NOTE ON "ESTIMATE_PROFILE_LOCATION" TOOL

We implemented the "Terrain-following interpolation for under-ice floats" method presented by Kaihe Yamazaki during ADMT 22 (https://docs.google.com/presentation/d/1XMSh20h8UEOt8D44iUuQur4-2Y7O44a3/edit?usp=sharing&ouid=106292473192519010847&rtpof=true&sd=true) and described in Appendix A of Kaihe Yamazaki et al. article (https://doi.org/10.1029/2019JC015406). The "Terrain-following" method explained in this paper should be understood before reading this note.

We then discovered that some results of the method may not cope with the principles of the "Terrain-following" (which is to follow the bathymetry) and the obtained trajectory may cross shallow isobaths. This is due to the principle of the method to always go forward to look for the path which "minimalize the difference between the local and the reference depth". The bathymetry can have dead ends for expected isobaths and, in such cases, we must go backwards to find a new path (see below example of cycles 82 to 124 of float WMO 2900114).

We also tried to improve the method by considering in situ data measured by the float. The average drift depth, the max depth of the profile and the grounded flag are then used to constrain the algorithm.

The tool provides (in a CSV file) information on the trajectory found in both directions (forward and backward along the trajectory) so that each DMQC operator can choose the relevant points to be used to replace RT profile position generally estimated from linear interpolation of the available positions.

TABLE OF CONTENTS

1. Tool requirements and configuration

The tool is composed of a unique standalone Matlab script named "estimate_profile_locations.m".

1.1. Tool requirements

The only requirement needed is the GEBCO worldwide bathymetric atlas.
We use the 2022 NetCDF version of this file available at
https://www.bodc.ac.uk/data/open_download/gebco/gebco_2022/zip/.

The tool can also generate few plots to provide an overview of the geographic constraints and the results of the algorithm. We then highly recommend to download and install the M_Map Matlab package (https://www.eoas.ubc.ca/~rich/map.html).

Float input data are Argo NetCDF files (version 3.1) available on the GDAC.

1.2. Tool configuration

The configuration of the tool is done inline, in the upper part of the script between the "CONFIGURATION - START" and "CONFIGURATION - END" comments.

The configuration variables are listed and described in the following table.

Configuration variable name	Configuration variable description
FLOAT_LIST_FILE_NAME	Default list of floats to process (list of WMO numbers).
DIR_INPUT_NC_FILES	Directory of input float data (Argo NetCDF files).
DIR_OUTPUT_FILES	Directory to store tool outputs (CSV file and PNG plots).
DIR_LOG_FILE	Directory to store tool log file.
GEBCO_FILE	File path name of the GEBCO atlas.
DIFF_DEPTH_TO_START	Maximum difference (in meters) between sea bottom and float RPP (Representative Park Pressure) to start the process.
FLOAT_VS_BATHY_TOLERANCE	Tolerance (in meters) used to compare reference depths and GEBCO depths.
FLOAT_VS_BATHY_TOLERANCE_FOR_GRD	Tolerance (in meters) used to compare reference depths and GEBCO depths when the float grounded.
FIRST_RANGE	First (half) length (in kilometers) of the search range segment.
LAST_RANGE	Last (half) length (in kilometers) of the search range segment.
RANGE_PERIOD	Period used to change search range segment size.
PLOT_FLAG	Flag to generate plots (M_Map Matlab package required).

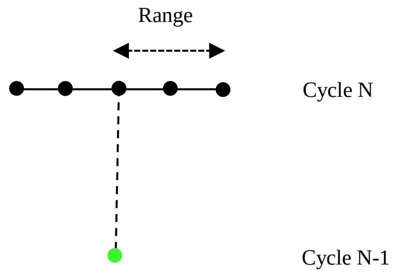
2. TOOL IMPROVEMENTS

Two main modifications have been done on the algorithm proposed by Kaihe Yamazaki.

