How to set a DOXY\_ADJUSTED\_ERROR in Coriolis NetCDF files

This note explains how the DOXY\_ADJUSTED\_ERROR information should be sent to Coriolis (to [Vincent.Bernard@ifremer.fr](mailto:Vincent.Bernard@ifremer.fr)) so that it will be propagated in Real Time into the profiles files and how the SCIENTIFIC\_CALIB\_COMMENT (of the DOXY parameter) will be set accordingly.

The PI must provide the estimated DOXY\_ADJUSTED\_ERROR of their floats together with the concerned case number.

A case number is defined as “*I*\_*J*” where “*I*” is the estimation method number used (by the PI to determine the DOXY\_ADJUSTED\_ERROR value) and “*J*” the propagation ERROR method number (to be used by the decoder to propagate the error at the profile levels). Three cases are defined so far: “1\_1”, “2\_1” and “3\_1” (see below).

The case number is then used by the decoder: 1- to determine the method used to propagate the provided error; 2- to set the SCIENTIFIC\_CALIB\_COMMENT associated to DOXY parameter.

# List of available propagation error methods

**Case 1 : Error in PPOX**

Propagation ERROR Method # = 1

Description of the method: propagation error of 10mbar by default or provided by PI in mbar

DOXY\_ADJUSTED\_ERROR = [X] µmol/kg is recomputed from CALIB\_RT\_ADJUSTED\_ERROR

# List of available estimation methods

**Case 1\_1:** Adjustment by comparison of in water float data to WOA based on PSAT or PPOX, error in PPOX

Method # = 1

Description of the method: gain estimated from the comparison between in water PSAT or PPOX from float and PSAT or PPOX from WOAat mostinthe upper 20 dbar of the water column. WOA PPOX is computed from WOA PSAT and from TEMP and PSAL float data at the atmospheric pressure of 1 atm.

SCIENTIFIC\_CALIB\_COMMENT = "DOXY\_ADJUSTED is estimated from an adjustment of in water PSAT or PPOX float data at surface by comparison to WOA PSAT climatology or WOA PPOX in using PSATWOA and TEMPand PSALfloat at 1 atm, DOXY\_ADJUSTED\_ERROR is estimated from a PPOX\_ERROR of [xx] mbar"

Propagation ERROR Method # = 1

**Case 2\_1: Adjustment by comparison of in air float data to NCEP reanalysis atmospheric data based on PPOX, error in PPOX**

Method # = 2

Description of the method: gain estimated from the comparison between in air PPOXfloat and PPOXNCEP.

SCIENTIFIC\_CALIB\_COMMENT = "DOXY\_ADJUSTED is estimated from an adjustment of in air PPOX float data by comparison to NCEP reanalysis, DOXY\_ADJUSTED\_ERROR is recomputed from a PPOX\_ERROR = [xx] mbar"

Propagation ERROR Method # = 1

**Case 3\_1: Adjustment based on last valid DM adjustment, error in PPOX**

Method # = 3

Description of the method: gain estimated from the last valid cycle with DM adjustment.

SCIENTIFIC\_CALIB\_COMMENT = "DOXY\_ADJUSTED is estimated from the last valid cycle with DM adjustment, DOXY\_ADJUSTED\_ERROR is recomputed from a PPOX\_ERROR = [xx] mbar"

Propagation ERROR Method # = 1

# Matlab code of “Propagation ERROR Method #1”

## compute\_DOXY\_ADJUSTED

% ------------------------------------------------------------------------------

% Adjust DOXY measurements.

% DOXY\_ADJUSTED is estimated from an adjustment of PPOX\_DOXY at surface on WOA

% climatology.

%

% SYNTAX :

% [o\_DOXY\_ADJUSTED, o\_DOXY\_ADJUSTED\_ERROR] = compute\_DOXY\_ADJUSTED( ...

% a\_PRES, a\_TEMP, a\_PSAL, a\_DOXY, ...

% a\_PRES\_fillValue, a\_TEMP\_fillValue, a\_PSAL\_fillValue, a\_DOXY\_fillValue, ...

