EGO Quality Control manual

Quality Control tests on glider time series and associated profile data

Version 1.4

March 18th, 2022

http://doi.org/10.13155/51485

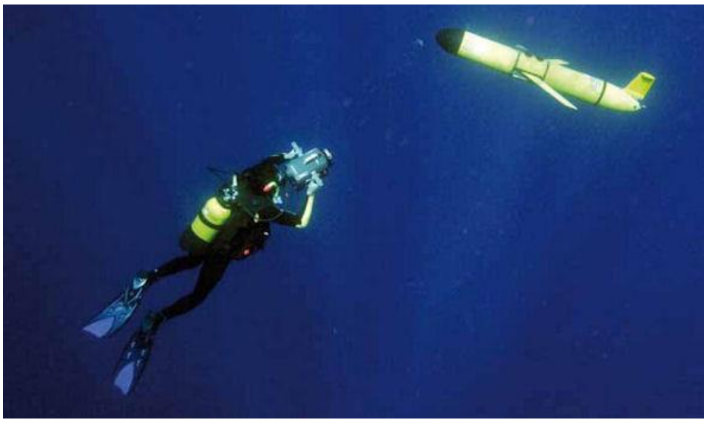


Table of contents

History

Version	Date	Comment
1.0	06/09/2017	JPR: creation of the document
1.1	27/08/2020	JPR: updated for decoder V008a, mainly: - Use of GEBCO bathymetric atlas for RTQC test #4 - Add RTQC test #25 (MEDD test)
1.2	28/04/2021	JPR: updated for decoder V009a: - Add RTQC test #57 (DOXY specific test)
1.3	17/08/2021	JPR: updated for decoder V010c (RTQC Version 1.6): - New test application order (tests 15 and 19 have moved). - A measurement with QC = '3' is tested by other quality control tests. - To follow Virginie Racapé's recomendation, TEST 25: MEDD test is not performed on (PRES, TEMP_DOXY) timeseries
1.4	18/03/2022	JPR: DOWN_IRRADIANCE532 and DOWN_IRRADIANCE555 parameters added in RTQC tests.

1 Introduction

This document is the EGO quality control manual for CTD (Conductivity, Temperature, Depth) and BGC (BioGeoChemical) data. It describes two levels of quality control:

- The first level is the real-time system that performs a set of agreed automatic checks.
  - Adjustment in real-time can also be performed and the real-time system can evaluate quality flags for adjusted fields.
- The second level is the delayed-mode quality control system.

Please note that at the present time, official quality control procedures don't exist for all BGC parameters. This document will be updated as soon as other procedures are available.

2 Real-time quality controls

The quality control procedures on real-time data are limited and automatic.

At the present time, real-time tests are defined for the following parameters:

- TIME,
- LATITUDE,
- LONGITUDE,
- PRES,
- TEMP,
- PSAL,
- CNDC,
- DOXY,
- TEMP_DOXY,
- CHLA,
- BBP700,
- BBP470,
- BBP532,
- PH_IN_SITU_TOTAL,
- NITRATE,
- DOWN_IRRADIANCE380,
- DOWN_IRRADIANCE412,
- DOWN_IRRADIANCE443,
- DOWN_IRRADIANCE490,
- DOWN_IRRADIANCE532,
- DOWN_IRRADIANCE555,
- DOWNWELLING_PAR.

QC of other parameters should be set to 0 (no QC was performed).

Each defined test has a unique number used to report the passed and failed tests list in the HISTORY_QCTEST variable of the EGO format. EGO and Argo projects share the same test numbers.

The list of the already defined tests is the following.

Test number	Test name	Applied on
2	Impossible Date Test	Time series
3	Impossible Location Test	Time series
4	Position on Land Test	Time series
6	Global Range Test	Time series
7	Regional Range Test	Time series
8	Pressure Increasing Test	Generated profiles
9	Spike Test	Time series
11	Gradient Test	Time series
12	Digit Rollover Test	Generated profiles
13	Stuck Value Test	Generated profiles
14	Density Inversion	Generated profiles
15	Grey List	Time series
19	Deepest pressure test	Time series
20	Questionable Argos position test	Time series
25	MEDian with a Distance (MEDD) test	Time series
57	DOXY specific test	Time series

2.1 Real-time quality control tests on glider time series

2.1.1 Common real-time quality control tests on glider time series

This section lists the real-time tests that are common to CTD data and BGC data.

2. Impossible date test

This test requires that the Julian days of glider data 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 fails this test, associated JULD_QC and TIME_QC should be flagged as bad data (=4).

3. Impossible location test

This test requires that the observation latitude and longitude of glider data be sensible. Latitude in range −90 to 90 Longitude in range −180 to 180

Action: If either LATITUDE or LONGITUDE fails this test, the associated POSITION_QC should be flagged as bad data (=4).

4. Position on land test

This test requires that the observation latitude and longitude of glider data 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 use of GEBCO bathymetry file that is freely available (https://www.bodc.ac.uk/data/open_download/gebco/gebco_2020/zip/).

