DATA PRODUCT SPECIFICATION FOR FAST DISSOLVED OXYGEN

Version 1-02
Document Control Number 1341-00521
2014-03-06

Consortium for Ocean Leadership
1201 New York Ave NW, 4th Floor, Washington DC 20005
www.OceanLeadership.org

in Cooperation with

University of California, San Diego
University of Washington
Woods Hole Oceanographic Institution
Oregon State University
Scripps Institution of Oceanography
Rutgers University

Document Control Sheet

Version	Date	Description	Author
0-01	2012-12-16	Initial Release	M. Vardaro
0-02	2013-02-07	Incorporates comments from Focused Review	M. Vardaro
0-03	2013-03-07	Incorporates comments from Formal Review and adds final test data set to section 4.6	M. Vardaro
1-00	2013-03-13	Initial Release	E. Griffin
1-01	2013-08-29	Updates to referenced TEOS-10 code	M. Vardaro
1-02	2014-03-06	Updates to include frequency output	M. Vardaro

Signature Page

This document has been reviewed and approved for release to Configuration Management.

OOI Senior Systems Engineer: ______________________________

Date:2014-03-07

This document has been reviewed and meets the needs of the OOI Cyberinfrastructure for the purpose of coding and implementation.

OOI CI Signing Authority: ______________________________

Date: 2013-03-13

Table of Contents

1 Abstract

This document describes the computation used to calculate the OOI Level 2 Fast Dissolved Oxygen core data product (DOCONCF), which is calculated using an algorithm based on that of Owens and Millard (1985) that incorporates raw voltage data from the Sea-Bird Electronics SBE 43 and 43F Dissolved Oxygen Sensor (DOFST) family of instruments along with L1 and L2 data products from the conductivity, temperature and depth (CTD) family of instruments. This document is intended to be used by OOI programmers to construct appropriate processes to create the L2 Fast Dissolved Oxygen data product.

2 Introduction

2.1 Author Contact Information

Please contact Michael Vardaro (mvardaro@coas.oregonstate.edu) or the Data Product Specification lead (DPS@lists.oceanobservatories.org) for more information concerning the computation and other items in this document.

2.2 Metadata Information

2.2.1 Data Product Name

The OOI Core Data Product Name for this product is
- DOCONCF

The OOI Core Data Product Descriptive Name for this product is
- Fast Dissolved Oxygen

2.2.2 Data Product Abstract (for Metadata)

The OOI Level 2 Fast Dissolved Oxygen core data product (DOCONCF) is the oxygen concentration (in micromoles per kg, µmol/kg) which is produced using raw voltage data from the Sea-Bird Electronics SBE 43 or frequency data from the 43F Dissolved Oxygen Sensor (DOFST) family of instruments along with L1 and L2 data products from the conductivity, temperature and depth (CTD) instrument to which the SBE 43 is attached.

2.2.3 Computation Name

Not required for data products.

2.2.4 Computation Abstract (for Metadata)

This computation computes the OOI Level 2 Fast Dissolved Oxygen core data product, which is calculated using an algorithm based on that of Owens and Millard (1985) that incorporates raw voltage data from the Sea-Bird Electronics SBE 43 or frequency data from the 43F Dissolved Oxygen Sensor family of instruments along with L1 Water Temperature (TEMPWAT), L1 Pressure (Depth) Data Product (PRESWAT), L2 Practical Salinity Data Product (PRACSAL) from the conductivity, temperature and depth (CTD) family of instruments, and latitude/longitude from the platform metadata.

2.2.5 Instrument-Specific Metadata

See Section 4.4 for instrument-specific metadata fields that must be part of the output data.

2.2.6 Data Product Synonyms

Synonyms for this data product are
- Oxygen concentration

2.2.7 Similar Data Products

Similar products that this data product may be confused with are DOCONCS, Stable Dissolved Oxygen, which is derived from the DOSTA Aanderaa Optode family of instruments. Other measurements of oxygen concentration use different units (ml L^{-1}, % saturation), but the current convention is to use the units described here: micromoles per kg (µmol kg^{-1}).

2.3 Instruments

For information on the instruments from which the L2 Fast Dissolved Oxygen core data product inputs are obtained, see the DOFST Processing Flow document (DCN 1342-00521). This document contains information on instrument class and make/model; it also describes the flow of data from the DOFST through all of the relevant QC, calibration, and data product computations and procedures.

Please see the Instrument Application in the SAF for specifics of instrument locations and platforms.

2.4 Literature and Reference Documents

Garcia and Gordon (1992) Oxygen solubility in seawater: Better fitting equations. Limnology & Oceanography. 37(6): 1307-1312.

Owens, W.B., and R.C. Millard Jr. (1985) A new algorithm for CTD oxygen calibration. J. Physical Oceanography, 15: 621-631.

Pawlowicz, R. (2010). What every oceanographer needs to know about TEOS-10 (The TEOS-10 Primer). *Thermodynamic Equation Of Seawater - 2010 (TEOS-10) website: http://www.teos-10.org/*

Sea-Bird Electronics, Inc. (2011) SBE 43 User Manual. Bellevue, WA.

Sea-Bird Electronics, Inc. (2013) SBE 52-MP Moored Profiler CTD and Optional DO Sensor User's Manual. Bellevue, WA.

Sea-Bird (2009), SBE 16*plus* V2 SEACAT User's Manual. Manual Version #005.

2.5 Terminology

2.5.1 Definitions

The following terms are defined here for use throughout this document. Definitions of general OOI terminology are contained in the Level 2 Reference Module in the OOI requirements database (DOORS).