% a\_slope, a\_offset, a\_adjError, a\_profOptode)

%

% INPUT PARAMETERS :

% a\_PRES : input PRES data

% a\_TEMP : input TEMP data

% a\_PSAL : input PSAL data

% a\_DOXY : input DOXY data

% a\_PRES\_fillValue : fill value for input PRES data

% a\_TEMP\_fillValue : fill value for input TEMP data

% a\_PSAL\_fillValue : fill value for input PSAL data

% a\_DOXY\_fillValue : fill value for input DOXY data

% a\_DOXY\_fillValue : fill value for input DOXY data

% a\_slope : slope of PPOX\_DOXY adjustment

% a\_offset : slope of PPOX\_DOXY adjustment

% a\_adjError : error on PPOX\_DOXY adjusted values

% a\_profOptode : OPTODE profile structure

%

% OUTPUT PARAMETERS :

% o\_DOXY\_ADJUSTED : output DOXY adjusted data

% o\_DOXY\_ADJUSTED\_ERROR : output error on DOXY adjusted data

%

% EXAMPLES :

%

% SEE ALSO :

% AUTHORS : Jean-Philippe Rannou (Altran)(jean-philippe.rannou@altran.com)

% ------------------------------------------------------------------------------

% RELEASES :

% 07/04/2019 - RNU - creation

% ------------------------------------------------------------------------------

function [o\_DOXY\_ADJUSTED, o\_DOXY\_ADJUSTED\_ERROR] = compute\_DOXY\_ADJUSTED( ...

a\_PRES, a\_TEMP, a\_PSAL, a\_DOXY, ...

a\_PRES\_fillValue, a\_TEMP\_fillValue, a\_PSAL\_fillValue, a\_DOXY\_fillValue, ...

a\_slope, a\_offset, a\_adjError, a\_profOptode)

% output parameters initialization

o\_DOXY\_ADJUSTED = ones(length(a\_DOXY), 1)\*a\_DOXY\_fillValue;

if (~isnan(a\_adjError))

o\_DOXY\_ADJUSTED\_ERROR = ones(length(a\_DOXY), 1)\*a\_DOXY\_fillValue;

else

o\_DOXY\_ADJUSTED\_ERROR = [];

end

% retrieve global coefficient default values

global g\_decArgo\_doxy\_202\_205\_304\_d0;

global g\_decArgo\_doxy\_202\_205\_304\_d1;

global g\_decArgo\_doxy\_202\_205\_304\_d2;

global g\_decArgo\_doxy\_202\_205\_304\_d3;

global g\_decArgo\_doxy\_202\_205\_304\_b0;

global g\_decArgo\_doxy\_202\_205\_304\_b1;

global g\_decArgo\_doxy\_202\_205\_304\_b2;

global g\_decArgo\_doxy\_202\_205\_304\_b3;

global g\_decArgo\_doxy\_202\_205\_304\_c0;

global g\_decArgo\_doxy\_202\_205\_304\_pCoef2;

global g\_decArgo\_doxy\_202\_205\_304\_pCoef3;

if (isempty(a\_PRES) || isempty(a\_TEMP) || isempty(a\_PSAL) || isempty(a\_DOXY))

return

end

idDef = find( ...

(a\_PRES == a\_PRES\_fillValue) | ...

(a\_TEMP == a\_TEMP\_fillValue) | ...

(a\_PSAL == a\_PSAL\_fillValue) | ...

(a\_DOXY == a\_DOXY\_fillValue));

idNoDef = setdiff(1:length(a\_DOXY), idDef);

if (~isempty(idNoDef))

presValues = a\_PRES(idNoDef);

tempValues = a\_TEMP(idNoDef);

psalValues = a\_PSAL(idNoDef);

doxyValues = a\_DOXY(idNoDef);

% convert DOXY into DOXY\_in\_molar\_units

% units convertion (micromol/kg to micromol/L)

[measLon, measLat] = get\_meas\_location(a\_profOptode.cycleNumber, a\_profOptode.profileNumber, a\_profOptode);

rho = potential\_density\_gsw(presValues, tempValues, psalValues, 0, measLon, measLat);

rho = rho/1000;

molarDoxyValues = doxyValues .\* rho;

% pressure effect un-correction:

% at presValue, optode quenched by different pO2 inside membrane than pO2

% outside in seawater due to re-equilibration effect

% translate already corrected value (outside conditions) back to sensed value

% (inside membrane)

oxygenPresUncomp = calcoxy\_presuncomp(molarDoxyValues, presValues, tempValues, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef2, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef3 ...

