explanation.

L. 324-330. PyESPER_LIRs were within 2σ (~95% of measurements should fall within this uncertainty level) for most ocean regions, with a few exceptions which occurred predominantly in coastal areas or deep waters near the edges of the original MATLAB grid (Figs. 3 and 4). Spatial patterns in distribution of outliers shown in Fig. 4 appear to reflect locations where more edge-of-grid biogeochemical measurements were collected (e.g., near coasts and in deep waters). Hence, these locations aligned well with places where coefficients were extrapolated in the MATLAB implementation (see Sect. 2.1.1, "*Locally interpolated regressions*"; Figs. 3, 4, and 5; for $_{w}$ Fig. B2 and B3). Within regions where MATLAB was interpolating, far outliers were uncommon (Figs. 3, 4, 5, B2, and B3).

Figure B3: The colorbar should 'depth' but there are no labels or units?

The labels and units appear to the right of the figure (please see below).


90° N
45° N
0°
45° S
90° S
TA$_w$
DIC$_w$
pH$_{T, w}$
phosphate$_w$
nitrate$_w$
silicate$_w$
oxygen$_w$
(a)
(b)
(c)
(d)
(e)
(f)
(g)
315° W
270° W
225° W
180° W
135° W
90° W
45° W
0° W
Depth (m) of locations where Python – MATLAB estimates > 2σ