Potential Density (pot_rho_t) Potential density of seawater, calculated using Absolute salinity, in situ temperature, sea pressure, and a reference pressure (p_ref) of 0 dbar

Practical Salinity (S_P) The measure of salinity defined by the Practical Salinity Scale 1978 (PSS-78). Practical Salinity is a unitless quantity that is approximately equivalent to the mass fraction of dissolved solute in seawater, but is not interchangeable with Absolute Salinity. Practical Salinity is an analogue for conductivity of seawater adjusted for temperature and pressure.

Absolute Salinity (S_A): Measure of absolute salinity adopted and defined jointly by the Intergovernmental Oceanographic Commission (IOC), International Association for the Physical Sciences of the Oceans (IAPSO) and the Scientific Committee on Oceanic Research (SCOR) in 2010 as part of the new standard for calculating the thermodynamic properties of seawater. Units of Absolute Salinity are g kg^{-1}, the mass fraction of dissolved salts in seawater. Absolute Salinity represents, to the best available accuracy, the mass fraction of dissolved solute in a sample of Standard Seawater of the same density as the observed sample (Pawlowicz, 2010).

2.5.2 Acronyms, Abbreviations and Notations

General OOI acronyms, abbreviations and notations are contained in the Level 2 Reference Module in the OOI requirements database (DOORS). The following acronyms and abbreviations are defined here for use throughout this document.

CTD = Conductivity, Temperature, and Depth instrument
SBE 43 = moored mode voltage (V) output SeaBird 43 Dissolved Oxygen Sensor
SBE 43F = profiling mode frequency (Hz) output SeaBird 43F Dissolved Oxygen Sensor

2.5.3 Variables and Symbols

The following variables and symbols are defined here for use throughout this document.

Oxygen	Dissolved oxygen concentration (micromole kg^{-1})
T	CTD Temperature (°C)
P	CTD Pressure (decibars)
S	CTD Salinity (psu)
V	SBE 43 temperature-compensated output oxygen signal (volts)
F	SBE 43F temperature-compensated output oxygen signal (frequency)
Soc	Oxygen signal slope
Voffset	Voltage at zero oxygen signal
Foffset	Frequency at zero oxygen signal (43F only)
A, B, C	Residual temperature correction factors
E	Pressure correction factor
tau20	Sensor time constant tau (T,P) at 20 ºC, 1 atmosphere, 0 PSU; slope term in calculation of tau(T,P)
D1, D2	Temperature and pressure correction factors in calculation of tau(T,P)
Oxsol(T,S)	Oxygen saturation value after Garcia and Gordon (1992)
δV/δt	Time derivative of SBE 43 output oxygen signal (volts/second)
δF/δt	Time derivative of SBE 43 output oxygen signal (frequency/second)
tau(T,P)	Sensor time constant at temperature and pressure = tau20 * exp (D1 * P + D2 * [T – 20])
K	Absolute temperature (K = T + 273.15)
pot_rho_t	Potential density of seawater, calculated using Absolute salinity, in situ temperature, and sea pressure
p_ref	Reference pressure, in this case 0 dbar

3 Theory

3.1 Description

Sea-Bird uses an algorithm based on that of Owens and Millard (1985) to convert SBE 43 oxygen sensor voltage data or 43F frequency data to oxygen concentration. The Sea-Bird algorithm incorporates a term related to the offset voltage/frequency produced for zero oxygen signal. In addition, there is a third-order polynomial that compensates for changes in sensitivity with temperature and an exponential term that compensates for changes in sensitivity with pressure.

3.2 Mathematical Theory

Sea-Bird's modified algorithm has the following form (SBE 43 User Manual):

$$Oxygen\ (mL/L) = Soc * \{(V + Voffset + tau(T,P) * \partial V/\partial t)\} * Oxsol(T,S) * (1.0 + A * T + B * T^{\wedge}2 + C * T^{\wedge}3\ ) * e^{\wedge}(((E * P)/K)\ )$$

(the red text (tau and the derivative) indicates that these parameters are removed from the equation if tau is set equal to zero, as recommended by SeaBird)

and

Oxygen [micromole/Kg] = Oxygen [mL/L] * 44660 [micromole-DO/L] / (pot_rho_t)

where:

Description	Symbol	Definition
Computed	Oxygen	Dissolved oxygen concentration (micromole/kg)
Input Parameters	T	CTD Temperature (degC)
	P	CTD Pressure (dbars)
	S	CTD Salinity (psu)
	V	SBE 43 temperature-compensated output oxygen signal (volts)
	F	SBE 43F temperature-compensated output oxygen signal (frequency)
Calibration Coefficients	Soc	Oxygen signal slope
	Voffset	Voltage at zero oxygen signal
	Foffset	Frequency at zero oxygen signal (43F only)
	A, B, C	Residual temperature correction factors
	E	Pressure correction factor
	tau20	Sensor time constant tau (T,P) at 20 ºC, 1 atmosphere, 0 PSU; slope term in calculation of tau(T,P)
	D1, D2	Temperature and pressure correction factors in calculation of tau(T,P)
	44660	Conversion constant for oxygen (micromole-DO/L)
Calculated Value	Oxsol(T,S)	Oxygen saturation value after Garcia and Gordon (1992)
	δV/δt	Time derivative of SBE 43 output oxygen signal (volts/second) (If tau is set to zero as recommended, the derivative is removed from the equation)
	tau(T,P)	Sensor time constant at temperature and pressure = tau20 * exp (D1 * P + D2 * [T – 20]) (This is recommended to be set to zero, which removes the derivative from the equation)
	K	Absolute temperature (in Kelvin)
	pot_rho_t	Potential density of seawater, calculated using Absolute salinity, in situ temperature, and sea pressure

3.3 Known Theoretical Limitations

The derivative term [tau(T,P) * δV/δt] function is used to improve the response of the measured signal in regions of large oxygen gradients. However, this term also amplifies residual noise in the signal (especially in deep water), and in some situations this negative consequence overshadows the gains in signal responsiveness. It is recommended to set Tau = 0, deleting the entire derivative term from the equation for calculated oxygen.