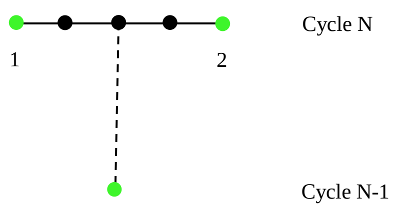
2.1. Going backwards in case of dead end in the isobaths

In the original algorithm, each new estimated position is chosen on the path which "minimalize the difference between the local and the reference depth", whatever this difference is. This means that it always goes forward and then can cross shallow bathymetry in specific sea bottom configurations (dead ends in the isobath).

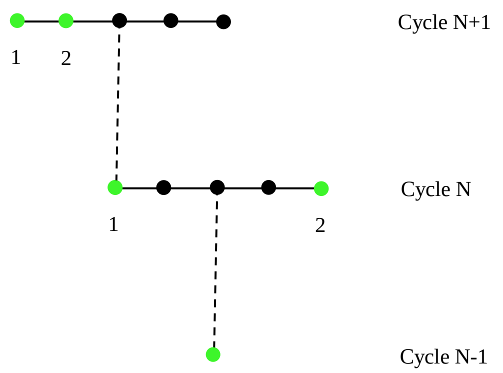
In our algorithm, the positions on the search segment are first decided to be eligible or not. In an eligible position the float is always above the bathymetry.
Then, the eligible positions are sorted according to the difference between the bathymetry and the float depth (the minimum differences come first).
The algorithm then tries to find a path from first cycle to last cycle between these eligible positions by checking all possible ways.
If all cycle #N eligible positions of the current path search segment have been checked without finding any eligible position in the cycle #N+1 associated search segment, it can go backwards for another try on the next eligible position of the cycle #N-1 associated segment.



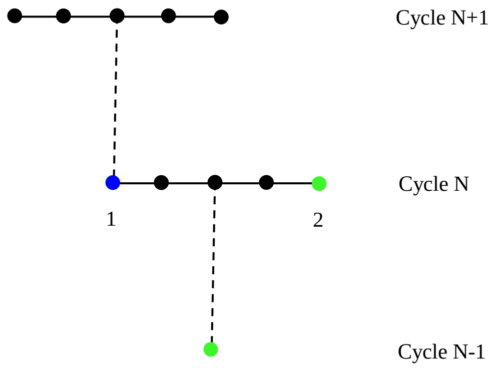
Step #1
For a given point of cycle #N-1, we create a local set of points on the search segment for cycle #N. We create a point each kilometre, thus in the example (with range = 2 km), 2*2+1= 5 points are created.



Step #2
Only the eligible points of the search segment are considered (in green). Moreover, they are sorted according to the difference between the bathymetry and the float depth.



Step #3
If points are eligible on the local search segment of cycle #N+1. The algorithm will try these points according to their sorted number.



Step #4
If no point is eligible on the local search segment of cycle #N+1. We have a dead end for this point of cycle #N (in blue). In that case, the next point (#2) of the Cycle #N search segment is checked.

Note that the result of the algorithm may fail to find a path from first to last cycles under these constraints.

2.2. Using in-situ measurements provided by the float as a constraint

The position estimation process is done on a given point only if the associated "average drift depth" (called RPP for Representative Park Pressure) is "not too far" from sea bottom. The `DIFF_DEPTH_TO_START` configuration value is used for this comparison.

A reference depth is assigned to each position to be estimated.

In the original algorithm it is a linear interpolation of the bottom depths of the first and last cycle available positions.

In our algorithm, we replace some of these reference depths by the maximum profile pressure of the cycle in 2 cases:

- If this max pressure profile is deeper that the reference depth,
- If the float grounded during the cycle.

3. TOOL USAGE, WORK AND OUTPUTS

3.1. Tool usage

Using the command without argument:

```
>> estimate_profile_locations
```

will process all the floats of the `FLOAT_LIST_FILE_NAME` file.

You can also choose to provide the floats to be processed by providing their WMO numbers as input parameters:

```
>> estimate_profile_locations(2900114, 2900129)
```

3.2. How does the tool work?

The algorithm will try to find a path to go forward, from first to last cycle, and then backward, from last to first cycle.

A reference depth is assigned to each cycle. It is initialized with a linear interpolation of end points bottom depths.
If a reference depth is shallower that the profile max pressure measured by the float, this later value is set as reference depth, same thing if the float grounded during the cycle.