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

6. Global range test

This test applies a gross filter on observed values for PRES, TEMP, PSAL, DOXY, TEMP_DOXY, CHLA, BBP, PH_IN_SITU_TOTAL, NITRATE, DOWN_IRRADIANCE, and DOWNWELLING_PAR. The ranges need to accommodate all of the expected extremes encountered in the oceans.

- PRES cannot be less than –5 dbar. PRES in the range [–5 to –2.4] dbar should be considered 'probably bad'.
- TEMP in range [−2.5, 40.0] °C
- PSAL in range [2, 41.0] PSU
- DOXY in range [−5, 600] micromoles/kg
- TEMP_DOXY in range [−2.5, 40.0] °C
- CHLA in range [−0.1, 50] mg/m^3
- BBP700 in range [−0.000025, 0.1] m^{-1}
- BBP470 in range [−0.000005, 0.1] m^{-1}
- BBP532 in range [−0.000005, 0.1] m^{-1}
- PH_IN_SITU_TOTAL in range [7.3, 8.5]
- NITRATE in range [-2, 50] micromole/kg
- DOWN_IRRADIANCE380 in range [-1, 1.7] W.m-2 .nm-1
- DOWN_IRRADIANCE412 in range [-1, 2.9] W.m-2 .nm-1
- DOWN_IRRADIANCE443 in range [-1, 3.2] W.m-2 .nm-1
- DOWN_IRRADIANCE490 in range [-1, 3.4] W.m-2 .nm-1
- DOWN_IRRADIANCE532 in range [-1, 3.3] W.m-2 .nm-1
- DOWN_IRRADIANCE555 in range [-1, 3.2] W.m-2 .nm-1
- DOWNWELLING_PAR in range [-1, 4672] micromoleQuanta.m-2 .s-1

Action:

For PRES:

If PRES < –5 dbar, then PRES_QC = 4, TEMP_QC = 4, PSAL_QC = 4.

If –5dbar <= PRES <= –2.4 dbar, then PRES_QC = 3, TEMP_QC = 3, PSAL_QC = 3.

For other parameters:

Values that fail this test should be flagged with a QC = 4 for TEMP, PSAL, DOXY, TEMP_DOXY, CHLA, PH_IN_SITU_TOTAL, DOWN_IRRADIANCE and DOWNWELLING_PAR and with a QC = 3 for BBP.

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

- TEMP in range [21, 40.0] °C
- PSAL in range [2, 41.0] PSU
- TEMP_DOXY in range [21, 40.0] °C

Mediterranean Sea

- TEMP in range [10.0, 40.0] °C
- PSAL in range [2, 40.0] PSU
- TEMP_DOXY in range [10.0, 40.0] °C

Action: If a value fails this test, it should be flagged as bad data (QC = 4).

9. Spike test

Spike test for TEMP, PSAL, DOXY and TEMP_DOXY

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 depth, but assumes a sampling that adequately reproduces changes in TEMP, PSAL, DOXY and TEMP_DOXY with depth.

Test value = | V2 − (V3 + V1)/2 | − | (V3 − V1) / 2 |

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

For TEMP and TEMP_DOXY: The V2 value is flagged when

- the test value exceeds 6 °C for pressures less than 500 dbar, or
- the test value exceeds 2 °C for pressures greater than or equal to 500 dbar.

For PSAL: 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.

For DOXY: The V2 value is flagged when

- the test value exceeds 50 micromol/kg for pressures less than 500 dbar, or
- the test value exceeds 25 micromol/kg for pressures greater than or equal to 500 dbar.

Action: Values considered as a spike should be flagged as bad data (QC = 4).

Spike test for CHLA

Difference between sequential measurements, where one measurement is quite different than adjacent ones, is considered as a spike. With respect to biogeochemistry, most of the time, spikes contain information, mainly in case of positive spikes. This is the reason why we set up a test to discriminate negative spikes.

We calculate the difference between the chlorophyll-A at a certain depth (V2) and a running median (5 values, V0, V1, V2, V3, V4) along the whole profile:

- RES = V2 - median(V0,V1,V2,V3,V4)

Then we calculate the percentile10 of this difference for the entire profile. (To get the percentile10, sort RES in ascending order, then find the value of RES at the index i where i = 10% * number of samples in the profile). If the difference between the chlorophyll-A and the running median is smaller than 2*percentile10:

- RES < 2*percentile10(RES)

then it is considered as a spike.

Action: Values considered as a spike should be flagged as bad data (QC = 4).

Spike test for BBP (<u>not used yet in the Coriolis EGO processing chain</u>)

Difference between sequential measurements, where one measurement is quite different than adjacent ones, is considered as a spike. With respect to biogeochemistry, most of the time, spikes contain information, mainly in case of positive spikes. This is the reason why we set up a test to discriminate negative spikes.

We calculate the difference between the BBP at a certain depth (V2) and a running median (5 values, V0, V1, V2, V3, V4) along the whole profile:

- RES = V2 - median(V0,V1,V2,V3,V4)

Then we calculate the percentile10 of this difference for the entire profile. (To get the percentile10, sort RES in ascending order, then find the value of RES at the index i where i = 10% * number of samples in the profile). If the difference between the BBP and the running median is smaller than 2*percentile10:

- RES < 2*percentile10(RES)

then it is considered as a spike.