Response time of the SBE 43 and 43F sensor is variable, and temperature dependent. Response time varies from 7 seconds at 29.2 degC to 28 seconds at 1.7 degC. SeaBird recommends a minimum pump time of 15 seconds for 15 degC and warmer water, and reference the 1% curve in the SBE 43 user manual for colder water.

3.4 Revision History

No revisions to date.

4 Implementation

4.1 Overview

The L2 Fast Dissolved Oxygen computation is implemented using a third-order polynomial expression (SBE 43 User Manual), which incorporates raw voltage/frequency data from the SBE 43/43F Dissolved Oxygen Sensor along with L1 Water Temperature (TEMPWAT), L1 Pressure (Depth) Data Product (PRESWAT), L2 Practical Salinity Data Product (PRACSAL) from the CTD to which the DOFST instrument is connected, and latitude/longitude from the platform metadata. This is followed by a subsequent conversion from ml/l units to µmol/kg units. This product is calculated using the potential density, or “pot_rho_t,” so as to correct for compression-related changes in the volume and temperature of the parcel of water being measured. Potential density is different from the OOI L2 DENSITY data product; L2 DENSITY is computed using *in situ* temperature and pressure, while potential density is calculated using a reference pressure of 0 dbar, *in situ* temperature, and Absolute Salinity. Timing will be dealt with by interpolating the nearest CTD data to the time of sampling of the oxygen data using the INTERP1 QC algorithm (1341-10002). The oxygen measurement will be delayed until the following CTD measurement.

4.2 Inputs

Inputs are:

- L0 Fast Dissolved Oxygen measurement [counts, representing volts or frequency], in hex format
- Calibration coefficients provided by SeaBird with instrument
- DOFST time stamp

Inputs from the attached or collocated CTD:

- L1 Temperature [degrees C] (see 1341-00010_DPS_TEMPWAT_OOI)
- L1 Pressure (sea pressure) [dbar] (see 1341-00020_DPS_PRESWAT_OOI)
- L2 Practical Salinity (PSS-78) [unitless] (see 1341-00040_DPS_PRACSAL_OOI)
- Latitude and longitude where the input data was collected. This information is the lat/long of the mooring or profiler on which the instrument is fixed and is part of the metadata.
- CTD time stamp

When available use L1b PRESWAT, L1b TEMPWAT, and L2b PRACSAL data products instead of L1a and L2a data products.

Moored Mode Input format:

Output from the SeaBird 16plus that will pass through the data from the SBE 43 Dissolved Oxygen sensor is a four-character hex code (vvvv/13,107), as per this example from the SBE

16plus Manual (NOTE: in this example the SBE 43 output is the "First external voltage" out of two):

example scan = ttttttccccccppppppvvvvvvvvvvvvssssssss = 0A53711BC7220C14C17D82030505940EC4270B

• Temperature = tttttt = 0A5371 (676721 decimal); temperature A/D counts = 676721
• Conductivity = 1BC722 (1820450 decimal); conductivity freq. = 1820450 / 256 = 7111.133 Hz
• Internally mounted strain gauge pressure = pppppp = 0C14C1 (791745 decimal);
Strain gauge pressure A/D counts = 791745
• Internally mounted strain gauge temperature compensation = vvvv = 7D82 (32,130 decimal);
Strain gauge temperature = 32,130 / 13,107 = 2.4514 volts
• First external voltage = vvvv = 0305 (773 decimal); voltage = 773 / 13,107 = 0.0590 volts
• Second external voltage = vvvv = 0594 (1428 decimal); voltage = 1428 / 13,107 = 0.1089 volts
• Time = ssssssss = 0EC4270B (247,736,075 decimal); seconds since January 1, 2000 = 247,736,075

Profiling Mode Input Format:
Output from the SeaBird 52-MP profiling CTD that will pass through the data from the SBE 43F Dissolved Oxygen sensor is a four-character hex code (oooo), as per this example from the SBE 52 Manual. NOTE: in profiling mode, oxygen data is output as frequency (Hz), and the conversion equation will use "F" and "Foffset" in place of "V" and "Voffset"):

Example: example scan = ccccctttttppppppoooo = 5C98D0E2D628E8E3056

- Conductivity = ccccc = 5C98D (379277 decimal); conductivity (mmho/cm) = (379277 / 10,000) – 0.5 = 37.4277
- Temperature = ttttt = 0E2D6 (58070 decimal); temperature (°C, ITS-90) = (58070 / 10,000) – 5 = 0.8070
- Pressure = ppppp = 28E8E (167566 decimal); pressure (decibars) = (167566 / 100) - 10 = 1665.66
- Oxygen = oooo = 3056 (12374 decimal); oxygen (Hz) = 12374

4.3 Processing Flow

The specific steps necessary to create all calibrated and quality controlled data products for each OOI core instrument are described in the instrument-specific Processing Flow documents (DCN 1342-XXXXX). These processing flow documents contain flow diagrams detailing all of the specific procedures (data product and QC) necessary to compute all levels of data products from the instrument and the order in which these procedures should be followed.