In each new local search segment, the positions are first set eligible or not.
The difference between local sea bottom depth and reference depth is first computed:
`freeWater = seaBottomDepth – referenceDepth`

If the float grounded during the cycle, a position is eligible when
`freeWater ≥ -FLOAT_VS_BATHY_TOLERANCE_FOR_GRD` and `freeWater ≤ FLOAT_VS_BATHY_TOLERANCE_FOR_GRD`
If the float didn't ground during the cycle, a position is eligible when
`freeWater ≥ -FLOAT_VS_BATHY_TOLERANCE`

Only eligible positions are checked, moreover the eligible positions are sorted (`freeWater` increasing values) and checked in the obtained order.

If a path is successfully found, the algorithm stopped, otherwise the range of the search segment (originally set to `FIRST_RANGE`) is increased (by `RANGE_PERIOD`) and so on until it succeeds or until the try at `LAST_RANGE` fails.

3.3. Tool output

The too generates a CSV file with the estimated positions and few additional information that may help the operator to choose the estimated positions.

The information provided in the CSV file are the following:

For all cycles.

Column name	Column content
WMO	Float WMO number.
CyNum	Cycle number.
Dir	Profile direction.
Juld	Profile date.
JuldQc	Profile date QC.
JuldLoc	Profile position date.
Lat	Profile latitude.
Lon	Profile longitude.
PosQc	Profile position QC.
Speed	Average speed between profiles (in cm/s).
ProfPresMax	Max pressure of the profile.
Rpp	RPP (Representative Park Pressure) of the cycle.
Grd	Grounded flag.
GrdPres	Grounded pressure.
GebcoDepth	GEBCO depth at profile location.

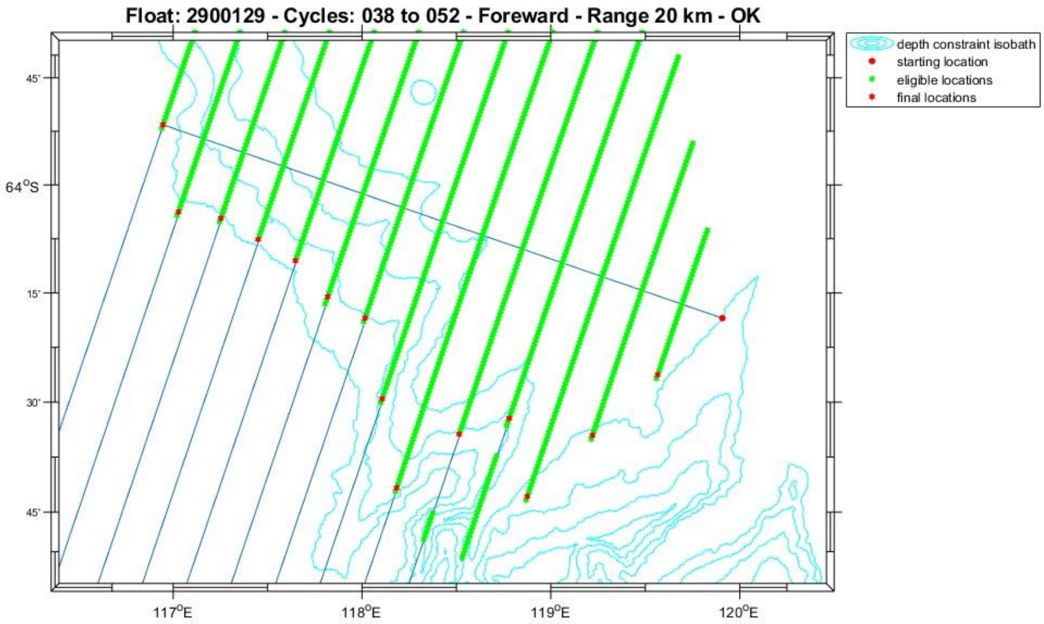
For cycles where positions are estimated.

