argo data management

Argo data management

DOI : http://dx.doi.org/10.13155/33951

Argo Quality Control Manual for CTD and Trajectory Data

1

Version 3.9

20th February 2025

ARGO

part of the integrated global observation strategy

Title:
Argo Quality Control Manual for CTD and Trajectory Data

Authors:
Annie Wong, Robert Keeley, Thierry Carval, and the Argo Data Management Team

How to cite this document:
Annie Wong, Robert Keeley, Thierry Carval, and the Argo Data Management Team (2025). Argo Quality Control Manual for CTD and Trajectory Data. http://dx.doi.org/10.13155/33951

Corresponding author:
Annie P. Wong, apsw.uw@gmail.com

Table of contents

History

Date (dd/mmm/yyyy)	Comment
01/Jan/2002	Creation of the document.
28/Mar/2003	Changed lower limit of temperature in the Mediterranean Sea to 10.0°C.
08/Jun/2004	Modified spike and gradient tests according to advice from Yasushi. Added inversion test.
24/Oct/2003	Real-time qc tests 15 and 16 proposed at ADMT4 in Monterey. Test 10 removed.
07/Oct/2004	1. Real-time and delayed-mode qc manuals merged in "Argo quality control manual". 2. Frozen profile real time qc test 17 proposed at ADMT5 in Southampton. 3. Deepest pressure real time qc test 18 proposed at ADMT5 Southampton. 4. Order list for the real time qc tests. 5. "Regional Global Parameter Test" renamed "Regional range test", test 7. 6. Grey list naming convention and format, test 15. 7. Real time qc on trajectories.
23/Nov/2004	§1: New introduction from Annie Wong. §3: Delayed mode quality control manual from Annie Wong.
26/Nov/2004	§3: update of summary flow chart for salinity delayed mode qc.
31/Aug/2005	§2.1.2: update on test 17, visual qc.
17/Nov/2005	§2.3: added a section on real-time salinity adjustment. §3.3: added usage of SURFACE PRESSURE from APEX floats. §3.5.6: added guidelines for PSAL_ADJUSTED_QC = '2'. §3.5: clarified that PROFILE_<PARAM>_QC should be recomputed when <PARAM>_ADJUSTED_QC becomes available.
16/Oct/2006	§3: updated delayed-mode section based on DMQC-2 Workshop.
20/Nov/2006	§2.1.2: test 19: deepest pressure delta set to 10%. §2.1.2: test 14: density inversion test applied downward and upward.
14/Nov/2007	§2.1.2: test 6: minimum salinity set to 2 PSU instead of 0 PSU. §2.1.2: test 7: minimum salinity set to 2 PSU instead of 0 PSU. This change was decided during ADMT8 in Hobart.
14/Nov/2007	§3.3: use "known pressure drift" instead of delta P > 5dbar. This change was decided during ADMT8 meeting in Hobart.
14/Nov/2007	§3.1: delayed-mode operators can edit real-time QC flags. This change was decided during ADMT8 in Hobart.
14/Nov/2007	§2.1.4: values with QC flag = ‘4’ are ignored by real-time quality control tests. This change was decided during ADMT8 in Hobart.
14/Nov/2007	§2.1.4: when salinity is calculated from temperature and conductivity, if temperature is flagged as bad then salinity is flagged as bad. This change was decided during ADMT8 in Hobart.
21/Jan/2008	§2.1: test 6: minimum salinity set to 2 PSU instead of 0 PSU. §2.1: test 7: minimum salinity set to 2 PSU instead of 0 PSU.
04/Nov/2008	§2.1.2: “Sigma0” specified in density inversion test. §2.2.1: added a section on real-time pressure adjustment for APEX floats. §3.5: updated delayed-mode section based on DMQC-3 Workshop.
14/Feb/2009	§3.3: added a section on delayed-mode pressure adjustment for APEX floats.
19/Oct/2009	§2.1.2: test 6: minimum P set to −5 dbar. §3.1: added a section on editing raw qc flags in delayed-mode. §3.3: updated delayed-mode treatment for APEX TNPDs, after DMQC-4.
15/Jul/2010	§2.4: added a section on Feedback from Statistical Test at Coriolis.
05/Nov/2010	§3.3: revised definition for TNPD after ADMT11 in Hamburg.

Date (dd/mmm/yyyy)	Comment
01/Dec/2011	§2.1.2: added threshold of 0.03 kg m^{-3} to RT Test 14, Density Inversion Test, for profile data, following ADMT12 in Seoul. Test to use potential density referenced to mid-point pressure between the two levels to be compared. §2.2: Added RT Test 20, "Questionable Argos position test" from JAMSTEC, as a new real-time qc test for trajectory data, following 3rd Trajectory Workshop and ADMT12 in Seoul.
03/Jan/2013	§2.3: clarified real-time pressure adjustment for non auto-correcting floats when SP=0.
03/Jan/2013	§2.1.2: addition of real-time quality control for dissolved oxygen DOXY. The real-time qc tests 6, 9, 11 and 13 are applied to DOXY.
04/Nov/2013	§2.5: addition of real-time qc test procedures for near-surface data.
04/Nov/2013	§3.5: recommended upper limit of delayed-mode salinity adjustment of 0.05.
18/Nov/2014	§2.6: addition of interim real-time qc flag scheme for float data deeper than 2000 dbar. This is RT Test 23.
10/Nov/2015	Argo QC Manual split into Argo QC Manual for CTD & Trajectory Data and Argo QC Manual for Biogeochemical Data, after ADMT16 in Bermuda.
07/Dec/2015	RT Test 2, Impossible Date Test, re-written in JULD instead of day-month-year.
07/Dec/2015	§2.1.1: description of the BUFR format added by Anh Tran.
08/Dec/2017	Reference Table 2, Argo qc flag scale: '8' meaning changed from "interpolated" to "estimated".
08/Dec/2017	§2.1.2: action for RT Test 15, the "grey list", updated to include no GTS distribution by TESAC or BUFR.
08/Dec/2017	§2.7: addition of interim real-time qc flag scheme for float data from pilot sensors, including the RBR CTD.
16/Jan/2019	RT Test 15, greylist definition revised and allow qc='2'.
16/Jan/2019	RT Test 7, Red Sea temperature lower limit changed from 21.7°C to 21.0°C.
16/Jan/2019	§3.5.8: clarified that the delayed-mode operator can decide how recent a profile to send out in 'D' mode.
16/Jan/2019	§3.7.2: added examples on how to fill the SCIENTIFIC_CALIB section in delayed-mode.
08/Jan/2020	RT Test 11, Gradient Test, declared obsolete at ADMT20 at Villefrache. New RT Test 25, MEDD test, was approved. Test 25, MEDD test, to replace Test 11, Gradient Test, in the real-time test application order table in §2.1.3.
08/Jan/2020	RT Test 6, Global Range Test, updated for –5 <= PRES <= –2.4 dbar.
08/Jan/2020	§2: RT Test 15, Greylist Test, added to the trajectory section.
08/Jan/2020	§2.4: Feedback from Quality Codes updates in Coriolis database, updated by Delphine Dobler and Christine Coatanoan.
08/Jan/2020	§3: added qc flag and greylist instructions for large P ($\|\Delta P\| > 10$ dbar) and large S ($\|\Delta S\| > 0.05$ PSU) delayed-mode adjustments.
08/Jan/2020	§3.8: Added new section to describe how to record dmqc operators in D- and BD- files, following decisions from ADMT20 at Villefranche.
29/Jan/2021	Appendix 6.1: updated Argo QC flag table with improved comments and accompanying notes.
29/Jan/2021	§2.6: added real-time adjustment procedures for vertical profiles from Deep-Argo floats.
10/May/2021	§2.1.3: revised the application order of the real-time QC tests, so that the Greylist Test is the first test to be performed on the vertical profiles.
10/May/2021	§3.10: added new section on delayed-mode procedures for salinity from Deep-Argo floats with SBE CTDs.

Date (dd/mmm/yyyy)	Comment
10/Jan/2022	§1: Introduction updated. RT Test 19: Deepest Pressure Test, revised. §2.3: Added new section on how to manage intermediate parameters in the R- files. §3.9: Added new section on how to manage intermediate parameters in the D- files. §3.6: Added new section on best practices and examples on how to treat salinity sensor drift in delayed-mode, including Abrupt Salty Drifts. §2.7: Added new section on RBR real-time qc flagging and adjustment. RT Test 24 updated to flag PSAL_QC='1' for post-April2021 RBR 2K CTDs. §3.11: Added new section on RBR delayed-mode salinity adjustment. §4: Section on real-time procedures on trajectory file data, previously numbered §2.2, re-numbered as §4. §5: Added new section on delayed-mode procedures on trajectory file data. Appendix 6.2: Updated comments in Ref Table 2a. Appendix 6.8: Added information on how to estimate float ascent rates.
26/July/2022	§2.7.1: RT Test 26 added for identifying profiles with bad TEMP_CNDC. §2.7.2: 'A' mode RBR salinity adjustment, ctcoeff changed to 0.014. §3.11: Instructions added to deal with bad TEMP_CNDC in delayed-mode.
17/Mar/2023	§2.4: Updated section on Feedback from Quality Code Updates in Coriolis database. §2.6 and §3.10: Updated instructions to use optimized CPcor_new values in 'A' and 'D' mode salinity adjustment, for Deep Argo data from SBE CTDs. §3.11: Updated Step 2(d) in thermal inertia adjustment for RBR salinity data to include an option to estimate TEMP_CNDC when the original data are bad.
07/Feb/2024	§2.1.2: Modified RT Test 8 (Pressure increasing test), by Henry Bittig. §2.1.4: Clarified the general rule that when PRES_QC = '4' or '9', all PARAM_QC = '4'. §5.1: Added definition of a Dtraj file.
20/Feb/2025	§2.1.2 and §4: RTQC Test 15 Greylist test renamed to Supplemental sensor exclusion list test. §2.6.1: RTQC Test 23 for Deep Argo 'R' mode QC flags PRES_QC and TEMP_QC changed from '2' to '1'. §2.6.2: Deep Argo 'A' mode QC flags PRES_ADJUSTED_QC, TEMP_ADJUSTED_QC and PSAL_ADJUSTED_QC changed from '2' to '1'. §2.6.2 and §3.10: Deep Argo recommended value for CPcor_new for 'A' mode and 'D' mode adjustments changed to -11.7e-8 $dbar^{-1}$ for both the Deep SBE-41CP and SBE-61. §3.11.2: added section on Correction for the square-root salinity discrepancy for RBRargo3 CTD.

1. Introduction

A Conductivity-Temperature-Depth (CTD) device measures temperature and salinity versus pressure. Trajectory data involve positions and time. This document is the Argo Quality Control Manual for CTD and Trajectory data. It describes two levels of quality control and adjustment procedures:

- The first level is the real-time system that performs a set of automatic checks and adjustments.
- The second level is the delayed-mode system that consists of evaluation and adjustment of the data by experts.

These quality control and adjustment procedures are applied to the Argo parameters <PARAM> = JULD, LATITUDE, LONGITUDE, PRES, TEMP, PSAL. For biogeochemical parameters, please refer to *"Argo Quality Control Manual for Biogeochemical Data"*, http://dx.doi.org/10.13155/40879.

In every Argo data file, parameter values are stored in two variables: <PARAM> and <PARAM>_ADJUSTED. The variable <PARAM> stores the raw data telemetered directly from the float, without any post-deployment adjustment. The variable <PARAM>_ADJUSTED stores the data that have been evaluated/adjusted, either in real-time (DATA_MODE = 'A') or in delayed-mode (DATA_MODE = 'D'). When DATA_MODE = 'R', it means no evaluated/adjusted value is available.

Each of these variables has an associated quality control flag, _QC. In addition, when available, <PARAM>_ADJUSTED_ERROR provides the uncertainty of the evaluated/adjusted values.

Users are advised that the raw data can contain instrument errors. Therefore, when available, Argo evaluated/adjusted values, in conjunction with their QC flags and uncertainties, should be used.

Argo evaluation and adjustments are refreshed periodically. Users should therefore sync their data holding with the Argo GDACs frequently to obtain the most up-to-date evaluated/adjusted values, QC flags, and uncertainty estimates.

This manual is arranged as follows:

Section 2 describes the real-time procedures applicable to the single-cycle R files.
Section 3 describes the delayed-mode procedures applicable to the single-cycle D files.
Section 4 describes the real-time procedures applicable to the multi-cycle Rtraj files.
Section 5 describes the delayed-mode procedures applicable to the multi-cycle Dtraj files.

2. Real-time procedures on vertical profiles

2.1. Argo real-time quality control tests on vertical profiles

2.1.1. Introduction

Due to the requirement of delivering data to users within 12-24 hours of the floats reaching the sea surface, the real-time quality control test procedures on Argo data are automatic but limited. These tests are described here. All the data and their QC flags, including those having failed the tests, are distributed on the GTS in the BUFR format, and are converted to the agreed netCDF format and forwarded to the Argo Global Data Assembly Centers (GDACs).

Before 2018, Argo sent data on the GTS in the TESAC format. The TESAC format did not allow inclusion of QC flags, hence only good profile data (QC='1') were put on the GTS. Profile data were sent as a function of depth, not pressure. The UNESCO routines were used to convert pressure to depth (Algorithms for computation of fundamental properties of seawater, N.P. Fofonoff and R.C. Millard Jr., UNESCO Technical Papers in Marine Science #44, 1983). If the position of a profile was deemed wrong, or the date was deemed wrong, or the platform identification was in error, then none of the data was sent on the GTS. For other failures, only the offending values were removed from the TESAC message.

Since 2018, Argo has been sending data on the GTS in the BUFR format. In the BUFR format, all the data (including the bad data) with their QC flags are distributed on the GTS. The BUFR format contains data as a function of pressure. If the date or platform identification is wrong, then none of the data is sent on the GTS. However, if the position of a profile is questionable, the data will be distributed on the GTS with the appropriate QC flag for the position. The appropriate actions to take are noted with each real-time test.

2.1.2. Quality control tests on vertical profiles

The following tests are applied in real-time on vertical profile data.

Test 1. Platform identification test

Every data centre handling float data and posting them on the GTS will need to prepare a metadata file for each float and in this is the WMO ID number that corresponds to each float ptt. There is no reason why, except because of a mistake, an unknown float ID should appear on the GTS.

Action: If the correspondence between the float ptt cannot be matched to the correct WMO ID number, then none of the data from the float should be distributed on the GTS.

Test 2. Impossible date test

This test requires that the Julian day of a float profile be later than 1st January 1997 and earlier than the current date of the check (in UTC time). Hence if JULD = number of days elapsed since 1st January 1950, then this test checks that:

17167 <= JULD < UTC date of check

Action: If JULD of a profile fails this test, the date of the profile should be flagged as bad data ('4'), and none of the data from the profile should be distributed on the GTS.

Test 3. Impossible location test

This test requires that the observation latitude and longitude from a float profile be sensible.

- Latitude in range −90° to 90°
- Longitude in range −180° to 180°

Action: If either latitude or longitude fails this test, the position should be flagged as bad data ('4'), and the data from the profile should be distributed on the GTS.

Test 4. Position on land test

This test requires that the observation latitude and longitude from a float profile be located in an ocean. Use can be made of any topography/bathymetry file that allows an automatic test to check if a position is located on land. We suggest the use of at least the 5-minute bathymetry file that is generally available. This is commonly called ETOPO5/TerrainBase and can be downloaded from:

https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ngdc.mgg.dem:3141

Action: If a position is not located in an ocean, the position should be flagged as bad data ('4'), and the data from the profile should be distributed on the GTS.

Test 5. Impossible speed test

Drift speeds for floats can be generated by using the positions and times of the floats when they are at the sea surface and between profiles. In all cases we would not expect the drift speed to exceed 3 ms^{-1}. If it does, it means either the positions or times are bad data, or a float is mislabeled. Using the multiple positions that are normally available for a float while at the sea surface, it is often possible to isolate the one position or time that is in error.

Action: If an acceptable position and time can be used from the available suite, then the data can be distributed on the GTS. Otherwise, the position, the time, or both, should be flagged as bad data (‘4’), and the data from the profile should be distributed on the GTS.

(Note: Floats that use the ARGOS system to obtain positions can use Test 20 in Section 4.)

Test 6. Global range test

This test applies a gross filter on the observed values of pressure (PRES), temperature (TEMP) and salinity (PSAL). The ranges need to accommodate all the expected extremes encountered in the ocean.

- Pressure cannot be less than –5 dbar.
- Pressure in the range –5 to –2.4 dbar should be considered 'probably bad'. The limit of 2.4 dbar is determined based on feedback from operational users.
- Temperature should be in the range –2.5 to 40.0°C.
- Salinity should be in the range 2 to 41.0 psu.

Action:
If PRES < –5dbar, then PRES_QC = '4', TEMP_QC = '4', PSAL_QC = '4'.
If –5dbar <= PRES <= –2.4dbar, then PRES_QC = '3', TEMP_QC = '3', PSAL_QC = '3'.
If TEMP values are outside of the range –2.5 to 40.0°C, then TEMP_QC = '4'.
If PSAL values are outside of the range 2 to 41.0 psu, then PSAL_QC = '4'.

Test 7. Regional range test

This test applies to certain regions of the world where conditions can be further qualified. In this case, specific ranges for observations from the Mediterranean Sea and the Red Sea further restrict what are considered sensible values. The Red Sea is defined by the region 10N, 40E; 20N, 50E; 30N, 30E; 10N, 40E. The Mediterranean Sea is defined by the region 30N, 6W; 30N, 40E; 40N, 35E; 42N, 20E; 50N, 15E; 40N, 5W; 30N, 6W.
Red Sea

- Temperature should be in the range 21.0 to 40.0°C.
- Salinity should be in the range 2 to 41.0 psu.

Mediterranean Sea

- Temperature should be in the range 10.0 to 40.0°C.
- Salinity should be in the range 2 to 40.0 psu.

Action: If a value fails this test, it should be flagged as bad data (‘4’). If temperature and salinity values at the same pressure both fail this test, both values should be flagged as bad data (‘4’).

Test 8. Pressure increasing test

This test checks PRES data sanity based on expected float behavior, taking into account expected behavior of higher resolution data.

From deep to shallow, a measurement PRES fails the test if the pressure value V_1 is the same or larger than any previous value V_{pre} plus a plausible pressure reversal threshold, PRES_reversal, i.e.,

if $V_1 >= \min(V_{pre}) +$ PRES_reversal is true, then V_1 fails the test.

From shallow to deep, a measurement fails the test if the pressure value V_1 is the same or smaller than any previous value V_{pre} minus a plausible pressure reversal threshold, PRES_reversal, i.e.,

if $V_1 <= \max(V_{pre}) -$ PRES_reversal is true, then V_1 fails the test.

If a pressure value fails this test with PRES_reversal = 20 dbar, that pressure value should be flagged as bad with PRES_QC = ‘4’.

For near-surface profiles, this test should be run from the deepest pressure to the shallowest pressure. For all other profiles, this test should be run from the middle of the profile to the shallowest pressure, and from the middle of the profile to the deepest pressure. The middle of the profile is the pressure at length(profile)/2.

Action: If a PRES value fails this test with PRES_reversal = 20 dbar, then that PRES value should be flagged as bad data (PRES_QC = ’4’).

Test 9. Spike test

The difference between sequential measurements, where one measurement is significantly different from adjacent ones, is a spike in both size and gradient. This test does not consider differences in pressure, but assumes a sampling that adequately reproduces changes in temperature and salinity with pressure. The algorithm is used on vertical profiles of temperature and salinity.

$$\text{Test value} = |\, V2 - (V3 + V1)/2 \,| - |\, (V3 - V1) / 2 \,|$$

where V2 is the measurement being tested, and V1 and V3 are the values above and below.

Temperature: The V2 value is flagged when

- the test value exceeds 6.0°C for pressures less than 500 dbar, or
- the test value exceeds 2.0°C for pressures greater than or equal to 500 dbar.

Salinity: The V2 value is flagged when

- the test value exceeds 0.9 psu for pressures less than 500 dbar, or
- the test value exceeds 0.3 psu for pressures greater than or equal to 500 dbar.

Action: If the value V2 fails this test, it should be flagged as bad data (‘4’). If temperature and salinity values at the same pressure both fail this test, both values should be flagged as bad data (‘4’).

Test 10. Top and bottom spike test: *declared obsolete in October 2003 at ADMT4*

Test 11. Gradient test: *declared obsolete in October 2019 at ADMT20*

~~This test is failed when the difference between vertically adjacent measurements is too steep. The test does not consider differences in pressure but assumes a sampling that adequately reproduces changes in temperature and salinity with pressure. The algorithm is used on vertical profiles of temperature and salinity.~~

~~Test value = | V2 − (V3 + V1)/2 |~~

~~where V2 is the measurement being tested, and V1 and V3 are the values above and below.~~
~~Temperature: The V2 value is flagged when~~

- ~~the test value exceeds 9.0°C for pressures less than 500 dbar, or~~
- ~~the test value exceeds 3.0°C for pressures greater than or equal to 500 dbar.~~

~~Salinity: The V2 value is flagged when~~

- ~~the test value exceeds 1.5 PSU for pressures less than 500 dbar, or~~
- ~~the test value exceeds 0.5 PSU for pressures greater than or equal to 500 dbar.~~

~~Action: If the value V2 fails this test, it should be flagged as bad data ('4'), and only that value should be removed from TESAC distribution on the GTS. If temperature and salinity values at the same pressure both fail this test, both values should be flagged as bad data ('4'), and values for pressure, temperature and salinity should be removed from TESAC distribution on the GTS.~~

Test 12. Digit rollover test

Only so many bits are allowed to store temperature and salinity values in a profiling float. This range is not always large enough to accommodate conditions that are encountered in the ocean. When the range is exceeded, stored values rollover to the lower end of the range. This rollover should be detected and compensated for when profiles are constructed from the data stream from the float. This test is used to make sure the rollover is properly detected.

- Temperature difference between adjacent pressures > 10°C
- Salinity difference between adjacent pressures > 5 psu

Action: Values that fail the test should be flagged as bad data ('4'). If temperature and salinity values at the same pressure both fail this test, both values should be flagged as bad data ('4').

Test 13. Stuck value test

This test looks for measurements of temperature and salinity in a profile being identical.

Action: If this occurs, all the values of the affected parameter should be flagged as bad data ('4'). If both temperature and salinity are affected, then all observed values from the profile should be flagged as bad data ('4').

Test 14. Density inversion test

This test compares potential density between valid measurements in a profile, in both directions, ie. from top to bottom, and from bottom to top. Values of temperature and salinity at the same pressure level P_i should be used to compute potential density ρ_i (or $\sigma_i = \rho_i - 1000$)

kg m^{-3}, referenced to the mid-point between P_i and the next valid pressure level. A threshold of 0.03 kg m^{-3} should be allowed for small density inversions.

Action: From top to bottom, if the potential density calculated at the greater pressure P_{i+1} is less than that calculated at the lesser pressure P_i by more than 0.03 kg m^{-3}, both the temperature and salinity values at P_i should be flagged as bad data ('4'). From bottom to top, if the potential density calculated at the lesser pressure P_{i-1} is greater than that calculated at the greater pressure P_i by more than 0.03 kg m^{-3}, both the temperature and salinity values at P_i should be flagged as bad data ('4').

Test 15. Supplemental sensor exclusion list test

This test is implemented as a mechanism for data assembly centers (DACs) to flag, in real-time, sensors that are potentially not working properly.

The exclusion list contains the following 7 items:

- Platform: Float WMO ID
- Parameter: name of the affected parameter
- Start date: from that date, all measurements for this parameter are flagged as bad ('4') or probably bad ('3') or probably good ('2')
- End date: from that date, measurements are not flagged as bad ('4') or probably bad ('3') or probably good ('2')
- Flag: value of the flag to be applied to all measurements of the parameter
- Comment: comment from the PI on the problem
- DAC: data assembly center for this float

Example:

Float WMO ID	Parameter	Start date	End date	Flag	Comment	DAC
1900206	PSAL	20030925		3		IF

Each DAC manages an exclusion list in the following format:

- Exclusion list file: ascii csv (comma separated values)
- Naming convention: xxx_exclusionlist.csv
  xxx: DAC name (e.g. aoml_exclusionlist.csv, coriolis_exclusionlist.csv)
- PLATFORM, PARAMETER, START_DATE, END_DATE, QC, COMMENT, DAC
  e.g. 4900228, TEMP, 20030909, , 3, , AO
  e.g 1900206, PSAL, 20030925, , 3, , IF

The decision to insert a float parameter in the exclusion list comes from the PI or the delayed-mode operator. A float parameter should be put in the exclusion list when sensor drift or offset is too big to be corrected adequately in real-time, or when the sensor is judged to be potentially not working properly.

Another means when a float parameter can be put in the exclusion list is when it fails the MinMax test, which is documented in the Coriolis monthly report. DACs should put the offending float parameters in the exclusion list automatically, without contacting the PI or the delayed-mode operator, by using the start cycles and suggested QC flags in the MinMax test section of the Coriolis monthly report.

The exclusion list only concerns real-time files (R-files). When an anomalous float is dead and the offending parameter has been adjusted in delayed-mode, it should be removed from the exclusion list. When an anomalous float is active and the offending parameter has been partially adjusted in delayed-mode, it should remain in the exclusion list if real-time adjustment is not adequate.

Action: Affected parameters in the exclusion list are flagged as probably good data (‘2’) or probably bad data (‘3’) or bad data (‘4’), as determined by the PI or the delayed-mode operator or the MinMax test in the Coriolis monthly report. DACs should make sure to revisit the QC flags of the profiles whose observation dates are between the exclusion list “start date” and the exclusion list insertion date.

Test 16. Gross salinity or temperature sensor drift test

This test is implemented to detect a sudden and significant sensor drift. It calculates the average temperature and salinity from the deepest 100 dbar of a profile and the previous good profile. Only measurements with good QC flags are used.

Action: For salinity, if the difference between the two average values is more than 0.5 psu, then all salinity values from the profile are flagged as probably bad data (‘3’). For temperature, if the difference between the two average values is more than 1°C, then all temperature values from the profile are flagged as probably bad data (‘3’).

Test 17. Visual QC test

This is subjective visual inspection of float measurements by an operator. To avoid delays, this test is not mandatory before real-time distribution.

Test 18. Frozen profile test

This test is used to detect a float that reproduces the same profile (with very small deviations) repeatedly. Typically, the differences between two profiles are of the order of 0.001 psu for salinity and of the order of 0.01°C for temperature.

A). Derive temperature and salinity profiles by averaging the original profiles to get mean values for each profile in 50 dbar slabs (T_prof, T_previous_prof; S_prof, S_previous_prof). This is necessary because the floats do not sample at the same level for each profile.

B). Obtain absolute values of the difference between the averaged temperature and salinity profiles as follows:

- deltaT = abs(T_prof − T_previous_prof)
- deltaS = abs(S_prof − S_previous_prof)

C). Find the maximum, minimum and mean of the absolute values of the averaged differences between profiles for temperature and salinity:

- mean(deltaT), max(deltaT), min(deltaT)
- mean(deltaS), max(deltaS), min(deltaS)

D). To fail the test, require that:

- max(deltaT) < 0.3°C
- min(deltaT) < 0.001°C
- mean(deltaT) < 0.02°C
- max(deltaS) < 0.3 psu
- min(deltaS) < 0.001 psu
- mean(deltaS) < 0.004 psu

Action: If a profile fails this test, all measurements from the profile are flagged as bad data (‘4’). If a float fails this test on 5 consecutive cycles, it is inserted in the grey list.

Test 19. Deepest pressure test

This test requires that a profile has pressures that are not greater than a pressure threshold, where PRESSURE_THRESHOLD = CONFIG_ProfilePressure_dbar + tolerance. The tolerance is a percentage of CONFIG_ProfilePressure_dbar, where the percentage decreases linearly from 150% at CONFIG_ProfilePressure_dbar = 10 dbar, to 10% at CONFIG_ProfilePressure_dbar = 1000 dbar. For CONFIG_ProfilePressure_dbar > 1000 dbar, the tolerance is 100 dbar. The varying threshold is to allow for floats having difficulties to stabilize precisely at shallower pressures.

The value of CONFIG_ProfilePressure_dbar comes from the meta.nc file of the float. DACs are reminded to check that these records are correct; otherwise, profiles will fail or pass this test erroneously.

Action: When PRES > PRESSURE_THRESHOLD, those pressures and their corresponding temperature and salinity measurements should be flagged with PRES_QC = '3', TEMP_QC = '3', PSAL_QC = '3'.

The following table provides the various pressure threshold values.

CONFIG_ProfilePressure_dbar	**percentage (%)**	**tolerance (dbar)**	**PRESSURE_THRESHOLD (dbar)**
10	150.00	15.00	25.00
20	148.59	29.72	49.72
30	147.17	44.15	74.15
40	145.76	58.30	98.30
50	144.34	72.17	122.17
60	142.93	85.76	145.76
70	141.52	99.06	169.06
80	140.10	112.08	192.08
90	138.69	124.82	214.82
100	137.27	137.27	237.27
150	130.20	195.30	345.30
200	123.13	246.26	446.26
250	116.06	290.15	540.15
300	108.99	326.97	626.97

350	101.92	356.72	706.72
400	94.85	379.39	779.39
450	87.78	395.00	845.00
500	80.71	403.54	903.54
550	73.64	405.00	955.00
600	66.57	399.39	999.39
650	59.49	386.72	1036.72
700	52.42	366.97	1066.97
750	45.35	340.15	1090.15
800	38.28	306.26	1106.26
850	31.21	265.30	1115.30
900	24.14	217.27	1117.27
950	17.07	162.17	1112.17
1000	10.00	100.00	1100.00
>1000	n/a	100.00	CONFIG_ProfilePressure_dbar + 100.00

Test 25. MEDian with a Distance (MEDD) test

This test is a set of algorithms based on three main steps:

- First, the computation of a sliding median with some customizations.
- Then, limits are computed that are at relative 2-dimensional distance *d* from the median.
- Finally, these limits are also computed for the density profile.

There is a spike if both the density profile and the temperature or salinity profile are out of limits. If there is no conductivity sensor, then the spikes in temperature are evaluated using a bigger *d* value.

Detailed specifications and Matlab codes for this test can be found on:

https://github.com/ArgoRTQC/matlab_MEDD

Action: Temperature and salinity values that fail this test should be flagged as bad data ('4').

2.1.3. Tests application order on vertical profiles

The Argo real-time QC tests on vertical profiles are applied in the order described in the following table.

A measurement with QC flag ‘4’ (bad data) is ignored by other quality control tests.
A measurement with QC flag '2' or '3' is tested by other quality control tests.

Application Order	Test Number (n)	Test Name
1	1	Platform Identification Test
2	2	Impossible Date Test
3	3	Impossible Location Test
4	4	Position on Land Test
5	5	Impossible Speed Test
6	15	Grey List Test
7	19	Deepest Pressure Test
8	6	Global Range Test
9	7	Regional Range Test
10	8	Pressure Increasing Test
11	9	Spike Test
12	25	MEDD Test
13	12	Digit Rollover Test
14	13	Stuck Value Test
15	14	Density Inversion Test
16	16	Gross Salinity or Temperature Sensor Drift Test
17	18	Frozen Profile Test
18	23	*specific test for Deep SBE CTD data > 2000 dbar*
19	26	*specific test for RBRargo^3|2K TEMP_CNDC data*
20	24	*specific test for RBRargo^3|2K CTD data < 2000 dbar*
21	17	Visual QC Test

Please note that the Test Number (n) is different from the Application Order. The Test Number (n) is a unique number assigned permanently to each QC test. It is used to fill HISTORY_QCTEST in the Argo profile files (see Section 6.3). Therefore, each Test Number (n) is associated uniquely to a QC test, and is never replaced, changed, or duplicated.

2.1.4. Quality control flag application policy

The following are some general rules on how to apply QC flags.

(a). The QC flag value assigned by a test cannot override a higher value from a previous test.

Example: a QC flag ‘4’ (bad data) set by the Grey List Test cannot be decreased to QC flag ‘3’ (bad data that are potentially correctable) set by the Gross Salinity or Temperature Sensor Drift Test.

(b). Because salinity (PSAL) is calculated from temperature (TEMP) and conductivity (CNDC), if temperature is flagged ‘4’ (or ‘3’), then salinity is flagged ‘4’ (or ‘3’).

For floats where CNDC values are reported, CNDC_QC = PSAL_QC.

(c). For all Argo real-time QC tests, when PRES_QC = ‘4’ or ‘9’ at a level, then all <PARAM>_QC = ‘4’ at that level, while <PARAM>_QC at other levels should be left unchanged.

Here, all <PARAM> means all core parameters (c param), all BGC parameters (b param), and all intermediate parameters (ic and ib param) of the profile.

Note that when a <PARAM> (c, b, ic, ib) does not have any RTQC tests, and therefore <PARAM>_QC = ‘0’, this general rule should be applied as follows:

- set <PARAM>_QC = '4' where PRES_QC = '4' or ‘9’ and set <PARAM>_QC = '0' on other levels

2.2. Argo real-time adjustments on vertical profiles

2.2.1. Real-time pressure adjustment for non auto-correcting floats

There are many float types in Argo, each has its own way of treating the pressure measurements. For example, while PROVOR and SOLO floats internally correct for pressure offsets, APEX and NAVIS floats do not make any internal pressure correction. All Argo float types and their treatment of pressure are summarized in the "Surface Pressure Correction Table", which can be found at http://www.argodatamgt.org/Documentation.

Floats that do not adjust their pressure data onboard before data telemetry are referred to as *non auto-correcting* floats. Pressure measurements from non auto-correcting floats need to be treated in real-time whenever valid surface pressure values are available. This section describes the real-time pressure adjustment procedures for APEX and NAVIS floats, which are the largest groups of non auto-correcting floats in Argo.