The processing flow for the Fast Dissolved Oxygen computation is as follows (in Matlab syntax):

Step 1:
Download and convert raw hex code data from SBE 43 to a decimal voltage value and divide by 13107 to produce a floating point number "V" in volts with four decimal places %.4f.

If using an SBE 43F in profiling mode, convert the hex code data to a decimal frequency value to produce a floating point number "F" in Hz with two decimal places %.2f.

Step 2:
Absolute salinity (S_A) is calculated from practical salinity (S_P, L2 input), sea pressure (p, L1 input), latitude (`lat`, L1 input metadata) and longitude (`long`, L1 input metadata) using the function

```
[SA, in_ocean] = gsw_SA_from_SP(SP,p,long,lat)
```

where `in_ocean` is a flag indicating that the lat/long is well inside the boundaries of dry land. See Appendix A.1 for example code.

Step 3:
Potential density is calculated using a reference pressure of 0 dbar (p_ref = 0), Absolute Salinity (SA, step 1 output), in situ Temperature (t, L1 input) and sea pressure (p, L1 input).

```
pot_rho_t_exact = gsw_pot_rho_t_exact(SA,t,p,p_ref)
```

See Appendix A.2 for example code.

Step 4:
Oxygen solubility (Oxsol) is calculated by the equations

$$\mathrm{Ts} = \ln\left[(298.15 - \mathrm{T}) / (273.15 + \mathrm{T})\right]$$

$$\mathrm{Oxsol}(\mathrm{T},\mathrm{S}) = \exp\text{^}\{\mathrm{A0} + \mathrm{A1}(\mathrm{Ts}) + \mathrm{A2}(\mathrm{Ts})\text{^}2 + \mathrm{A3}(\mathrm{Ts})\text{^}3 + \mathrm{A4}(\mathrm{Ts})\text{^}4 + \mathrm{A5}(\mathrm{Ts})\text{^}5 + [\mathrm{S} * (\mathrm{B0} + \mathrm{B1}(\mathrm{Ts}) + \mathrm{B2}(\mathrm{Ts})\text{^}2 + \mathrm{B3}(\mathrm{Ts})\text{^}3)] + \mathrm{C0}(\mathrm{S})\text{^}2\}$$

where
Oxsol(T,S) = oxygen saturation value = volume of oxygen gas at standard temperature and pressure conditions (STP) absorbed from humidity-saturated air at a total pressure of one atmosphere, per unit volume of the liquid at the temperature of measurement (ml/l)
S = practical salinity (using the L2 PRACSAL data product)
T = *in situ* water temperature (^{o}C)
A0 = 2.00907
A1 = 3.22014
A2 = 4.0501
A3 = 4.94457
A4 = -0.256847
A5 = 3.88767
B0 = -0.00624523
B1 = -0.00737614
B2 = -0.010341
B3 = -0.00817083
C0 = -0.000000488682

Step 5:
Fast dissolved oxygen is calculated by the function described in section 3.2 (with the derivative term removed by setting Tau = 0, to prevent amplification of residual noise):

$$Oxygen\ (mL/L) = Soc * (V + Voffset) * Oxsol(T,S) * (1.0 + A*T + B*T\text{^}2 + C*T\text{^}3\ ) * e\text{^}(((E * P)/K)\ )$$

This function is the same for 43F frequency data, but with "*F*" and "*Foffset*" in place of "*V*" and "*Voffset*".

Step 6:
The oxygen concentration is converted from mL/L into micromole/Kg by the function

$$\mathrm{Oxygen\ [micromole/Kg]} = \mathrm{Oxygen\ [ml/l]} * 44660/(\mathrm{pot\_rho\_t\_exact})$$

Step 7:
Perform the necessary QC steps as outlined in the processing flow document

4.4 Outputs

The outputs of the Fast Dissolved Oxygen computation are

- Oxygen concentration in µmol kg^{-1} as a floating point number with two decimal places %.2f.

The metadata that must be included with the output are
- The source of the PRACSAL, TEMPWAT, and PRESWAT inputs (which should be from the CTD to which the DOFST instrument is attached)
- The interpolated PRACSAL, TEMPWAT, and PRESWAT measurements used to perform the calculation
- Location (lat/long)
- Time

See Appendix D for a discussion of the accuracy of the output.

4.5 Computational and Numerical Considerations

4.5.1 Numerical Programming Considerations

There are no numerical programming considerations for this computation. No special numerical methods are used.

4.5.2 Computational Requirements

L2 PRACSAL, L1 TEMPWAT, and L1 PRESWAT Data Products must be available from the attached CTD instrument, as well as the timestamp information

Assuming we are reprocessing the data upon recovery of the various assets, and that one sample is a single data point from any Fast Dissolved Oxygen sensor attached to a profiling CTD, and example number of samples are as follows:
- For a deep profiler on RSN or Endurance: 1 sample/second for a 1000m profile with a profiler moving at 0.5 m/s operating 48 times per day (assumes that a CTD profile is taken on both down and up casts and profiler is operating continuously) for 365 days = $3.5 * 10^7$ samples.

4.6 Code Verification and Test Data Set

The code will be verified using the test data set provided, which contains inputs and their associated correct outputs. CI will verify that the code is correct by checking that the output, generated using the test data inputs, is identical to the test data density output.