Column name	Column content
SetNum	Number of the set of positions to be estimated.
DepthConstraint	Reference depth.
ForwLat	Estimated latitude of the forward path.
ForwLon	Estimated longitude of the forward path.
ForwGebcoDepth	GEBCO depth at estimated position of the forward path.
ForwDiffDepth	ForwGebcoDepth - DepthConstraint.
BackwLat	Estimated latitude of the backward path.
BackwLon	Estimated longitude of the backward path.
BackwGebcoDepth	GEBCO depth at estimated position of the backward path.
BackwDiffDepth	BackwGebcoDepth - DepthConstraint.
TrajLat	Latitude on the final trajectory.
TrajLon	Longitude on the final trajectory.
SpeedEst	Average speed on the final trajectory (in cm/s).

The last 4 columns DIFF_DEPTH_TO_START, FLOAT_VS_BATHY_TOLERANCE, FLOAT_VS_BATHY_TOLERANCE_FOR_GRD and TOOL_VERSION are used to store configuration parameter values used during the run.

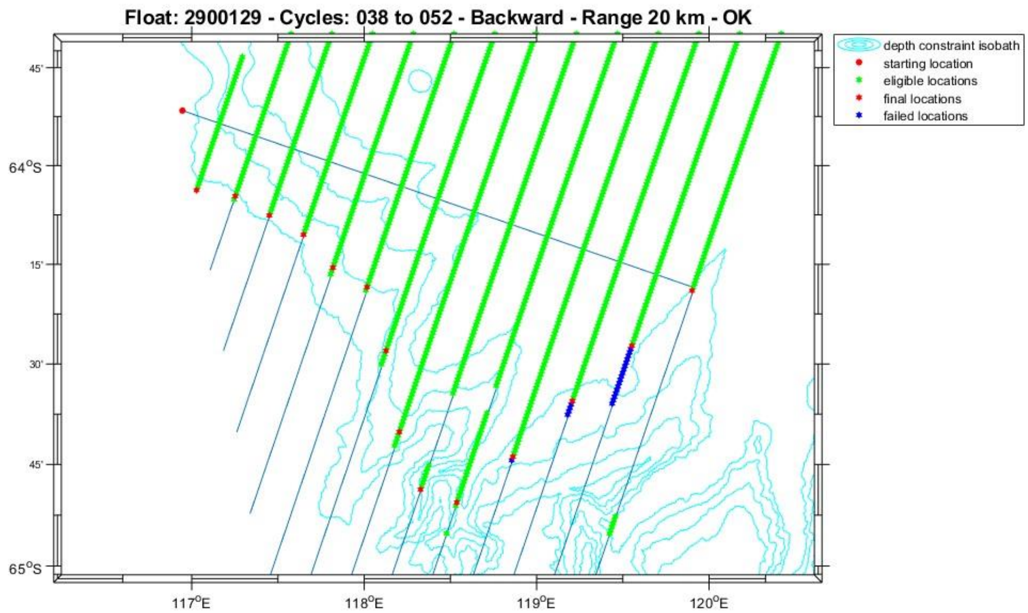
If `PLOT_FLAG` is set to 1, it also generates 3 plots per set of estimated positions:

