**Readme for the MATLAB functions ESPER\_LIR(…), ESPER\_NN(…), and ESPER\_Mixed(…)**

These functions estimate seawater properties from other seawater property measurements.

**How to set up:** Running the ESPER estimation routines in MATLAB is a 4-step procedure:

1. Extract all remaining zipped folders, if any.
2. Add all folders onto the current working directory or the MATLAB path.\* Be sure to add the downloaded folder “with subfolders.” The “private” folder should not need to be added if you are adding folders individually by command line.
3. Call the functions according to the instructions below. There are example calculations provided to verify your installation is correct. Full test output is provided as “TestOutput.m.”
4. Please cite the [CSIRO seawater toolbox](https://publications.csiro.au/rpr/download?pid=procite:49de56bc-b045-4cb4-9eeb-f7025bb06f5c&dsid=DS1) and CO2SYS (which are called by ESPER) appropriately in your documentation. The included version of CO2SYS is a renamed distribution (CO2SYS\_ESPER.m) of the code by van Hueven, Pierrot, Rae, Lewis, & Wallace (2011). CO2SYS code is only included because CO2SYS versions can sometimes differ locally.

\**Not sure what the MATLAB path is?* [*Check this out*](https://www.mathworks.com/help/matlab/matlab_env/what-is-the-matlab-search-path.html)*. An easy way get the LIR files on the path (without putting them in your working directory) is to navigate to the folder in the command window, then enter the command “addpath pwd”. You may need to do this again after each MATLAB restart.*

**Questions and Updates:** For questions not addressed by the papers or this readme file, please contact Brendan Carter, [Brendan.carter@gmail.com](mailto:Brendan.carter@gmail.com), and he will get back to you as he is able. Suggestions, bug reports, and queries regarding code updates are welcome. We intend to continue to update algorithms as new quality controlled datasets become available. Please indicate versions if you use routines for any publications (for reproducibility). Contact the corresponding author for updates. Unlike the LIRs, updates will be tracked on Zenodo instead of GitHub.

**Versioning and documentation**: ESPER\_LIR is an update to LIRv2 and is sometimes called LIRv3. ESPER\_NN is a neural network implementation of the LIR code that is inspired by CANYON-B (Bittig et al., 2018), but also somewhat modified. ESPER\_Mixed calls both functions and averages the outputs. The algorithms are described and assessed in a manuscript in preparation for submission at Earth Systems Science Data. The initial release of ESPER\_LIR will be version 3.0 and of ESPER\_NN and ESPER\_Mixed will be version 1.0. The first line of the comments will indicate if the version has been updated (e.g. it will say “version 3.1” and have some update notes).

*LIARv1: Carter et al., 2016, doi: 10.1002/lom3.10087*

*LIARv2, LIPHR, LINR citation:* [*https://doi.org/10.1002/lom3.10232*](https://doi.org/10.1002/lom3.10232)

*LIPR, LISIR, LIOR, first described/used: Carter et al., 2021,* [*https://doi.org/10.1029/2020GB006623*](https://doi.org/10.1029/2020GB006623)

*ESPER\_LIR, ESPER\_NN, and ESPER\_Mixed, submitted.*

*ESPER\_NN is inspired by CANYON-, which also uses neural networks: Bittig et al. 2018:* [*https://doi.org/10.3389/fmars.2018.00328*](https://doi.org/10.3389/fmars.2018.00328)

Notable changes from LIRv2:

1. LIRv2 had separate function calls for each property. LIRv3 uses a single function call. However, the function will only estimate the desired properties, so a new required input indicates which property estimate(s) is (are) desired.
2. To work with a variety of estimate types, the vector specifying which predictors are being provided was modified to use a single key (rather than a different key for each property type). The numbers for each property type have therefore changed.
3. Previously if the ‘Equations’ optional input was not provided, the routine would try to use the provided uncertainty information with all 16 possible equations to provide a single estimate with minimal uncertainty. This proved unintuitive to some users and problematic in a few use cases, so ESPER\_LIR and ESPER\_NN now default to attempting to use all 16 equations in this instance and returning all estimated values.
4. Potential temperature and AOU are no longer valid input predictors. The internal calculations still use these values, but the routines no longer accept them as inputs. This is to ensure that potential temperature and AOU are calculated within the routine using the same equations used for the training data.
5. The OAAdjustTF flag has been removed and the calculation is now always on. If no EstDates input is provided then the routine will report all values specific to January 1st, 2002 (i.e., the date of the Canth data product used as a basis for Canth adjustments: Lauvset et al., 2016).

**Function call explanation:** The function calls are nearly identical for all 3 variants. Once the code is on the Matlab path a detailed explanation of the function call can be found in the code comments and viewed by typing help ESPER\_NN or help ESPER\_LIR or help ESPER\_Mixed into the MATLAB command line and pressing enter/return.