SBE 43 Test Data Set
COEFFICIENTS
Soc = 0.4396
Voffset = -0.5186
Tau20 = 5.08
A = -3.1867e-003
B = 1.7749e-004
C = -3.5718e-006
E nominal = 0.036
NOMINAL DYNAMIC COEFFICIENTS
D1 = 1.92634e-4
D2 = -4.64803e-2

DOFST Voltage Test Data							
do_raw	salinity	temp	pressure	lat	lon	Intermediate DO (ml/l)	DO (umol/kg)
0	0.0	0.0	0.0	50.0	145.0	-2.332525266	-104.1869283

6798	33.4	-30.1	307.5	-42.0	-42.0	0.000797115	0.03494869
16384	31.2	30.3	201.2	39.0	-70.5	1.412078813	61.89990653
32768	20.1	10.1	5.2	60.0	39.0	5.934280027	261.0228351
65535	35.2	20.2	112.1	45.0	-125.0	10.06589881	438.6325206
0	35.2	20.2	5.2	60.0	39.0	-1.149671963	-50.09861089
6798	0.0	30.3	307.5	39.0	-70.5	0.000125639	0.005635974
16384	31.2	-30.1	201.2	45.0	-125.0	10.82518961	475.5984302
32768	20.1	0.0	0.0	50.0	145.0	7.744491469	340.3897211
65535	33.4	10.1	112.1	-42.0	-42.0	12.49523919	544.0600381
0	35.2	20.2	5.2	50.0	145.0	-1.149671963	-50.09857466
6798	0.0	30.3	0.0	39.0	-70.5	0.000121139	0.005434191
16384	20.1	10.1	112.1	60.0	39.0	2.2205184	97.67068802
32768	33.4	0.0	307.5	45.0	-125.0	7.347649726	319.5738329
65535	31.2	-30.1	201.2	-42.0	-42.0	66.32586768	2914.002444
0	33.4	-30.1	112.1	60.0	39.0	-7.415682793	-325.155281
6798	35.2	20.2	5.2	45.0	-125.0	0.000120053	0.005231489
16384	31.2	0.0	0.0	-42.0	-42.0	2.645019264	115.240647
32768	20.1	10.1	201.2	50.0	145.0	6.083964669	267.6054819
65535	0.0	30.3	307.5	39.0	-70.5	10.39697952	466.3908327
0	33.4	30.3	5.2	60.0	39.0	-0.966429789	-42.29682113
6798	31.2	0.0	0.0	45.0	-125.0	0.000195837	0.008532408
16384	35.2	-30.1	112.1	39.0	-70.5	10.2787545	449.9501918
32768	20.1	10.1	201.2	-42.0	-42.0	6.083964669	267.6060633
65535	0.0	20.2	307.5	50.0	145.0	12.68706213	567.6400574

SBE 43F Test Data Set

COEFFICIENTS
Soc = 2.9968e-04 (adj)
Foffset = -839.55
Tau20 = 1.72
A = -4.1168e-003
B = 2.4818e-004
C = -3.8820e-006
E nominal = 0.036
NOMINAL DYNAMIC COEFFICIENTS
D1 = 1.92634e-4
D2 = -4.64803e-2
Lat = 45
Long = -125

DOFST Frequency Test Data					
salinity	temp	pressure	freq	Intermediate DO (ml/l)	DO (μmol/kg)
34.1145	15.5257	60.520	4354	5.89891032167396	256.974348631580
34.2845	15.3317	72.580	4143	5.56780727769487	242.509215041926
33.2464	11.9239	31.420	4583	6.76187243958794	294.548757813511
33.5524	12.8940	70.820	4476	6.458117534861	281.302611659343

33.5619	12.9011	74.870	4481	6.46897458201929	281.773833754618
33.2512	11.9350	29.330	4591	6.77275996877815	295.022573240991
33.2609	11.9715	30.950	4575	6.73969994032525	293.582249046689
33.2716	12.0110	43.580	4574	6.74263221709132	293.709591753566
33.4191	12.4553	65.370	4545	6.64102027182035	289.274637555853
33.2710	11.9932	29.460	4578	6.74036293148305	293.610068502014
33.2808	12.0196	31.030	4572	6.72675872808420	293.016695107650
33.5483	12.8647	74.570	4505	6.51674462650798	283.855524405738
33.5424	12.8448	75.070	4383	6.30302843255881	274.546683447082
33.3458	12.2084	57.920	4555	6.68981312176670	291.402846954730
0.0000	12.0996	42.980	4569	8.28303866101280	370.109515168768
37.7843	-10.1230	42.980	4023	10.78098598783980	467.353919671818
35.7594	0.0000	42.980	4569	8.95549253591715	388.785846175276
33.3313	12.0996	0.000	4569	6.68181215593754	291.052144126035
33.3132	12.0996	42.980	0	-1.51252046989329	-65.8845095713829
33.3132	12.0996	42.980	841	0.00261229787546345	0.113790171971304
33.3132	12.0996	42.980	1000	0.289064271805584	12.5914710984794
33.3132	12.0996	42.980	2000	2.09064901350446	91.0674517683417
33.3132	12.0996	42.980	4000	5.69381849690220	248.019413108066
33.3132	12.0996	42.980	5000	7.49540323860107	326.495393777929
33.3132	12.0996	42.980	6000	9.29698798029994	404.971374447791

Appendix A Example Code

A.1 Absolute Salinity from Practical Salinity

USAGE: [SA, in_ocean] = gsw_SA_from_SP(SP,p,long,lat)

DESCRIPTION: Calculates absolute salinity from practical salinity. Since SP is non-negative by definition, this function changes any negative input values of SP to be zero.