Action: Values considered as a spike should be flagged as bad data (QC = 4).

Spike test for PH_IN_SITU_TOTAL

The pH sensor can generate occasional spikes due to electrical noise and despiking is appropriate. The default Argo spike tests in core variables (Wong et al., 2018) is

Test value 1 = | V2 − (V3 + V1)/2 | − | (V3 − V1) / 2 |

where V2 is the measurement being tested as a spike, and V1 and V3 are the values above and below. This test does not work well for pH. The criteria depends strongly on the vertical gradient, making it regionally dependent. The test is also sensitive to the vertical resolution of the sensor data, which is platform dependent. We do not recommend this approach.

The spike test recommended for chlorophyll (Schmechtig et al., 2014) is more appropriate for pH. The value tested is

Test value 2 = |V2 – median(V0, V1, V2, V3, V4)|

where the test value represents the anomaly of the observed pH from the median of the surrounding data. Schmechtig et al. (2014) recommend that the criteria used to determine if a data value is a spike be based on the statistical distribution of the 5 data points. Perhaps because pH has a smaller dynamic range than chlorophyll, we find that a constant value of 0.04 pH works well. A data point is considered a spike and marked with quality flag 4 (data bad) if Test value 2 > 0.04 pH.

Action: Values considered as a spike should be flagged as bad data (QC = 4).

Spike test for NITRATE

We calculate the absolute difference between the nitrate concentration at a certain depth (V2) and a running median (5 values, V0, V1, V2, V3, V4) along the whole profile:

TestValue = ABS[V2 - MEDIAN(V0,V1,V2,V3,V4)]

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.

Action:

In Red Sea and in Mediterranean sea:

IF TestValue > 1 micromole/kg THEN the test failed and V2 should be flagged as bad data (QC = 4).

Other places:

IF TestValue > 5 micromole/kg THEN the test failed and V2 should be flagged as bad data (QC = 4).

11. Gradient test

Note that the gradient test is no more applied on TEMP, PSAL parameter values. It has been replaced by the MEDian with a Distance (MEDD) test.

Gradient test for DOXY

This test is failed when the difference between vertically adjacent measurements is too steep. The test does not consider differences in depth, but assumes a sampling that adequately reproduces changes in DOXY with depth.

Test value = | V2 − (V3 + V1)/2 |

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

The V2 value is flagged when

- the test value exceeds 50 micromol/kg for pressures less than 500 dbar, or
- the test value exceeds 25 micromol/kg for pressures greater than or equal to 500 dbar.

Action: Values that fail this test should be flagged as bad data (QC = 4).

15. Grey list

This test is implemented to stop the real-time distribution of measurements from a sensor that is not working correctly.

The grey list contains the following 7 items:

- Platform: Glider WMO Id
- Parameter: name of the grey listed parameter
- Start date: from that date, all measurements for this parameter are flagged according to provided flag value
- End date: from that date, measurements are not flagged according to provided flag value
- 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 glider

Example:

Glider WMO Id	Parameter	Start date	End date	Flag	Comment	DAC
69456	PSAL	20130424		3		IF

19. Deepest pressure test

This test requires that the time series have pressures that are not higher than 1000 dbar plus 10%.

Action: If there is a region of incorrect pressures, all pressures and corresponding parameter measurements should be flagged as bad data (QC = 4).

20. Questionable Argos position test

This test identifies questionable Argos position data collected during surface drift of a glider deployment by considering the glider 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:

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

$$1.0 * \sqrt{Er_a^2 + Er_b^2}$$

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.

This test has been extended to Iridium fixes (to which a 30 m precision has been assigned).

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 (QC = 4).

This test is applied on (PRES, TEMP, PSAL) time series but not anymore on (PRES, TEMP_DOXY) time series.

2.1.2 Specific real-time quality control tests on glider time series

This section lists the real-time tests that are specific to BGC data.

57. DOXY specific Argo real-time quality control tests

PRES, TEMP and PSAL are used to compute DOXY.

Considering the impact of PRES and TEMP on DOXY calculation, when PRES_QC=4 and

TEMP_QC=4, DOXY_QC should bet set to 4

When PSAL_QC=4, DOXY_QC should be set to 3 because in general PSAL is not bad enough to justify to put a QC=4 to DOXY.

Action:

If TEMP_QC=4 or PRES_QC=4, then DOXY_QC=4

If PSAL_QC=4, then DOXY_QC=3

2.1.3 Test application order on glider time series

The rea-time quality control tests on glider time series 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.

Order	Test number	Test name
1	2	Impossible Date test
2	3	Impossible Location test
3	4	Position on Land test
4	20	Questionable Argos position test
5	15	Grey List test
6	19	Deepest pressure test
7	6	Global Range test
8	7	Regional Range test
9	9	Spike test
10	11	Gradient test
11	25	MEDian with a Distance (MEDD) test
12	57	DOXY specific test

2.2 Real-time quality control tests on glider profiles

This section lists the real-time tests that are applied to profiles generated from glider time series. Thus, the data are supposed to be already checked by real-times quality control tests on glider time-series.

2.2.1 Common real-time Quality Control tests on glider profiles

This section lists the real-time tests that are common to CTD data and BGC data.