);

% convert DOXY\_in\_molar\_units\_and\_inside\_conditions into PPOX\_DOXY

% units convertion (micromol/L to hPa)

ppoxDoxyValues = O2ctoO2p(oxygenPresUncomp, tempValues, psalValues, presValues, ...

g\_decArgo\_doxy\_202\_205\_304\_d0, ...

g\_decArgo\_doxy\_202\_205\_304\_d1, ...

g\_decArgo\_doxy\_202\_205\_304\_d2, ...

g\_decArgo\_doxy\_202\_205\_304\_d3, ...

g\_decArgo\_doxy\_202\_205\_304\_b0, ...

g\_decArgo\_doxy\_202\_205\_304\_b1, ...

g\_decArgo\_doxy\_202\_205\_304\_b2, ...

g\_decArgo\_doxy\_202\_205\_304\_b3, ...

g\_decArgo\_doxy\_202\_205\_304\_c0 ...

);

% adjust PPOX\_DOXY

ppoxDoxyAdjValues = ppoxDoxyValues \* a\_slope + a\_offset;

% convert PPOX\_ADJUSTED into DOXY\_ADJUSTED\_in\_molar\_units\_and\_inside\_conditions

% units convertion (hPa to micromol/L)

oxygenAdjPresUncomp = O2ptoO2c(ppoxDoxyAdjValues, tempValues, psalValues, presValues, ...

g\_decArgo\_doxy\_202\_205\_304\_d0, ...

g\_decArgo\_doxy\_202\_205\_304\_d1, ...

g\_decArgo\_doxy\_202\_205\_304\_d2, ...

g\_decArgo\_doxy\_202\_205\_304\_d3, ...

g\_decArgo\_doxy\_202\_205\_304\_b0, ...

g\_decArgo\_doxy\_202\_205\_304\_b1, ...

g\_decArgo\_doxy\_202\_205\_304\_b2, ...

g\_decArgo\_doxy\_202\_205\_304\_b3, ...

g\_decArgo\_doxy\_202\_205\_304\_c0 ...

);

% pressure effect re-correction:

% at presValue, optode quenched by different pO2 inside membrane than pO2

% outside in seawater due to re-equilibration effect

% translate adjusted sensed value (inside membrane) to adjusted corrected

% value (outside conditions)

molarDoxyAdjValues = calcoxy\_prescomp(oxygenAdjPresUncomp, presValues, tempValues, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef2, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef3 ...

);

% convert DOXY\_ADJUSTED\_in\_molar\_units into DOXY\_ADJUSTED

% units convertion (micromol/L to micromol/kg)

doxyAdjValues = molarDoxyAdjValues ./ rho;

o\_DOXY\_ADJUSTED(idNoDef) = doxyAdjValues;

% compute DOXY\_ADJUSTED\_ERROR

if (~isnan(a\_adjError))

% use PPOX\_DOXY\_ADJUSTED\_ERROR from META-DATA

ppoxDoxyAdjErrValues = a\_adjError;

% convert PPOX\_ADJUSTED\_ERROR into DOXY\_ADJUSTED\_ERROR\_in\_molar\_units\_and\_inside\_conditions

% units convertion (hPa to micromol/L)

oxygenAdjErrPresUncomp = O2ptoO2c(ppoxDoxyAdjErrValues, tempValues, psalValues, presValues, ...

g\_decArgo\_doxy\_202\_205\_304\_d0, ...

g\_decArgo\_doxy\_202\_205\_304\_d1, ...

g\_decArgo\_doxy\_202\_205\_304\_d2, ...

g\_decArgo\_doxy\_202\_205\_304\_d3, ...

g\_decArgo\_doxy\_202\_205\_304\_b0, ...

g\_decArgo\_doxy\_202\_205\_304\_b1, ...

g\_decArgo\_doxy\_202\_205\_304\_b2, ...

g\_decArgo\_doxy\_202\_205\_304\_b3, ...

g\_decArgo\_doxy\_202\_205\_304\_c0 ...