Real-time pressure adjustment for APEX and NAVIS floats

APEX and NAVIS floats return "raw" pressures, which are stored in the variable PRES in the Argo data files. Pressure adjustment should be applied in real-time to all APEX and NAVIS floats by using SURFACE PRESSURE (SP) values returned by these floats. The SP measurement is taken while the float is at the sea surface just before descent, and hence is different from the shallowest measured pressure in the vertical profile, which is taken on ascent and while the float is beneath the sea surface. These SP values are stored in the Argo technical files. Depending on the type of controller used, the variables are:

- PRES_SurfaceOffsetTruncatedPlus5dbar_dbar, or
- PRES_SurfaceOffsetNotTruncated_dbar.

Subtract 5 dbar from the SP values in PRES_SurfaceOffsetTruncatedPlus5dbar_dbar. SP values in PRES_SurfaceOffsetNotTruncated_dbar are used as they are, without needing to subtract 5 dbar. Then, erroneous outliers in SP need to be removed. This is done in real-time in two steps:

(1). Discard SP values > 20 dbar or < –20 dbar, then revert to the last valid SP.

(2). If the most recent SP value, SP(i), is different from the last valid SP value by more than 5 dbar, that is, if abs[SP(i) – last valid SP] > 5 dbar, revert to the last valid SP.

When there are valid SP values, real-time adjusted pressures should be recorded in the variable PRES_ADJUSTED, where PRES_ADJUSTED = PRES – SP.

When no valid SP value is available, no real-time pressure adjustment is available.

Zero is a valid SP value. Therefore, when SP = 0, PRES_ADJUSTED should be filled. In this case, PRES_ADJUSTED = PRES.

PRES should always record the *raw* data.

PRES_ADJUSTED_QC should be filled with the same values as PRES_QC.

PRES_ADJUSTED_ERROR and all variables in the SCIENTIFIC CALIBRATION section of the Argo data files should be filled with FillValue.

DATA_MODE should record 'A'.

There is no need to re-calculate salinity in real-time by using the real-time adjusted pressure values. This is because the difference in salinity due to real-time pressure adjustment is small. Pressure adjustments of < 20 dbar will result in salinity differences < 0.01.

When the SP value exceeds 20 dbar (or −20 dbar) for more than 5 consecutive cycles, PRES from the float should be put on the grey list because of pressure error.

When available, real-time adjusted values should be distributed on the GTS, instead of the raw values.

2.2.2. Real-time salinity adjustment

When delayed-mode salinity adjustment (see Section 3.3) becomes available for a float and cycle N is the last available D cycle, real-time DACs should extract the adjustment from the last D-file as an additive constant and apply it to the salinity data in the R-files, starting from cycle N+1. In this manner, intermediate-quality salinity data will be available to users in real-time.

The values of this real-time adjustment should be recorded in PSAL_ADJUSTED.

PSAL_ADJUSTED_QC should be filled with the same values as PSAL_QC.

PSAL_ADJUSTED_ERROR and all variables in the SCIENTIFIC CALIBRATION section of the Argo data files should be filled with FillValue.

DATA_MODE should record ‘A’.

When available, real-time adjusted values should be distributed on the GTS, instead of the raw values.

2.2.3. Real-time files with DATA_MODE = ‘A’

When real-time R-files have DATA_MODE = ‘A’, it means real-time adjustments are available for one or more parameters. Unfortunately, since there is not a separate variable to indicate which parameter has real-time adjustment, all PARAM_ADJUSTED variables should therefore be filled, where PARAM = PRES, TEMP, PSAL. (*Note: in the B-files, a separate variable PARAMETER_DATE_MODE is available to indicate the data mode of each parameter.*)

PARAM_ADJUSTED = real-time adjusted values, or PARAM if no real-time adjustment is available.

PARAM_ADJUSTED_QC = PARAM_QC.

PARAM_ADJUSTED_ERROR = FillValue.

Users should be aware that even though the _ADJUSTED_ fields are filled in ‘A’ mode, the adjustments are applied in an automated manner in real-time and are not checked by delayed-mode operators.

2.3. How to manage intermediate parameters in the R files

There are some intermediate parameters that can be stored in the core profile files as optional parameters. These are information from the CTD or the float that contribute to the understanding or computation of the core parameters. (For example: NB_SAMPLE_CTD, MTIME and TEMP_CNDC in relation to sensor response; PRES_MED, TEMP_MED, PSAL_MED, TEMP_STD, and PSAL_STD in relation to bin-averaging statistics.) Unless specified (e.g. TEMP_CNDC, Section 2.7.1), they do not receive any QC tests or adjustments.

These core intermediate parameters are denoted by "ic" in Reference Table 3. Their associated _QC and _ADJUSTED variables in the R files should be managed as follows:

- No need to include their _ADJUSTED, _ADJUSTED_QC, _ADJUSTED_ERROR variables.
- Unless specified (e.g. TEMP_CNDC, Section 2.7.1), <PARAM>_QC = '0'. If the sensor fails completely, PIs can instruct DACs to record <PARAM>_QC = '4'.
- PROFILE_<PARAM>_QC should be computed from <PARAM>_QC. If <PARAM>_QC = '0', then PROFILE_<PARAM>_QC = ' '. If the sensor fails completely and <PARAM>_QC = '4' is recorded, then PROFILE_<PARAM>_QC = 'F'.

For completeness and for adherence to the dimension N_PARAM, intermediate parameters should be included in PARAMETER in the SCIENTIFIC_CALIB section, and should be managed as follows:

- When the R file has DATA_MODE = 'R', no real-time adjustment is available and none of the _ADJUSTED_ variables for the core parameters are filled. Hence there is no need to fill the SCIENTIFIC_CALIB section. For the "ic" <PARAM>:

SCIENTIFIC_CALIB_EQUATION	= FillValue
SCIENTIFIC_CALIB_COEFFICIENT	= FillValue
SCIENTIFIC_CALIB_COMMENT	= FillValue
SCIENTIFIC_CALIB_DATE	= FillValue

- When the R file has DATA_MODE = 'A', all the _ADJUSTED_ variables for the core parameters are filled. Hence none of the SCIENTIFIC_CALIB information should be set to FillValue (ie. empty). For the "ic" <PARAM>:

SCIENTIFIC_CALIB_EQUATION	= 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENT	= 'not applicable'
SCIENTIFIC_CALIB_COMMENT	= 'not applicable'
SCIENTIFIC_CALIB_DATE	= YYYYMMDDHHMISS

Special case: MTIME for floats with multiple controller boards

MTIME is an optional "ic" parameter used to report raw timing information for some profile measurements. It is a relative time, relative to JULD. The purpose of using a relative time instead of an absolute time is to reduce the QC effort needed for MTIME. By using JULD as the reference for MTIME, only JULD needs to be adjusted in the event of clock drift, while MTIME does not need to be adjusted. See https://doi.org/10.13155/47998 for details.

Due to the above reason, MTIME_ADJUSTED should not be used, and MTIME should contain the best timing information available at the time of the profile file creation. For floats that have multiple controller boards (e.g. for multiple sensors), MTIME can be adjusted during real-time decoding to resolve inconsistencies between the different controller boards. The relevant information related to the real-time decoding adjustment can be recorded in the SCIENTIFIC_CALIB section for MTIME. Please note the DATA_MODE of the profile file is not affected by this real-time decoding adjustment between controller boards for MTIME.

2.4. Feedback from quality code updates in Coriolis database

When a file is detected to have been uploaded/updated on the GDAC, it is also uploaded/updated on the Coriolis Oracle database. These files then receive supplementary tests. The resulting alerts are checked visually by using the Scoop software and the QC flags are modified in the Coriolis database where appropriate. These alerts are based on the MinMax test (Gourrion et al., 2020):

- tested every hour on profiles not tested yet
- checked visually daily
- daily results from this test are compiled in a report once a month

The MinMax test can detect contaminated portions of a profile, as well as anomalies caused by sensor drift. Please refer to this paper for more details:

Gourrion J., Szekely T., Killick R., Owens W.B., Reverdin G., Chapron B. (2020). Improved Statistical Method for Quality Control of Hydrographic Observations. Journal Of Atmospheric and Oceanic Technology, 37(5), 789-806.
https://doi.org/10.1175/JTECH-D-18-0244.1

Whenever a quality code is changed in the Coriolis database, an automatic message (.csv file) is generated and sent to the corresponding DAC. The list of generated messages is also available on the Ifremer ftp site:

ftp://ftp.ifremer.fr/ifremer/argo/etc/ObjectiveAnalysisWarning

csv file naming conventions

ar_scoop_XX_YYYYMMDDHHMISS.csv

- ar: bigram code for Argo
- scoop: software that is used to make the change of QC flag
- XX: bigram code for DAC
- YYYYMMDDHHMISS: Date and time of generation of the .csv file

Example: ar_scoop_AO_20090121125021.csv

These .csv files are ftp daily to DACs and compiled into monthly pdf reports.

csv file content description

One line is generated for every station and vertically continuous {PARAM, QC_old, QC_new} change done in Coriolis database using Scoop software.

Definition of the fields in the line:

- DAC_CODE: bigram code of the Data Assembly Center of the platform
- PLATFORM_CODE: WMO code of the platform
- CYCLE: Cycle number of the station
- DATE_UPDATE: Last ingestion in Coriolis database of the GDAC file (this should correspond to the file submission on GDAC)
- DIRECTION: ‘A’ = Ascending and ‘D’ = descending
- WEB_URL QC: URL to access the web display of the station

- PARAMETER: parameter for which the QC flag has been changed. The used parameter codes are the following:
    - GF3codes for physical parameters (PSAL, PSAL_ADJUSTED, TEMP, TEMP_ADJUSTED, etc.)
    - POS$ for position
    - DAT$ for date
- START_IMMERSION: pressure in dbar corresponding to the shallowest pressure where the QC flag has been changed for a vertically continuous {PARAM, QC_old, QC_new}
- STOP_IMMERSION: pressure in dbar corresponding to the deepest pressure where the QC flag has been changed for a vertically continuous {PARAM, QC_old, QC_new}
- OLD_QC: QC flag before change
- NEW_QC: QC flag after change
- VERTICAL_SAMPLING_SCHEME: ‘Primary sampling’ or ‘Secondary sampling’ or ‘Near-surface sampling’
- START_IMMLEVEL: measure number corresponding to START_IMMERSION
- STOP_IMMLEVEL: measure number corresponding to STOP_IMMERSION
- PROFILE_NUMBER: profile index in the GDAC file

If PARAMETER is ‘POS$’, then START_IMMERSION, STOP_IMMERSION, START_IMMLEVEL and STOP_IMMLEVEL are left empty.

A report (**Report on anomalies on Argo profiles**) is compiled and sent monthly to the argo-dm mailing list with details for each DAC of the floats for which anomalies have been detected during the month. Three files are also provided each month corresponding to the table available in this report. These files correspond to:

- drift_<*month*><*year*>_new: new floats detected drifting during this month drift_january2023_new
- drift_<*month*><*year*>_previous_reports: floats already on alert during the previous months (up to 2 months back)
- drift_<*month*><*year*>_listed: floats that were greylisted

The report and files are also available respectively on the Ifremer web server:

https://data-argo.ifremer.fr/etc/Report_ObjectiveAnalysisWarning/

https://data-argo.ifremer.fr/etc/Report_ObjectiveAnalysisWarning/csv_drift/

2.5. Real-time quality control tests on near-surface data

The near-surface data described in this section are specialised data that are collected with vertical sampling methods that are different from the primary CTD profiles. For most profiling floats, the CTD pump is normally switched off at around 5 dbar during ascent to avoid contamination of the conductivity cell. Several float types continue to sample up to the sea surface (with the pump off, or with the pump on closer to the sea surface) or carry auxiliary modules for high-resolution near-surface sampling. These specialised near-surface data are focused on the top 5 dbar of the ocean. They may extend deeper than 5 dbar so as to overlap with the primary CTD profiles for the purpose of cross-calibration. They are stored as additional profiles (N_PROF > 1) in the single-cycle core Argo profile files, and are identifiable by **VERTICAL_SAMPLING_SCHEME = "Near-surface sampling"**.

(**Note**: the full character string is "Near-surface sampling: averaged/discrete/mixed, pumped/unpumped [optional description]". Please refer to Table 16 in the Argo Users Manual for details of the various vertical sampling schemes and their full character strings.)

The following tests are applied in real-time to these specialised near-surface data.

Test 6. Global range test
Same as in Section 2.1.2.

Test 7. Regional range test
Same as in Section 2.1.2.

Test 8. Pressure increasing test
Same as in Section 2.1.2.

Test 9. Spike test
Same as in Section 2.1.2.

~~**Test 11. Gradient test**~~

Test 19. Deepest pressure test
Same as in Section 2.1.2.

Test 25. MEDD test
Same as in Section 2.1.2.

Test 21. Near-surface unpumped CTD salinity test
Details described in this section.

Test 22. Near-surface mixed air/water test
Details described in this section.

Test 21. Near-surface unpumped CTD salinity test

While unpumped temperature data from the primary CTD are of good quality, unpumped salinity data from the primary CTD are of dubious quality because the flow rate through the conductivity cell is wrong when the pump is switched off. This test specifies that unpumped (or partially pumped) salinity data returned by the primary CTD should be flagged as "probably bad data" in real-time.

Action: Unpumped (or partially pumped) salinity data returned by the primary CTD are flagged as "probably bad data" in real-time. That is, PSAL_QC = '3'.

(a). PROVOR/ARVOR (bin-averaged data)
Data returned by some PROVOR/ARVOR floats are bin-averaged and are not separated into pumped and unpumped types. This separation is done during data processing at the DACs by checking when the CTD pump is switched off. In addition, for some older versions of PROVOR/ARVOR floats, some bin-averaged data can contain a mixture of pumped and unpumped measurements if the pump cut-off pressure falls in the middle of the bin. Please refer to the document "Argo DAC profile cookbook" on how to identify unpumped or partially pumped near-surface data from PROVOR/ARVOR floats.

(b). PROVOR/ARVOR (spot-sampled data)
Some PROVOR/ARVOR floats use the primary CTD to measure spot-sampled near-surface data in the umpumped mode. Spot-sampled near-surface data are identified as unpumped when PRES<=Pcutoff, where Pcutoff is the pressure at which the CTD pump is switched off.

(c). NOVA
Some NOVA floats collect near-surface data by using the primary CTD in the unpumped mode, with the CTD pump switched off at 4 dbar. Near-surface data from NOVA floats are bin-averaged and are identified as unpumped when PRES <= 4 dbar.

(d). APEX
Test 21 does not apply to near-surface data returned by APEX floats because APEX (ARGOS) floats with the NST firmware do not report unpumped salinity from the primary CTD. APEX (Iridium) floats equipped with the STS module return unpumped salinity from the auxiliary module. The STS module is designed to measure salinity in a free-flushing mode and so is different from the primary CTD.

Test 22. Near-surface mixed air/water test

Most near-surface profiles extend all the way to the sea surface. Therefore, the shallowest part of a near-surface profile will contain some mixed air/water measurements. This test identifies broadly the pressures at which this shallowest part of a near-surface profile takes place and specifies that data in that pressure range are “probably bad data”.

Action: Data from the shallowest part of a near-surface profile, which may contain mixed air/water measurements, are flagged as “probably bad data” in real-time.

(a). PROVOR/ARVOR (bin-averaged data)
For PROVOR/ARVOR floats that return bin-averaged data, if the first bin closest to the sea surface has PRES <= 1 dbar, then the temperature value from that first bin is suspected to contain averages of mixed air/water measurements and should be flagged as “probably bad data”. That is, TEMP_QC = ‘3’.

(b). PROVOR/ARVOR (spot-sampled data)
For PROVOR/ARVOR floats that return spot-sampled data, temperature observed at PRES <= 0.5 dbar should be flagged as “probably bad data”. That is, TEMP_QC = ‘3’.

(c). NOVA
For NOVA floats, if the first bin closest to the sea surface has PRES <=1 dbar, then the temperature value from that first bin is suspected to contain averages of mixed air/water measurements and should be flagged as “probably bad data”. That is, TEMP_QC = ‘3’.

(d). APEX
The shallowest part of near-surface profiles collected by APEX floats will contain pressure readings that are not necessarily monotonic with time.

For APEX (ARGOS) floats with the NST firmware, when near-surface data have pressures shallower than 5 dbar, check difference in pressure between two successive measurements. If the difference is less than 0.5 dbar from the previous measurement, then data from that level and all levels after that should be flagged as “probably bad data”. That is, PRES_QC = ‘3’, TEMP_QC = ‘3’.

For APEX (Iridium) floats equipped with the STS module, when near-surface data have pressures shallower than 5 dbar, check that pressure readings decrease monotonically with time. Data from the level when monotonicity stops and all levels after that should be flagged as “probably bad data”. That is, PRES_QC = ‘3’, TEMP_QC = ‘3’, PSAL_QC = ‘3’.

Various types of specialised near-surface vertical profiles in Argo that can be found under VERTICAL_SAMPLING_SCHEME = "Near-surface sampling"

Float type	primary CTD or auxiliary module	operating mode	averaged or discrete	Test 21	Test 22
PROVOR/ARVOR (bin-averaged)	primary CTD	unpumped T & S	averaged	a	a
PROVOR/ARVOR (spot-sampled)	primary CTD	unpumped T & S	discrete	a	a
NOVA	primary CTD	unpumped T & S	averaged	a	a
APEX (ARGOS) NST	primary CTD	unpumped T only	discrete	n/a	a
APEX (Iridium) STS	auxiliary module	free-flushing T & S	discrete	n/a	a
SOLO II/S2A	primary CTD	pumped T & S	discrete	n/a	n/a
Deep SOLO	primary CTD	pumped T & S	discrete	n/a	n/a

a = applicable
n/a = not applicable

2.6. Real-time quality control tests and adjustments for Deep-Argo vertical profiles from SBE CTDs

The Deep-Argo program aims to sample temperature and salinity over the full ocean depth to 6000 dbar. There are two Sea-Bird CTDs that are capable of deep sampling: the SBE-61 (used on Deep APEX and Deep SOLO), and the Deep SBE-41CP (used on Deep NINJA and Deep ARVOR). Users are warned that their salinity data show a bias that is dependent on pressure, due to compressibility effects.

2.6.1. Real-time QC flag scheme for Deep-Argo data from SBE CTDs

This section describes the real-time quality control flag scheme that should be applied to the Deep-Argo data in 'R' mode, to warn users that the raw data from Deep-Argo CTDs may not be suitable for research applications requiring high data accuracy.

Test 23. Real-time quality control flag scheme for Deep SBE CTD data > 2000 dbar

This test specifies that for float data deeper than 2000 dbar:

- real-time QC flag = ‘1’ for pressure (PRES) and temperature (TEMP);
- real-time QC flag = ‘3’ for salinity (PSAL); and
- distribute all raw data and their QC flags on GTS.

Action: For profiles from Deep-Argo SBE CTDs that extend deeper than 2000 dbar, real-time QC Tests 1 to 19 and Test 25 should be applied to the full-depth profiles. Float data shallower than 2000 dbar should be flagged as in Section 2.1. Float data deeper than 2000 dbar that have passed all other real-time QC tests should be flagged with PRES_QC = ‘1’, TEMP_QC = ‘1’, PSAL_QC = ‘3’. All float data and their QC flags should be distributed on the GTS in BUFR.

2.6.2. Real-time adjustments for Deep-Argo data from SBE CTDs

This section describes the adjustment procedures that should be applied to the Deep-Argo vertical profiles from SBE CTDs in real-time, 'A' mode.

Pressure and temperature profiles from Deep-Argo floats should be adjusted in real-time, 'A' mode, as per the vertical profiles from the 2000-dbar floats (Section 2.2).

Salinity profiles from Deep-Argo floats with Sea-Bird CTDs will need real-time adjustment to remove the pressure-dependent salinity bias due to compressibility. This is done by using a new CPcor term to correct conductivity. CPcor is the correction term for pressure effects on conductivity, or conductivity cell compressibility. Comparisons of data from the SBE-61 and the Deep SBE-41CP against shipboard CTD data showed that the CPcor value from Sea-Bird, **CPcor_SBE**, was too small, thus resulting in a significant fresh salinity bias at high pressures. A new CPcor value, **CPcor_new**, should be used in real-time, 'A' mode, to adjust salinity from Deep-Argo floats with Sea-Bird CTDs, according to the following procedure.

Real-time salinity adjustment for Deep-Argo floats with SBE CTDs

(a). Fill PRES_ADJUSTED and TEMP_ADJUSTED

Follow the same 'A' mode procedures as for the 2000-dbar floats, as in Section 2.2.

(b). Compute original conductivity, Co

This is done by using PRES, TEMP, PSAL. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox, then Co = gsw_C_from_SP (PSAL, TEMP, PRES).

(c). Compute new conductivity, Cnew

Cnew = Co*(1 + d*TEMP + CPcor_SBE*PRES) / (1 + d*TEMP_ADJUSTED + CPcor_new*PRES_ADJUSTED); where

d = 3.25e–6;
CPcor_SBE = –9.57e–8 dbar^{-1};
unless an optimized CPcor_new value is provided by the float's PI,
CPcor_new = –11.7e–8 dbar^{-1} for SBE-61 data;
CPcor_new = –11.7e–8 dbar^{-1} for Deep SBE-41CP data.

(d). Compute new salinity, PSAL_ADJUSTED_Cnew

This is done by using PRES_ADJUSTED, TEMP_ADJUSTED, Cnew. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox, then PSAL_ADJUSTED_Cnew = gsw_SP_from_C (Cnew, TEMP_ADJUSTED, PRES_ADJUSTED).

** Please note that by using PRES_ADJUSTED, this step also takes into account the effect of APEX/NAVIS pressure adjustment on salinity. **

(e). Compute final PSAL_ADJUSTED

If real-time 'A' mode salinity sensor drift adjustment is available (see Section 2.2.2), then PSAL_ADJUSTED = PSAL_ADJUSTED_Cnew + salinity sensor drift 'A' mode adjustment.

If real-time 'A' mode salinity sensor drift adjustment is not available, then PSAL_ADJUSTED = PSAL_ADJUSTED_Cnew.

In 'A' mode, ADJUSTED_QC and ADJUSTED_ERROR for Deep-Argo data should be filled as follows

PRES_ADJUSTED_ERROR = (2.5/6000) * PRES + 2.
PRES_ADJUSTED_QC = '1' below 2000 dbar, if all rtqc tests are passed.

TEMP_ADJUSTED_ERROR = 0.002°C.
TEMP_ADJUSTED_QC = '1' below 2000 dbar, if all rtqc tests are passed.

PSAL_ADJUSTED_ERROR = 0.004.
PSAL_ADJUSTED_QC = '1' below 2000 dbar, if all rtqc tests are passed.

Note that in 'A' mode, all raw <PARAM>_QC should remain the same as they are in 'R' mode. When available, the real-time adjusted values <PARAM>_ADJUSTED, together with <PARAM>_ADJUSTED_QC, should be distributed on the GTS in the BUFR format.

In 'A' mode, the SCIENTIFIC_CALIB section for 'PSAL' should be filled as follows when CPcor correction has been applied to Deep-Argo salinity data

N_PARAM = index for 'PSAL'

N_CALIB = 1

SCIENTIFIC_CALIB_EQUATION = 'new conductivity = original conductivity * (1 + delta*TEMP + CPcor_SBE*PRES) / (1 + delta*TEMP_ADJUSTED + CPcor_new*PRES_ADJUSTED)'

SCIENTIFIC_CALIB_COEFFICIENT = 'CPcor_new = xxx; CPcor_SBE = –9.57e–8; delta = 3.25e–6'

SCIENTIFIC_CALIB_COMMENT = 'New conductivity computed by using a different CPcor value from that provided by Sea-Bird.'

N_CALIB > 1

Other real-time salinity adjustments, such as those from sensor drifts, can be recorded in N_CALIB = 2, 3, ..., etc.

Note to DACs

To provide sufficient space for delayed-mode operators to record all calibration information, it is recommended that DACs preset N_CALIB > 1 (= 2 at least) for Deep-Argo profiles during the creation of the profile files. Please coordinate with the delayed-mode operator.

2.6.3. Summary of real-time QC flags for Deep-Argo data from SBE CTDs

Two steps to adjust Deep-Argo salinity from SBE CTDs in real-time:

- CPcor correction (see Section 2.6.2)
- 'A' mode salinity sensor drift adjustment (see Section 2.2.2)

If all real-time QC tests are passed:

RAW DATA ('R' MODE)	ADJUSTED DATA ('A' MODE)
> 2000 dbar	
PRES_QC = '1'	PRES_ADJUSTED_QC = '1'
TEMP_QC = '1'	TEMP_ADJUSTED_QC = '1'
PSAL_QC = '3' (because of increased CPcor-related error at high pressures)	PSAL_ADJUSTED_QC = '1'
Distribute raw data on GTS if real-time adjusted data are not available.	Distribute real-time adjusted data on GTS if available.
<= 2000 dbar	
PRES_QC = '1'	PRES_ADJUSTED_QC = '1'
TEMP_QC = '1'	TEMP_ADJUSTED_QC = '1'
PSAL_QC = '1'	PSAL_ADJUSTED_QC = '1'
Distribute raw data on GTS if real-time adjusted data are not available.	Distribute real-time adjusted data on GTS if available.

2.7. Real-time quality control tests and adjustments for RBR*argo*^{3}|2K vertical profiles

Some Argo floats are equipped with the RBR*argo*^{3}|2K CTD. Users are warned that salinity from the RBR*argo*^{3}|2K CTD calibrated pre-April 2021 can have a bias that is due to compressibility effects, which are not fully resolved by the suboptimal class-based calibration coefficients. This has been rectified in the RBR CTD calibrated post-April 2021, when RBR changed to unit-based compressibility calibration. The list of WMO ID of Argo floats with RBR*argo*^{3}|2K CTD calibrated pre-April 2021 can be found in Appendix 6.9 and in: https://github.com/ArgoDMQC/RBRargo_DMQC/blob/main/Compressibility_correction/RBRargo3_compressibility_table.csv

2.7.1. Real-time QC Flag Scheme for RBR*argo*^{3}|2K data

The following real-time quality control flag scheme should be applied to RBR*argo*^{3}|2K CTD data (< 2000 dbar) in 'R' mode after all the other RTQC tests. Please refer to the Tests Application Order in Section 2.1.3, and the QC flags Application Policy in Section 2.1.4.

Test 26. TEMP_CNDC test

TEMP_CNDC is used in the long timescale thermal inertia correction for RBR salinity data. This test aims to identify vertical profiles where TEMP_CNDC is of bad quality and thus unsuitable for use in the real-time adjustment of RBR salinity data.

A vertical profile of TEMP_CNDC is considered bad ('4') and unsuitable for use in the real-time application of the long timescale thermal inertia correction when the difference between TEMP_CNDC and TEMP exceeds Tlimit on more than 2 measurement levels. Tlimit = 1.5°C at PRES > 500 dbar; Tlimit = 10°C at PRES <= 500 dbar. For these vertical profiles, the long timescale thermal inertia correction should **not** be applied in the real-time adjustment for RBR salinity data. In addition, bad TEMP_CNDC may affect conductivity; therefore, PSAL should be marked as '3' (probably bad).

Action: For each vertical profile

- if |TEMP_CNDC – TEMP| > Tlimit on more than 2 measurement levels, then TEMP_CNDC_QC = '4' and PSAL_QC = '3' for all levels of the profile; skip Step 2 (long timescale thermal inertia correction) in the real-time adjustment of RBR salinity;
- otherwise, TEMP_CNDC_QC = '1'.

Tlimit = 1.5°C at PRES > 500 dbar; Tlimit = 10°C at PRES <= 500 dbar.

Test 24. RTQC flag scheme for RBR*argo*^{3}|2K CTD data (< 2000 dbar)

This test specifies that for RBR*argo*^{3}|2K CTD data that have passed all the other RTQC tests:

- real-time QC flag = ‘1’ for PRES and TEMP;
- for RBR*argo*^{3}|2K CTD calibrated pre-April2021, real-time QC flag = '3' for PSAL;
- for RBR*argo*^{3}|2K CTD calibrated post-April2021, real-time QC flag = '1' for PSAL.

Action: Float data (< 2000 dbar) from RBR*argo*^{3}|2K CTD that have passed all other real-time QC tests should be flagged with PRES_QC = '1' and TEMP_QC = '1'. For RBR*argo*^{3}|2K CTD calibrated pre-April 2021, PSAL_QC = '3'. For RBR*argo*^{3}|2K CTD calibrated post-April 2021, PSAL_QC = '1'. Data can be distributed on the GTS in BUFR format with QC flags.

2.7.2. Real-time adjustments for RBR*argo*^{3}|2K data

This section describes the adjustment procedures that should be applied to RBR*argo*^{3}|2K CTD data (< 2000 dbar) in real-time, 'A' mode.

Pressure measurements from RBR*argo*^{3}|2K CTDs should be adjusted in real-time, 'A' mode, as per the 2000-dbar floats, described in Section 2.2.

Salinity data from RBR*argo*^{3}|2K CTDs need real-time adjustment to remove the salinity bias associated with the long timescale thermal inertia error. In addition, salinity from RBR*argo*^{3}|2K CTDs calibrated pre-April 2021 need to be re-computed with new compressibility coefficients. These new RBR compressibility coefficients can be found at: https://github.com/ArgoDMQC/RBRargo_DMQC/blob/main/Compressibility_correction/RBRargo3_compressibility_table.csv

The following describes the steps involved in adjusting salinity from RBR*argo*^{3}|2K CTDs in real-time, 'A' mode.

Step 1. Re-compute salinity due to compressibility effect

For pre-April 2021 calibrated RBR CTDs with new compressibility coefficients, follow Step 1: a), b), c), d1), then proceed to Steps 2 and 3.

For post-April 2021 calibrated RBR CTDs, and pre-April 2021 calibrated RBR CTDs with no new compressibility coefficients available (NaN), follow Step 1: a), b), d2), then proceed to Steps 2 and 3.

(a). Fill PRES_ADJUSTED and TEMP_ADJUSTED

Follow the same 'A' mode procedures as per other 2000-dbar CTDs, described in Section 2.2.

(b). Compute original conductivity, Co

This is done by using PRES, TEMP, PSAL. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox, Co = gsw_C_from_SP (PSAL, TEMP, PRES).

(c). Compute new conductivity, Cnew

$$\mathrm{Cnew} = \mathrm{Co} * \frac{1 + \mathrm{X2old} * \mathrm{PRES} + \mathrm{X3old} * \mathrm{PRES}^2 + \mathrm{X4old} * \mathrm{PRES}^3}{1 + \mathrm{X2new} * \mathrm{PRES\_ADJUSTED} + \mathrm{X3new} * \mathrm{PRES\_ADJUSTED}^2 + \mathrm{X4new} * \mathrm{PRES\_ADJUSTED}^3}$$

where

$\mathrm{X2old} = 1.8732 \times 10^{-6}$, X2new = see lookup table for each RBR*argo*^{3}|2K CTD;
$\mathrm{X3old} = -7.7689 \times 10^{-10}$, X3new = see lookup table for each RBR*argo*^{3}|2K CTD;
$\mathrm{X4old} = 1.4890 \times 10^{-13}$, X4new = see lookup table for each RBR*argo*^{3}|2K CTD.

(d1). Compute new salinity, PSAL_ADJUSTED_Padj

This is done by using PRES_ADJUSTED, TEMP_ADJUSTED, and Cnew. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox:
PSAL_ADJUSTED_Padj
= gsw_SP_from_C (Cnew, TEMP_ADJUSTED, PRES_ADJUSTED).

** Please note that by using PRES_ADJUSTED, this step also takes into account the effect of APEX/NAVIS pressure adjustment on salinity. **

(d2). PSAL_ADJUSTED_Padj
= gsw_SP_from_C (Co, TEMP_ADJUSTED, PRES_ADJUSTED).

Step 2. Apply long timescale thermal inertia correction

This step requires the intermediate parameter TEMP_CNDC (the internal temperature of the conductivity cell). If the CTD malfunctions and causes data corruption in TEMP_CNDC (TEMP_CNDC_QC = '4'), then this step should be skipped. In that case, PSAL_QC = '3' (see Test 26) and PSAL_ADJUSTED_QC = '3'.

The following should be applied when TEMP_CNDC_QC = '1'.

(a). Calculate conductivity, Cadj

This is done by using PRES_ADJUSTED, TEMP_ADJUSTED, and PSAL_ADJUSTED_Padj from Step1. For example: Cadj
= gsw_C_from_SP (PSAL_ADJUSTED_Padj, TEMP_ADJUSTED, PRES_ADJUSTED).

(b). Compute TEMP_longanomaly

TEMP_longanomaly = ctcoeff * (TEMP_CNDC − TEMP), where ctcoeff = **0.014**.

(c). Use TEMP_longanomaly to compute salinity PSAL_ADJUSTED_Padj_CTM

PSAL_ADJUSTED_Padj_CTM
= gsw_SP_from_C (Cadj, TEMP_ADJUSTED + TEMP_longanomaly, PRES_ADJUSTED).

Step 3. Compute final PSAL_ADJUSTED in 'A' mode

If real-time 'A' mode salinity sensor drift adjustment is available (see Section 2.2.2), then PSAL_ADJUSTED = PSAL_ADJUSTED_Padj_CTM + salinity drift 'A' mode adjustment.

If real-time 'A' mode salinity sensor drift adjustment is not available, then PSAL_ADJUSTED = PSAL_ADJUSTED_Padj_CTM.

In 'A' mode, ADJUSTED_QC and ADJUSTED_ERROR for RBR*argo*^{3}|2K CTD data should be filled as follows

In 'A' mode, all <PARAM>_QC should remain the same as they are in 'R' mode.

PRES_ADJUSTED_QC = '1', if all rtqc tests are passed.
TEMP_ADJUSTED_QC = '1', if all rtqc tests are passed.
PSAL_ADJUSTED_QC = '1', if all rtqc tests are passed.

If available, the real-time adjusted values <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_QC should be distributed on the GTS in the BUFR format.

PRES_ADJUSTED_ERROR = FillValue.
TEMP_ADJUSTED_ERROR = FillValue.
PSAL_ADJUSTED_ERROR = FillValue.