8. Pressure increasing test

This test requires that the profile has pressures that are monotonically increasing (assuming the pressures are ordered from smallest to largest).

Action: If there is a region of constant pressure, all but the first of the consecutive levels of constant pressure should be flagged as bad data (=4). If there is a region where pressure reverses, all of the pressures in the reversed part of the profile should be flagged as bad data (=4).

12. Digit rollover test

Only so many bits are allowed to store sensor output values in a glider. 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 glider.

This test is used to make sure the rollover is properly detected:

- TEMP or TEMP_DOXY difference between adjacent pressures > 10 °C
- PSAL difference between adjacent pressures > 5 PSU.

Action: When a rollover is detected, both values used to detect the jump in the data should be flagged with a QC = 4, other values of the profile with a QC = 3.

13. Stuck value test

This test looks for a sensor output in a vertical profile being identical.

Action: Stuck values should be flagged as bad data (QC = 4).

14. Density inversion

This test compares potential density between valid measurements in a profile, in both directions, i.e. 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).

2.2.2 Specific real-time quality control tests on glider profiles

This section lists the real-time tests that are specific to BGC data.

No such tests has been defined yet for glider profiles.

2.2.3 Test application order on glider profiles

The rea-time Quality Control tests on glider 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.

Order	Test number	Test name
1	8	Pressure Increasing Test
2	12	Digit Rollover test
3	13	Stuck Value test

4	14	Density Inversion

2.3 Quality control flag application policy

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 Test 19 (deepest pressure test) cannot be decreased to QC flag 3 (bad data that are potentially correctable) set by Test 6 (global range test).

For gliders where salinity (PSAL) is calculated from temperature (TEMP) and conductivity (CNDC), if temperature is flagged 4 (or 3), then salinity is flagged 4 (or 3).

When a BGC parameter is calculated from other (intermediate or BGC) data, its associated QC is initialized to the worse QC value of the input data.

For example, CHLA is calculated from FLUORESCENCE_CHLA, then if FLUORESCENCE_CHLA_QC = 4 after the stuck value test, the corresponding CHLA_QC is initialized to 4.

3 Delayed-mode quality controls

TBD.

4 Annex A: Implementation of real-time quality control tests at Coriolis

This annex describes how the Real Time Quality Control (RTQC) tests on glider data are implemented at the Coriolis data centre (in the Matlab processing chain http://doi.org/10.17882/45402).

4.1 Test #3: Impossible location test

At Coriolis the checked interval is [-90, 90] for latitudes and [-180, 180] for longitudes.

4.2 Test #4: Position on land test

At Coriolis we use the GEBCO file (https://www.bodc.ac.uk/data/open_download/gebco/gebco_2020/zip/).

The detailed description of the test is the following:

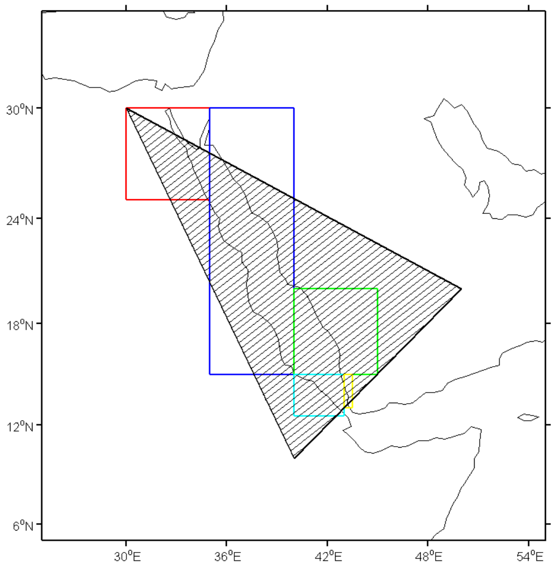
For a given position the elevations are retrieved from the bathymetry file (1, 2 or 4 elevations depending on the position location on the GEBCO grid). The test fails if the average value of the retrieved elevations is ≥ 0.

4.3 Test #7: Regional range test

At Coriolis the regions are defined by rectangular areas.

For the Red Sea Region, 5 rectangular areas:

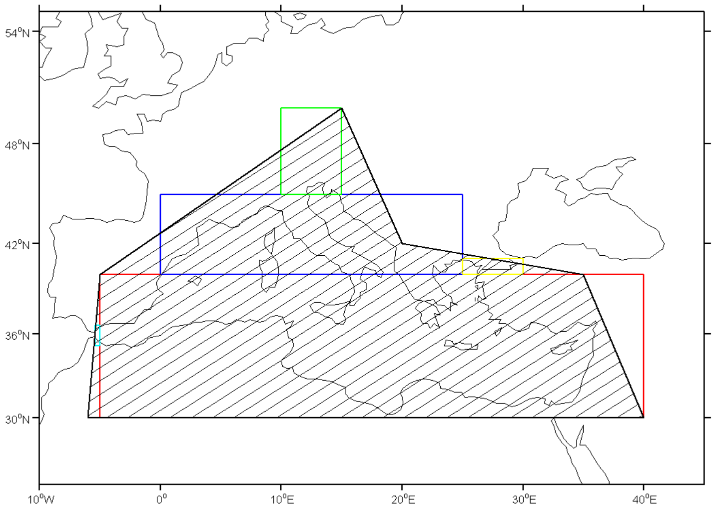
- 30N, 30E ; 25N, 35E,
- 30N, 35E ; 15N, 40E,
- 20N, 40E ; 15N, 45E,
- 15N, 40E ; 12.55N, 43E,
- 15N, 43E ; 13N, 43.5E.