INPUT:
SP = Practical Salinity (PSS-78) [unitless]
p = sea pressure [dbar] (i.e., absolute pressure - 10.1325 dbar)
long = longitude in decimal degrees [0 ... +360] or [-180 ... +180]
lat = latitude in decimal degrees north [-90 ... +90]

p, lat & long may have dimensions 1x1 or Mx1 or 1xN or MxN, where SP is MxN.

OUTPUT:
SA = Absolute Salinity [g/kg]
in_ocean = 0, if long and lat are a long way from the ocean
= 1, if long and lat are in the ocean

Note. This flag is only set when the observation is well and truly on dry land; often the warning flag is not set until one is several hundred kilometres inland from the coast.

AUTHOR: Trevor McDougall, Paul Barker & David Jackett

VERSION NUMBER: 3.0 (23rd May, 2011)

REFERENCES:

IOC, SCOR and IAPSO, 2010: The international thermodynamic equation of seawater - 2010: Calculation and use of thermodynamic properties. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO (English), 196 pp. Available from the TEOS-10 web site. See section 2.5 and appendices A.4 and A.5 of this TEOS-10 Manual.

McDougall, T.J., D.R. Jackett and F.J. Millero, 2010: An algorithm for estimating Absolute Salinity in the global ocean. Submitted to Ocean Science. A preliminary version is available at Ocean Sci. Discuss., 6, 215-242. http://www.ocean-sci-discuss.net/6/215/2009/osd-6-215-2009-print.pdf

```
function [SA, in_ocean] = gsw_SA_from_SP(SP,p,long,lat)

% gsw_SA_from_SP              Absolute Salinity from Practical Salinity
%==========================================================================
%
% USAGE:
%  [SA, in_ocean] = gsw_SA_from_SP(SP,p,long,lat)
%
% DESCRIPTION:
%  Calculates Absolute Salinity from Practical Salinity.  Since SP is
%  non-negative by definition, this function changes any negative input
```

```
%  values of SP to be zero.
%
% INPUT:
%  SP   =  Practical Salinity  (PSS-78)                    [ unitless ]
%  p    =  sea pressure                                    [ dbar ]
%        ( i.e. absolute pressure - 10.1325 dbar )
%  long =  longitude in decimal degrees                    [ 0 ... +360 ]
%                                                   or  [ -180 ... +180 ]
%  lat  =  latitude in decimal degrees north              [ -90 ... +90 ]
%
%  p, lat & long may have dimensions 1x1 or Mx1 or 1xN or MxN,
%  where SP is MxN.
%
% OUTPUT:
%  SA        =  Absolute Salinity                          [ g/kg ]
%  in_ocean  =  0, if long and lat are a long way from the ocean
%            =  1, if long and lat are in the ocean
%  Note. This flag is only set when the observation is well and truly on
%    dry land; often the warning flag is not set until one is several
%    hundred kilometres inland from the coast.
%
% AUTHOR:
%  David Jackett, Trevor McDougall & Paul Barker      [ help_gsw@csiro.au ]
%
% VERSION NUMBER: 3.0 (31st May, 2011)
%
% REFERENCES:
%  IOC, SCOR and IAPSO, 2010: The international thermodynamic equation of
%   seawater - 2010: Calculation and use of thermodynamic properties.
%   Intergovernmental Oceanographic Commission, Manuals and Guides No. 56,
%   UNESCO (English), 196 pp.  Available from http://www.TEOS-10.org
%    See section 2.5 and appendices A.4 and A.5 of this TEOS-10 Manual.
%
%  McDougall, T.J., D.R. Jackett and F.J. Millero, 2010: An algorithm
%   for estimating Absolute Salinity in the global ocean.  Submitted to
%   Ocean Science. A preliminary version is available at Ocean Sci. Discuss.,
%   6, 215-242.
%   http://www.ocean-sci-discuss.net/6/215/2009/osd-6-215-2009-print.pdf
%
%  The software is available from http://www.TEOS-10.org
%
%==========================================================================

%--------------------------------------------------------------------------
% Check variables and resize if necessary
%--------------------------------------------------------------------------

if ~(nargin==4)
   error('gsw_SA_from_SP:  Requires four inputs')
end %if

[ms,ns] = size(SP);
[mp,np] = size(p);

if (mp == 1) & (np == 1)              % p is a scalar - fill to size of SP
   p = p*ones(size(SP));
```

```
elseif (ns == np) & (mp == 1)         % p is row vector,
   p = p(ones(1,ms), :);              % copy down each column.
elseif (ms == mp) & (np == 1)         % p is column vector,
   p = p(:,ones(1,ns));               % copy across each row.
elseif (ns == mp) & (np == 1)         % p is a transposed row vector,
   p = p.';                           % transposed then
   p = p(ones(1,ms), :);              % copy down each column.
elseif (ms == mp) & (ns == np)
   % ok
else
   error('gsw_SA_from_SP: Inputs array dimensions arguments do not agree')
end %if

[mla,nla] = size(lat);

if (mla == 1) & (nla == 1)            % lat is a scalar - fill to size of SP
   lat = lat*ones(size(SP));
elseif (ns == nla) & (mla == 1)       % lat is a row vector,
   lat = lat(ones(1,ms), :);          % copy down each column.
elseif (ms == mla) & (nla == 1)       % lat is a column vector,
   lat = lat(:,ones(1,ns));           % copy across each row.
elseif (ns == mla) & (nla == 1)       % lat is a transposed row vector,
   lat = lat.';                       % transposed then
   lat = lat(ones(1,ms), :);          % copy down each column.
elseif (ms == mla) & (ns == nla)
   % ok
else
   error('gsw_SA_from_SP: Inputs array dimensions arguments do not agree')
end %if

[mlo,nlo] = size(long);
[Iwest] =find(long < 0);
if ~isempty(Iwest)
   long(Iwest) = long(Iwest) + 360;
end

if (mlo == 1) & (nlo == 1)            % long is a scalar - fill to size of SP
   long = long*ones(size(SP));
elseif (ns == nlo) & (mlo == 1)       % long is a row vector,
   long = long(ones(1,ms), :);        % copy down each column.
elseif (ms == mlo) & (nlo == 1)       % long is a column vector,
   long = long(:,ones(1,ns));         % copy across each row.
elseif (ns == mlo) & (nlo == 1)       % long is a transposed row vector,
   long = long.';                     % transposed then
   long = long(ones(1,ms), :);        % copy down each column.
elseif (ms == nlo) & (mlo == 1)       % long is a transposed column vector,
   long = long.';                     % transposed then
   long = long(:,ones(1,ns));         % copy down each column.
elseif (ms == mlo) & (ns == nlo)
   % ok
else
   error('gsw_SA_from_SP: Inputs array dimensions arguments do not agree')
end %if

if ms == 1
   SP = SP.';
```

```
    p = p.';
    lat = lat.';
    long = long.';
    transposed = 1;
else
    transposed = 0;
end

[Iout_of_range] = find(p < 100 & SP > 120);
SP(Iout_of_range) = NaN;
[Iout_of_range] = find(p >= 100 & SP > 42);
SP(Iout_of_range) = NaN;

[Inan] = find(abs(SP) == 99999 | abs(SP) == 999999);
SP(Inan) = NaN;
[Inan] = find(abs(p) == 99999 | abs(p) == 999999);
p(Inan) = NaN;
[Inan] = find(abs(long) == 9999 | abs(long) == 99999);
long(Inan) = NaN;
[Inan] = find(abs(lat) == 9999 | abs(lat) == 99999);
lat(Inan) = NaN;

if ~isempty(find(p < -1.5 | p > 12000))
    error('gsw_SA_from_SP: pressure is out of range')
end
if ~isempty(find(long < 0 | long > 360))
    error('gsw_SA_from_SP: longitude is out of range')
end
if ~isempty(find(abs(lat) > 90))
    error('gsw_SA_from_SP: latitude is out of range')
end

%--------------------------------------------------------------------------
% Start of the calculation
%--------------------------------------------------------------------------

% These few lines ensure that SP is non-negative.
[I_neg_SP] = find(SP < 0);
if ~isempty(I_neg_SP)
    SP(I_neg_SP) = 0;
end

[Iocean] = find(~isnan(SP.*p.*lat.*long));

SA = nan(size(SP));
SAAR = nan(size(SP));
in_ocean = nan(size(SP));

% The following function (gsw_SAAR) finds SAAR in the non-Baltic parts of
% the world ocean.  (Actually, this gsw_SAAR look-up table returns values
% of zero in the Baltic Sea since SAAR in the Baltic is a function of SP,
% not space.
[SAAR(Iocean), in_ocean(Iocean)] = gsw_SAAR(p(Iocean),long(Iocean),lat(Iocean));

SA(Iocean) = (35.16504/35)*SP(Iocean).*(1 + SAAR(Iocean));
```

```
% Here the Practical Salinity in the Baltic is used to calculate the
% Absolute Salinity there.
SA_baltic(Iocean) = gsw_SA_from_SP_Baltic(SP(Iocean),long(Iocean),lat(Iocean));

[Ibaltic] = find(~isnan(SA_baltic(Iocean)));

SA(Iocean(Ibaltic)) = SA_baltic(Iocean(Ibaltic));

if transposed
   SA = SA';
   in_ocean = in_ocean';
end

end
```