In 'A' mode, the SCIENTIFIC_CALIB section should be filled as follows

N_PARAM = index for 'PRES' and 'TEMP'

As per other CTDs.

N_PARAM = index for 'TEMP_CNDC'

Please fill these with 'not applicable'. See Section 2.3.

N_PARAM = index for 'PSAL'

N_CALIB = 1. Compressibility correction.

Example a). RBR CTDs calibrated pre-April 2021 and have new compressibility coefficients available in lookup table

SCIENTIFIC_CALIB_EQUATION = 'new conductivity = original conductivity * (1 + X2old*PRES + X3old*PRES^2 + X4old*PRES^3) / (1 + X2new*PRES_ADJUSTED + X3new*PRES_ADJUSTED^2 + X4new*PRES_ADJUSTED^3)'

SCIENTIFIC_CALIB_COEFFICIENT = 'X2old = 1.8732e-6, X3old = -7.7689e-10, X4old = 1.4890e-13, X2new = ..., X3new = ..., X4new = ...'

SCIENTIFIC_CALIB_COMMENT = 'Pre-April 2021 calibrated RBR CTD. Salinity re-computed by using new compressibility coefficients provided by RBR.'

Example b). RBR CTDs calibrated pre-April 2021 but have no new compressibility coefficients available (N/A) in lookup table

SCIENTIFIC_CALIB_EQUATION = 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENTS = 'not applicable'
SCIENTIFIC_CALIB_COMMENTS = 'Pre-April 2021 calibrated RBR CTD. No new compressibility coefficients available.'

Example c). RBR CTDs calibrated post-April 2021

SCIENTIFIC_CALIB_EQUATION = 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENTS = 'not applicable'
SCIENTIFIC_CALIB_COMMENTS = 'Post-April 2021 calibrated RBR CTD. No compressibility correction needed.'

N_CALIB = 2. Long timescale thermal inertia correction.

Example a). Correction applied

SCIENTIFIC_CALIB_EQUATION = 'PSAL_ADJUSTED = gsw_SP_from_C (Cadj, TEMP_ADJUSTED + TEMP_longanomaly, PRES_ADJUSTED); TEMP_longanomaly = ctcoeff * (TEMP_CNDC – TEMP); Cadj is conductivity from re-computed salinity due to compressibility effect.'

SCIENTIFIC_CALIB_COEFFICIENT = 'ctcoeff = 0.014'

SCIENTIFIC_CALIB_COMMENT = 'Long timescale thermal inertia correction applied to RBR salinity.'

Example b). Correction not applied

SCIENTIFIC_CALIB_EQUATION = 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENTS = 'not applicable'
SCIENTIFIC_CALIB_COMMENTS = 'Long timescale thermal inertia correction not applied to RBR salinity.'

N_CALIB = 3. Real-time salinity adjustment for sensor drift and offset.

As per other CTDs.

<u>Note to DACs</u>

To provide sufficient space for delayed-mode operators to record all calibration information, it is recommended that DACs preset N_CALIB > 1 (= 3 at least) for RBR profiles at the creation of the profile files. Please coordinate with the delayed-mode operators.

2.7.3. Summary of real-time QC flags for RBR*argo*³|2K data

Three steps to adjust RBR*argo*³|2K salinity in real-time:

- Compressibility correction for RBR*argo*³|2K CTDs calibrated pre-April 2021 (see Section 2.7.2)
- Thermal inertia correction (long timescale only, see Section 2.7.2)
- 'A' mode salinity sensor drift adjustment (see Section 2.2.2)

If all real-time QC tests are passed:

RAW DATA ('R' MODE)	ADJUSTED DATA ('A' MODE)
PRES_QC = '1'	PRES_ADJUSTED_QC = '1'
TEMP_QC = '1'	TEMP_ADJUSTED_QC = '1'
PSAL_QC = '1' for RBR*argo*³|2K CTDs calibrated post-April 2021	PSAL_ADJUSTED_QC = '1' RBR*argo*³|2K CTDs calibrated post-April 2021
PSAL_QC = '3' for RBR*argo*³|2K CTDs calibrated pre-April 2021 (because of known compressibility error)	PSAL_ADJUSTED_QC = '1' RBR*argo*³|2K CTDs calibrated pre-April 2021
Distribute raw data on GTS if real-time adjusted data are not available.	Distribute real-time adjusted data on GTS if available.

3. Delayed-mode procedures on vertical profiles

3.1. Editing raw QC flags in delayed-mode

The QC flags determined in delayed-mode should replace those determined in real-time. This is because some bad data cannot be detected by the real-time tests, and some good data can be identified wrongly as bad by the real-time tests.

For vertical profile data in PARAM, where PARAM = PRES, TEMP, PSAL and CNDC, delayed-mode operators should examine them for pointwise errors (such as spikes) and flag them appropriately. If an error is identified, both PARAM_QC and PARAM_ADJUSTED_QC should record ‘4’. Conversely, if good data have been identified wrongly as bad by the real-time tests, then both PARAM_QC and PARAM_ADJUSTED_QC should record ‘1’.

3.2. Delayed-mode procedures for JULD, LATITUDE, LONGITUDE

Delayed-mode operators should check that JULD in the profiles are in chronological order. Erroneous or missing JULD values should be replaced with another telemetered value if available, or replaced with interpolated values and marked with JULD_QC = ‘8’.

Profile positions in LONGITUDE, LATITUDE should be checked for outliers. Erroneous or missing LONGITUDE, LATITTUDE values should be replaced with another telemetered value if available, or replaced with estimated values and marked with POSITION_QC = ‘8’. Two optional variables: POSITION_ERROR_ESTIMATED and POSITION_ERROR_ESTIMATED_COMMENT, can be filled.

3.3. Delayed-mode procedures for pressure

Delayed-mode QC for pressure is done by visual assessment of vertical profile plots of TEMP vs. PRES, and PSAL vs. PRES. This assessment should be done in relation to measurements from the same float, as well as in relation to nearby floats and historical data. The assessment should aim to identify: (a) erroneous data points that cannot be detected by the real-time qc tests, and (b) vertical profiles that have the wrong shape, which could indicate erroneous pressure measurements.

Bad data points identified by visual inspection in delayed-mode are recorded with PRES_ADJUSTED_QC = ‘4’ and PRES_QC = '4'. For these bad data points, TEMP_QC, TEMP_ADJUSTED_QC, PSAL_QC, PSAL_ADJUSTED_QC should also be set to ‘4’. Please note that whenever <PARAM>_ADJUSTED_QC = ‘4’, both <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_ERROR should be set to FillValue.

3.3.1. Delayed-mode pressure adjustment for APEX and NAVIS floats

Similar to the real-time procedure, pressures from APEX and NAVIS floats should be adjusted for offsets by using SURFACE PRESSURE (SP) values in delayed-mode. These SP values are stored in the Argo technical files, depending on the type of controller, as:

- PRES_SurfaceOffsetNotTruncated_dbar, or
- PRES_SurfaceOffsetTruncatedPlus5dbar_dbar.

The SP time series is examined and treated in delayed-mode as follows:

(1). Subtract 5 dbar from PRES_SurfaceOffsetTruncatedPlus5dbar_dbar. SP values in PRES_SurfaceOffsetNotTruncated_dbar are used as they are, without needing to subtract 5dbar.
(2). Despike the SP time series to 1 dbar relative to neighbouring points. This is most effectively done by first removing the more conspicuous spikes that are bigger than 5 dbar (as in the real-time procedure), then the more subtle spikes that are between 1 to 5 dbar by comparing the SP values with those derived from a 5-point median filter. For standard Argo floats that sample every 10 days, a 5-point filter represents a filter window of 40 days (+/− 20 days from a profile), which is an appropriate time scale for retaining effects from the atmospheric seasonal cycle.
(3). Replace the spikes and any other missing SP values by interpolating between good neighbouring points. If missing values occur at the ends of the SP time series, extrapolate from the nearest good points.
(4). The resulting SP time series should then be inspected visually to make sure there are no more erroneous points.

The clean SP value from cycle *i*+1 is then used to adjust pressures from cycle *i* by:

PRES_ADJUSTED (cycle *i*) = PRES (cycle *i*) – SP (cycle *i*+1).

The CTD profile and the associated SP are staggered by one cycle for all APEX and NAVIS floats, because the SP measurement is taken after the end of telemetry for cycle *i*, just prior to descent to begin cycle *i*+1. It is then stored in the memory and telemetered in the message file for cycle *i*+1. This feature has remained the same for all controllers (Apf5, Apf7, Apf8, Apf9a, Apf9i, Apf11i, N1, N2). The real-time procedure does not match SP value from cycle *i*+1 with PRES from cycle *i*, because real-time adjustment cannot wait 10 days. However, in delayed-mode, it is important to match the CTD profile with the staggered telemetry of SP, because SP values can contain synoptic atmospheric variations, and because a missing CTD profile is often associated with an erroneous SP point. By this scheme, SP(1), which is taken before cycle 1 and therefore before the float has had its first full dive, is not used in delayed-mode.

Note that the real-time procedure does not adjust for pressure offsets that are greater than 20 dbar (or less than −20 dbar). This is because the real-time automatic procedure cannot determine whether SP values greater than 20 dbar (or less than −20 dbar) represent true sensor drift or erroneous measurements. Instead, in real-time, floats that return SP values greater than 20 dbar (or less than −20 dbar) for more than 5 consecutive cycles are greylisted in consultation with the PI. In delayed-mode, operators can visually inspect the SP time series and the float measurements when severe pressure sensor drift occurs, to determine if the pressure measurements are adjustable or not. Note that the method of using SP values to adjust pressure works well when the sensor drift has no significant variation. In general, when the magnitude of pressure sensor drift exceeds 10 dbar, a depth-independent offset adjustment may not be sufficient. Therefore, 10 dbar should be considered the upper limit of good pressure adjustment in delayed-mode.

If the magnitude of pressure sensor drift exceeds 10 dbar, ie. $|\Delta P| > 10$ dbar, the delayed-mode operator should consider the following two options:

(1). Pressure data are considered bad and unadjustable in delayed-mode. Therefore,

PRES_ADJUSTED_QC = '4'; PRES_QC = '4'.

(2). Pressure data are considered adjustable in delayed-mode, but PRES_ADJUSTED_QC and PRES_QC should be set appropriately for such large presure adjustments. Therefore,

PRES_ADJUSTED_QC = '2'; PRES_QC = '3'.

In both cases, pressure should be greylisted.

Other points to note:

- After pressure adjustment, delayed-mode operators should check that PRES_ADJUSTED > 0. If PRES_ADJUSTED < 0, please check for decoding error in SP or in the CTD pressures.
- PRES should always record the *raw* data.
- PRES_ADJUSTED_ERROR = 2.4 dbar is the recommended error to quote, with 2.4 dbar being the manufacturer quoted accuracy of the pressure sensor.
- Salinity should be re-calculated by using PRES_ADJUSTED and recorded in PSAL_ADJUSTED. Salinity error due to pressure uncertainty is negligible and can be ignored in the consideration of PSAL_ADJUSTED_ERROR.
- Please use the SCIENTIFIC CALIBRATION section in the netCDF files to record details of the delayed-mode adjustment.

Note to users: The 1-dbar despiking threshold (relative to neighbouring points) for SP assumes that spikes greater than 1 dbar relative to neighbouring points represent noise in the SP measurement that should not be integrated into float pressures. After despiking to 1 dbar relative to neighbouring points, the remaining SP values contain sea surface atmospheric pressure variations and variations due to other high-frequency surface processes. While sea surface atmospheric pressure variations affect the whole water column and therefore should be adjusted for, high-frequency surface processes do not affect the whole water column. Therefore, users should be aware that PRES_ADJUSTED contains noise from high-frequency surface processes that are of the order < 1 dbar. In addition, other more subtle pressure errors such as those due to non-linear hysteresis and other temperature- and pressure-dependent effects are not accounted for in PRES_ADJUSTED. Hence users should always heed the error bars quoted in PRES_ADJUSTED_ERROR.

3.3.2. Truncated Negative Pressure Drift (TNPD) in APEX floats

APEX floats with the Apf-5, Apf-7, or Apf-8 controllers that set negative SURFACE PRESSURE (SP) to zero (then add an artificial 5 dbar) present a challenge to delayed-mode qc because information from SP on any negative pressure offset is lost, thus making the pressure data unadjustable[1]. The problem with some of these APEX floats having unknown negative pressure error escalated with the discovery of the oil microleak defect in Druck pressure sensors. The Druck oil microleak defect manifests itself as increasingly negative offset at all pressures and will eventually end the useful life of the float.

During delayed-mode qc of pressure measurements from APEX floats with the Apf-5, Apf-7, or Apf-8 controllers, operators should first remove erroneous SP values and any isolated spikes in the time series by following the procedure described in Section 3.2.1. The delayed-mode operator should then examine the resulting valid and despiked SP time series, and determine whether there are long periods of zero SP readings (after removing the artificial 5 dbar) that qualify as "Truncated Negative Pressure Drift", which has the following definition:

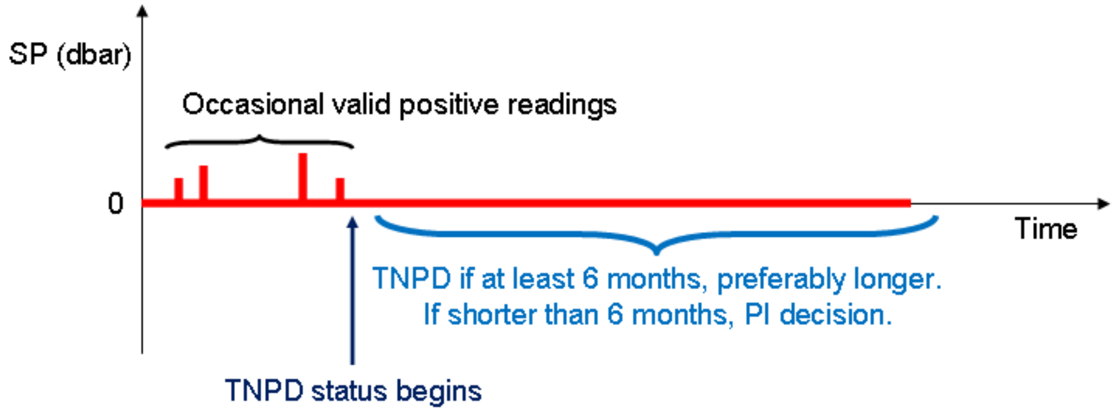
*Truncated Negative Pressure Drift (TNPD) refers to the part of a float's time series from which **valid and despiked** SP (after removing the artificial 5 dbar) reads continuously zero without reverting back to positive values or containing any occasional positive values. The continuous valid zero-reading period needs to span at least 6 months, preferably longer. This captures the microleakers whose oil leak rates are fastest and allows for seasonal variability from half of an annual cycle when surface pressure values may read just below zero. For floats whose useful life is less than 6 months or when the continuous valid zero-reading period is shorter than 6 months, the qualifying time span is at the PI's discretion.*

Examples (a) to (d) below illustrate some cases that should or should not be classified as TNPD. Please note that in all of the following schematic examples, SP represents valid and despiked values (after removing the artificial 5 dbar).

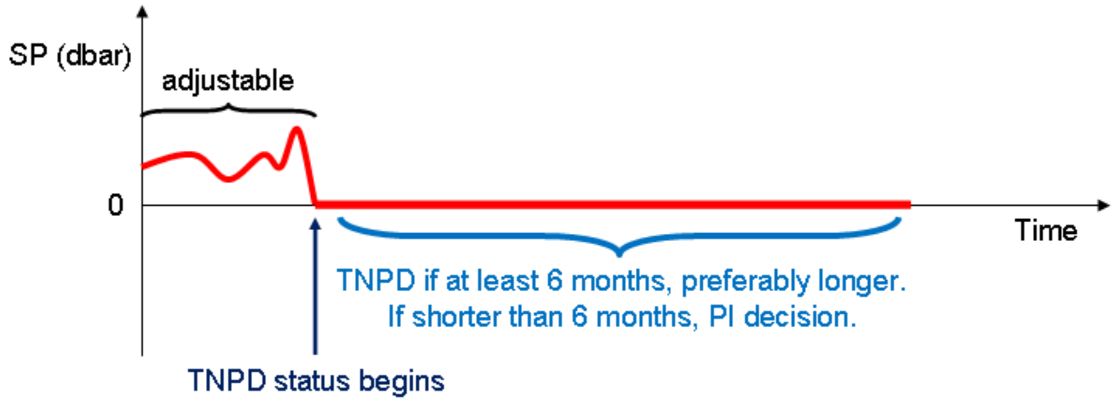
Example (a). 100% of the time series is TNPD.

[1] This feature was corrected in the Apf-9 and later versions of the controller.

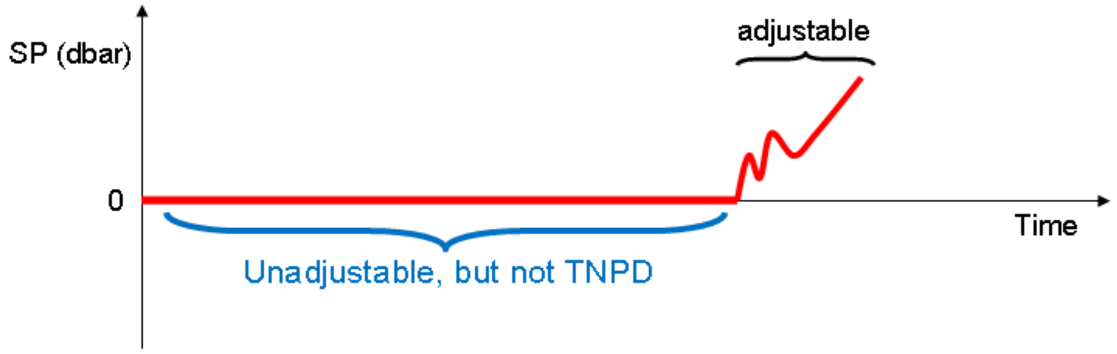
Example (b). There are occasional valid positive SP readings in the first part of the time series, followed by a continuous zero-reading period that does not contain any occasional valid positive readings.



Example (c). The time series starts with continuous valid positive readings, then becomes continuously zero with no occasional valid positive readings.



Example (d). The time series starts with a continuous zero-reading period, then reverts back to valid positive values. The initial zero-reading period does not qualify as TNPD. This is because pressure drifts are typically monotonic and therefore a reversal back to positive values indicates that the pressure sensor is not likely to have developed a negative drift.



After determining which part of the time series qualifies as TNPD, the delayed-mode operator should then determine the probability of the TNPD data being affected by the Druck oil microleak problem. According to SeaBird, the date of manufacturing change at Druck that led to the oil microleak defect occurred sometime in mid-2006. The microleak failure rate jumped from 3% before 2006 to 30% in 2007. Any Druck pressure sensor with serial number greater than **2324175** falls into the group that has 30% likelihood of being affected by oil microleaks. Cross checking between the various APEX groups within Argo indicated that deployment of floats with Druck serial number greater than 2324175 occurred after October 2006. Since July 2009 SeaBird has begun screening Druck pressure sensors in order to identify those transducers that have microleaks.

One way to identify affected floats is by T/S analysis, since severe pressure error will lead to observable T/S anomalies. Anomalies associated with severe negative pressure drift include:
(a). Positive salinity drift; e.g. pressure error of −20 dbar will cause a positive salinity error of approximately 0.01 PSS-78. Comparison methods that are used to determine conductivity sensor drift (e.g. OWC) can be used as diagnostic tools for these cases. Please refer to Section 3.5 for descriptions of these comparison methods for salinity.
(b). Cold temperature anomaly whose size depends on vertical temperature gradient.
(c). Float-derived dynamic height anomalies significantly lower than satellite-derived sea level anomalies.
(d). Shoaling of isotherm depths independent of time/space migration of the float.

In addition to T/S analysis, delayed-mode operators should also observe when a float begins telemetering highly erratic data. This is a sign that it may be suffering from the Druck oil microleak problem, and that the pressure sensor may be about to fail completely. Note that symptoms of this failure are very similar to those of the Druck "snowflakes" problem, which affected floats that were manufactured in 2002 and 2003. Please refer to Appendix 6.3 for a brief description of the Druck "snowflakes" problem.

In light of these events, the following 2 categories should be considered in assigning delayed-mode qc flags and error bars for data classified as TNPD.

1. <u>When float data do not show observable T/S anomalies that are consistent with increasingly negative pressure drift.</u> This means that the TNPD data may have unknown negative pressure error that is not severe.

For these less severe cases, the adjusted variables should receive a dmqc flag of '2':

PRES_ADJUSTED_QC = '2'
TEMP_ADJUSTED_QC = '2'
PSAL_ADJUSTED_QC = '2'.

(Note that TEMP_ADJUSTED_QC and PSAL_ADJUSTED_QC can change to '3' or '4' if TEMP and PSAL contain additional errors that are independent of the pressure error, e.g. pointwise temperature spike, conductivity cell drift, etc.)

For these less severe cases, two groups should be considered in assigning the pressure error:

(a). For TNPD data belonging to floats that used Druck pressure sensors with serial numbers less than 2324175, or were deployed before 1 October 2006 if the Druck serial numbers are unknown, the likelihood of them being affected by the oil microleak problem is low, about 3%. Hence it is reasonable to cite the manufacturer quoted accuracy of 2.4 dbar as the pressure error for this group:

PRES_ADJUSTED_ERROR = 2.4 dbar.

(b). For TNPD data belonging to floats that used Druck pressure sensors with serial numbers greater than 2324175, or were deployed after 1 October 2006 if the Druck serial numbers are unknown, the likelihood of them being affected by the microleak disease is elevated, about 30%. For these suspicious data, an *upper bound* of the estimated error should be cited. Since a negative 20 dbar pressure error will cause a positive 0.01 salinity error, at which point T/S anomalies will become observable and data should be flagged as '4' as described in Category 2(b) below, 20 dbar has been chosen as the upper bound of the data error for this group:

PRES_ADJUSTED_ERROR = 20 dbar.

Note that SeaBird will eventually provide a list of serial numbers that represents Druck sensors that have been screened as healthy. These healthy Druck sensors should be excluded from receiving the larger pressure error bar. Moreover, SeaBird has records that connect the Druck serial number to CTD number, and Teledyne WRC can make the connection to the float hull number.

2. <u>When float data show observable T/S anomalies that are consistent with increasingly negative pressure drift after cycle-*n*.</u> This means that the TNPD data have unknown negative pressure error, and that the error becomes severe after cycle-*n*.

(a). For the less severe part of the TNPD data before cycle-*n*, the adjusted variables should receive a dmqc flag of '2':

PRES_ADJUSTED_QC = '2'
TEMP_ADJUSTED_QC = '2'
PSAL_ADJUSTED_QC = '2',

while the pressure error should increase to 20 dbar:

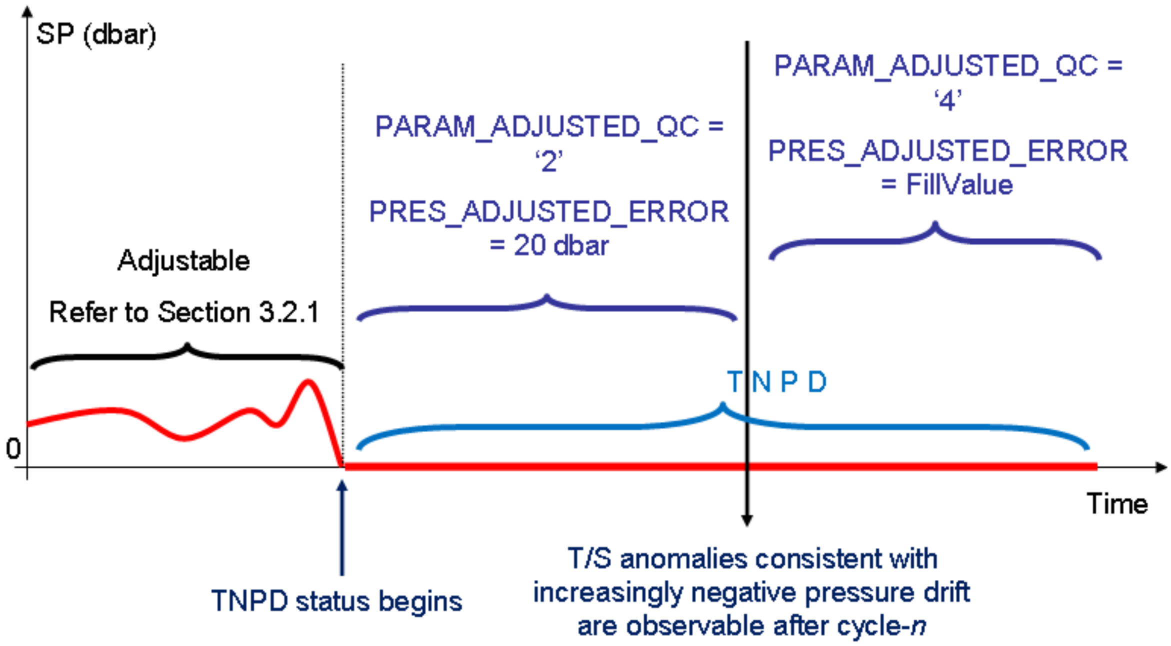
PRES_ADJUSTED_ERROR = 20 dbar.

(b). For the severe part of the TNPD data after cycle-*n*, the adjusted variables should receive a dmqc flag of '4':

PRES_ADJUSTED_QC = '4'
TEMP_ADJUSTED_QC = '4'
PSAL_ADJUSTED_QC = '4'.

Please note that whenever PARAM_ADJUSTED_QC = '4', PARAM_ADJUSTED = FillValue, and PARAM_ADJUSTED_ERROR = FillValue.

Example (e). A complex case belonging to Category 2, where T/S anomalies consistent with increasingly negative pressure drift are observed after cycle-*n*, part way through the TNPD portion of the time series.



All TNPD data should receive a standard label in SCIENTIFIC_CALIB_COMMENT in the Argo single-cycle netcdf files, in the dimension corresponding to PRES. The standard label consists of the character string **"TNPD: APEX float that truncated negative pressure drift".** The delayed-mode operator may append to the end of this character string any other comments regarding PRES that he/she wishes to make.

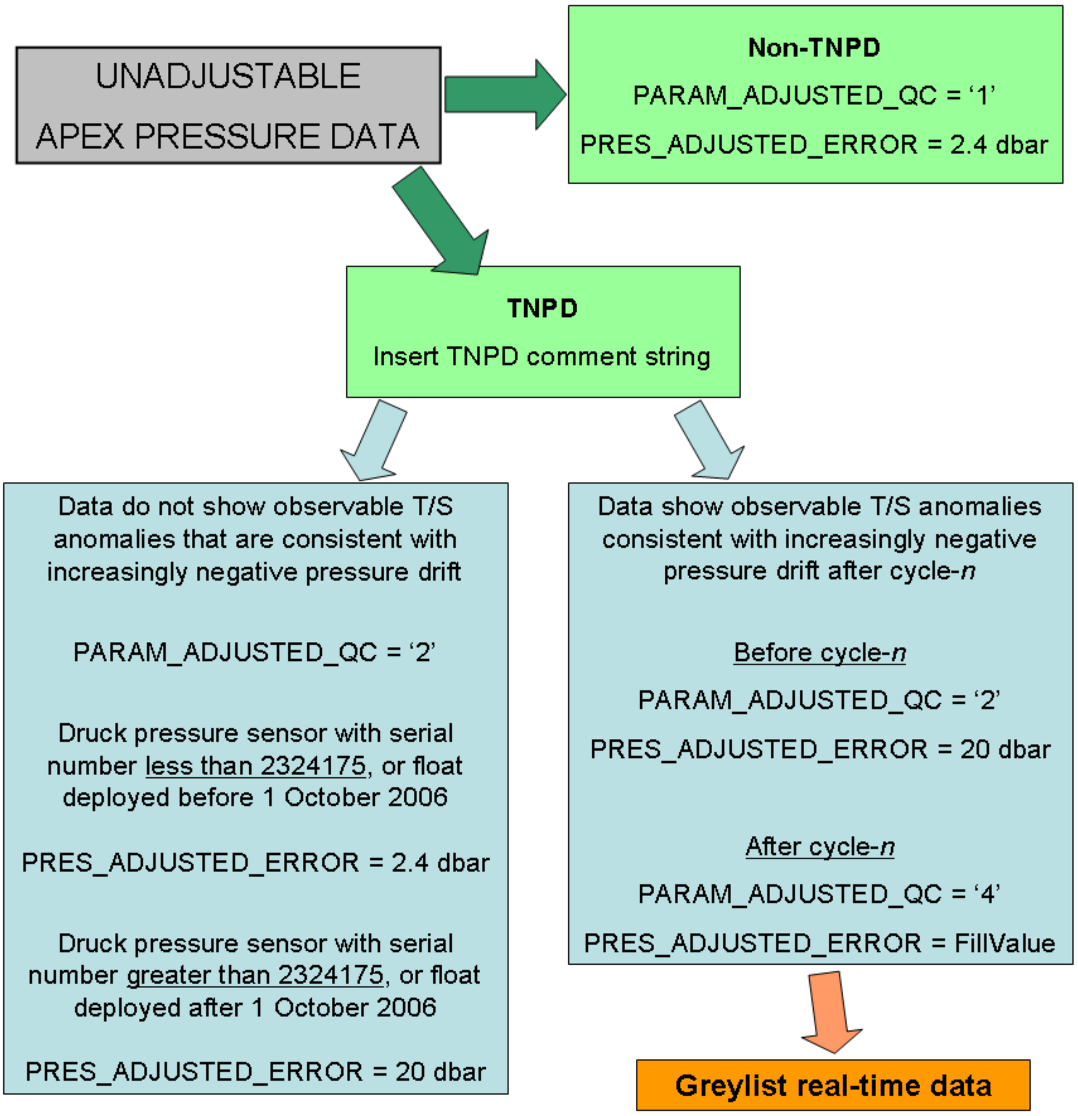
For the portion of the time series that contains occasional valid positive SP readings (Example b), it is the PI's decision, based on the frequency of occurrence of the valid positive SP readings, on whether or not to adjust those profiles.

For the unadjustable but non-TNPD data (Example d, and cases in Example b where the PI decides to not adjust), any negative pressure offset is likely to be less than the manufacturer quoted accuracy of 2.4 dbar. For these unadjustable but non-TNPD data, if no additional error is found, then:

PRES_ADJUSTED_QC = '1'
TEMP_ADJUSTED_QC = '1'
PSAL_ADJUSTED_QC = '1'
PRES_ADJUSTED_ERROR = 2.4 dbar.

3.3.3. Summary flowchart for unadjustable APEX pressures

Procedures for processing unadjustable APEX pressure data in delayed-mode



3.4. Delayed-mode procedures for temperature

Delayed-mode QC for temperature is done by visual assessment of vertical profile plots of TEMP vs. PRES, and PSAL vs. TEMP. This assessment should be done in relation to measurements from the same float, as well as in relation to nearby floats and historical data. The assessment should aim to identify: (a) erroneous data points that cannot be detected by the real-time qc tests, and (b) vertical profiles that have the wrong shape.

Bad data points identified by visual inspection in delayed-mode are recorded with TEMP_ADJUSTED_QC = ‘4’ and TEMP_QC = '4'. Please note that whenever <PARAM>_ADJUSTED_QC = ‘4’, <PARAM>_ADJUSTED = FillValue and <PARAM>_ADJUSTED_ERROR = FillValue.

TEMP_ADJUSTED, TEMP_ADJUSTED_ERROR, and TEMP_ADJUSTED_QC should be filled even when the data are good and no adjustment is needed. In these cases, TEMP_ADJUSTED_ERROR can be the manufacturer’s quoted accuracy at deployment, which is 0.002°C.

Please use the SCIENTIFIC CALIBRATION section in the netCDF files to record details of the delayed-mode assessment.

3.5. Delayed-mode procedures for salinity

3.5.1. Conductivity cell thermal inertia correction

The conductivity cell thermal inertia error is a dynamic error that is due to the CTD sampling through a temperature gradient. Salinity reported immediately after a float has crossed a strong temperature gradient can be in error, as the thermal inertia in the flow duct can alter the temperature of water entering the conductivity cell, thus inducing a conductivity error. This salinity error can exceed 0.01 (PSS-78) for strong thermal gradients, and can result in unstable fresh spikes at the base of the mixed layer. This salinity error can be corrected if the ascent rate of the float is known. A correction algorithm for Sea-Bird CTDs, based on *Johnson et al. (2007)*, is available at http://github.com/ArgoDMQC/celltm. Please refer questions to Dr. Greg Johnson at NOAA/PMEL (email: Gregory.C.Johnson@noaa.gov).

Delayed-mode operators should attempt to estimate the ascent rate of the float if possible (see Appendix 6.8), and correct for the conductivity cell thermal inertia error.

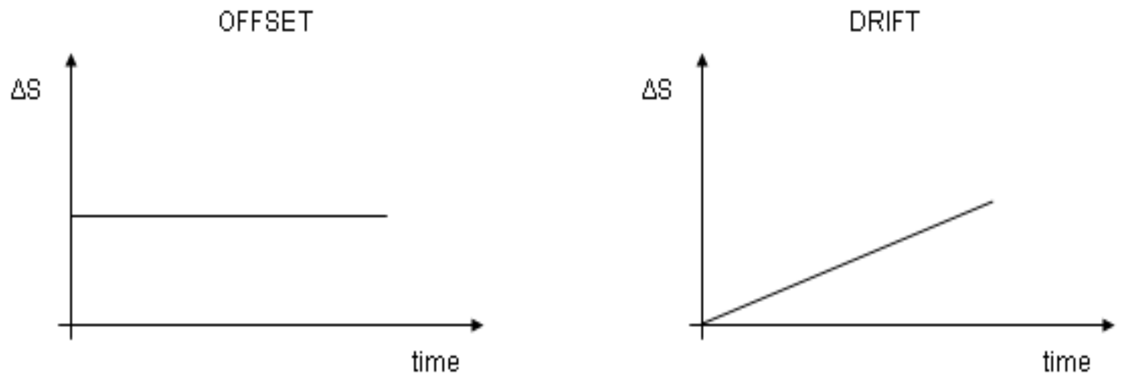
3.5.2. Despiking the salinity profile

Each salinity profile should be inspected visually for pointwise errors, such as spikes and other random irregularities. These pointwise errors are bad data points that are not adjustable in delayed-mode, and should be flagged with PSAL_ADJUSTED_QC = '4' and PSAL_QC = '4'. Please note that whenever <PARAM>_ADJUSTED_QC = ‘4’, <PARAM>_ADJUSTED = FillValue and <PARAM>_ADJUSTED_ERROR = FillValue.