For the Mediterranean Sea region, 5 rectangular areas:

- 40N, 5W ; 30N, 40E,
- 45N, 0; 40N, 25E,

- 50N, 10E; 45N, 15E,
- 41N, 25E; 40N, 30E,
- 36.6N, 5.4W; 35.2N, 5W.



The borders are considered as part of the region.

4.4 Test #8: Pressure increasing test

Some gliders have the ability to transmit 'raw' sampled data, i.e. direct sensor output without any averaging or decimation of the measurements. Some parts of these 'raw' sampled profiles will probably be flagged as bad by this test (at the beginning and end parts of the profile).

As shown in the table below, the flagged data depend on the implementation (if we start the check from shallow values, from deep values or from the middle of the profile). Then, we should specify how to implement this test.

PRES	QC (start shallow)	QC (start deep)	QC (start middle)
0	1	1	1
1	1	4	4
2	1	4	4
1	4	1	1
2	4	4	4
3	1	4	4
4	1	4	4
3	4	4	4
2	4	1	1
3	4	4	4
3	4	1	1
4	4	1	1
5	1	1	1

6	1	4	4
7	1	4	4
8	1	4	4
9	1	4	4
10	1	4	4
9	4	4	4
8	4	4	4
7	4	4	4
6	4	1	1
7	4	1	1
8	4	1	1
9	4	1	1
10	4	1	1
11	1	1	1
12	1	1	1
13	1	1	1
14	1	1	1
15	1	1	1
…	…	…	…
1995	1	1	1
1996	1	1	1
1997	1	4	1
1998	1	4	1
1999	1	4	1
2000	1	4	1
2001	1	4	1
2002	1	4	1
2002	4	4	4
2001	4	4	4
2000	4	4	4
2000	4	4	4
1999	4	4	4
1998	4	4	4

1997	4	1	4

At Coriolis, we start from the middle of the profile and check increasing pressures to deep measurements and decreasing pressures to shallow measurements.

4.5 Test #14: Density inversion

The density inversion test has been implemented thanks to the following Matlab code (provided by Cécile Cabanes).

```
% [delta_upbot,delta_botup,is_inv_dens] = test_density14(P,T,S,threshold)
% INPUT:
% P(1,nlevel) pression (dB)
% T(1,nlevel) T in situ (°C)
% S(1,nlevel) S in situ (PSU)
% treshold : treshold used for detecting density inversion (0.03kg/m3)
% OUTPUT:
% delta_upbot (1,nlevel) : top to bottom test :  if delta_upbot >= 0.03kg/m3 flag 4 of the
level i
% delta_botup (1,nlevel) : bottom to top test:  if delta_botup <= -0.03kg/m3 flag of the level
i
% is_inv_dens (1,nlevel) : is_inv_dens=1 if level i should be flagued as bad (flag 4) ; ==0
otherwise

% You will need The ITS-90 version of the CSIRO SEAWATER library, which you can obtain from
% http://www.cmar.csiro.au/datacentre/ext_docs/seawater.htm.  Version 3.3   22-Sep-2010
%

function [delta_upbot,delta_botup,is_inv_dens] = test_density14(P,T,S,treshold)

if size(P,1)> 1
P=P';
end
if size(T,1)> 1
T=T';
end
if size(S,1)> 1
S=S';
end


%%
% From top to bottom :  two consecutive levels, i and i+1, are checked (Pi < Pi+1 )
%%
Pref = (P(1:end-1)+P(2:end))/2;


% rho_i(Pi, Ti, Si, Pref): potential density referenced to the mid-point Pref
```

```
rhop_i = sw_pden(S(1:end-1),T(1:end-1),P(1:end-1),Pref);

% rho_ip1(Pi+1, Ti+1, Si+1, Pref): potential density referenced to the mid-point Pref

rhop_ip1 = sw_pden(S(2:end),T(2:end),P(2:end),Pref);

delta_upbot = [rhop_i-rhop_ip1, NaN];


%%
% From bottom to top:  two consecutive levels, i and i+1, are checked (Pi > Pi+1 )
%%
Pref = (P(2:end)+P(1:end-1))/2;


% rho_i(Pi, Ti, Si, Pref): potential density referenced to the mid-point Pref

rhop_i = sw_pden(S(2:end),T(2:end),P(2:end),Pref);

% rho_ip1(Pi+1, Ti+1, Si+1, Pref): potential density referenced to the mid-point Pref

rhop_ip1 = sw_pden(S(1:end-1),T(1:end-1),P(1:end-1),Pref);

delta_botup = [NaN, rhop_i-rhop_ip1];


% test if there is density inversion with the 0.03kgm-3 criterion

is_inv_dens = delta_upbot>=treshold | delta_botup<=-treshold;
```

4.6 Test #19: Deepest pressure test

At Coriolis, we decided to apply this test to parameters that already have at least one (other) RTQC test defined (see list in §2).