[Estimates,Uncertainties,Coefficients]=ESPER\_LIR(DesiredVariables,OutputCoordinates,PredictorMeasurements,PredictorTypes,...Optional Inputs...)

[Estimates,Uncertainties]=ESPER\_NN(DesiredVariables,OutputCoordinates,PredictorMeasurements,PredictorTypes,...Optional Inputs...)

[Estimates,Uncertainties]=ESPER\_Mixed(DesiredVariables,OutputCoordinates,PredictorMeasurements,PredictorTypes,...Optional Inputs...)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Locally Interpolated Regressions (LIRs) for seawater property

estimation: Estimates seawater properties and estimate uncertainty from

combinations of other parameter measurements. LIRs refers specifically

to code that uses interpolated regressions as opposed to collections of

neural networks (such as CANYON or ESPER\_NN).

Updated 2020.08.19 (v3)

-Combined all LIRs

-Implemented first-principles adjustments for Ocean Acidification

-Implemented regression uncertainty quantification (vs. interpolation)

-Refit to GLODAPv2.2020 (subset)

-Implemented distance-weighting in regressions with wider windows

Documentation and citations:

LIARv1: Carter et al., 2016, doi: 10.1002/lom3.10087

LIARv2, LIPHR, LINR citation: Carter et al., 2018, https://doi.org/10.1002/lom3.10232

LIPR, LISIR, LIOR, first described/used: Carter et al., 2021, https://doi.org/10.1029/2020GB006623

ESPER\_LIR and ESPER\_NN, submitted.

ESPER\_NN is inspired by CANYON-B, which also uses neural networks:

Bittig et al. 2018: https://doi.org/10.3389/fmars.2018.00328

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Input/Output dimensions:

.........................................................................

p: Integer number of desired property estimate types (e.g., TA, pH, NO3-)

n: Integer number of desired estimate locations

e: Integer number of equations used at each location

y: Integer number of parameter measurement types provided by the user.

n\*e: Total number of estimates returned as an n by e array

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Required Inputs:

DesiredVariables:

Specifies which variables will be returned, excepting unitless pH,

all outputs are in umol/kg.

1. Total Titration Seawater Alkalinity (TA)

2. Total Dissolved Inorganic Carbon (DIC)

3. in situ pH on the total scale

4. Phosphate

5. Nitrate

6. Silicate

7. Dissolved Oxygen (O2)

OutputCoordinates (required n by 3 array):

Coordinates at which estimates are desired. The first column should

be longitude (degrees E), the second column should be latitude

(degrees N), and the third column should be depth (m).

PredictorMeasurements (required n by y array):

Parameter measurements that will be used to estimate alkalinity. The

column order (y columns) is arbitrary, but specified by PredictorTypes.

Concentrations should be expressed as micromol per kg seawater unless

PerKgSwTF is set to false in which case they should be expressed as

micromol per L, temperature should be expressed as degrees C, and

salinity should be specified with the unitless convention. NaN

inputs are acceptable, but will lead to NaN estimates for any

equations that depend on that parameter.

PredictorTypes (required 1 by y vector):

Vector indicating which parameter is placed in each column of the

'PredictorMeasurements' input. Note that salinity is required for

all equations. If O2 is provided, then temperature or potential

temperature must also be provided to convert O2 to AOU. For example,

if the first three columns of 'PredictorMeasurements' contain

salinity, silicate, and temperature, then PredictorTypes should equal [1 5

7].

Input Parameter Key:

1. Salinity

2. Temperature

3. Phosphate

4. Nitrate

5. Silicate

6. O2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Optional inputs: All remaining inputs must be specified as sequential

input argument pairs (e.g. "..., 'Equations',[1:16], 'VerboseTF', false,

etc...")

EstDates (Semi-optional n by 1 array or 1 by 1 value, default 2002.0):

A vector of decimal dates for the estimates (e.g. July 1 2020 would

be "2020.5"). If only a single date is supplied that value is used

for all estimates. It is highly recommended that date(s) be provided

for estimates of DIC and pH. This version of the code will accept 1

by n inputs as well.

Equations (optional 1 by e vector, default []):

Vector indicating which equations will be used to estimate

alkalinity. This input also determines the order of the columns in

the 'Estimates' output. If [] is input or the input is not specified

then all 16 equations will be used and only the outputs from the

equation with the lowest uncertainty estimate will be returned.

(S=salinity, Theta=potential temperature, N=nitrate, Si=silicate,

T=temperature, O2=dissolved oxygen molecule... see

'PredictorMeasurements' for units).

...........................................................