3.5.3. Evaluating salinity sensor drift and offset

The free-moving nature of profiling floats means that most float salinity measurements are without accompanying in-situ “ground truth” data for absolute calibration (such as those afforded by shipboard CTD measurements). Therefore, Argo delayed-mode procedures for checking salinity sensor calibration drift and offset rely on reference datasets and linear regression analysis. However, since the ocean has inherent spatial and temporal variabilities, these adjustments are subject to uncertainties. Users should include the supplied error estimates in their usage of Argo delayed-mode salinity data.

Sensor calibration drift and offset can be identified in the trend of ΔS over time, where ΔS is the difference in salinity between float-measured values and objective estimates. If $\Delta S = a + bt$, where t is time, then a is the offset and b is the drift rate. Note that these drifts and offsets can be sensor-related, or they can be real ocean events that appear like sensor drifts and offsets. Evaluation by ocean salinity experts is needed to distinguish between sensor errors and real ocean events.



Four methods are available for comparing float salinity against objective estimates in order to check for sensor calibration drift and offset, and for computing an adjustment and related uncertainties:

1. *Wong, Johnson, Owens (2003)* estimates background salinity on a set of fixed standard isotherms, then calculates drifts and offsets by time-varying weighted least squares fit between vertically interpolated float salinity and estimated background salinity.
2. *Boehme and Send (2005)* takes into account planetary vorticity in its estimates of background salinity and chooses a set of desirable isotherms for calculations. This method suits float data from regions where water mass distribution is affected by topographic barriers.
3. *Owens and Wong (2009)* improves the objective mapping scheme of WJO based on the method suggested by BS and performs an optimal linear piecewise continuous fit in potential conductivity space.
4. *Cabanes et al (2016)* modifies the covariance function used in the objective mapping and improves the calculation of the error on the fit.

The combined Matlab code from *Owens and Wong (2009)* and *Cabanes et al (2016)* (OWC) can be found in http://github.com/ArgoDMQC/matlab_owc. Instructions on how to use OWC are described in http://github.com/ArgoDMQC/matlab_owc/doc/README.doc.

These methods require an adequate reference database and an appropriate choice of spatial and temporal decorrelation scales, as well as input of good/adjusted float pressure, temperature, position, and date of sampling. Therefore, delayed-mode operators should first check the reference database for adequacy and determine a set of appropriate spatial and temporal scales before using these methods. Delayed-mode operators should also ensure that other float measurements (PRES, TEMP, LATITUDE, LONGITUDE, JULD) are accurate or adjusted before they input them into these tools for estimating reference salinity. In particular, the input salinity profile should be free of spikes and other random irregularities that can skew the least squares fit in these methods.

The outputs of these methods are a set of float salinity time series on isotherms and a set of estimated reference salinity. An estimate of ΔS is also computed by least squares fit. Delayed-mode operators should determine if the differences are due to sensor problems or real ocean features such as eddies, fronts, water mass boundaries, and natural ocean changes. Care should be taken to not confuse real ocean events with sensor drift and offset. This is done by inspecting as long a float time series as possible, and by evaluating other independent information.

Some of the diagnostic tools are:

- Inspecting the trend of ΔS over time. Trends that reverse directions, or oscillate, are difficult to explain in terms of systematic sensor malfunction. These are often caused by the float sampling oceanographic features (e.g. eddies, fronts, etc.) that are not adequately described in the reference database.
- Visually checking the float migration path with reference to oceanographic features such as eddies and fronts that can introduce complications to the objective maps.
- Inspecting contour plots of float salinity anomaly time series versus pressure. Systematic sensor malfunction should show up as salinity anomalies over several water masses.
- Using other independent oceanographic atlases to anticipate water mass changes along the float migration path that can be misinterpreted as sensor drift.
- Cross-checking with nearby good Argo floats.
- Here is a list of Argo DMQC regional experts you can contact: http://www.argodatamgt.org/DMQC/Mentors-for-Argo-CTD

3.5.4. When to adjust float salinity for sensor drift and offset

If the delayed-mode operator determines that salinity sensor drift and/or offset has occurred, then ΔS should be used to adjust float salinity, starting from when ΔS = 0 (or from cycle 0 for offsets), so as to model the continous nature of sensor drift. In general, the CTD_for_DMQC reference data can resolve sensor drift above background variability to 2000 dbar when the magnitude of sensor drift eventually exceeds 0.01 at some time in the float's life. For these general cases, if the sensor drift does not exceed the 0.01 uncertainty throughout the float's life, then it is recommended to not adjust float salinity. Please note that there are special cases where this uncertainty can be reduced; for example: if there are nearby high-quality contemporaneous CTD casts that can be used to check float salinity, or if the float is in an area of the world's oceans with very little T-S variability. For these special cases with reduced uncertainty, the decision to apply salinity adjustment can be refined and the delayed-mode operator may consider adjusting for salinity sensor drift of magnitude < 0.01.

More examples on how to treat salinity sensor drift in delayed-mode can be found in Section 3.6. Additional resources can be found in *"DMQC Cookbook for core Argo parameters"* (http://doi.org/10.13155/78994).

3.5.5. Splitting the float time series and length of calibration window

This step has been taken into account by the Akaike Information Criterion in *Owens and Wong* (2009). The material is retained here for the purpose of historical record-keeping.

If a float exhibits changing behavior during its lifetime, the time series of ΔS should be split into separate segments according to the different behaviors, so that one behavior does not contaminate another during the least squares fit process (e.g. the slowly-fouling segment does not contaminate the stable segment).

The following is a step-by-step guide on how to deal with float salinity time series with changing behaviors.

1). Identify different regimes in the time series. These can be:

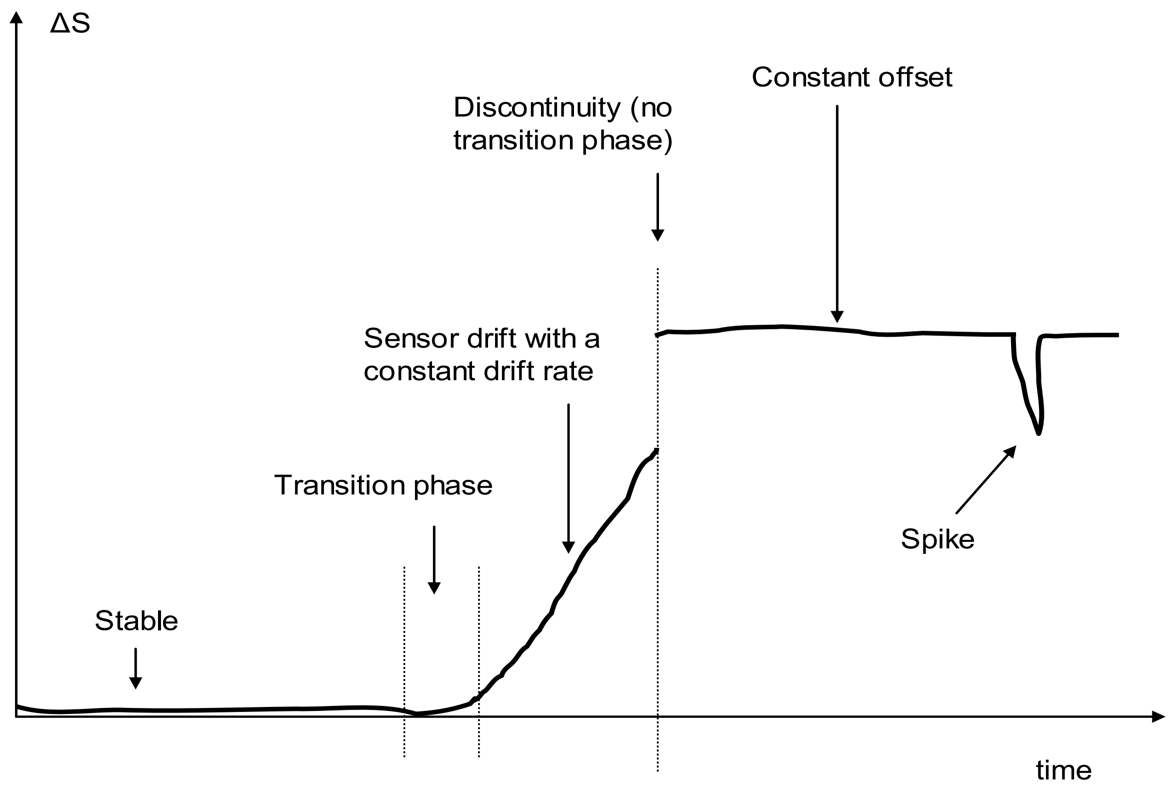
• Stable measurements (no sensor drift), including constant offsets.

• Sensor drift with a constant drift rate.

• Spikes (random error).

• Transition phase where drift rate changes rapidly, e.g. ‘elbow region’ between stable measurements and constant drift, or initial biocide wash-off.

2). Split the time series into discrete segments according to these different regimes or when there are too many missing cycles. Here is an example:



3). Choose length of sliding calibration window for each segment. These can be:

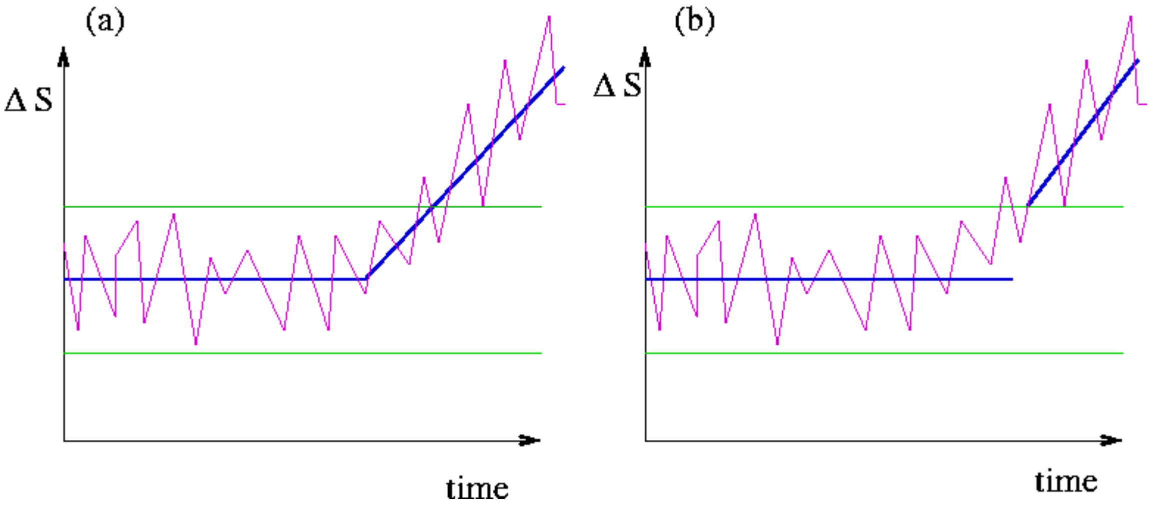
• Long window (+/− 6 months or greater) for the stable regime, or highly variable regimes where a long window is required to average over oceanographic variability to detect slow sensor drift, or period of constant drift rate.

• Short window (can be as short as +/− 10 days) for the transition phase.

• Zero length window for spikes. That is, adjust single profile.

4). Calculate proposed adjustment for each segment. The assembled proposed adjustments for the entire float series should be continuous and piecewise-linear within error bars, except where the delayed-mode operator believes there is a genuine discontinuity. This is to ensure that no artificial jump is introduced where the separate segments join.

In the following example, the float experiences salinity sensor drift after a stable period. The time series has been split for calibration. However, the time series has no discontinuity, so the final assembled adjustment should be continuous. Adjustment continuity is achieved by using model (a), not (b).



In general, the delayed-mode operator should aim to use as long a calibration window as possible, because a long calibration window (where the least squares fit is calculated over many cycles) will average over oceanographic noise and thus give a stable calibration. Hence splitting the float series into short segments is to be avoided (short segments mean short calibration windows, hence unstable calibrations).

3.5.6. Assigning adjusted salinity, error estimates, and QC flags

After evaluating all available information, the delayed-mode operator then assigns adjusted salinity, error estimates, and QC flags. In Argo data files, these are found respectively in the variables PSAL_ADJUSTED, PSAL_ADJUSTED_ERROR, PSAL_ADJUSTED_QC and PSAL_QC. The following is a set of guidelines for assigning their values.

<u>For float salinity that are considered good and need no adjustment in delayed-mode</u>

These are measurements that are unaffected by any instrument errors, such as sensor drift or calibration offset. For these good salinity data:

PSAL_ADJUSTED = PSAL (raw value);
PSAL_ADJUSTED_QC = ‘1’ or ‘2’;
PSAL_ADJUSTED_ERROR = maximum [adjustment uncertainty, 0.01].

Note that a subset of these good salinity data is selected for inclusion in an Argo-based reference database. In cases where a float is considered to need no salinity adjustment in delayed-mode, but that the data may not be suitable for use as reference because uncertainty is high, PSAL_ADJUSTED_ERROR should increase to > 0.015. Please refer to Section 6.6 for the list of selection criteria used in compiling the Argo reference database.

For float salinity that are considered bad and unadjustable in delayed-mode

For example, large jumps in salinity from one cycle to another, salinity offset that has significant verical variation, or extreme behavior where the relative vertical T-S shape does not match good data. These measurements are bad and unadjustable:

PSAL_ADJUSTED = FillValue;
PSAL_ADJUSTED_QC = ‘4’; PSAL_QC = '4';
PSAL_ADJUSTED_ERROR = FillValue.

For float salinity affected by instrument errors that are considered adjustable in delayed-mode

These are measurements that are affected by instrument errors whose effects can be rectified by delayed-mode adjustments. These include minor calibration offset and minor sensor drift that do not have significant vertical variation, where the data still retain a relative vertical T-S shape that is close to good data. When an adjustment has been applied:

PSAL_ADJUSTED = original value + adjustment recommended by regression analysis, or adjustment provided by PI.

PSAL_ADJUSTED_QC = ‘1’ or ‘2’.

For PSAL_ADJUSTED_ERROR, if a measurement has been adjusted for more than one instrument error, operators should propagate the uncertainties from all the adjustments.

PSAL_ADJUSTED_ERROR = maximum [$(\sum \text{adjustment\_error}^2)^{1/2}$, 0.01], where “adjustment_error” is the uncertainty from each adjustment. These can be uncertainty from sensor drift adjustment, uncertainty from conductivity cell thermal mass adjustment, etc.

Large magnitude delayed-mode salinity adjustment: $|\Delta S| > 0.05$ - ***proceed with caution***

The methods in Section 3.5.3 correct salinity sensor drift by applying a pressure-independent offset adjustment to the vertical profiles. These methods work well when the sensor drift has no significant vertical variation. In general, based on past experience with the SBE41Cp CTD, when the magnitude of sensor drift exceeds 0.05 psu, the salinity data will have some vertical errors, or become erratic. Therefore, in general, 0.05 should be considered the upper limit of good salinity adjustment in delayed-mode. Delayed-mode operators should examine float data with these upper limits of sensor drift carefully to determine the suitability of adjustment. If the float data are still coherent and the large magnitude sensor drift has not caused significant vertical errors, then the delayed-mode operator may consider salinity adjustments larger than 0.05.

If the magnitude of salinity sensor drift exceeds 0.05 psu, the delayed-mode operator should consider the following two options.

(1). Salinity data are considered unadjustable in delayed-mode, because a pressure-independent offset adjustment is not sufficient.

PSAL_ADJUSTED = FillValue;
PSAL_ADJUSTED_QC = ‘4’; PSAL_QC = '4';
PSAL_ADJUSTED_ERROR = FillValue.

PSAL should be greylisted with flag '4' and START_DATE that corresponds to the beginning of the sensor problem.

(2). Salinity data are considered adjustable with a pressure-independent offset adjustment, but confidence in the adjustment is low.

PSAL_ADJUSTED = original value + adjustment > 0.05;
PSAL_ADJUSTED_QC = ‘2’; PSAL_QC = '3';
PSAL_ADJUSTED_ERROR = uncertainty provided by PI, preferably > 0.015.

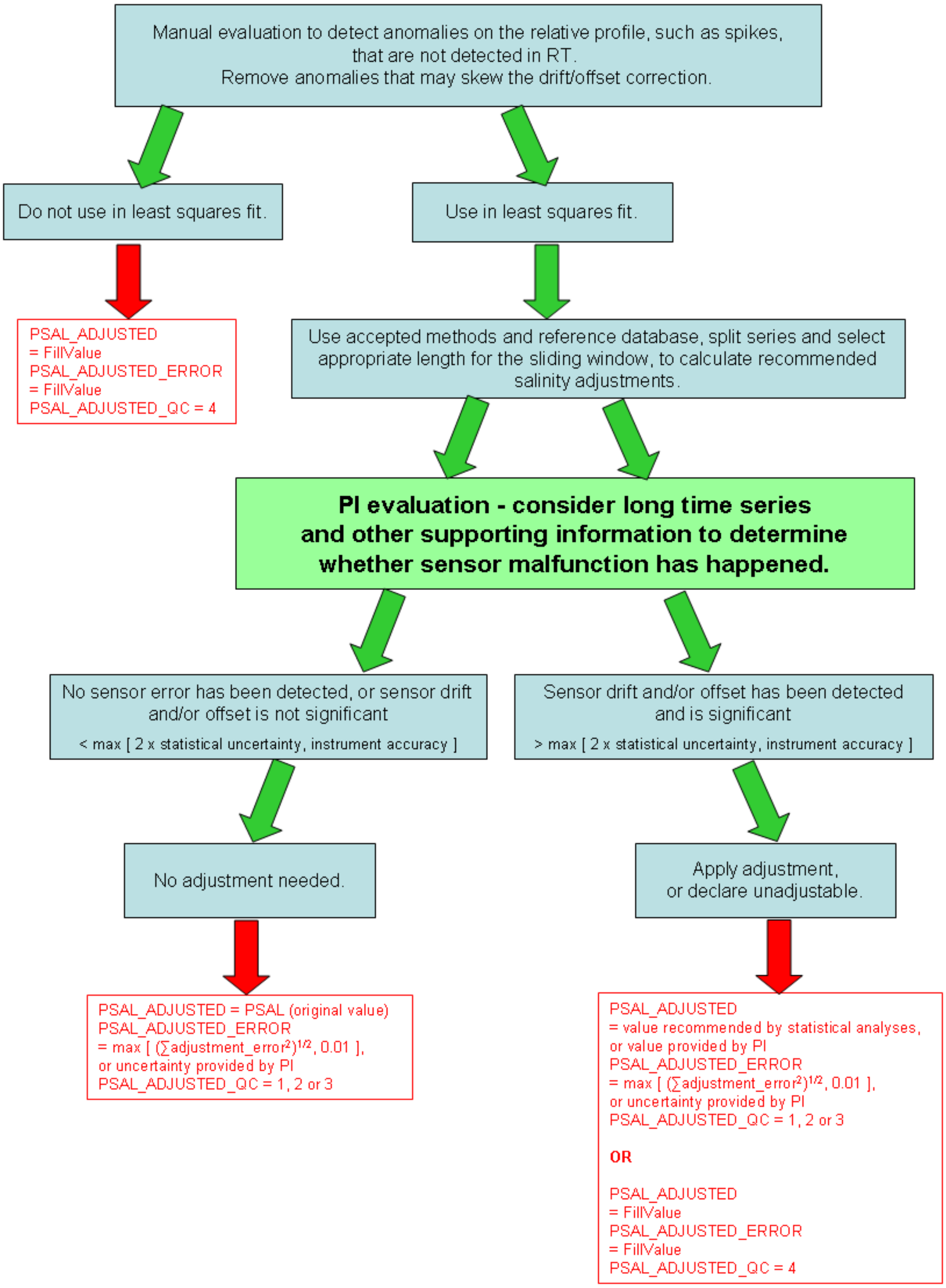
PSAL should be greylisted with flag '3' and START_DATE that corresponds to the beginning of the sensor problem.

<u>The following are some cases where PSAL_ADJUSTED_QC = ‘2’ should be assigned</u>

- Adjustment is based on unsatisfactory reference database.
- Adjustment is based on a short calibration window (because of sensor behaviour transition, or end of sensor life) and therefore may not be stable.
- Evaluation is based on insufficient information.
- Sensor is unstable (e.g. magnitude of salinity adjustment is greater than 0.05, or sensor has undergone too many sensor behavior changes) and therefore data are inherently of mediocre quality.
- When a float exhibits problem with its pressure measurements, e.g. APEX TNPD.
- As soon as the delayed-mode operator decides to assign PSAL_ADJUSTED_QC = '2', he/she should consider greylisting PSAL with flag '3'.

3.5.7. Summary flowchart of salinity delayed-mode procedures

Argo salinity sensor drift & offset QC procedures



3.5.8. Timeframe of availability for delayed-mode salinity data

The methods used in the Argo delayed-mode process for checking sensor drift and offset in salinity require the accumulation of a time series for reliable evaluation of the sensor trend. Timeframe for availability of delayed-mode salinity data is therefore dependent on the sensor trend. Some floats need a longer time series than others for stable calibration and evaluation. Thus delayed-mode salinity data for the most recent profile may not be available until sufficient subsequent profiles have been accumulated. The *default* length of time series for evaluating sensor drift is 12 months (6 months before and 6 months after the profile). This means that, in general, the timeframe of availability of drift-adjusted delayed-mode salinity data is 6+ months after a profile is sampled. However, in instances of known sensor stability or known adjustment stability, the delayed-mode operator can decide how recent a profile to send out.

Users should also be aware that changes may be made to delayed-mode files at any time by DACs and delayed-mode operators. For example, delayed-mode files may be revised when new CTD or float data become available after the original delayed-mode assessment and adjustment. The date of latest adjustment of a parameter can be found in SCIENTIFIC_CALIB_DATE.

Anytime an Argo file is updated for any reason, the DATE_UPDATE variable will reflect the date of the update. The "profile index file" on the GDACs contains the DATE_UPDATE information (along with other information) for every file on the GDACs and can be used to monitor updates. The profile index file is maintained in the top-level GDAC directory and is named "ar_index_global_prof.txt"; index files also exist for the meta-data and trajectory files.

3.5.9. References

Böhme, L. and U. Send, 2005: Objective analyses of hydrographic data for referencing profiling float salinities in highly variable environments. *Deep-Sea Research II*, **52/3-4**, 651-664.

Cabanes, C., V. Thierry, and C. Lagadec, 2016: Improvement of bias detection in Argo float conductivity sensors and its application in the North Atlantic. *Deep Sea Research Part I: Oceanographic Research Papers*, **114,** 128-136.

Johnson, G.C., J.M. Toole, and N.G. Larson, 2007: Sensor corrections for Sea-Bird SBE-41CP and SBE-41 CTDs. *Journal of Atmospheric and Oceanic Technology*, **24**, 1117-1130.

Owens, W.B. and A.P.S. Wong, 2009: An improved calibration method for the drift of the conductivity sensor on autonomous CTD profiling floats by θ-S climatology. *Deep-Sea Research*, Part I: Oceanographic Research Papers, **56(3)**, 450-457.

Wong, A.P.S., G.C. Johnson, and W.B. Owens, 2003: Delayed-mode calibration of autonomous CTD profiling float salinity data by θ-S climatology. *Journal of Atmospheric and Oceanic Technology*, **20**, 308-318.

3.6. Best practices and examples on how to treat salinity sensor drift in delayed-mode

This section provides some examples on how to treat float salinity data that are affected by sensor drift in order to produce optimal adjustment in delayed-mode. More regional examples can be found in *"DMQC Cookbook for core Argo parameters"* (http://doi.org/10.13155/78994).

Sensor drift is the temporal drift of measurements caused by physical changes in the sensor that progress over time. In addition to sensor drift, measurements can also be affected by random irregularities, such as spikes, that are caused by trapped particles or contamination, but these will usually clear with time. Please refer to Section 3.5 for the complete set of Argo delayed-mode procedures for salinity.

3.6.1. Do not confuse natural variability with sensor drift

Salinity sensor drift in profiling float data is diagnosed by comparing time series of float data to nearby reference data. The comparisons are best done on isotherms that have the least salinity variability. These are usually deep isotherms where water masses have relatively uniform salinity ranges that are stable over time. Ideally (even though this is not always possible), sensor drift should be diagnosed by examining multiple water masses, or multiple isotherms, so that natural variability in one water mass is not misinterpreted as sensor drift. Natural variability can include floats moving to another area with different water masses, or floats crossing oceanic fronts and eddies, or temporal variability in the reference data, etc.

Tips: In OWC, you can constrain the temperature range and/or the pressure range in the least squares fit by using the variables "use_pres_gt", "use_pres_lt", "use_theta_gt", "use_theta_lt", in the calseries_ mat file.

3.6.2. Adjustable or unadjustable

(a). Use profile-to-profile stability to determine the health of the conductivity cell

After natural variability has been excluded and sensor drift detected, the delayed-mode operator should determine when the conductivity cell has deteriorated to a point when it is no longer producing stable salinity data. Profile-to-profile stability is an indication of sensor health and can be determined by examining the time series of float data on isotherms. Imagine an experiment where a sensor samples a constant body of water - if the body of water does not change, a healthy sensor should produce consistent measurements over time. Poor profile-to-profile stability, or temporal instability, such as large jumps in measured values between profiles, or erratic sensor drift trends, are usually signs that the conductivity cell is damaged - these data should be considered bad and unadjustable in delayed-mode (Figure 1).

On the other hand, a conductivity cell can be affected by mild sensor drift and still produces data with good profile-to-profile stability - these data should be considered as adjustable in delayed-mode (Figure 2).

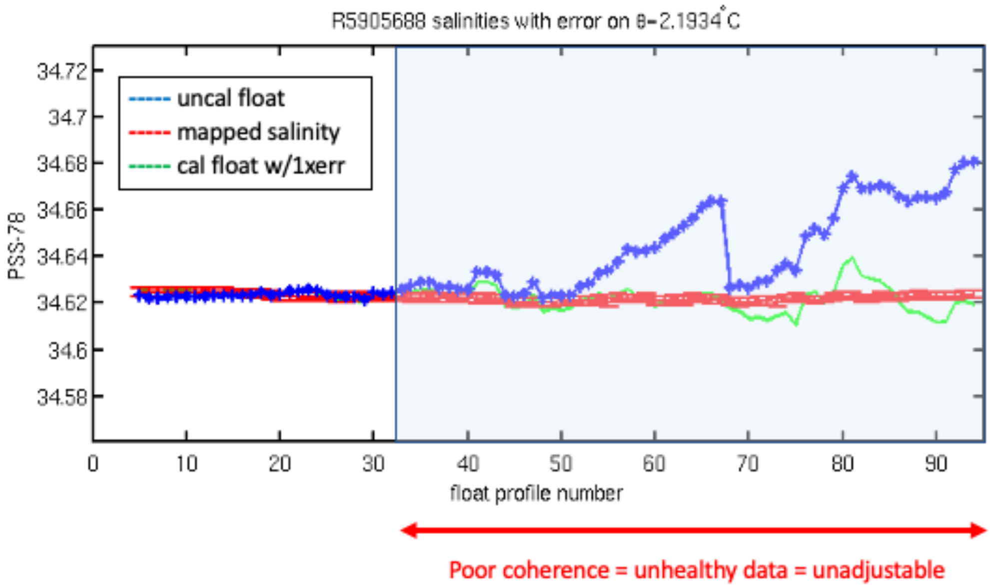


Figure 1. WMO ID 5905688, DAC = AOML/SIO. Starting from cycle 32, float salinity data drifted salty in an unstable manner, with many reversals. These signs indicate that the conductivity cell has failed. Salinity from cycle 32 onwards should be considered as bad and unadjustable in delayed-mode.

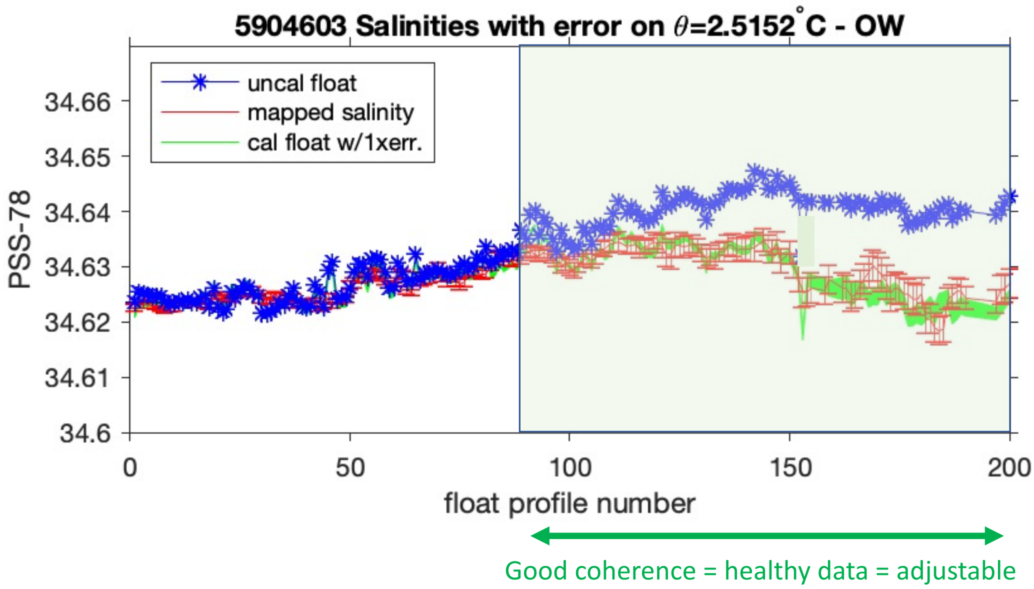


Figure 2. WMO ID 5904603, DAC = AOML/UW. Starting from around cycle 80, float salinity data drifted salty in a consistent manner. This indicates that the conductivity cell is still relatively stable, and so these salinity data should be considered as adjustable in delayed-mode.

(b). Vertical variations

Some sensor drifts have significant vertical variations in addition to the offset (Figure 3). Currently there is not a method that can consistently model the vertical distribution of salinity errors in sensor drifts that have a vertical component. Hence this type of sensor drifts is currently unadjustable in delayed-mode.

The OWC method aims to correct the offset component, and therefore can be used to adjust salinity data affected by sensor drifts whose vertical variations are negligible.

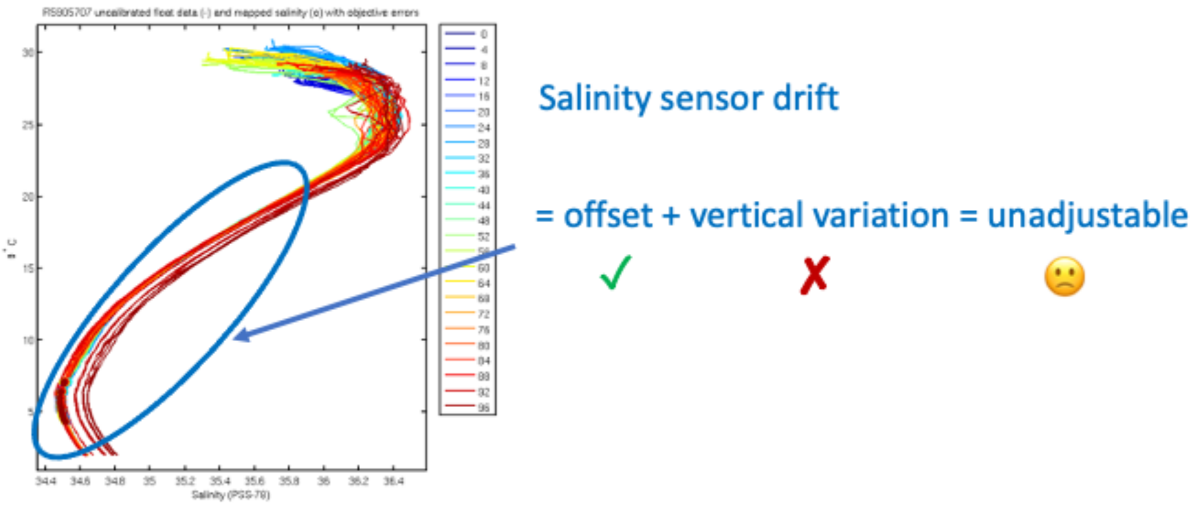


Figure 3. WMO ID 5905707, DAC = AOML/SIO. Starting from cycle 90, float salinity drifted salty, with the sensor drift consisting of an offset plus significant vertical variation. OWC can correct the offset component, but not the vertical variation. Hence salinity data from cycle 90 onwards are currently unadjustable in delayed-mode, due to the absence of method that can correct the vertical variation.

Note that in Deep-Argo salinity data, the vertical skewness due to the compressibility effect can be corrected by applying the CPcor adjustment. This step should be done prior to OWC adjustment, so that the vertical skewness does not affect the least squares fit in OWC. Please refer to Section 3.10 for the delayed-mode procedure for Deep-Argo salinity data.

(c). Upper limit of salinity adjustment

The recommendation to consider 0.05 as the upper limit of salinity adjustment (see Section 3.5.6) is arrived at through analyzing many float data in delayed-mode. It is found that, in general, when the magnitude of salinity offset exceeds 0.05, the data will show poor profile-to-profile coherence, or vertical errors, or both, and hence become unadjustable with our current method. However, the delayed-mode operator should still examine float data in these upper limits carefully, in case there are occasions where salinity offset adjustment > 0.05 can be applied.