4.7 Test #20: Questionable Argos position test

The JAMSTEC trajectory quality control method is described in Nakamura et al (2008), "Quality control method of Argo float position data", JAMSTEC Report of Research and Development, Vol. 7, 11-18 (http://www.godac.jamstec.go.jp/catalog/data/doc_catalog/media/JAM_RandD07_02.pdf).

This method checks the surface trajectory of an Argos float by considering the speeds induced by the successive Argos fixes. The test can flag the surface position as '3' or '4'.

In the following, we propose a detailed description of the algorithm to implement.

4.7.1 Inputs

The inputs of the algorithm are:

- The surface trajectory to be checked (N Argos location dates, latitudes, longitudes and classes),
- The last good (flagged as '1') surface location of the (already checked) previous (received) cycle.

4.7.2 Algorithm

Assuming that the location dates have not been flagged as bad by the test #2 "Impossible date test", we first chronologically sort the surface positions.

The whole surface trajectory is used to initialize the (checked) current trajectory.

The current trajectory is processed in an infinite loop in which the following steps are performed:

4.7.2.1 Step 1

The subsurface drift speed is computed between the last good surface position of the previous cycle and the first position of the current trajectory.

If this speed is greater than 3 m/s, the first position of the current trajectory is flagged as '4', this position is then excluded from the current trajectory and a new iteration of the infinite loop starts.

4.7.2.2 Step 2

Speeds are computed for the second position to the last position of the current trajectory. Each speed is computed between position #i-1 and position #i and affected to position #i.

In case of duplicated positions (i.e. if position #i-1 and position #i have the same latitude, longitude and date): the position #i is flagged as '4', it is then excluded from the current trajectory and a new iteration of the infinite loop starts.

In case of an erroneous cycle number of the position #i (i.e. if the times difference between position #i and position #i-1 is greater than one day): the position #i is flagged as '4', it is then excluded from the current trajectory and a new iteration of the infinite loop starts.

4.7.2.3 Step 3

The position #iMax is found as the position with the maximum speed.

If this maximum speed is greater than 3 m/s, the position #iMax is 'questionable' and the **speed test** (see §4.7.3) is performed on it over the current trajectory.

The **speed test** should lead to define position #iMax or/and position #iMax-1 as 'abnormal'.

4.7.2.4 Step 4

If the **distance test** (see §4.7.4) between position #iMax and position #iMax-1 is verified, the 'abnormal' position(s) is (are) flagged as '3'.

The 'abnormal' position(s) is(are) then deleted from the current trajectory (even when the distance test is not verified) and a new iteration of the infinite loop starts.

The infinite loop ends when no 'abnormal' position has been detected or when the current trajectory has less than 2 positions.

4.7.3 Speed test

The speed test is performed on a 'questionable' position over a given trajectory.

The 'questionable' position (called B in the following) can be all but the first position of the trajectory. The position which precedes B on the trajectory is called A in the following.

4.7.3.1 Case of different Argos classes

If positions A and B have different Argos classes, the position with the less accurate Argos class is defined as 'abnormal' by the speed test.

Remember that the accuracy of the Argos location classes is the following:

more accurate <= 3, 2, 1, 0, A, B, Z => less accurate

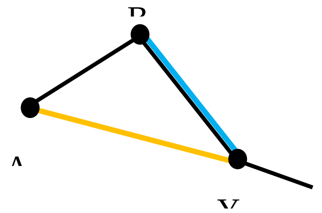
4.7.3.2 Case of identical Argos classes

If positions A and B have the same Argos classes:

- If the trajectory only comprises the two positions A and B, both positions are defined as 'abnormal' by the speed test,
- Otherwise the speed test depends on the position of the location B on the trajectory, 3 cases are possible.

Case 1: If B is the second position of the trajectory

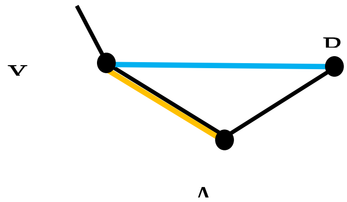
In this case: A is the first position, B the second one and there is a position Y following the position B on the trajectory.



Speeds on the segments A-Y (orange) and B-Y (blue) are computed: if $speed_{A\text{-}Y}$ is greater than $speed_{B\text{-}Y}$, the position A is defined as 'abnormal' by the speed test otherwise B is defined as 'abnormal' by the speed test.

Case 2: If B is the last position of the trajectory

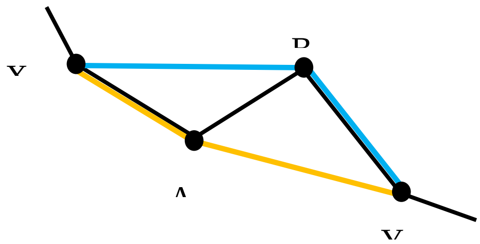
In this case: A is the last but one position, B is the last position and there is a position X preceding the position A on the trajectory.



Speeds on the segments X-A (orange) and X-B (blue) are computed: if $speed_{X-A}$ is greater than $speed_{X-B}$, the position A is defined as 'abnormal' by the speed test otherwise B is defined as 'abnormal' by the speed test.