A.2 Potential Density

```
function pot_rho_t_exact = gsw_pot_rho_t_exact(SA,t,p,p_ref)

% gsw_pot_rho_t_exact                               potential density
%==========================================================================
==
%
% USAGE:
%  pot_rho_t_exact = gsw_pot_rho_t_exact(SA,t,p,p_ref)
%
% DESCRIPTION:
%  Calculates potential density of seawater.  Note. This function outputs
%  potential density, not potential density anomaly; that is, 1000 kg/m^3
%  is not subtracted.
%
% INPUT:
%  SA     =  Absolute Salinity                                  [ g/kg ]
%  t      =  in-situ temperature (ITS-90)                      [ deg C ]
%  p      =  sea pressure                                       [ dbar ]
%           ( i.e. absolute pressure - 10.1325 dbar )
%  p_ref  =  reference pressure                                 [ dbar ]
%           ( i.e. reference absolute pressure - 10.1325 dbar )
%
%  SA & t need to have the same dimensions.
%  p & p_ref may have dimensions 1x1 or Mx1 or 1xN or MxN, where SA & t
%  are MxN
%
% OUTPUT:
%  pot_rho_t_exact  =  potential density (not potential density anomaly)
%                                                            [ kg/m^3 ]
%
% AUTHOR:
%  David Jackett, Trevor McDougall and Paul Barker     [ help@teos-10.org ]
%
% VERSION NUMBER: 3.02 (15th November, 2012)
%
% REFERENCES:
%  IOC, SCOR and IAPSO, 2010: The international thermodynamic equation of
%   seawater - 2010: Calculation and use of thermodynamic properties.
%   Intergovernmental Oceanographic Commission, Manuals and Guides No. 56,
%   UNESCO (English), 196 pp.  Available from http://www.TEOS-10.org
%    See section 3.4 of this TEOS-10 Manual.
%
%  The software is available from http://www.TEOS-10.org
%
%==========================================================================
==

%--------------------------------------------------------------------------
% Check variables and resize if necessary
%--------------------------------------------------------------------------
```

```
if ~(nargin == 4 )
  error('gsw_pot_rho_t_exact:  Requires four inputs')
end %if

[ms,ns] = size(SA);
[mt,nt] = size(t);
[mp,np] = size(p);

if (mt ~= ms | nt ~= ns)
   error('gsw_pot_rho_t_exact: SA and t must have same dimensions')
end

if ~isscalar(unique(p_ref))
   error('gsw_pot_rho_t_exact: The reference pressures differ, they should be unique')
end

if (mp == 1) & (np == 1)              % p scalar - fill to size of SA
   p = p*ones(size(SA));
elseif (ns == np) & (mp == 1)         % p is row vector,
   p = p(ones(1,ms), :);              % copy down each column.
elseif (ms == mp) & (np == 1)         % p is column vector,
   p = p(:,ones(1,ns));               % copy across each row.
elseif (ns == mp) & (np == 1)         % p is a transposed row vector,
   p = p.';                           % transposed then
   p = p(ones(1,ms), :);              % copy down each column.
elseif (ms == mp) & (ns == np)
   % ok
else
   error('gsw_pot_rho_t_exact: Inputs array dimensions arguments do not agree')
end %if

if ms == 1
   SA = SA.';
   t = t.';
   p = p.';
   transposed = 1;
else
   transposed = 0;
end

upr = unique(p_ref);
p_ref = upr*ones(size(SA));

%--------------------------------------------------------------------------
% Start of the calculation
%--------------------------------------------------------------------------

pt = gsw_pt_from_t(SA,t,p,p_ref);

pot_rho_t_exact = gsw_rho_t_exact(SA,pt,p_ref);

if transposed
   pot_rho_t_exact = pot_rho_t_exact.';
end

end
```