3.6.3. Adjustable salinity data - evaluating the model fit of dS versus time

After determining that sensor drift is adjustable by an offset correction, the next step is to estimate the sensor drift trend by finding a good model for dS versus time (t). Here, dS is the difference in salinity between the float measured value and the objectively mapped value from reference data. As described in 3.6.1, dS is evaluated by using multiple isotherms.

It is important to evaluate the model fit, because adjusting the float data with a wrong model will create anomalous data. In general, there are 3 scenarios of data fitting (Figure 4) to consider: (a) under fit, (b) over fit, and (c) optimal fit.

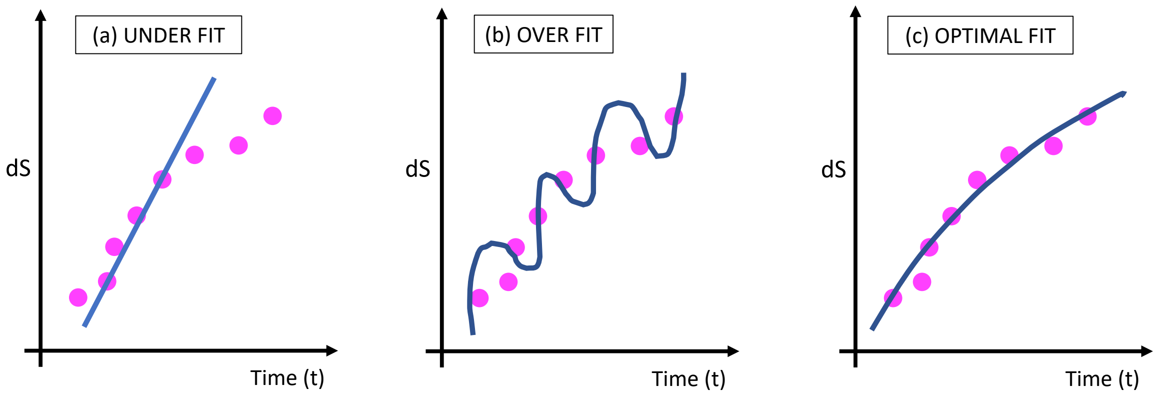


Figure 4. Three scenarios of data fitting: (a) under fit, (b) over fit, and (c) optimal fit.

(a) Under fit
A model that underfits the data means the model is too simplistic, which makes it inadequate to represent the dominant sensor drift trend. This will produce delayed-mode adjustments that have a high bias for some profiles (too salty or too fresh).

(b) Over fit
A model that overfits the data means the model is capturing noise in the data, which obscures the dominant sensor drift trend. This will produce delayed-mode adjustments that have a high variance and will result in anomalous data.

(c) Optimal fit
This is the desirable model that has a good bias-variance balance. This will produce sensible delayed-mode adjustments that agree well with reference data.

There is not a formal validation step in the delayed-mode adjustment for salinity. But a good practice is to compare the adjusted data with the reference data to see if an optimal fit has been achieved.

Tips: OWC models the sensor drift with a series of linear segments (piecewise linear). To achieve an optimal fit, you can use two variables in the calseries_ mat file: "breaks" to specify your own break points, or "max_breaks" to specify the number of break points (Figure 5).

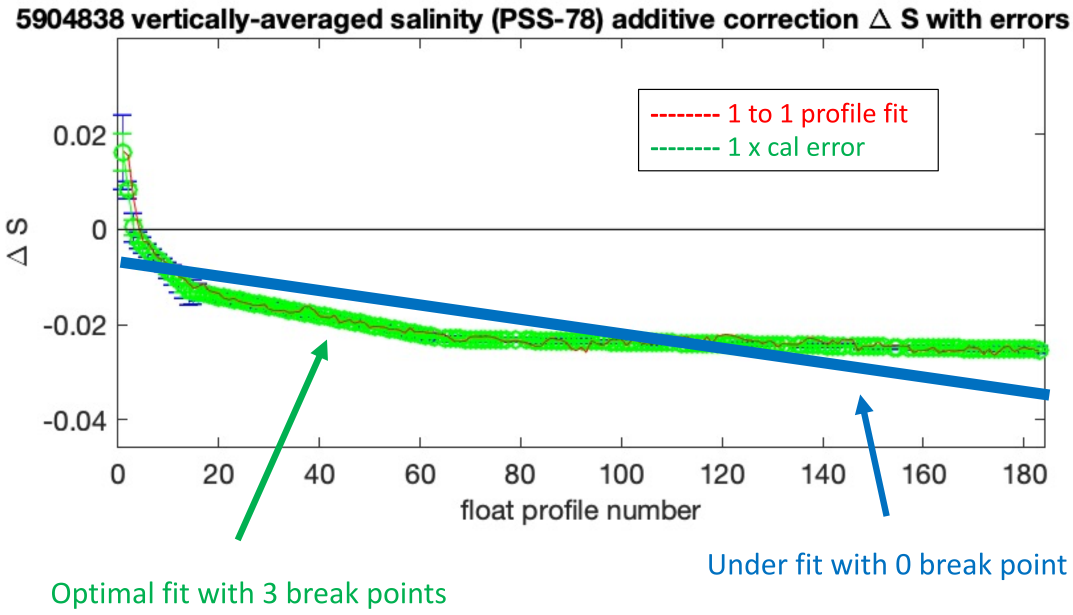


Figure 5. WMO ID 5904838, DAC = AOML/UW. Salinity from this float was affected by TBTO contamination in the first three profiles. Then salty drift began in a coherent manner starting from profile 4. With 0 break point (blue line), the model underfits the data, which will lead to anomalously fresh adjusted data in the beginning of the time series, and anomalously salty adjusted data at the end of the time series. A piecewise linear fit with 3 break points (green line) gives an optimal fit.

3.6.4. Examples of unadjustable salinity data - Abrupt Salty Drift (ASD)

Historically, about 10% of Sea-Bird CTDs used in Argo floats are expected to develop salinity sensor drift over time. However, starting around 2015, a larger than expected percentage of Sea-Bird CTDs (SBE41Cp with serial numbers > 6000, and all SBE61) developed salty drifts earlier than expected, and progressed to a failed/uncorrectable state earlier than expected. This has resulted in an increase in salinity data loss in Argo. To monitor these premature SBE CTD failures since 2015, a google spreadsheet has been set up that aims to record floats affected by salty drifts early in life (before cycle 80). Delayed-mode operators are urged to fill in this google spreadsheet at least once a year:

https://docs.google.com/spreadsheets/d/1TA7SAnTiUvCK7AyGtSTUq3gu9QFbVdONj9M9zAq8CJU/edit#gid=1096144849

In particular, many of these salty sensor drifts show a peculiar set of characteristics:

a) the sensor drift rate accelerates rapidly and abruptly;
b) the sensor drift trend is erratic, often with jumps between consecutive profiles, and can reverse in direction;
c) the onset of (a) and/or (b) can sometimes be seen as an inflection point or a jump in the float salinity time series.

These cases are referred to generally as "Abrupt Salty Drift" (ASD). These characteristics suggest failure of the conductivity cell to produce healthy salinity measurements. Hence float salinity data from the ASD phase should be considered as **bad and unadjustable** in delayed-mode (Figures 6, 7). Note that some afflicted conductivity cells can still produce healthy, adjustable salinity measurements prior to the onset of ASD (Figures 8, 9).

Some notes on ASD:

- In some cases, a handful of profiles prior to the onset of ASD can be anomalously fresh.
- In some cases, towards the advanced stage of ASD, the sensor drift has a vertical variation in addition to an offset, with larger salinity errors at higher pressures.
- In oceanic regions with high variability (Figure 10), or where reference data are sparse, it can be difficult to pinpoint the onset of ASD. Delayed-mode operators are reminded to be careful in distinguishing natural variability and erratic sensor behavior.
- In some extreme cases, ASD can manifest as a moderate salty drift initially, and then proceed to abruptly jump to grossly abnormal fresh measurements (Figure 11). These "fresh jumps" are an indication of the end state of ASD and should be marked as bad and unadjustable.

ASD was believed to be caused by failure of the encapsulant material (delamination of the urethane) in the SBE CTD. A manufacturing change at Sea-Bird was installed in 2018 in order to help decrease the occurrence of ASD in SBE CTDs. The CTDs after the 2018 manufacturing change have serial numbers > 11252 for the SBE41Cp, and > 5724 for the SBE61. Delayed-mode operators are urged to monitor the data quality from SBE CTDs in these serial number bands.

When ASD is confirmed, it is recommended that the information be recorded in the ANOMALY variable of the float metadata file with the char string “ASD:AbruptSaltyDrift”.

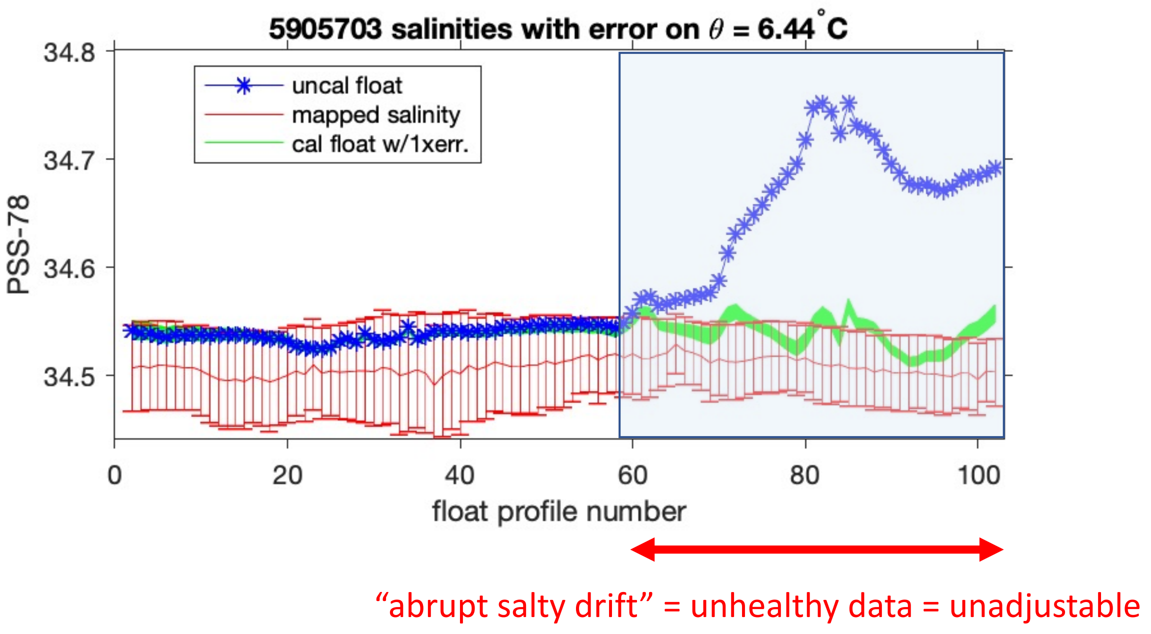


Figure 6. WMO ID 5905703, DAC = AOML/SIO. An example of "abrupt salty drift". Starting from cycle 59, float salinity data drifted salty abruptly and rapidly. There were large jumps between profiles and many reversals. Salinity data from cycle 59 onwards should be considered as bad and unadjustable.

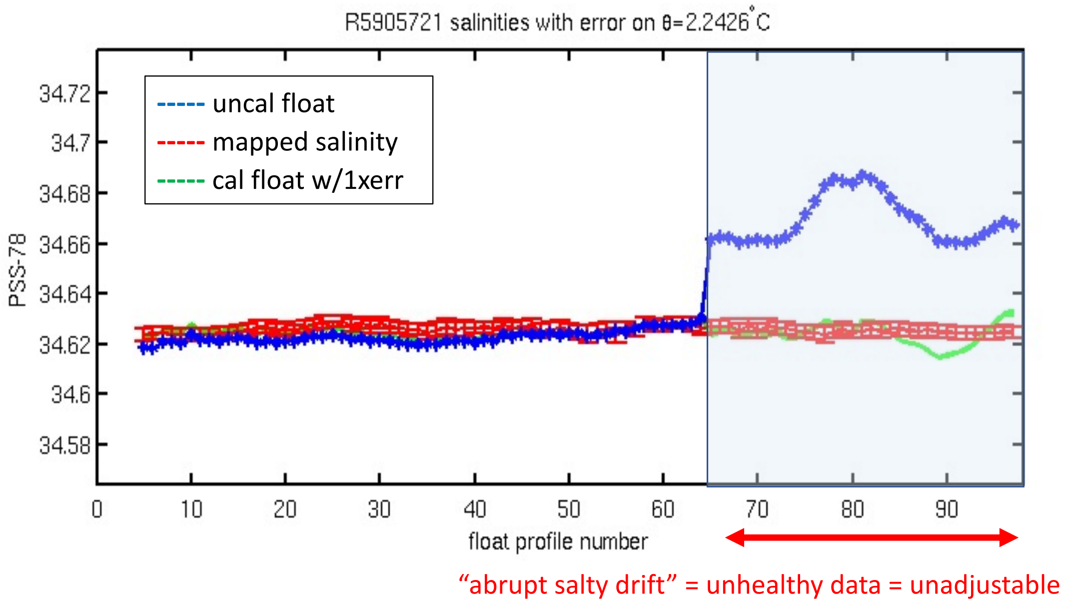
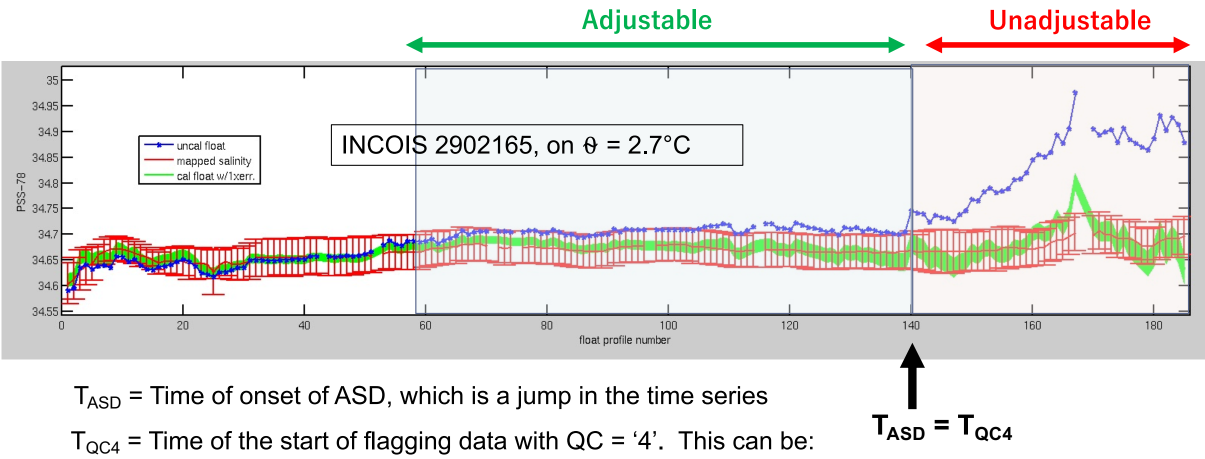


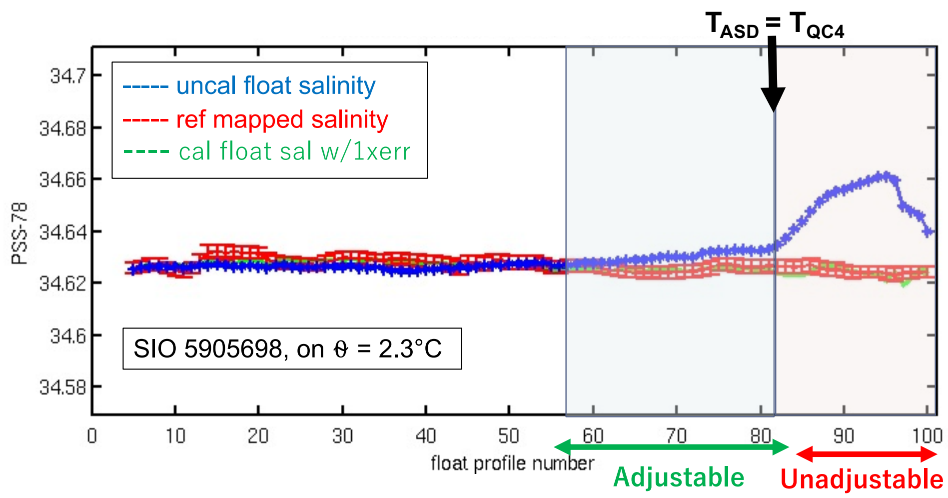
Figure 7. WMO ID 5905721, DAC = AOML/SIO. Another example of "abrupt salty drift". After cycle 65, salinity jumped salty by > 0.035, with no consistent sensor trend after that. Therefore salinity data from cycle 65 onward should be considered as bad and unadjustable.



T_{ASD} = Time of onset of ASD, which is a jump in the time series

T_{QC4} = Time of the start of flagging data with QC = '4'. This can be:
- when dS > 0.05, or
- when dS between consecutive profiles > 0.01, or
- **time as decided by the delayed-mode operator**.

Figure 8. WMO ID 2902165, DAC = INCOIS. This is an example of a CTD that produced adjustable salinity measurements before the onset of ASD. Starting from cycle 58, float salinity data drifted salty in a consistent manner until cycle 140. Profile-to-profile stability is good, and the data are adjustable in delayed-mode. After cycle 140, the salty drift jumps, accelerates rapidly, and then reverses direction. Therefore salinity data after cycle 140 are bad and unadjustable in delayed-mode.



T_{ASD} = Time of onset of ASD, which is the inflection point in the time series

T_{QC4} = Time of the start of flagging data with QC = '4'. This can be:
- when dS > 0.05, or
- when dS between consecutive profiles > 0.01, or
- **time as decided by the delayed-mode operator**.

Figure 9. WMO ID 5905698, DAC = AOML/SIO. This is another example of a CTD that produced adjustable salinity measurements before the onset of ASD. Starting from cycle 52, salinity drifted salty in a stable, consistent manner. Then, after cycle 82, ASD began, with accelerated drift rate and reversal of direction. Therefore salinity data from cycles 52 to 82 are adjustable, but are bad and unadjustable from cycle 82 onward.

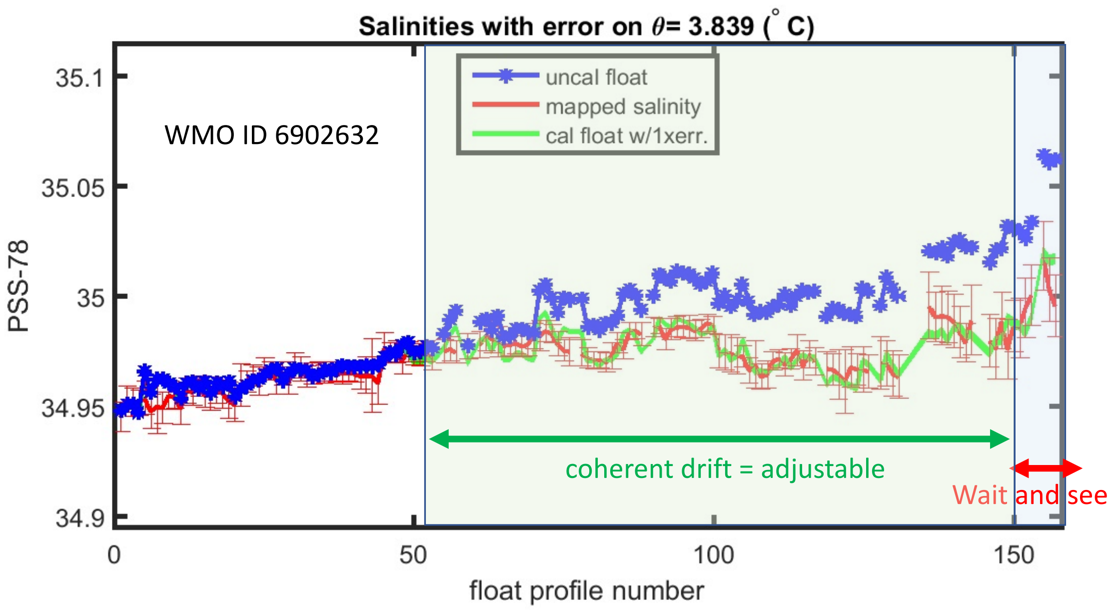


Figure 10. WMO ID 6902632, DAC = Coriolis. This is an example of a float sampling in the Gulf Stream in the North Atlantic Ocean. The high variability in the Gulf Stream gives rise to relatively noisy data. Adjustable salty drift starts from around cycle 51. Then, after cycle 150, there are some anomalous features that may be precursor to possible severe conductivity cell corruption, or may be noisy data from an eddy. Best practice would be to not issue D-files beyond cycle 150 until more data are accumulated, or flag PSAL_ADJUSTED_QC = '2'.

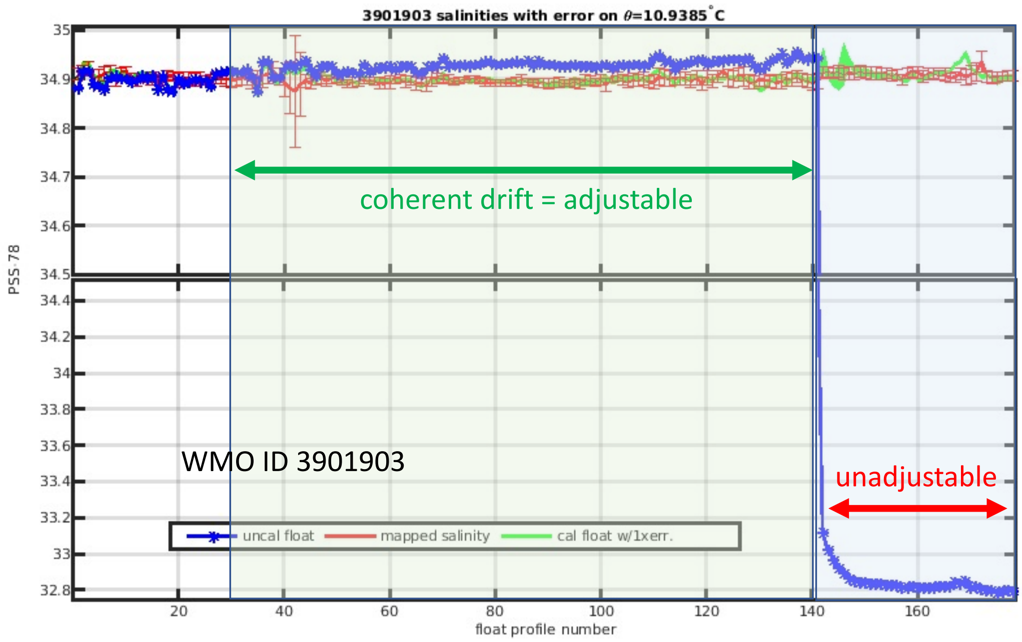
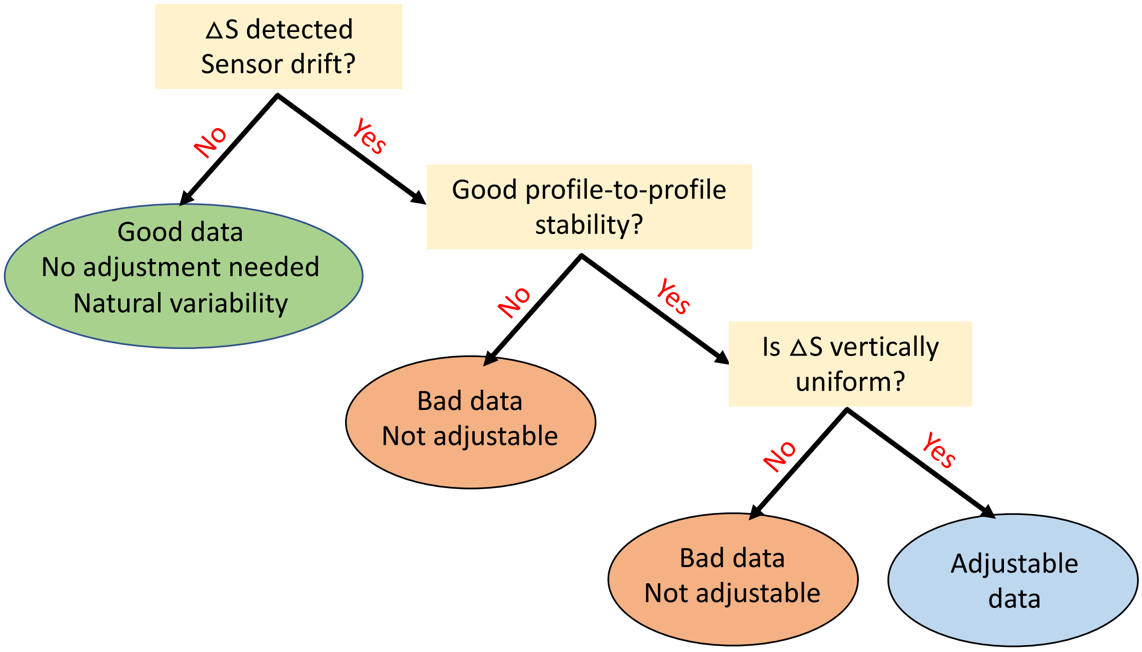


Figure 11. WMO ID 3901903, DAC = Coriolis. This is an example of a float with an adjustable salty drift, followed by a fresh jump. Salinity data from cycle 142 onward are bad and unadjustable in delayed-mode.

3.6.5. Summary: decision tree for how to treat salinity sensor drift



3.7. Compulsory variables to be filled in a D file

This section lists the compulsory variables that must be filled in a D file as part of the delayed-mode process.

3.7.1. Measurements for each profile

The following are compulsory measurement variables that must be filled in a D file:

- <PARAM>_ADJUSTED;
- <PARAM>_ADJUSTED_QC;
- <PARAM>_ADJUSTED_ERROR.

Here, <PARAM> denotes all the measurement parameters that are reported in the D file. <PARAM> = PRES, TEMP, PSAL are the core Argo measurement parameters. Please see Sections 3.3, 3.4, 3.5 on how to fill their _ADJUSTED_ variables.

The variable PROFILE_<PARAM>_QC should be recomputed when <PARAM>_ADJUSTED_QC becomes available.

3.7.2. Other variables in a D file

The variable DATA_MODE should record 'D'.

The variable DATA_STATE_INDICATOR should record '2C' or '2C+'.

The variable DATE_UPDATE should record the date of last update of the netcdf file, in the format YYYYMMDDHHMISS.

The name of the single-cycle profile file should be changed from R*.nc to D*.nc.

3.7.3. Scientific calibration information for each profile

Within each D file is a scientific calibration section that records details of the QC and adjustments. It is compulsory to fill the variables in the scientific calibration section.

In the scientific calibration section, every measurement parameter recorded in STATION_PARAMETER in the D file should be listed in the variable PARAMETER. For every measurement parameter listed in PARAMETER, there are four variables to record the scientific calibration details:

- SCIENTIFIC_CALIB_EQUATION;
- SCIENTIFIC_CALIB_COEFFICIENT;
- SCIENTIFIC_CALIB_COMMENT;
- SCIENTIFIC_CALIB_DATE.

Regardless of whether an adjustment has been applied or not, the date of delayed-mode qc for each measurement parameter should be recorded in SCIENTIFIC_CALIB_DATE, in the format YYYYMMDDHHMISS.

The following are some examples on how to fill the SCIENTIFIC_CALIB section.

Example 1. PARAMETER = PRES. APEX pressure adjustment has been applied.
SCIENTIFIC_CALIB_EQUATION = ‘PRES_ADJUSTED = PRES – dP’
SCIENTIFIC_CALIB_COEFFICIENT = ‘dP = –0.2 dbar’
SCIENTIFIC_CALIB_COMMENT = ‘Pressures adjusted by using pressure offset at the sea surface. The quoted error is manufacturer specified accuracy in dbar.’

Example 2. PARAMETER = TEMP. Data are good and no adjustment is needed.
SCIENTIFIC_CALIB_EQUATION = ‘none’
SCIENTIFIC_CALIB_COEFFICIENT = ‘none’
SCIENTIFIC_CALIB_COMMENT = ‘The quoted error is manufacturer specified accuracy with respect to ITS-90 at time of laboratory calibration.”

Example 3. PARAMETER = PSAL. Data are good; no adjustment has been applied.
SCIENTIFIC_CALIB_EQUATION = ‘none’
SCIENTIFIC_CALIB_COEFFICIENT = ‘none’
SCIENTIFIC_CALIB_COMMENT = ‘No significant salinity drift detected. The quoted error is max[0.01, adjustment uncertainty] in PSS-78.’

Example 4. PARAMETER = PSAL. Data show sensor drift; adjustment has been applied.
SCIENTIFIC_CALIB_EQUATION = ‘PSAL_ADJUSTED = PSAL + dS’
SCIENTIFIC_CALIB_COEFFICIENT = ‘vertically averaged dS = 0.012 +/- 0.003’
SCIENTIFIC_CALIB_COMMENT = ‘Significant salinity sensor drift detected. OW least squares fit adopted. Salinity also adjusted for effects of pressure adjustment. The quoted error is max[0.01, adjustment uncertainty] in PSS-78.’

Example 5. PARAMETER = PSAL. Data are bad and unadjustable.
SCIENTIFIC_CALIB_EQUATION = ‘none’
SCIENTIFIC_CALIB_COEFFICIENT = ‘none’
SCIENTIFIC_CALIB_COMMENT = ‘Salinity data are bad and unadjustable because of the Druck oil microleak problem.’

3.7.4. HISTORY section

A history record should be appended to the HISTORY section of a D file to indicate that the file has been through the delayed-mode process. Please refer to the Argo User’s Manual (§5 “Using the History section of the Argo netCDF Structure”) on usage of the History section. An example to record the completion of delayed-mode QC is here.

```
HISTORY_INSTITUTION(N_HISTORY, N_PROF, STRING4)        "UW  "
HISTORY_STEP(N_HISTORY, N_PROF, STRING4)               "ARSQ"
HISTORY_SOFTWARE(N_HISTORY, N_PROF, STRING4)           "UWQC" (locally defined char string)
HISTORY_SOFTWARE_RELEASE(N_HISTORY, N_PROF, STRING4) "V1.0" (locally defined char string)
HISTORY_REFERENCE(N_HISTORY, N_PROF, STRING64) "CTD & Argo      " (locally defined char string)
HISTORY_DATE(N_HISTORY, N_PROF, DATE_TIME)             "20200910155005"
HISTORY_ACTION(N_HISTORY, N_PROF, STRING4)             "IP  "
```

3.8. How to record the DMQC operator in the D and BD files

At the ADMT meeting in 2019, it was agreed to record the delayed-mode operator in the D and BD files by adding a comment line to the **//global_attributes** section of the Argo single-cycle profiles files. This comment line will contain the ORCID of the delayed-mode operator, as well as his/her name and institution.

For D and BD files where one person is in charge of the whole file, add the line:
:comment_dmqc_operator = "PRIMARY | https://orcid.org/16-digit-number | operator name, institution".

For BD files specifically where some B-PARAMs have dmqc operators who are different from that in PRIMARY and who want to be listed explicitly, add the line with the PRIMARY operator first, then add additional lines:
:comment_dmqc_operator1 = "PRIMARY | http://orcid.org/16-digit-number 1 | operator name 1, institution 1"
:comment_dmqc_operatorN = "B-PARAM | http://orcid.org/16-digit-number N | operator name N, institution N"
where N > 1. For example,
:comment_dmqc_operator1 = "PRIMARY | https://orcid.org/1111-1111-1111-1111 | Jane Smith, University A"
:comment_dmqc_operator2 = "DOXY | https://orcid.org/2222-2222-2222-2222 | Joe Doe, University B"
:comment_dmqc_operator3 = "NITRATE | https://orcid.org/3333-3333-3333-3333 | Jack Apple, University C"

Some notes:

- The first line of *comment_dmqc_operator* must list the PRIMARY operator in both the D and BD files.
- The core D files should not list multiple dmqc operators, because the delayed-mode process for all the core parameters should be overseen by one primary person.
- The bgc BD files must list a primary dmqc operator even if there are multiple dmqc operators for the various B-PARAMs.
- When a different dmqc operator has re-worked an existing D or BD file with new delayed-mode adjustments, *comment_dmqc_operator* should be re-written with a new entry. In other words, do not append additional lines with new entries.
- When an existing D or BD file receives minor updates that do not relate to delayed-mode adjustments, do not update *comment_dmqc_operator*.
- DMQC operators who need a record of their work should refer to the Argo GDAC monthly DOI snapshots.
- This is an optional step, because delayed-mode groups are not required to backfill existing D and BD files. Moreover, this is not required for real-time R and BR files.

3.9. How to manage intermediate parameters in the D files

There are some intermediate parameters that can be stored in the core profile files as optional parameters. These are information from the CTD or the float that contribute to the understanding or computation of the core parameters. (For example: NB_SAMPLE_CTD, MTIME and TEMP_CNDC in relation to sensor response; PRES_MED, TEMP_MED, PSAL_MED, TEMP_STD, and PSAL_STD in relation to bin-averaging statistics.) Unless specified (e.g. TEMP_CNDC, Section 3.11), they do not receive any QC tests or adjustments.

These core intermediate parameters are denoted by "ic" in Reference Table 3. Their associated _QC and _ADJUSTED variables in the D files should be managed as follows:

- No need to include their _ADJUSTED, _ADJUSTED_QC, _ADJUSTED_ERROR variables.
- Unless specified (e.g. TEMP_CNDC, Section 3.11), <PARAM>_QC = '0'. If the sensor fails completely, PIs can instruct DACs to record <PARAM>_QC = '4'.
- PROFILE_<PARAM>_QC should be computed from <PARAM>_QC. If <PARAM>_QC = '0', PROFILE_<PARAM>_QC = ' '. If the sensor fails completely and <PARAM>_QC = '4' is recorded, then PROFILE_<PARAM>_QC = 'F'.