Case 3: we are not in case 1 or 2

In this case: there is a position X preceding the position A on the trajectory and a position Y following the position B on the trajectory.



Speeds on the segments X-A-Y (orange trajectory) and X-B-Y (blue trajectory) are computed. If $speed_{X-A-Y}$ is greater than $speed_{X-B-Y}$, the position A is defined as 'abnormal' by the speed test otherwise B is defined as 'abnormal' by the speed test.

4.7.4 Distance test

The distance test is performed on two Argos locations A and B.

The distance test is verified if the distance between locations A and B is greater or equal to

$1.0 \times \sqrt{{Er_A}^2 + {Er_B}^2}$ where Er_A and Er_B are the radii of position error for locations A and B respectively.

These position errors, deduced from the position classes, are 150 m, 350 m and 1000 m for Argos class 3, 2 and 1 respectively. Moreover we have associated a position error of 1500 m, 1501 m, 1502 m and 1503 m for Argos classes 0, A, B and Z respectively.

4.7.5 Distance computation

As far as distance and speed are concerned in this trajectory QC method, we must specify an algorithm to compute distance between positions of the surface trajectory. This algorithm must be common to all the DACs so that the trajectory QC results will not depend on DAC's distance computation method.

We propose to use the distance algorithm from the *Laboratoire de Physiques des Océans* (*LPO*) at IFREMER.

This algorithm computes distance between points on the earth using the WGS 1984 ellipsoid, its Matlab implementation and some test points are provided below.