1. One to plot the results of the forward pass of the algorithm,

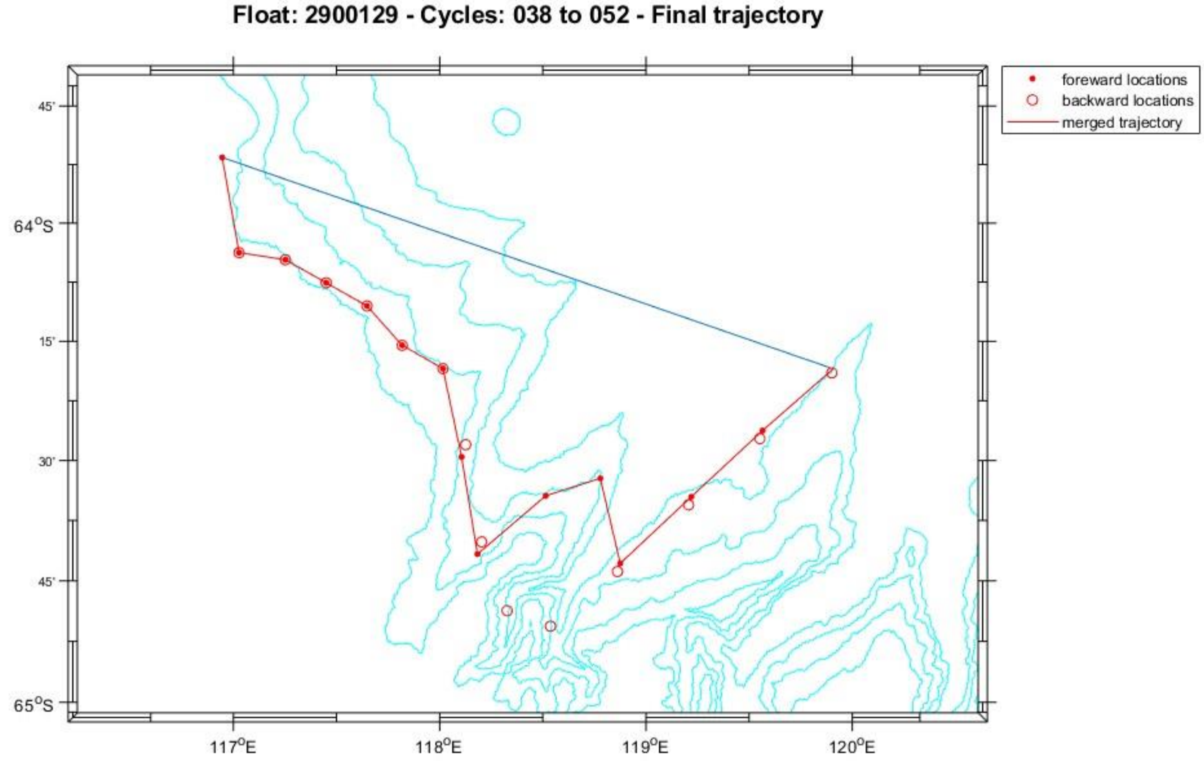


Isobath are generated from min and max values of reference depths.
Green stars are points on the search segments that have not been checked yet.
Blue stars are points that have been unsuccessfully checked (leads to a dead end).
Red stars are estimated positions (points of the final forward/backward trajectory).

2. One to plots the results of the backward pass of the algorithm,



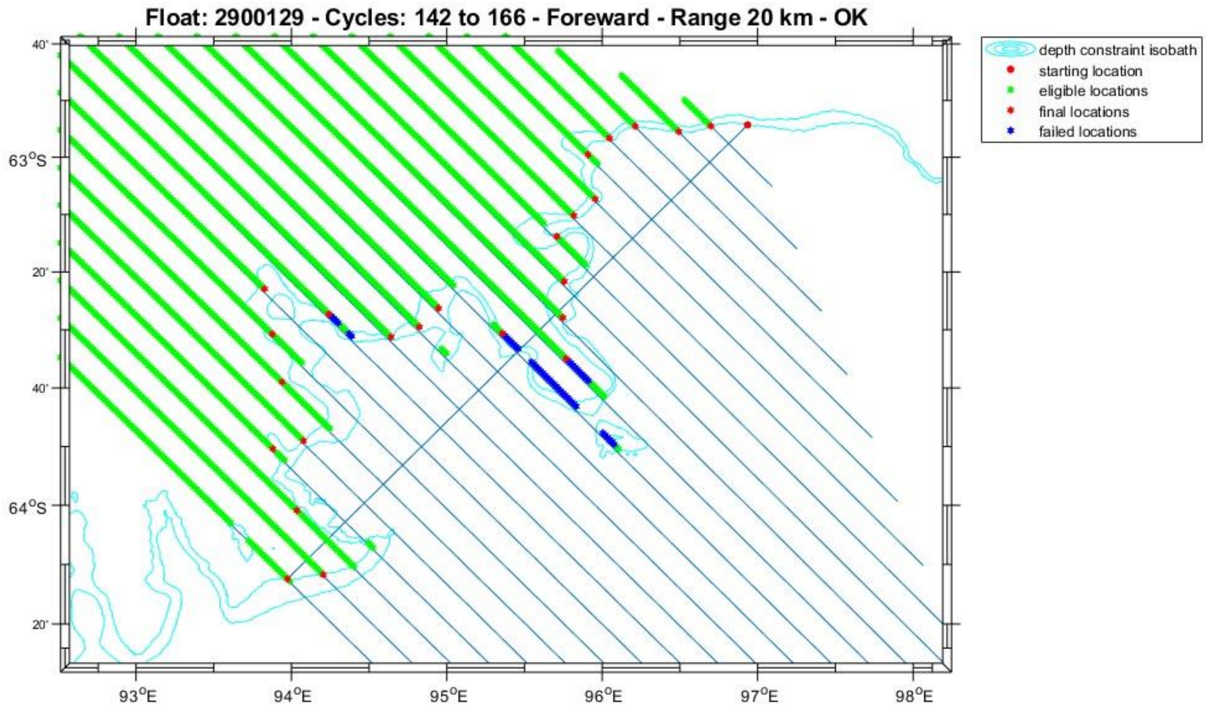
3. One with a proposed trajectory created in its first half with the forward estimated positions and in its second half with the backward estimated ones.