For completeness and for adherence to the dimension N_PARAM, intermediate parameters should be included in PARAMETER in the SCIENTIFIC_CALIB section, and should be managed as follows:

- In the D file, DATA_MODE = 'D' and all the _ADJUSTED_ variables for the core parameters are filled. Hence none of the SCIENTIFIC_CALIB information should be set to FillValue (ie. empty). For the "ic" <PARAM>:

  SCIENTIFIC_CALIB_EQUATION = 'not applicable'
  SCIENTIFIC_CALIB_COEFFICIENT = 'not applicable'
  SCIENTIFIC_CALIB_COMMENT = 'not applicable'
  SCIENTIFIC_CALIB_DATE = YYYYMMDDHHMISS

(**Note:** Please refer to Section 2.3 for the special case of MTIME for floats with multiple controller boards.)

3.10. Delayed-mode procedures for Deep-Argo data from SBE CTDs

The Deep-Argo program aims to sample temperature and salinity over the full ocean depth to 6000 dbar. There are two Sea-Bird CTDs that are capable of deep sampling: the SBE-61 (used on the Deep APEX and the Deep SOLO) and the Deep SBE-41CP (used on the Deep NINJA and the Deep ARVOR). However, their salinity data show a bias that is dependent on pressure, due to compressibility effects.

In the manufacturer calibration of the SBE CTDs, the correction term for compressibility effects on conductivity is referred to as CPcor. The nominal CPcor value used by Sea-Bird is based on the compressibility of the glass used in the conductivity cell. Comparisons of SBE deep profiles to reference shipboard CTD data show that the nominal CPcor value from Sea-Bird, **CPcor_SBE = –9.57e-08 dbar^{-1}**, is too large, thus resulting in a substantial fresh salinity bias at high pressure. More information can be found in these two references:

Kobayashi, T. (2021), Salinity bias with negative pressure dependency caused by anisotropic deformation of CTD measuring cell under pressure examined with a dual-cylinder cell model, *Deep Sea Research Part I: Oceanographic Research Papers*, *167*, 103420, doi: https://doi.org/10.1016/j.dsr.2020.103420

Kobayashi, T., K. Sato, and B. A. King (2021), Observed features of salinity bias with negative pressure dependency for measurements by SBE 41CP and SBE 61 CTD sensors on deep profiling floats, *Prog. Oceanogr.*, *198*, 102686, doi: https://doi.org/10.1016/j.pocean.2021.102686

In Argo delayed-mode adjustment of salinity data from Deep-Argo floats with the SBE CTDs, a new CPcor value, **CPcor_new**, should be used to re-compute salinity as the first step in 'D' mode. There are two options for CPcor_new: **(a)** the recommended value, and **(b)** the optimized value.

(a). The recommended CPcor_new value

The recommended CPcor_new value is **–11.7 e–08 dbar^{-1}** for both the SBE-61 and Deep SBE-41CP data. This recommended CPcor_new value was derived by the Deep-Argo team, by estimating the optimal CPcor values from 208 deep floats (71 equipped with the Deep SBE-41CP and 137 equipped with the SBE-61).

(b). The optimized CPcor_new value

The optimized CPcor_new value for a float can be obtained in delayed-mode by comparing a deep float profile to a reference profile (e.g. deployment CTD cast). The profiles used in this estimation should be free from other problems, such as sensor drift, spikes, TBTO contamination, etc. A Matlab routine for this estimation (COMPUTE_new_CPcor_brian.m) can be downloaded from https://github.com/ArgoDMQC/DM_CPcor. As this estimation is highly sensitive to the reference layer used, it is recommended to compute the optimized CPcor_new value over a vertical layer as thick as possible, while removing the outliers and the layers with large variability before doing the optimization. It is also recommended to estimate CPcor_new by using different reference layers. If the sensitivity of the CPcor_new estimates to the reference layers lies within +/- 0.5 e–08 dbar^{-1}, we can be confident in the robustness of the optimized CPcor_new value.

Please note that any optimized CPcor_new values should lie within the limits:

$$-20\text{e}{-08}\ \text{dbar}^{-1} \le \text{CPcor\_new} \le -5\text{e}{-08}\ \text{dbar}^{-1}.$$

These limits are based on values derived by the Deep-Argo team. The resulting salinity difference within these limits is estimated to be approximately 0.01 psu averaged over 0-2000 dbar, and slightly less if averaged over 0-6000 dbar. Hence the uncertainty in salinity within these limits is acceptable for core-Argo data. If the optimized estimates of CPcor_new fall outside these limits, the DMQC operator should consider that further investigations are required to demonstrate that such CPcor_new values are valid, or revert to using the recommended values instead.

In delayed-mode, the optimized CPcor_new value (described in **(b)** above) should be used whenever it is available and is checked to be robust. Otherwise, the recommended CPcor_new value (described in **(a)** above) should be used. If possible, PIs should estimate the optimized CPcor_new value soon after deployment and provide this value to the DAC for better 'A'-mode adjustment.

The following are the delayed-mode procedures for adjusting salinity data from Deep-Argo floats with the SBE CTDs.

Step 1. Use CPcor_new to re-compute salinity, PSAL_ADJUSTED_Cnew

The steps to re-compute salinity with CPcor_new are as follows.

(a). Fill PRES_ADJUSTED and TEMP_ADJUSTED

Follow the same 'D' mode procedures as for the 2000-dbar floats, as in Sections 3.3 and 3.4.

(b). Compute original conductivity, Co

This is done by using PRES, TEMP, PSAL. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox:

Co = gsw_C_from_SP (PSAL, TEMP, PRES).

(c). Compute new conductivity, Cnew

$$\mathrm{Cnew} = \mathrm{Co} * \frac{1 + \delta * \mathrm{TEMP} + \mathrm{CPcor\_SBE} * \mathrm{PRES}}{1 + \delta * \mathrm{TEMP\_ADJUSTED} + \mathrm{CPcor\_new} * \mathrm{PRES\_ADJUSTED}}$$

where δ = 3.25e–06,
CPcor_SBE = –9.57e–08 dbar^{-1},
CPcor_new = –11.7e–08 dbar^{-1} for SBE-61 data,
CPcor_new = –11.7e–08 dbar^{-1} for Deep SBE-41CP data, or
CPcor_new = as estimated by DMQC operator.

(d). Compute new salinity, PSAL_ADJUSTED_Cnew

This is done by using PRES_ADJUSTED, TEMP_ADJUSTED, and Cnew. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox:

PSAL_ADJUSTED_Cnew
= gsw_SP_from_C (Cnew, TEMP_ADJUSTED, PRES_ADJUSTED).

** Please note that by using PRES_ADJUSTED, this step also takes into account the effect of APEX/NAVIS pressure adjustment on salinity. **

Step 2. Assess sensor drift and offset by using PSAL_ADJUSTED_Cnew

Sensor drift and offset assessment should then be done by using PSAL_ADJUSTED_Cnew; i.e. *after* CPcor adjustment. If not, the pressure-dependent compressibility effect of CPcor could be interpreted erroneously as a pressure-dependent sensor drift.

Input PSAL_ADJUSTED_Cnew into OWC to check for possible sensor drift and offset.

If sensor drift or offset is detected and the salinity data are adjustable in 'D' mode, then
PSAL_ADJUSTED = PSAL_ADJUSTED_Cnew + OWC adjustment in 'D' mode.

If no sensor drift or offset is detected in 'D' mode, then
PSAL_ADJUSTED = PSAL_ADJUSTED_Cnew.

If the salinity data are assessed as bad and unadjustable in 'D' mode, then
PSAL_ADJUSTED = FillValue.

Some notes:

(a). Evaluation of salinity sensor drift for deep profiles (after CPcor adjustment) should include an assessment of whether a pressure-independent offset adjustment from OWC is sufficient, within uncertainty, to correct for the sensor drift in these deep profiles.

(b). The Deep-Argo expected uncertainty for salinity (after CPcor adjustment) is 0.004. Hence evaluation of salinity sensor drift should be done at the 0.004 level, and not the usual 0.01 level for 2K floats.

(c). The Argo_for_DMQC reference database should not be used in OWC for Deep floats, because it consists of almost all 2K Argo profiles. Instead, the CTD_for_DMQC reference database, 2021_V01 or later versions, contains the GO-SHIP Easy Product (QCLEVEL = GSD), and should be used in OWC for Deep floats. A Matlab tool is available for selecting deep reference CTD for use in OWC. Please refer questions to John.Lyman@noaa.gov, or see http://github.com/ArgoDMQC/Deep_CTD_selection.

(d). Tips: A 6000dbar profile with 2dbar bin-averaged data will contain 3000 vertical levels. You can subsample the profile to reduce the number of vertical levels, before input into OWC, to save some computer time during the objective mapping step.

Step 3. In 'D' mode, ADJUSTED_QC and ADJUSTED_ERROR for Deep-Argo data should be filled as follows

For pressure:
PRES_ADJUSTED_ERROR = (2.5/6000) * PRES + 2.

PRES_QC and PRES_ADJUSTED_QC should be filled as in Section 3.3.

For temperature:
TEMP_ADJUSTED_ERROR = 0.002°C.

TEMP_QC and TEMP_ADJUSTED_QC should be filled as in Section 3.4.

For salinity:
PSAL_ADJUSTED_ERROR = minimum 0.004. DMQC operators should consider increasing this error in case of sensor drift adjustment, to take into account possible pressure dependency of the sensor drift that cannot be detected with certainty.

PSAL_QC and PSAL_ADJUSTED_QC should be filled as in Section 3.5. In particular, salinity data that have been assigned PSAL_QC = '3' in real-time should remain as PSAL_QC = '3', to indicate that the raw salinity data need CPcor_new correction.

Step 4. In 'D' mode, SCIENTIFIC_CALIB for 'PSAL' should be filled as follows when CPcor correction has been applied to Deep-Argo salinity data

It is important to record details of the CPcor correction in the SCIENTIFIC_CALIB section of the D-files. This is because in future, when the delayed-mode processing for Deep Argo salinity data becomes mature enough, we would like to extract the CPcor value from the D-files for real-time adjustments in 'A' mode. Having the CPcor value recorded in the D-files in a parsable syntax will help facilitate that extraction.

Therefore, when CPcor correction has been applied to Deep Argo salinity data in 'D' mode, please add the following character strings to what you would normally write for the parameter PSAL in SCIENTIFIC_CALIB. (Please refer to Section 3.6.2.) In particular, please follow exactly the syntax for **CPcor_new** in SCIENTIFIC_CALIB_COEFFICIENT.

Character strings for recording CPcor correction

SCIENTIFIC_CALIB_EQUATION = 'new conductivity = original conductivity * (1 + delta*TEMP + CPcor_SBE*PRES) / (1 + delta*TEMP_ADJUSTED + CPcor_new*PRES_ADJUSTED)'

SCIENTIFIC_CALIB_COMMENT = 'New conductivity computed by using a different CPcor value from that provided by Sea-Bird.'
SCIENTIFIC_CALIB_COEFFICIENT = '**CPcor_new = xxx;** CPcor_SBE = –9.57e–8; delta = 3.25e–6'

[Note: By adhering to this syntax in SCIENTIFIC_CALIB_COEFFICIENT, where **xxx** is between two fixed char strings, '**CPcor_new =**' and '**;**', the value of xxx can be extracted for use in real-time. For example, if using Matlab, xxx = str2num(char(extractBetween(coeff, 'CPcor_new =', ';'))).]

N_CALIB = 1 or N_CALIB > 1

Delayed-mode operators have the option of either:

(a). Record all the delayed-mode salinity steps in one character string (maximum length 256) with N_CALIB = 1. If using this option, please record the CPcor correction first, then other delayed-mode salinity steps next. For example, SCIENTIFIC_CALIB_COMMENT = 'New conductivity computed by using a different CPcor value from that provided by Sea-Bird. No significant salinity sensor drift detected.'.

or

(b). Record the delayed-mode salinity steps with N_CALIB > 1. If using this option, please record the CPcor correction in N_CALIB index 1, then other delayed-mode salinity steps in N_CALIB index 2, 3, etc. For example:

- at N_CALIB index 1, SCIENTIFIC_CALIB_COMMENT = 'New conductivity computed by using a different CPcor value from that provided by Sea-Bird.';
- at N_CALIB index 2, SCIENTIFIC_CALIB_COMMENT = 'No significant salinity sensor drift detected.'.

Note to DACs

To provide sufficient space for delayed-mode operators to record all calibration information, it is recommended that DACs preset N_CALIB > 1 (= 2 at least) for Deep-Argo profiles at the creation of the profile files. Please coordinate with the delayed-mode operators.

Step 5. Record the DMQC operator in the D-file

Please record the DMQC operator in the //global-attributes section of the D-files. Please refer to Section 3.8 for the agreed format.

[Tips: In Matlab, this can be done by ncwriteatt (Dfile, '/', 'comment_dmqc_operator', 'xxx').]

Step 6. Record the DMQC step in the HISTORY section

After completion of the DMQC process, a record should be appended to the HISTORY section as per other floats. Please refer to Section 3.7.

If the optimized CPcor_new value is used in 'D' mode, then it is recommended that an additional step be recorded in the HISTORY section to record the reference profile used in the optimization. For example:

```
HISTORY_INSTITUTION(N_HISTORY, N_PROF, STRING4)        "IF  "
HISTORY_STEP(N_HISTORY, N_PROF, STRING4)               "ARSQ"
HISTORY_SOFTWARE(N_HISTORY, N_PROF, STRING4)           "DMCP"
(locally defined char string)
HISTORY_SOFTWARE_RELEASE(N_HISTORY, N_PROF, STRING4) "V1.0"
(locally defined char string)
HISTORY_REFERENCE(N_HISTORY, N_PROF, STRING64)         "OVIDE18,stationXX       "
(locally defined char string)
HISTORY_DATE(N_HISTORY, N_PROF, DATE_TIME)             "20200910155005"
HISTORY_ACTION(N_HISTORY, N_PROF, STRING4)             "IP  "
HISTORY_PARAMETER(N_HISTORY, N_PROF, STRING16)         "PSAL            "
```

3.10.1. Summary of delayed-mode QC flags for Deep-Argo data from SBE CTDs

Two steps to adjust Deep-Argo salinity data from SBE CTDs in delayed-mode:

- CPcor correction
- OWC sensor drift adjustment

If no other error is found in delayed-mode and OWC salinity adjustment is < 0.05:

ADJUSTED DATA ('D' MODE)	**RAW DATA (re-flag after DMQC evaluation)**
PRES_ADJUSTED_QC = '1'	PRES_QC = '1'
TEMP_ADJUSTED_QC = '1'	TEMP_QC = '1'
PSAL_ADJUSTED_QC = '1' <= 2000 dbar	PSAL_QC = '1' <= 2000 dbar
PSAL_ADJUSTED_QC = '1' > 2000 dbar	PSAL_QC = '3' > 2000 dbar (because of increased CPcor-related error at high pressures)

3.11. Delayed-mode procedures for RBR*argo*^{3}|2K data

This section describes the delayed-mode procedures for adjusting salinity data from RBR*argo*^{3}|2K CTDs used on Argo floats that profile to 2000 dbar.

Step 1. Re-compute salinity due to compressibility effect

Prior to April 2021, RBR CTDs were calibrated with a set of class-based compressibility coefficients for conductivity, which were shown to be not optimal and resulted in a pressure-dependent bias in salinity, peaking at about 0.01 at 2000 dbar. This has been mitigated in RBR CTDs calibrated post-April 2021, as RBR moved to unit-based compressibility calibration after April 2021. However, for RBR CTDs calibrated pre-April 2021, salinity needs to be re-computed in delayed-mode with new compressibility coefficients. Please note that the manufacturer pre-deployment calibration date of RBR CTDs can be found in the sensor calibration sheets, or via https://oem-lookup.rbr-global.com

The new compressibility coefficients are provided in an RBR lookup table in Appendix 6.9 for each Argo float that has the pre-April 2021 calibrated RBR*argo*^{3}|2K CTD. The lookup table can also be found at:
https://github.com/ArgoDMQC/RBRargo_DMQC/blob/main/Compressibility_correction/RBRargo3_compressibility_table.csv

For pre-April 2021 calibrated RBR CTDs with new compressibility coefficients, follow Step 1: a), b), c), d1), then proceed to Steps 2 and 3.

For post-April 2021calibrated RBR CTDs, and pre-April 2021 calibrated RBR CTDs with no new compressibility coefficients available (NaN), follow Step 1: a), b), d2), then proceed to Steps 2 and 3.

(a). Fill PRES_ADJUSTED and TEMP_ADJUSTED

Follow the same 'D' mode procedures as per the 2000-dbar floats in Sections 3.3 and 3.4.

(b). Compute original conductivity, Co

This is done by using PRES, TEMP, PSAL. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox, Co = gsw_C_from_SP (PSAL, TEMP, PRES).

(c). Compute new conductivity, Cnew

$$\mathrm{Cnew} = \mathrm{Co} * \frac{1 + \mathrm{X2old} * \mathrm{PRES} + \mathrm{X3old} * \mathrm{PRES}^2 + \mathrm{X4old} * \mathrm{PRES}^3}{1 + \mathrm{X2new} * \mathrm{PRES\_ADJUSTED} + \mathrm{X3new} * \mathrm{PRES\_ADJUSTED}^2 + \mathrm{X4new} * \mathrm{PRES\_ADJUSTED}^3}$$

where
X2old = 1.8732×10^{-6}, X2new = see lookup table for each specific unit;
X3old = -7.7689×10^{-10}, X3new = see lookup table for each specific unit;
X4old = 1.4890×10^{-13}, X4new = see lookup table for each specific unit.

(d1). Compute new salinity, PSAL_ADJUSTED_Padj

This is done by using PRES_ADJUSTED, TEMP_ADJUSTED, and Cnew. For example, if using the Gibbs-SeaWater (GSW) Oceanographic Matlab Toolbox:

PSAL_ADJUSTED_Padj
= gsw_SP_from_C (Cnew, TEMP_ADJUSTED, PRES_ADJUSTED).

** Please note that by using PRES_ADJUSTED, this step also takes into account the effect of APEX/NAVIS pressure adjustment on salinity. **

(d2). PSAL_ADJUSTED_Padj

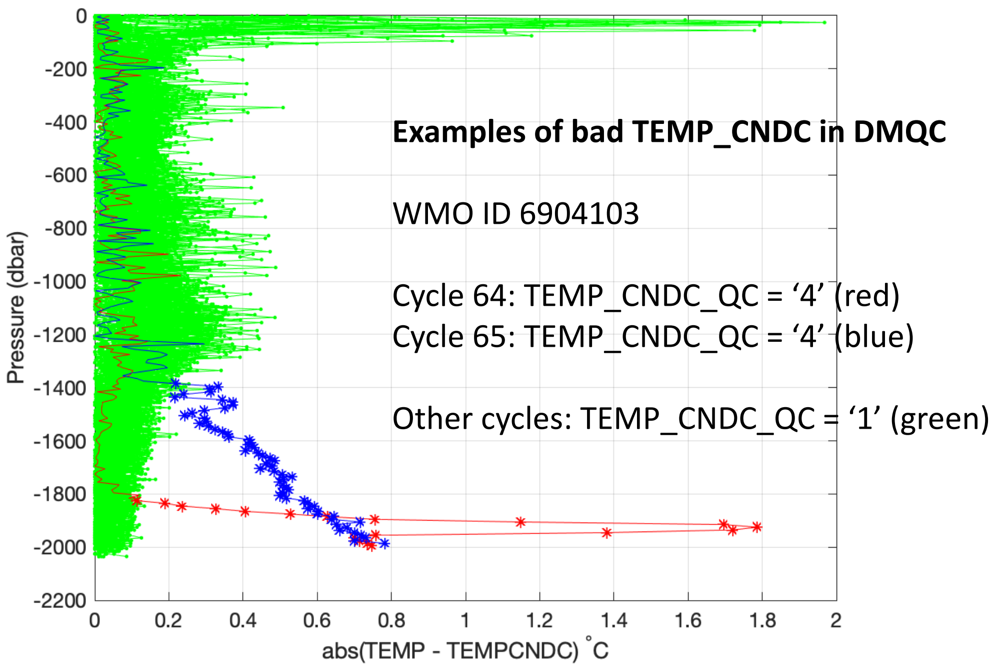
PSAL_ADJUSTED_Padj
= gsw_SP_from_C (Co, TEMP_ADJUSTED, PRES_ADJUSTED).

Step 2. Apply thermal inertia correction

This step requires the intermediate parameter TEMP_CNDC (the internal temperature of the conductivity cell), and the elapsed time at each sample since the start of the vertical profile.

(a). Check TEMP_CNDC visually

Sometimes the CTD can malfunction and corrupt the TEMP_CNDC data. Delayed-mode operators should check if the TEMP_CNDC data are bad by visually checking TEMP_CNDC against TEMP. Bad TEMP_CNDC data should be marked with TEMP_CNDC_QC '4'. Otherwise, TEMP_CNDC_QC = '1'.



(b). Calculate conductivity, Cadj

This is done by using PSAL_ADJUSTED_Padj from Step 1, with PRES_ADJUSTED and TEMP_ADJUSTED. For example: Cadj
= gsw_C_from_SP (PSAL_ADJUSTED_Padj, TEMP_ADJUSTED, PRES_ADJUSTED).

(c). Estimate elapsed time, elptime

The elapsed time at each measurement since the start of the vertical profile, *elptime* (in seconds), should be estimated by using different methods for different models of float. Please refer to Appendix 6.8.

(d). Compute TEMP_celltm

In delayed-mode, the thermal inertia correction should be done in temperature, and should consist of 3 components: the C-T lag adjustment, the long timescale thermal mass adjustment, and the short timescale thermal mass adjustment. A Matlab routine "RBRargo3_celltm.m" is available for computing all 3 components, and outputs TEMP_celltm. See Dever et al. (2022) and https://github.com/ArgoDMQC/RBRargo_DMQC/tree/main/Thermal_inertia

TEMP_celltm
= RBRargo3_celltm(TEMP_ADJUSTED, PRES_ADJUSTED, TEMP_CNDC, elptime).

Note (*optional step*):

Bad TEMP_CNDC can affect the output TEMP_celltm, and therefore the values in PSAL_ADJUSTED (see **(e)** below). If a profile has bad TEMP_CNDC on *too many* measurement levels, the delayed-mode operator can attempt to salvage the salinity profile by estimating a pseudo TEMP_CNDC profile. This can be done by using the routine "RBRargo3_TEMP_CNDC_from_TEMP_ADJUSTED.m", available at https://github.com/ArgoDMQC/RBRargo_DMQC/tree/main/TEMP_CNDC

Please note that this optional step is not meant for use to estimate single bad points. If the delayed-mode operator decides to take this optional step, then the entire pseudo TEMP_CNDC profile should be inputted into "RBRargo3_celltm.m" to obtain the TEMP_celltm output.

Please also note that this optional step *does not* replace the original measurements in <PARAM> = TEMP_CNDC. Values in <PARAM> are *never* altered.

(e). Use TEMP_celltm to compute salinity PSAL_ADJUSTED_Padj_CTM

PSAL_ADJUSTED_Padj_CTM
= gsw_SP_from_C (Cadj, TEMP_celltm, PRES_ADJUSTED).

(f). At levels where TEMP_CNDC_QC = '4' and no estimate is attempted from (d) above, PSAL_ADJUSTED_Padj_CTM should also be considered bad. These levels should be excluded from the input into OWC (Step 3) because they will skew the least squares fit.

Step 3. Assess sensor drift and offset with OWC

Delayed-mode sensor drift and offset assessment should be done by using PSAL_ADJUSTED_Padj_CTM. Input PSAL_ADJUSTED_Padj_CTM into OWC to check for possible sensor drift and offset. Follow "best practices" in Section 3.6.

If sensor drift or offset is detected and the salinity data are adjustable in 'D' mode, then PSAL_ADJUSTED = PSAL_ADJUSTED_Padj_CTM + OWC adjustment in 'D' mode.

If no sensor drift or offset is detected in 'D' mode, then PSAL_ADJUSTED = PSAL_ADJUSTED_Padj_CTM.

If the salinity data are assessed as bad and unadjustable in 'D' mode, then PSAL_ADJUSTED = FillValue.

Step 4. In 'D' mode, ADJUSTED_QC and ADJUSTED_ERROR for RBR*argo*^{3}|2K CTD data (< 2000 dbar) should be filled as follows

For pressure:

PRES_ADJUSTED_ERROR = 1 dbar

PRES_QC and PRES_ADJUSTED_QC should be filled as in Section 3.3. In particular, pressure data that have been assigned PRES_QC = '3' in real-time should be re-flagged as PRES_QC = '1' in delayed-mode if no error is found.

For temperature:

TEMP_ADJUSTED_ERROR = 0.002°C

TEMP_QC and TEMP_ADJUSTED_QC should be filled as in Section 3.4. In particular, temperature data that have been assigned TEMP_QC = '3' in real-time should be re-flagged with TEMP_QC = '1' in delayed-mode if no error is found.

For salinity:

PSAL_ADJUSTED_ERROR = maximum [$(\sum \text{adjustment\_error}^2)^{1/2}$, 0.01], where "adjustment_error" is the uncertainty from each adjustment applied to salinity.

The adjustment uncertainty from Step 1 (compressibility correction for pre-April 2021 calibrated RBR*argo*^{3}|2K CTDs) is 0.005 psu. If new compressibility coefficients are not available (NaN), delayed-mode operators should compare PSAL_ADJUSTED with nearby reference data to determine an appropriate error estimate.

The adjustment error from Step 2 (thermal inertia correction) is the size of the correction.

The adjustment error from Step 3 (sensor drift adjustment) is described in Section 3.5.6.

At levels where TEMP_CNDC_QC = '4', PSAL_QC = '4', and

- if a pseudo TEMP_CNDC profile is used to estimate TEMP_celltm in Step 2 (d), then PSAL_ADJUSTED_QC = '1' or '2', based on delayed-mode evaluation on the quality of the estimate;
- if no estimate is used, then PSAL_ADJUSTED_QC = '4'.

At levels where TEMP_CNDC_QC = '1', PSAL_ADJUSTED_QC should be filled as per Section 3.5, and PSAL_QC should be re-flagged as per Section 3.1.

Salinity data that have been assigned PSAL_QC = '3' in real-time should be re-flagged with PSAL_QC = '1' in delayed-mode if no other error is found. However, salinity data from pre-April 2021 calibrated RBR*argo*^{3}|2K CTDs that have been assigned PSAL_QC = '3' in real-time should remain with PSAL_QC = '3', to indicate that the raw salinity data from pre-April 2021 calibrated RBR*argo*^{3}|2K need compressibility correction.

For pre-April 2021 calibrated RBR*argo*^{3}|2K CTDs, if new compressibility coefficients are not available (NaN), delayed-mode operators should compare PSAL_ADJUSTED with nearby reference data to determine if PSAL_ADJUSTED_QC = '1' or '2'.

Step 5. In 'D' mode, the SCIENTIFIC_CALIB section should be filled as follows

N_PARAM = index for 'PRES' and 'TEMP'

N_CALIB = 1. Please fill these as per other CTDs.
N_CALIB = 2, 3. Please fill these with 'not applicable'.

N_PARAM = index for 'TEMP_CNDC'

Please fill these with 'not applicable'. See Section 3.9.

N_PARAM = index for 'PSAL'

N_CALIB = 1. Compressibility correction.

Example a). RBR CTDs calibrated pre-April 2021 and have new compressibility coefficients available in lookup table

SCIENTIFIC_CALIB_EQUATION = 'new conductivity = original conductivity * (1 + X2old*PRES + X3old*PRES^2 + X4old*PRES^3) / (1 + X2new*PRES_ADJUSTED + X3new*PRES_ADJUSTED^2 + X4new*PRES_ADJUSTED^3)'

SCIENTIFIC_CALIB_COEFFICIENT = 'X2old = 1.8732e-6, X3old = -7.7689e-10, X4old = 1.4890e-13, X2new = ..., X3new = ..., X4new = ...'

SCIENTIFIC_CALIB_COMMENT = 'Pre-April 2021 calibrated RBR CTD. Salinity re-computed by using new compressibility coefficients provided by RBR.'

Example b). RBR CTDs calibrated pre-April 2021 but have no new compressibility coefficients available (NaN) in lookup table

SCIENTIFIC_CALIB_EQUATION = 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENTS = 'not applicable'
SCIENTIFIC_CALIB_COMMENTS = 'Pre-April 2021 calibrated RBR CTD. No new compressibility coefficients available.'

Example c). RBR CTDs calibrated post-April 2021

SCIENTIFIC_CALIB_EQUATION = 'not applicable'
SCIENTIFIC_CALIB_COEFFICIENTS = 'not applicable'
SCIENTIFIC_CALIB_COMMENTS = 'Post-April 2021 calibrated RBR CTD. No compressibility correction needed.'

N_CALIB = 2. Thermal inertia correction with RBRargo3_celltm.

SCIENTIFIC_CALIB_EQUATION = 'PSAL_ADJUSTED = gsw_SP_from_C (Cadj, TEMP_celltm, PRES_ADJUSTED); TEMP_celltm = RBRargo3_celltm(TEMP_ADJUSTED, PRES_ADJUSTED, TEMP_CNDC, elapsed time); Cadj is conductivity from re-computed salinity due to compressibility effect.'

SCIENTIFIC_CALIB_COEFFICIENT = 'CellTM coefficients are computed as functions of float ascent rate Vp in cm/s; ctcoeff = 0.14*(Vp).^(-1.00); alpha = 0.37*(Vp).^(-1.03); tau = 16.02*(Vp).^(-0.26).'

SCIENTIFIC_CALIB_COMMENT = 'RBR thermal inertia correction applied to salinity during delayed-mode post-processing. See Dever et al. (2022).'

or

SCIENTIFIC_CALIB_COMMENT = 'RBR thermal inertia correction applied to salinity during delayed-mode post-processing. See Dever et al. (2022). TEMP_CNDC is inferred from TEMP because the original values are bad.'

N_CALIB = 3. Delayed-mode salinity adjustment for sensor drift and offset.

As per other CTDs. OWC adjustment applied, or no significant salinity sensor drift found.

Step 6. Record the DMQC operator in the D-file

Please record the DMQC operator in the //global-attributes section of the D-files. Please refer to Section 3.7 for the agreed format.

[Tips: In Matlab, this can be done by ncwriteatt(Dfile, '/', 'comment_dmqc_operator', 'xxx').]

Step 7. Record the DMQC step in the HISTORY section

After completion of the DMQC process, a record should be appended to the HISTORY section. Please refer to Section 3.7.3.

References

Dever, M., B. Owens, C. Richards, S. Wijffels, A. Wong, I. Shkvorets, M. Halverson, and G. Johnson (2022). Static and dynamic performance of the RBR*argo*^{3} CTD. Journal of Atmospheric and Oceanic Technology, 39 (10), 1525-1539. https://doi.org/10.1175/JTECH-D-21-0186.1

3.11.1. Summary of delayed-mode QC flags for RBRargo3|2K data

Three steps to adjust RBR*argo*^{3}|2K salinity in delayed-mode:

- Compressibility correction for RBR*argo*^{3}|2K CTDs calibrated pre-April 2021
- Thermal inertia correction (C-T lag, long timescale, short timescale)
- OWC sensor drift adjustment

If no other error is found in delayed-mode and OWC salinity adjustment is < 0.05:

ADJUSTED DATA ('D' MODE)	**RAW DATA (re-flag after DMQC evaluation)**
PRES_ADJUSTED_QC = '1'	PRES_QC = '1'
TEMP_ADJUSTED_QC = '1'	TEMP_QC = '1'
PSAL_ADJUSTED_QC = '1' for RBR*argo*^{3}|2K CTDs calibrated post-April 2021	PSAL_QC = '1' for RBR*argo*^{3}|2K CTDs calibrated post-April 2021
PSAL_ADJUSTED_QC = '1' for RBR*argo*^{3}|2K CTDs calibrated pre-April 2021	PSAL_QC = '3' for RBR*argo*^{3}|2K CTDs calibrated pre-April 2021 (because of known compressibility error)

3.11.2. Correction for the square-root salinity discrepancy for RBRargo3 CTD

In January 2024, RBR discovered an inconsistency between salinity readings reported by an instrument while streaming and the salinity readings calculated by Ruskin with the downloaded data. The cause of the discrepancy between the onboard calculated and the Ruskin calculated salinity was identified to be from an approximation in the square-root calculation onboard the instrument. The maximum salinity discrepancy has been established to be less than 0.002. This discrepancy only affects instruments streaming or being interfaced by a host controller, such as a float controller. The instruments' calibrations are unaffected.

The list of serial numbers affected by this error is provided here:

https://oem.rbr-global.com/floats/0017299

This error can be corrected during post-processing by using a correction algorithm provided by RBR here:

https://github.com/ArgoDMQC/RBRargo_DMQC/tree/main/SQRT_salinity_adjustment

If needed, this algorithm should be applied before other delayed-mode steps.

4. Real-time procedures on trajectory file data

The following quality control tests are applied in real-time to data in the trajectory files.

~~**Test 1. Platform identification test**~~ *(This test is not needed for traj files.)*
~~Every centre handling float data and posting them to the GTS will need to prepare a metadata file for each float and in this is the WMO number that corresponds to each float ptt. There is no reason why, except because of a mistake, an unknown float ID should appear on the GTS.~~

~~Action: If the correspondence between the float ptt cannot be matched to the correct WMO number, then none of the data from the float should be distributed on the GTS.~~

Test 2. Impossible date test

This test requires that the Julian day of a float profile be later than 1st January 1997 and earlier than the current date of the check (in UTC time). Hence if JULD = number of days elapsed since 1st January 1950, then this test checks that: 17167 <= JULD < UTC date of check.

Action: If JULD of a profile fails this test, the date of the profile should be flagged as bad data (‘4’).