A.3 DOCONCF Sample Code

```
function DO = dofst_calc(do_VorF, VorF_offset, Soc, A, B, C, E, T, P, SP, lat, lon)

% Description:
%     Conversion of SBE43 raw measurement(voltage or frequency) to
%     dissolved oxygen concentration, and applies a potential density
%     correction from co-located CTD data.  OOI L2 data product DOCONCF.
%
% Usage:
%
%     DO = dostf_calculation(do_VorF,VorF_offset,Soc,A,B,C,E,T,P,SP,lat,lon)
%
%         where
%
%     DO = corrected dissolved oxygen [micro-mole/kg].
%     do_VorF = Oxygen sensor voltage or frequency [V] or [Hz].
%     VorF_offset = Voltage or Frequency offset [V] or [Hz].
%     Soc = Oxygen signal slope
%     A = Residual temperature correction factor A
%     B = Residual temperature correction factor B
%     C = Residual temperature correction factor C
%     E = Pressure correction factor
%     T = TEMPWAT water temperature [deg C]. (see
%         1341-00010_Data_Product_Spec_TEMPWAT)
%     P = PRESWAT water pressure [dbar]. (see
%         1341-00020_Data_Product_Spec_PRESWAT)
%     SP = PRACSAL practical salinity [unitless]. (see
%         1341-00040_Data_Product_Spec_PRACSAL)
%     lat, lon = latitude and longitude of the instrument [degrees].

% Author: Mike Vardaro 2013-08-22

% Get the potential density using the TEOS-10 toolbox
SA = gsw_SA_from_SP(SP, P, lon, lat);
pot_rho_t = gsw_pot_rho_t_exact(SA, T, P, 0);

% Oxygen saturation value after Garcia and Gordon (1992)
temp_K = T + 273.15;
Ts = log((298.15 - T) ./ (temp_K));
% Empirical constants
A0 = 2.00907;
A1 = 3.22014;
A2 = 4.0501;
A3 = 4.94457;
A4 = -0.256847;
A5 = 3.88767;
B0 = -0.00624523;
B1 = -0.00737614;
B2 = -0.010341;
B3 = -0.00817083;
C0 = -0.000000488682;
Oxsol = exp(A0 + A1*Ts + A2*Ts.^2 + A3*Ts.^3 + A4*Ts.^4 + A5*Ts.^5 + ...
    SP .* (B0 + B1*Ts + B2*Ts.^2 + B3*Ts.^3) + ...
```

```
    C0*SP.^2);

% Intermediate step, Dissolved Oxygen concentration in [mL/L]
DO_int = Soc * (do_VorF + VorF_offset) .* Oxsol .* ...
    (1.0 + A*T + B*T.^2 + C*T.^3) .* exp((E * P)./temp_K);

% Correct DO_int for Potential Density and convert to [micromole/Kg]
DO = DO_int * 44660 ./ (pot_rho_t);
```

Appendix B Output Accuracy

The DOCONCF accuracy requirement is stated as:

"Upon initial deployment, the instrument shall measure dissolved O_2 concentrations with an accuracy within ± 2% of the value provided by a Winkler titration of a corresponding water sample. <L2-SR-RQ-3495, L4-CG-IP-RQ-182, L4-RSN-IP-RQ-312>"

This measurement will be performed by the operator and logged with CI prior to deployment.

Accuracy of the L2 Fast Dissolved Oxygen data product is dependent on the accuracy of the L1 input to the computation, which for the SBE43 is 2% of saturation.

Appendix C Sensor Calibration Effects

Not Applicable for Level 2 (L2) data products.