4.7.5.1 Matlab implementation of the LPO distance algorithm

```
function [range, A12, A21] = distance_lpo(lat, long)
%
% Computes distance and bearing between points on the earth using WGS 1984
% ellipsoid
%
% [range, A12, A21] = distance_lpo(lat, long) computes the ranges RANGE between
% points specified in the LAT and LONG vectors (decimal degrees with positive
% indicating north/east). Forward and reverse bearings (degrees) are returned
% in AF, AR.
%
% Ellipsoid formulas are recommended for distance d<2000 km,
% but can be used for longer distances.
%
% GIVEN THE LATITUDES AND LONGITUDES (IN DEG.) IT ASSUMES THE IAU SPHERO
% DEFINED IN THE NOTES ON PAGE 523 OF THE EXPLANATORY SUPPLEMENT TO THE
% AMERICAN EPHEMERIS.
%
% THIS PROGRAM COMPUTES THE DISTANCE ALONG THE NORMAL
% SECTION (IN M.) OF A SPECIFIED REFERENCE SPHEROID GIVEN
% THE GEODETIC LATITUDES AND LONGITUDES OF THE END POINTS
% *** IN DECIMAL DEGREES ***
%
% IT USES ROBBIN'S FORMULA, AS GIVEN BY BOMFORD, GEODESY,
% FOURTH EDITION, P. 122.  CORRECT TO ONE PART IN 10**8
% AT 1600 KM.  ERRORS OF 20 M AT 5000 KM.
%
% CHECK:  SMITHSONIAN METEOROLOGICAL TABLES, PP. 483 AND 484,
% GIVES LENGTHS OF ONE DEGREE OF LATITUDE AND LONGITUDE
% AS A FUNCTION OF LATITUDE. (SO DOES THE EPHEMERIS ABOVE)
%
% PETER WORCESTER, AS TOLD TO BRUCE CORNUELLE...1983 MAY 27
%

% On 09/11/1988, Peter Worcester gave me the constants for the
% WGS84 spheroid, and he gave A (semi-major axis), F = (A-B)/A
% (flattening) (where B is the semi-minor axis), and E is the
% eccentricity, E = ( (A**2 - B**2)**.5 )/ A
% the numbers from peter are: A=6378137.; 1/F = 298.257223563
% E = 0.081819191
A = 6378137.;
E = 0.081819191;
B = sqrt(A.^2 - (A*E).^2);
EPS = E*E/(1.-E*E);

NN = max(size(lat));
if (NN ~= max(size(long))),
```

```
   error('dist: Lat, Long vectors of different sizes!');
end

if (NN == size(lat))
   rowvec = 0;  % it is easier if things are column vectors,
else
   rowvec = 1; % but we have to fix things before returning!
end;

% convert to radians
lat = lat(:)*pi/180;
long = long(:)*pi/180;

% fixes some nasty 0/0 cases in the geodesics stuff
lat(lat == 0) = eps*ones(sum(lat == 0), 1);

% endpoints of each segment
PHI1 = lat(1:NN-1);
XLAM1 = long(1:NN-1);
PHI2 = lat(2:NN);
XLAM2 = long(2:NN);

% wiggle lines of constant lat to prevent numerical problems.
if (any(PHI1 == PHI2))
   for ii = 1:NN-1
      if (PHI1(ii) == PHI2(ii))
         PHI2(ii) = PHI2(ii) + 1e-14;
      end
   end
end
% wiggle lines of constant long to prevent numerical problems.
if (any(XLAM1 == XLAM2))
   for ii = 1:NN-1
      if (XLAM1(ii) == XLAM2(ii))
         XLAM2(ii) = XLAM2(ii) + 1e-14;
      end
   end
end

% COMPUTE THE RADIUS OF CURVATURE IN THE PRIME VERTICAL FOR EACH POINT
xnu = A./sqrt(1.0-(E*sin(lat)).^2);
xnu1 = xnu(1:NN-1);
xnu2 = xnu(2:NN);

% COMPUTE THE AZIMUTHS.
% A12 (A21) IS THE AZIMUTH AT POINT 1 (2) OF THE NORMAL SECTION CONTAINING THE POINT 2 (1)
TPSI2 = (1.-E*E)*tan(PHI2) + E*E*xnu1.*sin(PHI1)./(xnu2.*cos(PHI2));
PSI2 = atan(TPSI2);
```

```
% SOME FORM OF ANGLE DIFFERENCE COMPUTED HERE??
DPHI2 = PHI2-PSI2;
DLAM = XLAM2-XLAM1;
CTA12 = (cos(PHI1).*TPSI2 - sin(PHI1).*cos(DLAM))./sin(DLAM);
A12 = atan((1.)./CTA12);
CTA21P = (sin(PSI2).*cos(DLAM) - cos(PSI2).*tan(PHI1))./sin(DLAM);
A21P = atan((1.)./CTA21P);

% GET THE QUADRANT RIGHT
DLAM2 = (abs(DLAM)<pi).*DLAM + (DLAM>=pi).*(-2*pi+DLAM) + (DLAM<=-pi).*(2*pi+DLAM);
A12 = A12 + (A12<-pi)*2*pi-(A12>=pi)*2*pi;
A12 = A12 + pi*sign(-A12).*(sign(A12) ~= sign(DLAM2));
A21P = A21P + (A21P<-pi)*2*pi - (A21P>=pi)*2*pi;
A21P = A21P + pi*sign(-A21P).*(sign(A21P) ~= sign(-DLAM2));
% A12*180/pi
% A21P*180/pi

SSIG = sin(DLAM).*cos(PSI2)./sin(A12);

% At this point we are OK if the angle < 90 but otherwise
% we get the wrong branch of asin!
% This fudge will correct every case on a sphere, and *almost*
% every case on an ellipsoid (wrong handling will be when
% angle is almost exactly 90 degrees)
dd2 = [cos(long).*cos(lat) sin(long).*cos(lat) sin(lat)];
dd2 = sum((diff(dd2).*diff(dd2))')';
if (any(abs(dd2-2) < 2*((B-A)/A))^2),
   disp('dist: Warning...point(s) too close to 90 degrees apart');
end
bigbrnch = dd2>2;

SIG = asin(SSIG).*(bigbrnch==0) + (pi-asin(SSIG)).*bigbrnch;
A21 = A21P - DPHI2.*sin(A21P).*tan(SIG/2.0);

% COMPUTE RANGE
G2 = EPS*(sin(PHI1)).^2;
G = sqrt(G2);
H2 = EPS*(cos(PHI1).*cos(A12)).^2;
H = sqrt(H2);
TERM1 = -SIG.*SIG.*H2.*(1.0-H2)/6.0;
TERM2 = (SIG.^3).*G.*H.*(1.0-2.0*H2)/8.0;
TERM3 = (SIG.^4).*(H2.*(4.0-7.0*H2)-3.0*G2.*(1.0-7.0*H2))/120.0;
TERM4 = -(SIG.^5).*G.*H/48.0;

range = xnu1.*SIG.*(1.0 + TERM1 + TERM2 + TERM3 + TERM4);

% CONVERT TO DECIMAL DEGREES
```

```
A12 = A12*180/pi;
A21 = A21*180/pi;
if (rowvec),
   range = range';
   A12 = A12';
   A21 = A21';
end
```

4.7.5.2 Test points

The following table provides results of calculation distances from the LPO distance algorithm.

Test #	Longitude point #1	Latitude point #1	Longitude point #2	Latitude point #2	Distance (m)
1	59.137	81.450	132.862	-71.971	17452769.38
2	245.057	-75.309	331.764	-77.086	2110391.35
3	185.622	87.327	183.692	-17.999	11689986.02
4	182.640	20.009	49.196	5.048	14227739.39
5	150.579	41.603	208.973	39.188	4868529.07
6	0.000	0.000	332.341	19.629	3717195.47
7	356.228	79.610	254.896	-47.763	15364005.55
8	199.871	88.917	70.224	52.035	4312751.18
9	287.193	-35.107	200.803	52.926	12831368.01
10	102.486	-83.242	312.077	75.131	18753227.55
11	69.797	88.120	207.543	18.708	8087967.56
12	93.492	-16.942	304.265	20.978	16765984.94
13	199.115	-39.885	182.679	60.574	11263499.39
14	303.234	77.720	332.681	-0.149	8830419.21
15	152.391	-4.042	179.072	-21.859	3490115.84
16	38.772	-90.000	252.147	9.952	11097348.67
17	170.518	85.414	311.396	-28.009	13474193.18
18	83.708	44.039	273.558	48.297	9728568.10
19	325.393	4.457	60.402	-18.541	10702629.73
20	201.680	15.173	142.753	-29.194	8013018.54