Test 3. Impossible location test

This test requires that the observation latitude and longitude from a float profile be sensible.

- Latitude in range −90° to 90°
- Longitude in range −180° to 180°

Action: If either latitude or longitude fails this test, the position should be flagged as bad data (‘4’).

Test 4. Position on land test

This test requires that the observation latitude and longitude from a float profile be located in an ocean. Use can be made of any topography/bathymetry file that allows an automatic test to check if a position is located on land. We suggest the use of at least the 5-minute bathymetry file that is generally available. This is commonly called ETOPO5/TerrainBase and can be downloaded from:

https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ngdc.mgg.dem:3141

Action: If a position cannot be located in an ocean, the position should be flagged as bad data (‘4’).

Test 5. Impossible speed test

Drift speeds for floats can be generated given the positions and times of the floats when they are at the surface and between profiles. In all cases we would not expect the drift speed to exceed 3 ms^{-1}. If it does, it means either the positions or times are bad data, or a float is mislabeled. Using the multiple positions that are normally available for a float while at the surface, it is often possible to isolate the one position or time that is in error.

Action: If this test is failed, the position, the time, or both, should be flagged as bad data (‘4’).

(Note: Floats that use the ARGOS system to obtain position data can use Test 20 instead.)

Test 6. Global range test

This test applies a gross filter on the observed values of pressure (PRES), temperature (TEMP) and salinity (PSAL). The ranges need to accommodate all the expected extremes encountered in the oceans.

- Pressure cannot be less than –5 dbar.
- Pressure in the range –5 to –2.4 dbar should be considered 'probably bad'. The limit of 2.4 dbar is determined based on feedback from operational users.
- Temperature should be in the range –2.5 to 40.0°C.
- Salinity should be in the range 2 to 41.0 PSU.

Action:
If PRES < –5dbar, then PRES_QC = '4', TEMP_QC = '4', PSAL_QC = '4'.
If –5dbar <= PRES <= –2.4dbar, then PRES_QC = '3', TEMP_QC = '3', PSAL_QC = '3'.
If TEMP values are outside of the range –2.5 to 40.0°C, then TEMP_QC = '4'.
If PSAL values are outside of the range 2 to 41.0 PSU, then PSAL_QC = '4'.

Test 7. Regional range test

This test applies to certain regions of the world where conditions can be further qualified. In this case, specific ranges for observations from the Mediterranean Sea and the Red Sea further restrict what are considered sensible values. The Red Sea is defined by the region 10N, 40E; 20N, 50E; 30N, 30E; 10N, 40E. The Mediterranean Sea is defined by the region 30N, 6W; 30N, 40E; 40N, 35E; 42N, 20E; 50N, 15E; 40N, 5W; 30N, 6W.
Red Sea:

- Temperature should be in the range 21.0 to 40.0°C.
- Salinity should be in the range 2 to 41.0 PSU.

Mediterranean Sea:

- Temperature should be in the range 10.0 to 40.0°C.
- Salinity should be in the range 2 to 40.0 PSU.

Action: If a value fails this test, it should be flagged as bad data (‘4’). If temperature and salinity values at the same pressure both fail this test, both values should be flagged as bad data (‘4’).

Test 15. Supplemental sensor exclusion list test

This test is implemented as a mechanism for data assembly centres (DACs) to flag, in real-time, sensors that are potentially not working properly.

The exclusion list contains the following 7 items:

- Platform: Float WMO Id
- Parameter: name of the affected parameter
- Start date: from that date, all measurements for this parameter are flagged as bad (‘4’) or probably bad (‘3’) or probably good (‘2’)
- End date: from that date, measurements are not flagged as bad (‘4’) or probably bad (‘3’) or probably good (‘2’)
- Flag: value of the flag to be applied to all measurements of the parameter
- Comment: comment from the PI on the problem
- DAC: data assembly center for this float

Example:

Float WMO Id	Parameter	Start date	End date	Flag	Comment	DAC
1900206	PSAL	20030925		3		IF

Each DAC manages an exclusion list in the following format:

- Exclusion list file: ascii csv (comma separated values)
- Naming convention: xxx_exclusionlist.csv
  xxx: DAC name (e.g.: aoml_exclusionlist.csv, coriolis_exclusionlist.csv)
- PLATFORM, PARAMETER, START_DATE, END_DATE, QC, COMMENT, DAC
  e.g. 4900228, TEMP, 20030909, , 3, , AO
  e.g 1900206, PSAL, 20030925, , 3, , IF

The decision to insert a float parameter in the exclusion list comes from the PI or the delayed-mode operator. A float parameter should be put in the exclusion list when sensor drift or offset is too big to be corrected adequately in real-time, or when the sensor is judged to be potentially not working properly.

Another means when a float parameter can be put in the exclusion list is when it fails the MinMax test, which is documented in the Coriolis monthly report. DACs should put the offending float parameters in the exclusion list automatically without contacting the PI or the delayed-mode operator, by using the start cycles and suggested QC flags in the MinMax test section of the Coriolis monthly report.

The exclusion list only concerns real-time files (R-files). When an anomalous float is dead and the offending parameter has been adjusted in delayed-mode, it should be removed from the exclusion list. When an anomalous float is active and the offending parameter has been partially adjusted in delayed-mode, it should remain in the exclusion list if real-time adjustment is not adequate.

Action: Affected parameters in the exclusion list are flagged as probably good data (‘2’) or probably bad data (‘3’) or bad data (‘4’), as determined by the PI or the delayed-mode operator or the MinMax test in the Coriolis monthly report. DACs should make sure to revisit the QC flags of the profiles whose observation dates are between the exclusion list “start date” and the exclusion list insertion date.

Test 20. Questionable ARGOS position test

For floats that use the ARGOS system to obtain position data, this test can be used in lieu of Test 5 (Impossible Speed Test).

This test identifies questionable ARGOS position data collected during surface drift of a float cycle by considering the float speed at the sea surface and ARGOS position errors. Details of the method can be found in Nakamura et al (2008), “Quality control method of Argo float position data”, JAMSTEC Report of Research and Development, Vol. 7, 11-18.

A brief description of the procedure is summarized here.

A). Collect all ARGOS positions during surface drift of a float cycle. The distance between two positions *A* and *B* is referred to as a segment. A segment is considered questionable if:

(i) the float speed along the segment exceeds 3 m s^{-1}, **and**
(ii) the length of the segment is longer than the critical error length, defined as

$$1.0 \times \sqrt{Er_A^{\ 2} + Er_B^{\ 2}}$$

where Er_A^2 and Er_B^2 are the radii of position error of the ARGOS system (150m, 350m, and 1000m for ARGOS class 3, 2, and 1 respectively) at *A* and *B* respectively.

B). If a segment is not considered questionable, then both positions *A* and *B* are good.

C). If a segment is considered questionable, then:

- if the ARGOS class at *A* and *B* are different, then the position with the less accurate ARGOS class is flagged as '3';
- if the ARGOS class at *A* and *B* are the same, and there is one good position before and one good position after *A* and *B* (i.e. there are 4 positions for the check), then the position that gives the higher speed along the segment from the previous good position to the later good position is flagged as '3';
- if the ARGOS class at *A* and *B* are the same, and there is one good position either before or after *A* and *B* (i.e. there are 3 positions for the check), then the position that gives the higher speed along the segment either from the previous good position or to the later good position is flagged as '3';
- if the ARGOS class at *A* and *B* are the same, but there are no other good positions around *A* and *B* (i.e. there are 2 positions for the check), then both *A* and *B* are flagged as '3'.

5. Delayed-mode procedures on trajectory file data

5.1. Definition of a Dtraj file

A Dtraj file can exist in either the V3.1 or the V3.2 traj format, but it is *highly recommended* to switch to the V3.2 traj format as soon as any delayed-mode data are stored in a trajectory file. This is because two new variables, JULD_DATA_MODE and TRAJECTORY_PARAMETER_DATA_MODE, have been added to the V3.2 traj format to provide information on the DATA_MODE of the time measurements and each parameter measurement respectively. Also, a JULD_CALIB section and a SCIENTIFIC_CALIB section have been added to the V3.2 traj format to record calibration information.

For both the V3.1 & V3.2 traj formats

<u>Basic definition used by the GDAC file checker:</u>
A traj file is a Dtraj file when DATA_MODE = ‘D’ for at least one cycle.

<u>Tasks for delayed-mode operators:</u>
For each cycle, there are N_PARAMs (one for each PARAM) + positions/times/cycle numbers to check/adjust. DATA_MODE = ‘D’ when positions are checked **and** times/cycle numbers are checked/adjusted **or** <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC, <PARAM>_ADJUSTED_ERROR are filled with delayed-mode values for at least one <PARAM>.

For the V3.2 traj format

DATA_MODE(N_CYCLE) = ‘D’ when at least one TRAJECTORY_PARAMETER_DATA_MODE(N_MEASUREMENT, N_PARAM) **or** JULD_DATA_MODE(N_MEASUREMENT) in the concerned cycle is set to ‘D’.

JULD_DATA_MODE = ‘D’ when JULD_ADJUSTED are filled with delayed-mode values.

TRAJECTORY_PARAMETER_DATA_MODE = ‘D’ when <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC, <PARAM>_ADJUSTED_ERROR are filled with delayed-mode values for the concerned <PARAM>.

The following sections describe the delayed-mode procedures for various kinds of data in the trajectory files.

5.2. Cycle numbers

Delayed-mode operators should first check that the cycle numbers are correctly assigned. The cycle number is the link between the trajectory file and the profile files; hence it is important that the correspondence is correct. In particular, correctly assigned cycle numbers are needed for computing subsurface displacements and time periods, which in turn are used for estimating subsurface velocities. For example, if a float cycle is missing but CYCLE_NUMBER is not incremented correctly, (N and N+1 are assigned, but N and N+2 are what occurred), the deduced subsurface velocity will be inaccurate or spurious. Alternatively, if a float cycle is not missing but CYCLE_NUMBER is incremented by more than one, (N and N+2 are assigned, but N and N+1 are what occurred), the displacement at N+1 would be missed and would not be calculated. Finally, because files may have to be re-written if CYCLE_NUMBER is changed, it is suggested to do QC on cycle numbers first prior to other DMQC work.

Some floats transmit the cycle number, while others do not. However, even for floats that transmit the cycle number, it can be incorrect due to a bad transmission. For floats with two-way communication, it is important to keep track of the float's mission for each cycle when trying to determine if a cycle number is bad.

If a cycle number is correct, copy the correct cycle number into:

- CYCLE_NUMBER_ADJUSTED(N_MEASUREMENT)
- CYCLE_NUMBER_INDEX_ADJUSTED(N_CYCLE)

If a cycle number is determined to be incorrect, the DMQC operator has two options:

(a). Leave the raw, incorrect cycle number in CYCLE_NUMBER(N_MEASUREMENT) and CYCLE_NUMBER_INDEX(N_CYCLE), then put the correct cycle number in CYCLE_NUMBER_ADJUSTED(N_MEASUREMENT) and CYCLE_NUMBER_INDEX_ADJUSTED(N_CYCLE).

(b). Adjust CYCLE_NUMBER to the correct one and fill CYCLE_NUMBER_ADJUSTED with the same correct cycle number.

If option (b) is taken, *all corresponding profile, meta and tech files will need to be re-written* to ensure that the value in CYCLE_NUMBER in the Dtraj file corresponds to the cycle numbers in all the other files.

<u>How to identify bad cycle numbers</u>

1. Reference all surface times back to the first cycle

This method works well for floats with constant cycle durations (such as APEX Argos floats), but is also useful for other floats, since cycle duration variations are only a few minutes, while errors in cycle number result in cycle duration variations on the order of days.

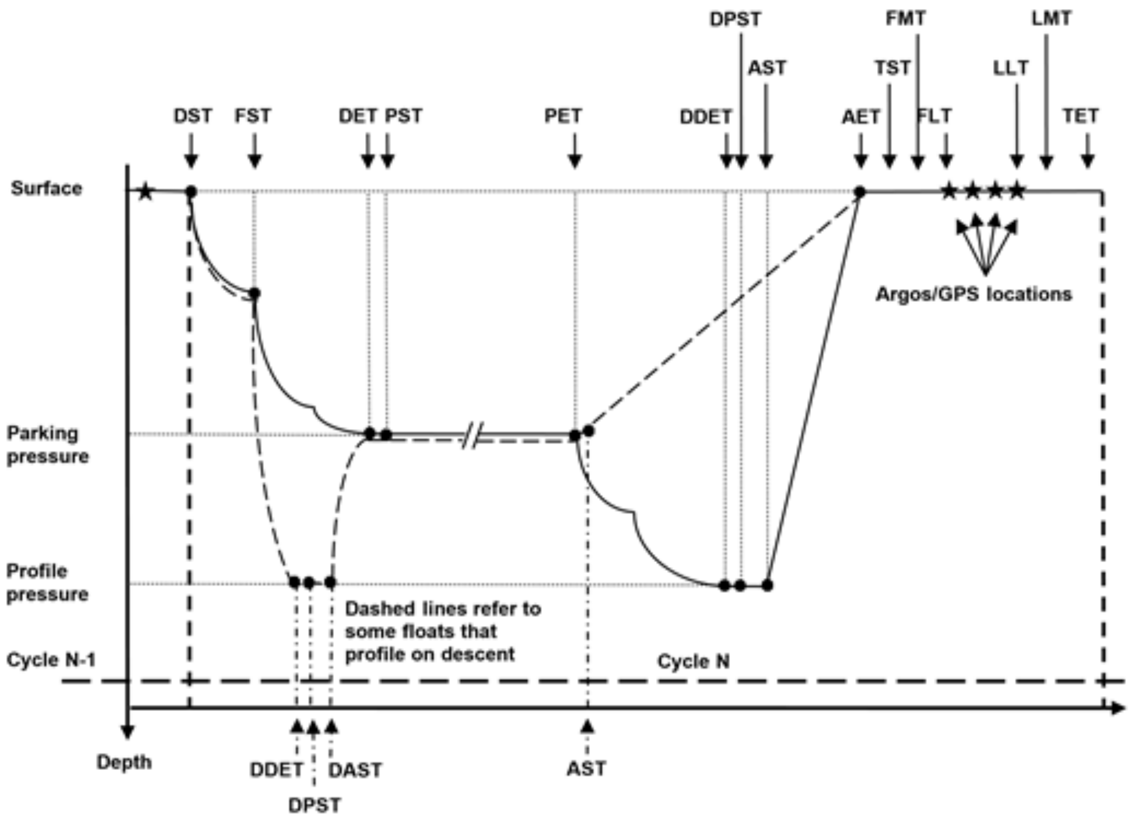
The difficulty with this method is that cycle durations are based on information provided in the configuration section of the meta file, which 1) can be erroneous, or 2) can differ from the effective cycle duration for some cycles. This can be due to unpredictable variation in float behavior, such as floats waiting at the surface for a longer period of time due to communication timeouts, or floats waiting to begin ascent to the surface because the profile pressure has not been reached. Depending on the float type, different configuration parameters need to be examined in the metafile. Nonetheless, this method can be used to detect errors in cycle numbers. Examples can be found in the "DAC trajectory cookbook" (https://doi.org/10.13155/29824, Section 4.3).

2. Other ways to identify bad cycle numbers

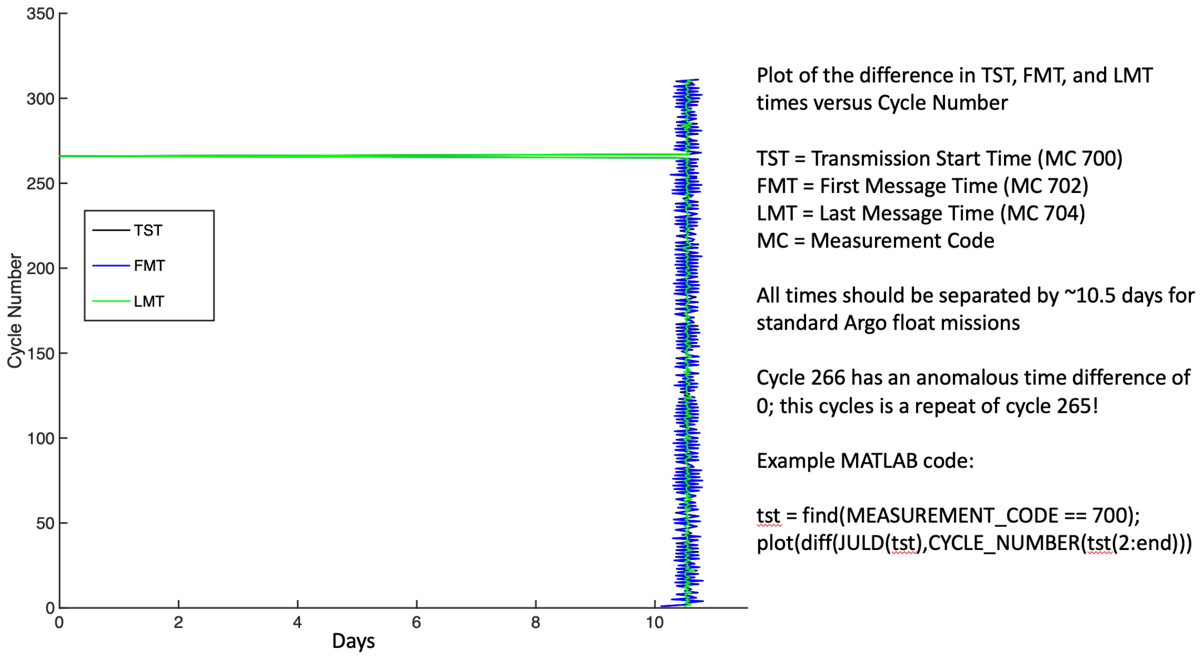
Compare the cycle times in the float's mission config parameters in the meta file to the times between the first surface fix of consecutive cycles in the traj file. Any anomalously long time periods between consecutive cycle could indicate errors in CYCLE_NUMBER.

3. Some examples

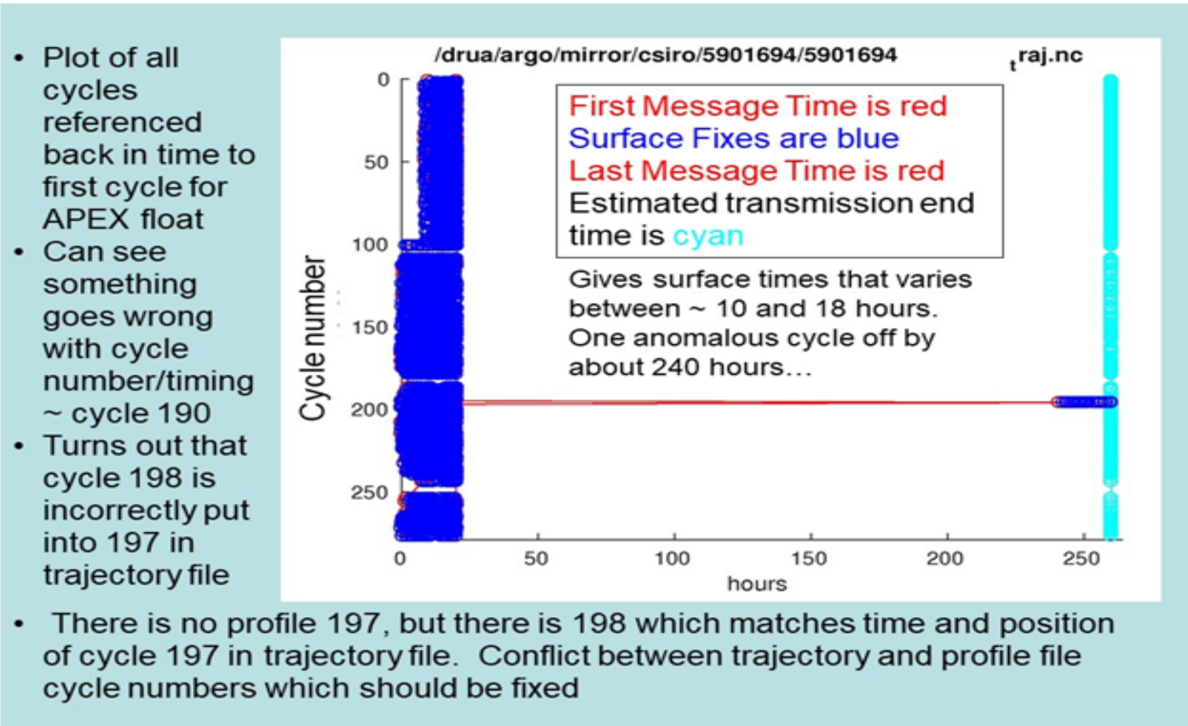
Here are examples of some basic checks that DMQC operators can do. These checks are based on known float cycling missions, as described in this schematic diagram. Please refer to Argo Reference Table 15 for the explanation of the various abbreviated terms in this diagram.



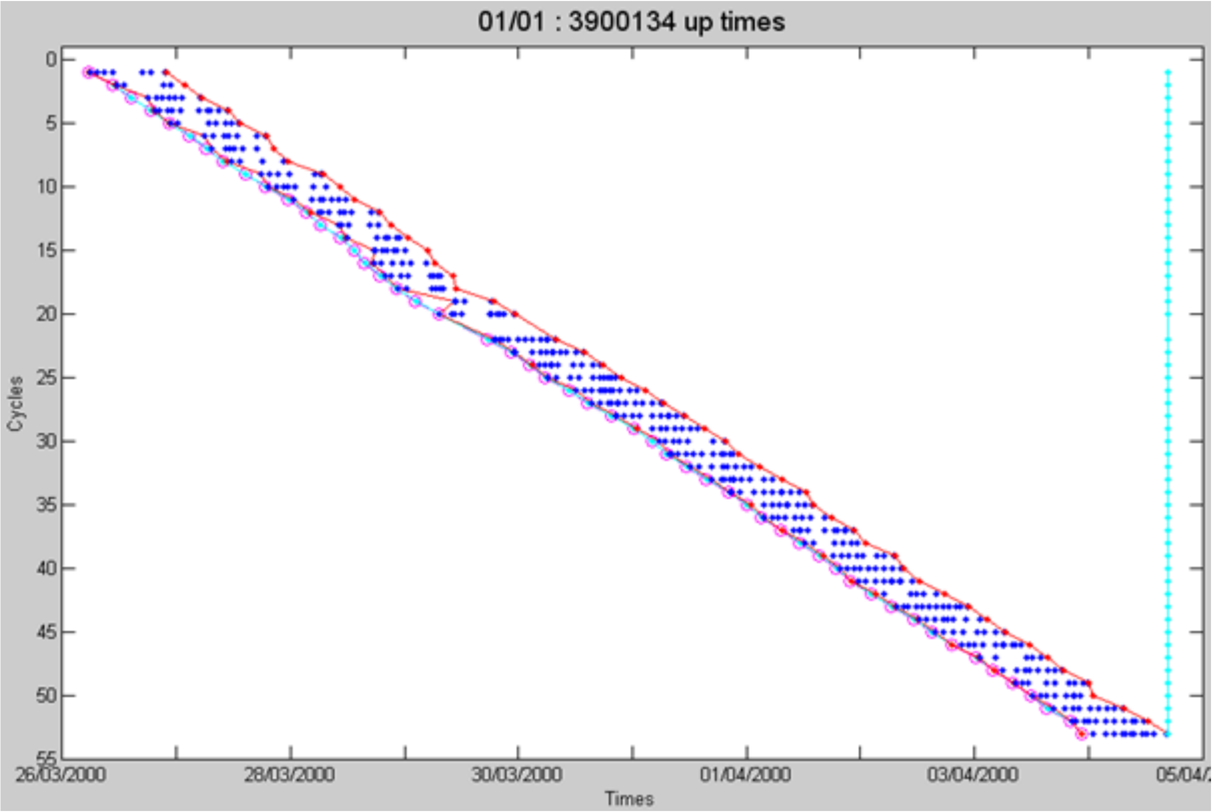
Example 1. This shows a jump in float surface times (TST, FMT, LMT) for an Argos float (WMO ID 5900634) from ~10.5 days to ~0 days at cycle 265. This outlier was caused by erroneously recording the same data in cycles 265 and 266, where in fact the data belong to cycle 266, and cycle 265 is missing.



Example 2. This shows that the surface time (WMO ID 5901694) at cycle 197 is offset from the rest by ~240 hours (~10 days). Checking with the profile files shows there is no cycle 197, but the time and position in profile file cycle 198 match the information in traj file cycle 197. So the trajectory file has the wrong cycle number and 197 should be corrected to 198.



Example 3. The method of referencing all surface times back to the first cycle relies on the trajectory file cycle times aligning with the config cycle times found in the meta files. When they are different, the cycles do not line up, but rather tilt one way or the other. This could be the result of clock drift, or could be due to the float's actual behaviour. Here is an example of a difference in cycle times that looks like a clock drift. In this example (WMO ID 3900134), one can see how a config cycle time of 240 hours, instead of the actual 244 hours, looks like a positive clock drift (estimated as +147:44:36 per year).



Note that there can be bad config cycle times in float meta files. A list of corrected cycle times was created when making the ANDRO Atlas and is available here: [CorrectedMetadata_20131010.xlsx (ifremer.fr)]

5.3. Positions

Argos floats

Delayed-mode operators should use the JAMSTEC position qc test, if not done in real-time, then adjust POSITION_QC accordingly. Details of the JAMSTEC position test can be found in Nakamura et al (2008), "Quality control method of Argo float position data", JAMSTEC Report of Research and Development, Vol. 7, 11-18.

Iridium floats

Delayed-mode operators should use the JAMSTEC position qc test with an associated error of 30m, if not done in real-time, then adjust POSITION_QC accordingly. Details of the JAMSTEC position test can be found in Nakamura et al (2008), "Quality control method of Argo float position data", JAMSTEC Report of Research and Development, Vol. 7, 11-18.

How to estimate float positions

If positions are missing, or not available because floats are prevented from surfacing as a result of the ice-avoidance algorithm, or the reported positions are bad, then an estimated position can be added to the trajectory file. The most commonly used estimation method is linear interpolation between known positions. Other more involved methods that have been published in peer-reviewed publications can also be used.

If a position has been estimated, insert it into the trajectory file as follows:

LATITUDE	= estimated latitude
LONGITUDE	= estimated longitude
POSITION_ACCURACY	= 'U' (unknown)
POSITION_QC	= '8' (estimated)
AXES_ERROR_ELLIPSE	= estimate of error (optional)

Update positions in the profile files:

- If the first surface position is found to be bad, swap the next best surface position and time into LATITUDE, LONGITUDE and JULD_LOCATION in the corresponding profile file. If there is no good surface position, consider estimating a position.
- If a position has been estimated, insert it into the corresponding profile file with POSITION_QC = '8'. Two optional variables: POSITION_ERROR_ESTIMATED and POSITION_ERROR_ESTIMATED_COMMENT, can be filled.

5.4. Measured parameters

Delayed-mode operators should take any adjustments made to <PARAM> in the D files (and BD files, for the V3.2 Dtraj) and apply them to the same parameters in the corresponding cycles in the trajectory file. This includes measurements made during all phases of a cycle (ie. upon descent, during park drift, etc).

In the V3.2 Dtraj files, the parameters are listed in the variable TRAJECTORY_PARAMETERS, and their corresponding data modes are listed in the variable TRAJECTORY_PARAMETER_DATA_MODE. When TRAJECTORY_PARAMETER_DATA_MODE = 'D', <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR for that parameter should be filled. If <PARAM> is found to be bad and unadjustable in delayed-mode, <PARAM>_ADJUSTED_QC should be set to '4', <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_ERROR should be set to FillValue. DATA_MODE should be set to 'D' when at least one TRAJECTORY_PARAMETER_DATA_MODE is set to 'D'.

In the V3.1 Dtraj files, the variable TRAJECTORY_PARAMETER_DATA_MODE does not exist. Hence when DATA_MODE = 'D', <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR for all parameters should all be filled. If <PARAM> is found to be bad and unadjustable in delayed-mode, <PARAM>_ADJUSTED_QC should be set to '4', <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_ERROR should be set to FillValue.

5.5. N_MEASUREMENT JULD variables

In the Dtraj files, JULD_ADJUSTED, JULD_ADJUSTED_QC and JULD_ADJUSTED_STATUS should be filled. For the V3.2 Dtraj files, an additional variable, JULD_DATA_MODE, should be filled.

The information for JULD can come from many sources: satellite times, the float clock times, or estimated times. How to fill the JULD_ADJUSTED* variables depends on the source of the JULD (reported in JULD_STATUS), the MEASUREMENT_CODE (MC), and the float type.

Satellite times

Satellite times cannot be corrected. For satellite times, the JULD_ADJUSTED* variables should be filled as follows:

JULD_ADJUSTED = JULD
JULD_ADJUSTED_QC = '1' or '4'
JULD_ADJUSTED_STATUS = '4' (determined by satellite)
JULD_DATA_MODE = 'D' (for V3.2 Dtraj files)

Typical satellite times are Argos fix times. For floats that are equipped to communicate with the GPS, please distinguish between a) GPS times, which are satellite times, and b) the times of GPS fixes according to the float clock, which are not satellite times. These float clock times are described below.

Float clock times

Some floats report clock times for the measurements. Over time, these internal float clocks can drift if they are not reset each cycle during the surface interval. For floats that are equipped to communicate with the GPS, they generally get the GPS time before asking for a GPS fix, but do not necessarily reset their clocks with the GPS time. It depends on the float's configuration (e.g. APEX Apf9 floats generally reset their internal clock when the offset is > 20 seconds, but NOVA floats generally do not reset it). Please refer to Section 1.2.5 in the DAC trajectory cookbook for details.

1. For floats that do not reset their clocks at each cycle

If a float sends back its real clock time, the clock offset can be determined by comparison to the true time (UTC) sent by the satellite. Note that Apex Argos floats do not send back their real clock time, but a method has been developed to estimate a float clock drift, which can be used to estimate a clock offset for each cycle by linear interpolation. (See Annex B of the DAC trajectory cookbook for details.)

These clock offsets should be recorded in the variable CLOCK_OFFSET in the N_CYCLE array:

CLOCK_OFFSET = float clock time - UTC.

If CLOCK_OFFSET(N_CYCLE) is known, JULD_ADJUSTED(N_MEASUREMENT) for cycle N can be obtained as follows:

(a). Select the times of cycle N: JULD(CYCLE_NUMBER == N).
(b). Select the corresponding CLOCK_OFFSET:
CLOCK_OFFSET(CYCLE_NUMBER_INDEX == N).
(c). Adjust the times of cycle N: JULD_ADJUSTED(CYCLE_NUMBER == N) =
JULD(CYCLE_NUMBER == N) – CLOCK_OFFSET(CYCLE_NUMBER_INDEX == N).

Please note that the resolution of CLOCK_OFFSET may not match JULD. When making an adjustment to JULD by using CLOCK_OFFSET, please round to the coarsest resolution (i.e. do not adjust to seconds if the provided time is only in minutes).

To fill the JULD_ADJUSTED* variables if clock offset is known:

JULD_ADJUSTED = JULD – CLOCK_OFFSET (see above)
JULD_ADJUSTED_QC = '1' or '4'
JULD_ADJUSTED_STATUS = '3' (calculated)
JULD_DATA_MODE = 'D' (for V3.2 Dtraj files)

2. For floats that reset their clocks at each cycle

For these floats, the clock offset will most often be zero: CLOCK_OFFSET = 0. However, clock drift can still occur occasionally; for example, when a float is unable to reset its clock, such as when it is sequestered under sea ice. In these cases, clock drift should be estimated (e.g. by interpolation).

Estimated times

Sometimes there are measurements from the float without a time or there are surface intervals for which the float does not receive any times (or positions). These times are not available either because the floats are not configured to record them, or because they are missing for some reasons. In these cases, delayed-mode operators can estimate the unknown times and add them to the Dtraj file.

For surface intervals without times (or positions), linear interpolation or other peer-reviewed methods can be used to estimate the unknown times (or positions). For measurements from the float types that do not have times because the floats are not configured to record them, methods to estimate their times have been developed based on known float behavior and are described in the DAC Trajectory Cookbook.

1. Estimated times for Argos APEX APF8 floats
Methods to fill JULD* for APEX APF8 floats are described in Section 2.2.4 of the DAC Trajectory Cookbook.

2. Estimated times for other Argos APEX floats
Methods to fill JULD* for MC = 800 are described in Section 4 of the DAC Trajectory Cookbook.

3. Estimated times for SOLO floats
Methods to fill JULD* for MC = 600 and MC = 800 are described in the DAC Trajectory Cookbook.

4. Estimated times for surface intervals
Linear interpolation or other peer-reviewed methods can be used to estimate the unknown times (or positions).

Note that if CLOCK_OFFSET is available and is used to adjust other timing information, then it should also be applied to the estimated times if they are based on float clock time.

For estimated times, the various JULD* variables should be filled as follows:
JULD = FillValue
JULD_QC
= FillValue (if the float is not configured to record a time, hence a value is not expected)
= '9' (a value is expected but it is missing for some reasons)
JULD_STATUS
= FillValue (if the float is not configured to record time, hence a value is not expected)
= '9' (a value is expected but it is missing for some reasons)
JULD_ADJUSTED = estimated time
JULD_ADJUSTED_QC = '8' (estimated)
JULD_ADJUSTED_STATUS = '1' (estimated)
JULD_DATA_MODE = 'D' (for V3.2 Dtraj files)

5.6. N_CYCLE JULD variables

When delayed-mode operators have finished making the necessary adjustments to the variables in the N_MEASUREMENT array, the N_CYCLE variables should be filled with the best values from the N_MEASUREMENT array. This means that if a JULD_ADJUSTED value is available, it should be used to fill the corresponding JULD_* variable in the N_CYCLE array. For example, JULD_ASCENT_END in the N_CYCLE array should match JULD_ADJUSTED for MC = 600 in the corresponding cycle.

5.7. Optional strongly encouraged tasks

These tasks add value to the trajectory file, but are based on information that are already in the traj file, and so can be done by the users.