Output Equation Key (See below for explanation of A, B, and C):

1. S, T, A, B, C

2. S, T, A, C

3. S, T, B, C

4. S, T, C

5. S, T, A, B

6. S, T, A

7. S, T, B

8. S, T

9. S, A, B, C

10. S, A, C

11. S, B, C

12. S, C

13. S, A, B

14. S, A

15. S, B

16. S

DesiredVar | A B C

\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TA | Nitrate Oxygen Silicate

DIC | Nitrate Oxygen Silicate

pH | Nitrate Oxygen Silicate

phosphate | Nitrate Oxygen Silicate

nitrate | Phosphate Oxygen Silicate

silicate | Phosphate Oxygen Nitrate

O2 | Phosphate Nitrate Silicate

MeasUncerts (Optional n by y array or 1 by y vector, default: [0.003 S,

0.003 degrees C (T or theta), 1% P, 1% O2, 1% Si]: Array of

measurement uncertainties (see 'PredictorMeasurements' for units).

Uncertainties should be presented in order indicated by PredictorTypes.

Providing these estimates will improve ESPERestimate

uncertainties in 'UncertaintyEstimates'. Measurement uncertainties

are a small part of ESPER estimate uncertainties for WOCE-quality

measurements. However, estimate uncertainty scales with measurement

uncertainty, so it is recommended that measurement uncertainties be

specified for sensor measurements. If this optional input argument

is not provided, the default WOCE-quality uncertainty is assumed. If

a 1 by y array is provided then the uncertainty estimates are assumed

to apply uniformly to all input parameter measurements.

pHCalcTF (Optional boolean, default false):

If set to true, LIPHR will recalculate the pH to be a better estimate

of what the seawater pH value would be if calculated from TA and DIC

instead of measured with purified m-cresol dye. This is arguably also

a better estimate of the pH that would be obtained from pre-2011

measurements with impure dyes. See the LIPHR paper for details

PerKgSwTF (Optional boolean, default true):

Many sensors provide measurements in micromol per L (molarity)

instead of micromol per kg seawater. Indicate false if provided

measurements are expressed in molar units (concentrations must be

micromol per L if so). Outputs will remain in molal units

regardless.

VerboseTF (Optional boolean, default true):

Setting this to false will make ESPER\_LIR stop printing updates to

the command line. This behavior can also be permanently disabled

below. Warnings and errors, if any, will be given regardless.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Outputs:

OutputEstimates:

A n by e array of ESPER\_LIR estimates specific to the coordinates and

parameter measurements provided as inputs. Units are micromoles per

kg (equivalent to the deprecated microeq per kg seawater).

UncertaintyEstimates:

A n by e array of uncertainty estimates specific to the coordinates,

parameter measurements, and parameter uncertainties provided.

CoefficientsUsed: (ESPER\_LIR only)

A n by 6 by e array of ESPER\_LIR coefficients specific to the

coordinates and provided as inputs. Units are those of the property

being predicted divided by the property doing the prediction (often

micromol per kg for both) and are provided in the order indicated in

the Output Equation Key above, with NaNs for predictors that are not

used. The first constant is multiplied by 1 in the prediction

equations (i.e., is just an added coefficient).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Missing data: should be indicated with a NaN. A NaN coordinate will

yield NaN estimates for all equations at that coordinate. A NaN

parameter value will yield NaN estimates for all equations that require

that parameter.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Please send questions or related requests to brendan.carter@gmail.com.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Example call:**

The following call can be input into the command line to test your installation of the ESPER code. This call requests estimates of all properties (1 through 7) at 3 locations (100 m depth at 0°N and 0°E, 1000 m depth at the same location, and 100 m depth at 150°W, or -150°E, and 0°N) and provides identical predictor properties for all locations: salinity (35), phosphate (0.5 µmol kg₋1), temperature (5°C), nitrate (10 µmol kg₋1), silicate (20 µmol kg₋1), and oxygen (200 µmol kg₋1). The 4th input tells the code the order of the input property measurements, and it appears slightly out of order because phosphate was given before temperature. The next 2 input argument pairs (4 inputs) are optional. The first limits the output to the three indicated equations and tells the code to give the outputs for equation 16 2nd after equation 1 and before equation 8. The last pair specifies the dates of the requested output estimates. In this case the last estimate is specific to 2030. The code assumes an exponential increase in Canth and is not trustworthy for estimates much beyond 2030… this is just an example calculation.

[a,b]=ESPER\_Mixed([1 2 3 4 5 6 7],[0 0 100;0 0 1000;-150 0 100],[35 0.5 5 10 20 200;35 0.5 5 10 20 200;32 0.5 5 10 20 200],[1 3 2 4 5 6],'Equations',[1 16 8],'EstDates',[1980;2002;2030]);

The full expected output is provided within ESPER\_files.zip as TestOutput.m. For a quick check, enter the code above and then follow it up with:

a.pH(2,2)

…. and you should get the response:

ans =

7.8812

… getting different values? Please contact me so I can verify that this Readme is up to date.