);

% pressure effect re-correction:

% at presValue, optode quenched by different pO2 inside membrane than pO2

% outside in seawater due to re-equilibration effect

% translate adjusted sensed value (inside membrane) to adjusted corrected

% value (outside conditions)

molarDoxyAdjErrValues = calcoxy\_prescomp(oxygenAdjErrPresUncomp, presValues, tempValues, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef2, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef3 ...

);

% convert DOXY\_ADJUSTED\_ERROR\_in\_molar\_units into DOXY\_ADJUSTED\_ERROR

% units convertion (micromol/L to micromol/kg)

doxyAdjErrValues = molarDoxyAdjErrValues ./ rho;

o\_DOXY\_ADJUSTED\_ERROR(idNoDef) = doxyAdjErrValues;

end

end

return

function oxygenPresUncomp = calcoxy\_presuncomp(molarDoxyValues, presValues, tempValues, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef2, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef3 ...

)

...

function oxygenPresUncomp = molarDoxyAdjValues = calcoxy\_prescomp(oxygenAdjPresUncomp, presValues, tempValues, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef2, ...

g\_decArgo\_doxy\_202\_205\_304\_pCoef3 ...

)

...

## calcoxy\_presuncomp

% ------------------------------------------------------------------------------

% Undo Correct DO (in micromol/L) from pressure effect.

%

% SYNTAX :

% [o\_oxygen] = calcoxy\_presuncomp(a\_oxygenPrescomp, a\_pres, a\_temp, ...

% a\_pCoef2, a\_pCoef3)

%

% INPUT PARAMETERS :

% o\_oxygenPrescomp : DO values (in micromol/L) corrected from pressure effect

% a\_pres : PRES values

% a\_temp : TEMP values

% a\_pCoef2 and a\_pCoef3 : additional coefficient values

%

% OUTPUT PARAMETERS :

% a\_oxygen : DO values

%

% EXAMPLES :

%

% SEE ALSO :

% AUTHORS : Jean-Philippe Rannou (Altran)(jean-philippe.rannou@altran.com)

% ------------------------------------------------------------------------------

% RELEASES :

% 05/20/2011 - Virginie THIERRY - creation

% 05/17/2016 - RNU - update

% ------------------------------------------------------------------------------

function [o\_oxygen] = calcoxy\_presuncomp(a\_oxygenPrescomp, a\_pres, a\_temp, ...

a\_pCoef2, a\_pCoef3)

% pressure compensation correction

o\_oxygen = a\_oxygenPrescomp ./ (1 + ((a\_pCoef2 .\* a\_temp) + a\_pCoef3) .\* a\_pres/1000);

return

## calcoxy\_prescomp

% ------------------------------------------------------------------------------

% Correct DO (in micromol/L) from pressure effect.

%

% SYNTAX :

% [o\_oxygenPrescomp] = calcoxy\_prescomp(a\_oxygen, a\_pres, a\_temp, ...

% a\_pCoef2, a\_pCoef3)

%

% INPUT PARAMETERS :

% a\_oxygen : DO values

% a\_pres : PRES values

% a\_temp : TEMP values

% a\_pCoef2 and a\_pCoef3 : additional coefficient values

%

% OUTPUT PARAMETERS :

% o\_oxygenPrescomp : DO values (in micromol/L) corrected from pressure effect

%

% EXAMPLES :

%

% SEE ALSO :

% AUTHORS : Jean-Philippe Rannou (Altran)(jean-philippe.rannou@altran.com)

% ------------------------------------------------------------------------------

% RELEASES :

% 05/20/2011 - Virginie THIERRY - creation

% 05/17/2016 - RNU - update

% ------------------------------------------------------------------------------

function [o\_oxygenPrescomp] = calcoxy\_prescomp(a\_oxygen, a\_pres, a\_temp, ...

a\_pCoef2, a\_pCoef3)

% pressure compensation correction

o\_oxygenPrescomp = a\_oxygen .\* (1 + ((a\_pCoef2 .\* a\_temp) + a\_pCoef3) .\* a\_pres/1000);

return