Fill MC 301/RPP

Determining the best pressure during the park-and-drift phase of the float cycle can be complicated depending on what information is reported by the float during park-and-drift and what adjustments need to be made to the pressure measurements. Therefore, during delayed-mode, please consider filling PRES(MC = 301) and/or REPRESENTATIVE_PARK_PRESSURE(N_CYCLE) and REPRESENTATIVE_PARK_PRESSURE_STATUS(N_CYCLE). Instructions on how to do this are described in Section 2.4.3 of the DAC Trajectory Cookbook.

Fill grounded flags

Identifying whether and where the float grounded during a cycle is useful to users who wish to calculate subsurface displacements and who are interested in better understanding the bathymetry of the ocean.

To address the needs of users in calculating subsurface displacements, the variable GROUNDED (reference table 20) includes flags that indicate if the float grounded during a part of the cycle that affected the free drift period or not. If the float did ground during the free drift period, the user will want to exclude that cycle from subsurface displacement calculations. If the float grounded outside of the free drift period, the user may want to include that cycle in subsurface displacement calculations.

Methods to detect grounding include: a) comparisons with resolved bathymetry, and b) comparisons of programmed park/profile pressures with actual pressures.
If method (a) is used, please indicate grounding using flags ‘B’ or ‘C’.
If method (b) is used, please indicate grounding using flags ‘Y’ or ‘P’.

For users interested in ocean bathymetry, it will be important to look at grounding flags of ‘Y’ or ‘P’, since no external bathymetry dataset was used to determine the grounding.

5.8. Record what has been done in delayed-mode

Record the DMQC operator

Please add a comment line to the //global_attributes section of the Dtraj files in the same format as in the D and BD files. See Section 3.7 in this manual.

Record data modes

In the V3.2 traj files, both core and bgc parameters are recorded. Their delayed-mode work can be done by different DMQC operators and at different times. The data mode of each parameter must therefore be indicated in TRAJECTORY_PARAMETER_DATA_MODE(N_MEASUREMENT, N_PARAM).

DATA_MODE(N_CYCLE) should record 'D'.

JULD_DATA_MODE(N_MEASUREMENT) should record 'D'.

DATA_STATE_INDICATOR should record '2C' or '2C+'.

DATE_UPDATE should record the date of the last update of the Dtraj file, in the format YYYYMMDDHHMISS.

The name of the trajectory file should change from wmoid_Rtraj.nc to wmoid_Dtraj.nc.

Record scientific calibration information

In the V3.2 traj files, there is a SCIENTIFIC_CALIB* section. These should be filled out with the proper information, similar to Section 3.6.2 in this manual, and preferably with reference to the particular measurement codes (MC). For example:

SCIENTIFIC_CALIB_COMMENT = 'MC<500 and MC>600: DOXY adjustment based on comparison of surface in-air data with NCEP atmospheric reanalysis.'

Record history

A history record should be appended to the HISTORY section of the Dtraj file to indicate the file has been through the delayed-mode process. See Section 3.7.3 in this manual.

6. Appendix

6.1. Reference Table 2: Argo quality control flag scale

This table describes the Argo QC flag scale. Please note that this table is used for all measured parameters. This table is Reference Table 2 in the Argo User's Manual.

Flag	Meaning	**Real-time comment** (applicable to <PARAM>_QC in 'R' mode and <PARAM>_ADJUSTED_QC in 'A' mode)	**Delayed-mode comment** (applicable to <PARAM>_ADJUSTED_QC in 'D' mode)
'0'	No QC is performed	No QC is performed.	No QC is performed.
'1'	Good data	Good data. All Argo real-time QC tests passed. These measurements are good within the limits of the Argo real-time QC tests.	Good data. No adjustment is needed, or the adjusted value is statistically consistent with good quality reference data. An error estimate is supplied.
'2'	Probably good data	Probably good data. These measurements are to be used with caution.	Probably good data. Delayed-mode evaluation is based on insufficient information. An error estimate is supplied.
'3'	Probably bad data that are potentially adjustable	Probably bad data. These measurements are not to be used without scientific adjustment, e.g. data affected by sensor drift but may be adjusted in delayed-mode.	Probably bad data. An adjustment may (or may not) have been applied, but the value may still be bad. An error estimate is supplied.
'4'	Bad data	Bad data. These measurements are not to be used. A flag '4' indicates that a relevant real-time qc test has failed. A flag '4' may also be assigned for bad measurements that are known to be not adjustable, e.g. due to sensor failure.	Bad data. Not adjustable. Adjusted data are replaced by FillValue.
'5'	Value changed	Value changed	Value changed
'6'	Not used	Not used	Not used
'7'	Not used	Not used	Not used
'8'	Estimated value	Estimated value (interpolated, extrapolated, or other estimation)	Estimated value (interpolated, extrapolated, or other estimation)
'9'	Missing value	Missing value. Data parameter will record FillValue.	Missing value. Data parameter will record FillValue.
' '	FillValue	Empty space in netcdf file.	Empty space in netcdf file.

Notes on the Argo QC flags:

(a). The QC flags are stored as a character variable, not a numeric variable, in the Argo netcdf files. Typically, a character variable will use ' ' as the FillValue, and a numeric variable will use 99999 as the FillValue. These are defined in the netcdf files.

(b). Please distinguish between QC = '9' and QC = ' '. QC = '9' denotes missing values, or missing data, from a measurement, which can occur as a result of instrument problem or telemetry problem. QC = ' ' denotes empty spaces to pad the rectangular shapes of the variables in the netcdf files.

(c). QC = '5' and '8' are only used in specific cases. QC = '5' is only used in CHLA_ADJUSTED_QC to denote adjusted data that have value changed due to the Non-Photochemical Quenching (NPQ) correction. QC = '5' is not used in <PARAM>_QC because the raw values should never be altered. QC = '8' is used in POSITION_QC and JULD_QC, to denote estimated locations and times, for profiles whose positions and Julian days are missing. In vertical profile data, <PARAM>_QC = '8' and <PARAM>_ADJUSTED_QC = '8' are only used in the synthetic S- files; they are not used in any R-, D-, BR-, BD- files.

(d). During real-time processing, when multiple QC tests are applied to the data, the QC flag assigned by a QC test cannot override a higher order flag. For example, a QC flag '4' assigned by a QC test cannot be decreased to '3' by another QC test. The QC flag order is: '4' > '3' > '5' or '8' > '2' > '1'.

(e). For error estimates of the adjusted data, please refer to the parameter's <PARAM>_ADJUSTED_ERROR in the profile files. For error estimates of the unadjusted raw data, please refer to PARAMETER_ACCURACY in the meta files, or the relevant BGC-Argo DAC processing cookbooks.

6.2. Reference Table 2a: overall profile quality flags

This table is used for all measured parameters. This table is Reference Table 2a in the Argo User's Manual.

N is defined as the percentage of levels with good data where:

- QC flag values of '1', '2', '5', '8' (only used in the S- files) are GOOD data
- QC flag values of '0' (no QC), '9' (missing), ' ' (FillValue) are NOT USED in the computation
- QC flag values of '3', '4' are BAD data

The computation should be taken from <PARAM>_ADJUSTED_QC if available and from <PARAM>_QC otherwise.

n	Meaning
' '	No QC is performrmed or no usable flag values are present
A	***N*** = 100%; All profile levels contain good data
B	75% <= ***N*** < 100%
C	50% <= ***N*** < 75%
D	25% <= ***N*** < 50%
E	0% < ***N*** < 25%
F	***N*** = 0%; No profile levels have good data

6.3. Reference Table 11: RTQC tests binary ID

This table lists the binary ID of the RTQC tests. This table is Reference Table 11 in the Argo User's Manual.

Each RTQC test has a unique Test Number (n). The binary ID (2^n) of the unique test number (n) is used to record rtqc tests performed and failed in the history variable HISTORY_QCTEST.

Test number (n)	Binary ID (2^n)	Test name
1	2	Platform identification test
2	4	Impossible date test
3	8	Impossible location test
4	16	Position on land test
5	32	Impossible speed test
6	64	Global range test
7	128	Regional global parameter test
8	256	Pressure increasing test
9	512	Spike test
~~*10*~~	~~*1024*~~	~~*Top and Bottom Spike test (obsolete)*~~
~~*11*~~	~~*2048*~~	~~*Gradient test (obsolete)*~~
12	4096	Digit rollover test
13	8192	Stuck value test
14	16384	Density inversion test
15	32768	Supplemental sensor exclusion list test
16	65536	Gross salinity or temperature sensor drift test
17	131072	Visual QC test
18	261144	Frozen profile test
19	524288	Deepest pressure test
20	1048576	Questionable Argos position test
21	2097152	Near-surface unpumped CTD salinity test
22	4194304	Near-surface mixed air/water test
23	8388608	RTQC flag scheme for Deep SBE CTD data > 2000 dbar
24	16777216	RTQC flag scheme for RBR*argo*^{3}|2K CTD data (< 2000 dbar)
25	33554432	MEDD test
26	67108864	TEMP_CNDC test

6.4. Common instrument errors and failure modes

This section describes some common instrument errors and failure modes that will cause error in float measurements.

1. TBTO leakage

TBTO (tributyltinoxide) is a wide spectrum poison that is used to protect conductivity cells from biofouling. However, accidental leakage of TBTO onto the conductivity cells can occur, but it usually gets washed off within the first few profiles. This will result in fresh salinity offsets in float data that gradually return to normal. Delayed-mode analysts should pay special attention to salinity data at the beginning of the float's life because TBTO leakage usually affects the first few CTD profiles after deployment before the contamination gets washed off.

2. Pollution/contamination events

Any pollution or contamination on the conductivity cell will result in erroneously fresh salinity measurements. When pollution washes off, reversal of sensor drift trend can occur. Delayed-mode analysts need to be careful in splitting the float time series in such cases.

3. Ablation events

Any ablation of the conductivity cell, such as etching, scouring, or dissolution of the glass surface, will result in erroneously salty salinity measurements.

4. Conductivity cell geometry changes

The geometry of conductivity cells can change, thus causing electrodes to change distance. This will result in either an increase or a decrease in salinity values.

5. Conductivity cell circuit changes

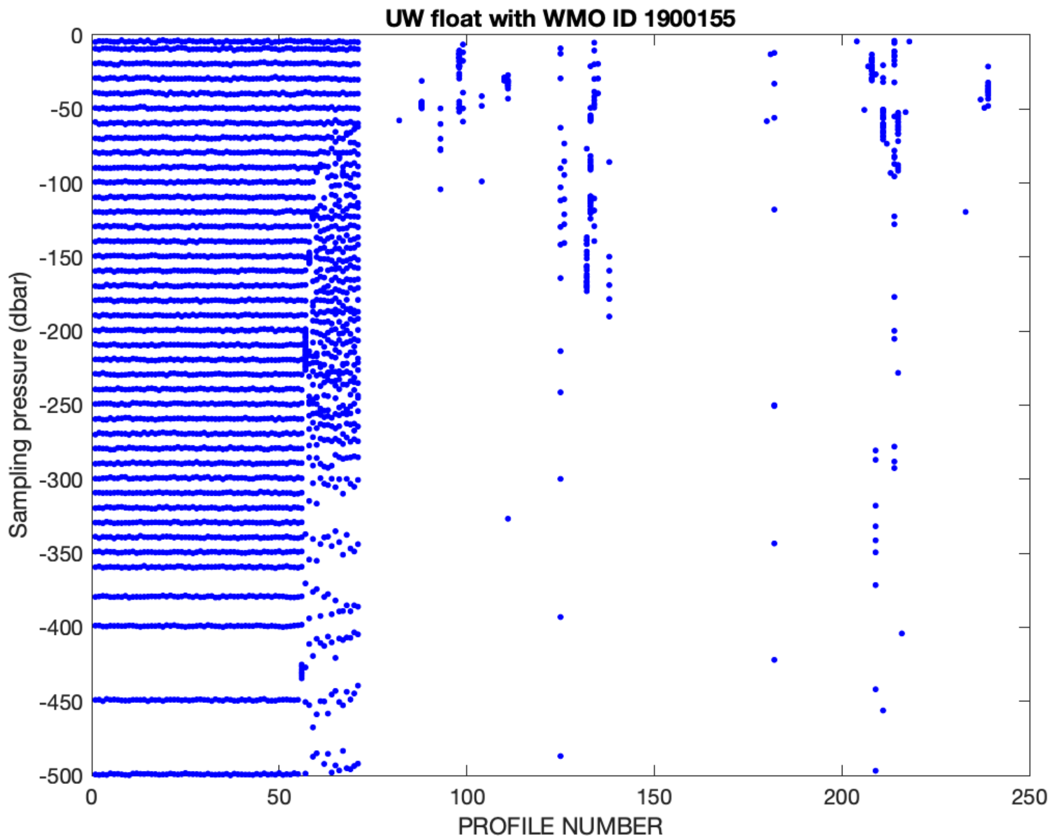
The circuit within the conductivity cell contains capacitors and resistors. Changes to any of these electrical components will affect electrical conductivity and thus will give erroneous (fresh or salty) salinity measurements. Electrical complication can result in sensor drifts that have varying drift rates (e.g. drift rates can change from slow and linear to exponential). Usually jumps in salinity measurements are an indication of electrical malfunction. If electrical complication is suspected, delayed-mode analysts should check the shape of the vertical salinity profiles for adjustability. Usually the vertical profiles after a measurement jump are wrong and so are unadjustable.

6. Low voltage at end of float life, and "Energy Flu"

Some floats experience a sudden rapid decrease in available battery energy reserves. This premature exhaustion of battery, known as "Energy Flu", usually starts about 2 years after deployment. The sharp drop in battery voltage related to "Energy Flu", as well as the low voltage towards the end of a float's natural life, will produce low-of-correct salinity values. Towards the end of float life, low voltage will result in large drift, followed by death. "Energy Flu" will cause spikes that get worse and more frequent, also followed by death.

7. Druck pressure sensor “snowflakes” problem

About 4% of SBE41 CTDs that were manufactured in late 2002 through end of 2003 have experienced the Druck pressure sensor “snowflakes” problem. SeaBird has fixed this problem in 2004, so this feature is only included in this section for identifying the historical profiles that have been affected. The Druck pressure sensor “snowflakes” problem is due to internal electrical shorting by the growth of titanium oxide particles (‘snowflakes’) in the oil-filled cavity in the pressure sensor, causing the pressure sensor to report erratic pressure measurements, or going to full scale, i.e. either report PRES ~ 3000 dbar or −3000 dbar. These erratic pressure measurements will preferentially report deeper than correct. The firmware tries to adjust the piston according to the erroneous deeper pressures, causing the float to park shallower. The float will thus progressively become a surface drifter. Erroneous deeper pressures will also result in the firmware placing the pointer at the deeper nominal sampling levels in the lookup table, thus causing the float to take a sample everytime the firmware performs a lookup (every 6 seconds). The result is a series of measurements from very close-together depth levels. Progressively shallower profiles and close-together measurements are therefore two ways to identify whether the Druck pressure sensor has been contaminated. When the Druck pressure sensor has been contaminated, pressure measurements become suspect and should be considered bad. The corresponding temperature and salinity measurements are therefore also suspect and should be considered bad.



This plot shows the sampling pressures of a float affected by the Druck "snowflakes" problem. Starting from profile #54, the sampling pressures become increasingly close together and erratic. The float CTD measurements also become erratic, bad, or missing.

8. Druck pressure sensor “oil microleak” problem

Another pathology in Druck pressure sensors is oil microleak past the glass/metal seal. This oil leak leads to an internal volume loss, which then exhibits itself as an increasing negative offset at all pressures. At the early stages of oil microleak, float measurements are still correctable and usable. However, as more and more oil is leaked, the flexible titanium diaphragm will dip so far down the oil chamber that it will short the electrical parts, causing erratic behaviour in float measurements. This is the end stage of oil microleak, and the data at this point are bad and unadjustable.

9. Incorrect pressure sensor coefficient

Incorrect scaling coefficient in the pressure sensor will give anomalous T-S curves at depth. The T-S relation will look acceptable, but at depth it will look as if the float is sampling an anomalous water mass relative to nearby floats. Delayed-mode analysts should try to re-scale pressure measurements (e.g. PRES’ = PRES * X) to see whether the T-S curve can be recovered. Air bubbles in the pressure transducer can also cause erroneous pressure measurements that are visible as anomalous T-S curves.

10. Abnormal salinity “hooks” at base of profiles

In some floats, salty or fresh salinity hooks may be observed at the base of the CTD profiles. The “hook” appearance occurs when the two deepest measurements are taken at nearly identical pressures (usually less than 5dbar apart) but salinity from the first measurment looks abnormal relative to the second measurement. The first (deeper) measurement is taken at the end point of descent; the second (shallower) measurement is the first deep sample taken during ascent. The abnormal (deeper) salinity reading is caused by water in the conductivity cell carried from the surface or park level to deep profile level not being flushed out completely before the descent-end-point sample is taken. Salty hooks are produced when surface or park-level water is saltier than deep water; fresh hooks are produced when surface or park-level water is fresher than deep water. These salinity hooks cannot be detected by the real-time tests, so delayed-mode analysts are urged to carefully examine the base of CTD profiles for these abnormal salinity values, and to flag them appropriately in delayed-mode.

6.5. Criteria for CTD profiles to be retained in the reference database

The following criteria are used to select CTD data to be included in the reference database CTD_for_DMQC, which is used in Argo delayed-mode qc for salinity.

1). For data from the Ocean Climate Library (WOD), use only data that have passed all NODC quality control tests for observed level data. For data from other sources, they are put through a Coriolis quality control.

2). Use all country codes.

3). Use only profiles that sampled deeper than 900 dbar, except for the Mediterranean Sea, where all profiles are kept even if they are shallower than 900 dbar.

4). Weed out all data points outside these ranges: $24 < S < 41$, $0.01 < P < 9999$, $0°C < T < 40°C$, except for WMO boxes with latitudes north of 60°N or south of 50°S, where $-2.5°C < T < 40°C$.

5). For WMO boxes that contain more than 10,000 profiles, only select profiles that are post-1995.

6). Eliminate nearby duplicates.

7). Do objective residual analysis using previously qc'd reference data to identify anomalies. Then do visual inspection of anomalies. An additional visual quality control step is done on sets of profiles.

6.6. Criteria for Argo profiles to be retained in the reference database

The following criteria are used to select Argo data to be included in the reference database Argo_for_DMQC, which is used in Argo delayed-mode qc for salinity.

1). No real-time data. Profiles must be from D-files.

2). Exclude all profiles that sample on descent.

3). Exclude floats that fail within 1 year of deployment.

4). No profiles within 12 months of end of record (death).

5). No profiles that have salinity drift adjustment ($\Delta S > .001$ in bottom data to distinguish from thermal lag adjustment at shallower levels).

6). Profiles must contain data deeper than 800 dbar.

7). Profiles must contain 90% good salinity values (PRES_ADJUSTED_QC = ‘1’, TEMP_ADJUSTED_QC = ‘1’, PSAL_ADJUSTED_QC = ‘1’).

8). No profiles within 2 months of deployment (due to the propensity of some floats to acquire TBTO contamination).

9). No profiles within 12 months of the onset of salinity drift ($\Delta S > .001$ in bottom data).

10). No profiles whose bottom data have PSAL_ADJUSTED_ERROR > 0.015.

6.7. Consistency checks for D files at the GDACs

The following is a list that is used at the GDACs for checking D files.

1). <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_ERROR must contain data, except when <PARAM>_ADJUSTED_QC = '4' or '9'. Here, <PARAM> = PRES, TEMP, PSAL, and CNDC.

2). If PRES_ADJUSTED_QC = '4', then TEMP_ADJUSTED_QC = '4' and PSAL_ADJUSTED_QC = '4'.

3). <PARAM>_ADJUSTED_QC cannot be '0'.

4). POSITION_QC and JULD_QC cannot be '0'.

5). No variable should be filled with the netCDF value of IEEE NaN.

6). In the Scientific Calibration section, PARAMETER should have N_PARAM entries equal to the number of measurement parameters recorded in the netCDF file.

7). In the Scientific Calibration section, SCIENTIFIC_CALIB_COMMENT should have non-FillValue entries in every N_PARAM dimension.

8). In the Scientific Calbration section, CALIBRATION_DATE should have non-FillValue entries in every N_PARAM dimension, and should have format YYYYMMDDHHMISS (seconds must be 0 to 59).

9). DATE_UPDATE should be equal to or later than any CALIBRATION_DATE, HISTORY_DATE, DATE_CREATION, JULD, and JULD_LOCATION.

10). There should be at least one HISTORY record.

11). All dates must be after 1st Jan 1997, and before GDAC file time.

12). All dates must be 14-digit strings, in the format YYYYMMDDHHMISS (seconds must be 0 to 59).

13). Character strings should not contain the NULL character.

6.8. How to estimate float ascent rates

For various sensor response corrections (e.g. the conductivity cell thermal inertia correction), the float ascent rate, or the elapsed time at each measurement since the start of the vertical profile (*elptime*, in seconds), needs to be estimated for different models of floats. When timing information is available, *elptime* can be obtained directly, or via interpolation. When timing information is not available, *elptime* will need to be estimated by other means. Here are some methods to estimate *elptime* per float type.

<u>APEX and NAVIS floats</u>

Most APEX and NAVIS floats do not return timing information with their vertical profiles, but those that return bin-averaged data also report the number of samples in each pressure bin. These are stored as an intermediate parameter NB_SAMPLE_CTD in the core R and D files. If the CTD samples at f Hz, then *elptime* (in seconds), at each pressure bin k can be estimated as:

$$elptime_k = \sum_{i=1}^{k} \frac{NB\_SAMPLE\_CTD_i}{f}$$

The sampling frequency of a CTD is hardcoded in the CTD firmware and is not documented anywhere. All SBE41cp CTDs sample at 1 Hz when in CP mode (f = 1 Hz). Sbe41cp with STS samples at either 1 sample per 1.1sec or 1 sample per 1.2sec (depending on hardware). Sbe41n on NAVIS floats sample at 1 Hz.

<u>SOLO floats</u>

a) SIO SOLO with Argos: The SIO SOLO pumps all oil into the external bladder at the base of the profile. Thus, the rise rate is high at the base of the profile and decreases through the thermocline and to the surface. There are no mid-depth buoyancy adjustments keeping the rise rate within a range. Timing data are not returned by the float and must be estimated by using the estimated float buoyancy and the total time that the CTD is sampling during the profile phase. Part of DMQC of the float is to estimate the primary and secondary measurement code timing. When an estimate is possible, it can be found in the Dtraj file.

b) SIO SOLO, SOLOII with Iridium: The SIO SOLOII and MRV S2A models are similar. During profile, these floats attempt to maintain a rise rate range, adding buoyancy at the start of profile and mid-depth when the rise rate falls below a threshold value. The floats sample pressure-time pairs occasionally during the profile to estimate whether a buoyancy increase is needed. Thus, the rise rate for these floats can be a saw-tooth pattern. The pressure-time data can be found in the Rtraj and Dtraj files (identified with a MEASUREMENT_CODE of 589 or 590). Timing for individual profile levels can be interpolated from these pressure-time pairs, and difference from the start of profile (MEASUREMENT_CODE=500).

<u>PROVOR floats</u>

Some PROVOR floats send back sparse timing information at only a few levels of the vertical profile, and these are stored in the Rtraj and Dtraj files. Other PROVOR floats send back full timing information for all profile observations, and these are stored as an intermediate parameter MTIME. For details, please refer to "*Processing BGC-Argo measurement timing information at the DAC level*", http://dx.doi.org/10.13155/47998.

6.9. New compressibility coefficients for RBR*argo*³|2K CTDs calibrated pre-April 2021

The following table represents the new compressibility coefficients for some of the RBR*argo*³|2K CTDs calibrated pre-April 2021. Work is still ongoing to fill this table. Please refer to this version in github for the most up-to-date information:

https://github.com/ArgoDMQC/RBRargo_DMQC/blob/main/Compressibility_correction/RBRargo3_compressibility_table.csv

WMO	Deployment Date	Float model	program	RBR SN	X2old	X3old	X4old	X2	X3	X4	Source
5904925	2015-07-24T08:23:00	APEX	Argo AUSTRALIA	60263	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	different CTD design
4902387	2016-05-08T17:22:00	NOVA	Argo CANADA		4.00E-07	0	0	N/A	N/A	N/A	
6901191	2016-06-13T02:44:00	APEX	Argo UK	40341	4.00E-07	0	0	N/A	N/A	N/A	different CTD design
2901891	2016-06-28T11:45:00	APEX	Argo UK	40343	4.00E-07	0	0	N/A	N/A	N/A	different CTD design
6901193	2016-07-11T00:35:00	APEX	Argo UK	40342	4.00E-07	0	0	N/A	N/A	N/A	different CTD design
2901901	2017-02-10T12:00:00	APEX	Argo UK	40344	4.00E-07	0	0	N/A	N/A	N/A	different CTD design
6901203	2017-10-02T14:25:00	APEX	Argo UK	40328	4.00E-07	0	0	N/A	N/A	N/A	different CTD design
2902730	2018-01-11T00:00:00	APEX	Argo CHINA	40528	4.00E-07	0	0				
2903005	2018-02-03T23:40:00	APEX	Argo JAMSTEC	40610	4.00E-07	0	0	1.8335E-06	-5.8275E-10	9.3534E-14	deployment cast
2903327	2018-02-03T23:41:00	APEX	Argo JAMSTEC	40611	4.00E-07	0	0	2.2308E-06	-7.0901E-10	1.1380E-13	deployment cast
5905468	2020-07-27T01:36:00	ALTO	Argo AUSTRALIA	203986	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	
5906298	2020-08-15T09:09:00	APEX	Argo UW	201597	1.8732E-06	-7.7689E-10	1.4890E-13				
5905482	2020-09-21T19:40:00	ALTO	Argo AUSTRALIA	203995	1.8732E-06	-7.7689E-10	1.4890E-13	1.2660E-06	-4.0237E-10	6.4583E-14	WMO5906251
5905483	2020-09-23T10:00:00	ALTO	Argo AUSTRALIA	203997	1.8732E-06	-7.7689E-10	1.4890E-13	7.8579E-07	-2.4975E-10	4.0086E-14	WMO5906267
5906614	2020-09-25T16:03:00	ALTO	Argo AUSTRALIA	203984	1.8732E-06	-7.7689E-10	1.4890E-13	1.4581E-06	-4.6343E-10	7.4382E-14	WMO5906256
5906615	2020-09-27T03:29:00	ALTO	Argo AUSTRALIA	203982	1.8732E-06	-7.7689E-10	1.4890E-13	1.2049E-06	-3.8295E-10	6.1465E-14	WMO5906149
5906616	2020-09-27T09:51:00	ALTO	Argo AUSTRALIA	203981	1.8732E-06	-7.7689E-10	1.4890E-13	1.3009E-06	-4.1348E-10	6.6365E-14	WMO5906263
5906617	2020-09-29T23:00:00	ALTO	Argo AUSTRALIA	203994	1.8732E-06	-7.7689E-10	1.4890E-13	1.4843E-06	-4.7175E-10	7.5718E-14	WMO5906272
5906618	2020-09-30T05:28:00	ALTO	Argo AUSTRALIA	203993	1.8732E-06	-7.7689E-10	1.4890E-13	1.3620E-06	-4.3290E-10	6.9482E-14	WMO5906153
5906619	2020-10-01T10:37:00	ALTO	Argo AUSTRALIA	203998	1.8732E-06	-7.7689E-10	1.4890E-13	1.4581E-06	-4.6343E-10	7.4382E-14	WMO5906269
4903275	2020-10-05T11:00:00	ALTO	Argo WHOI	203999	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	shallow float
5906383	2020-11-10T13:00:00	SOLO_II	Argo SIO	41059	1.8732E-06	-7.7689E-10	1.4890E-13	1.4755E-06	-4.6898E-10	7.5273E-14	WMO5906404
5906299	2020-11-21T21:54:00	APEX	Argo UW	201598	1.8732E-06	-7.7689E-10	1.4890E-13				
6903076	2020-12-11T08:35:00	ARVOR	Coriolis	203794	1.8732E-06	-7.7689E-10	1.4890E-13	1.5541E-06	-4.9395E-10	7.9281E-14	WMO6903010
6903075	2020-12-11T08:59:00	ARVOR	Coriolis	203793	1.8732E-06	-7.7689E-10	1.4890E-13	1.8597E-06	-5.9107E-10	9.4870E-14	WMO6903010
6903753	2020-12-18T18:51:00	APEX	Argo UK	203420	1.8732E-06	-7.7689E-10	1.4890E-13				deployment cast
6903752	2020-12-21T16:05:00	APEX	Argo UK	203419	1.8732E-06	-7.7689E-10	1.4890E-13				deployment cast
6903754	2020-12-23T13:23:00	APEX	Argo UK	203596	1.8732E-06	-7.7689E-10	1.4890E-13				deployment cast
6903727	2020-12-27T20:00:00	APEX	Argo UK	203597	1.8732E-06	-7.7689E-10	1.4890E-13				deployment cast
6904104	2021-01-12T12:51:00	ARVOR	Argo GERMANY 2025	205360	1.8732E-06	-7.7689E-10	1.4890E-13	1.5803E-06	5.0228E-10	8.0617E-14	deployment cast
6904105	2021-01-12T12:51:00	ARVOR	Argo GERMANY 2025	205361	1.8732E-06	-7.7689E-10	1.4890E-13	1.4144E-06	-4.4955E-10	7.2155E-14	deployment cast
6904103	2021-01-12T12:51:00	ARVOR	Argo GERMANY 2025	205362	1.8732E-06	-7.7689E-10	1.4890E-13	1.4493E-06	-4.6065E-10	7.3936E-14	deployment cast
6904101	2021-01-12T12:51:00	ARVOR	Argo GERMANY 2025	205365	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	
6904102	2021-01-12T12:51:00	ARVOR	Argo GERMANY 2025	205363	1.8732E-06	-7.7689E-10	1.4890E-13	1.4668E-06	-4.6620E-10	7.4827E-14	deployment cast
4902533	2021-04-26T02:32:00	ARVOR	Argo CANADA		1.8732E-06	-7.7689E-10	1.4890E-13	1.5716E-06	-4.9950E-10	8.0172E-14	WMO5906437 + deployment cast
4902534	2021-04-26T02:37:00	ARVOR	Argo CANADA		1.8732E-06	-7.7689E-10	1.4890E-13	1.6676E-06	-5.3003E-10	8.5071E-14	WMO5906437 + deployment cast
6903709	2021-05-19T06:00:00	ARVOR	Euro-Argo RISE	205366	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	shallow float
6903710	2021-05-21T17:00:00	ARVOR	Euro-Argo RISE	205364	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	shallow float
4903364	2021-08-09T19:23:00	ALTO	Argo WHOI	203996	1.8732E-06	-7.7689E-10	1.4890E-13	1.1350E-06	-3.6075E-10	5.7902E-14	WMO6901191
5906385	2021-09-04T00:00:00	APEX	Argo JAMSTEC	40767	4.00E-07	0	0	1.6109E-06	-5.1199E-10	8.2176E-14	WMO2902532
5906386	2021-09-06T00:00:00	APEX	Argo JAMSTEC	40662	4.00E-07	0	0	1.3358E-06	-4.2457E-10	6.8146E-14	WMO2902725
5906387	2021-09-06T00:00:00	APEX	Argo JAMSTEC	40661	4.00E-07	0	0	1.3839E-06	-4.3984E-10	7.0596E-14	WMO5905869
5906739	2021-09-26T08:00:00	SOLO_II	Argo SIO	206473	1.8732E-06	-7.7689E-10	1.4890E-13	N/A	N/A	N/A	N/A
6903077	2021-11-15T07:15:00	ARVOR	Coriolis	203914	1.8732E-06	-7.7689E-10	1.4890E-13	1.5803E-06	-5.0228E-10	8.0617E-14	WMO6902840
6903078	2021-11-24T02:00:00	ARVOR	Coriolis	204164	1.8732E-06	-7.7689E-10	1.4890E-13	1.4406E-06	-4.5788E-10	7.3491E-14	WMO6904128

Note: Manufacturer pre-deployment calibration date and other meta data for RBR sensors can be obtained via https://oem-lookup.rbr-global.com.

6.10. Argo QC tools on Github

Many Argo QC softwares are publicly available on Github. Here is a list of Argo QC and QC-related repositories on Github.

github.com/ArgoRTQC

matlab_MEDD
Matlab codes for implementing the MEDD test. The MEDD test is a spike detection test that aims to detect simple spikes and vertically short burst of spiky points. MEDD stands for MEDian with a Distance. Please send questions to Delphine Dobler (delphine.dobler@euro-argo.eu).

github.com/ArgoDMQC

matlab_owc
Matlab codes for implementing the OWC float salinity calibration system. Please send questions to Cecile Cabanes (cecile.cabanes@ifremer.fr) and Annie Wong (apsw.uw@gmail.com).

PMEL_gui
A Matlab GUI developed by PMEL for checking OWC output. Please send questions to John Lyman (john.lyman@noaa.gov).

celltm
Matlab code for correcting the conductivity cell thermal mass error in SBE CTDs. Please send questions to Gregory C. Johnson (gregory.c.johnson@noaa.gov) or John Lyman (john.lyman@noaa.gov).

Deep_Argo_DMTools
Codes to post-process salinity data from Deep CTDs. **DM_CPcor** contains Matlab code for estimating optimal CPcor values for SBE CTDs used in Deep-Argo; please send questions to Cecile Cabanes (cecile.cabanes@ifremer.fr). **Deep_CTD_selection** contains Matlab code that can be swapped into OWC to select only reference CTD deeper than a set of pressure levels; please send questions to John Lyman (john.lyman@noaa.gov).

RBRargo_DMQC
Codes to post-process salinity data from RBR CTDs. Please send questions to Mathieu Dever (mathieu.dever@rbr-global.com).

github.com/euroargodev

dm_floats
Matlab tools for preparing OWC input files and write netCDF Argo D-files. Please send questions to Cecile Cabanes (cecile.cabanes@ifremer.fr).