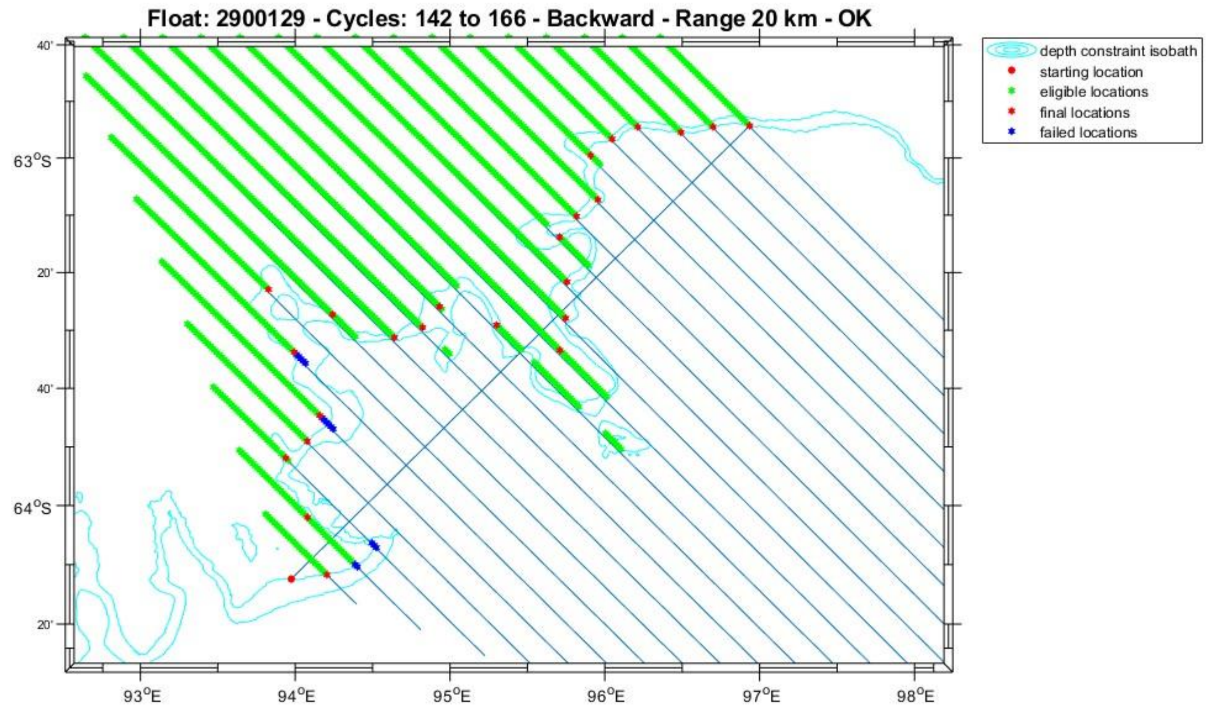


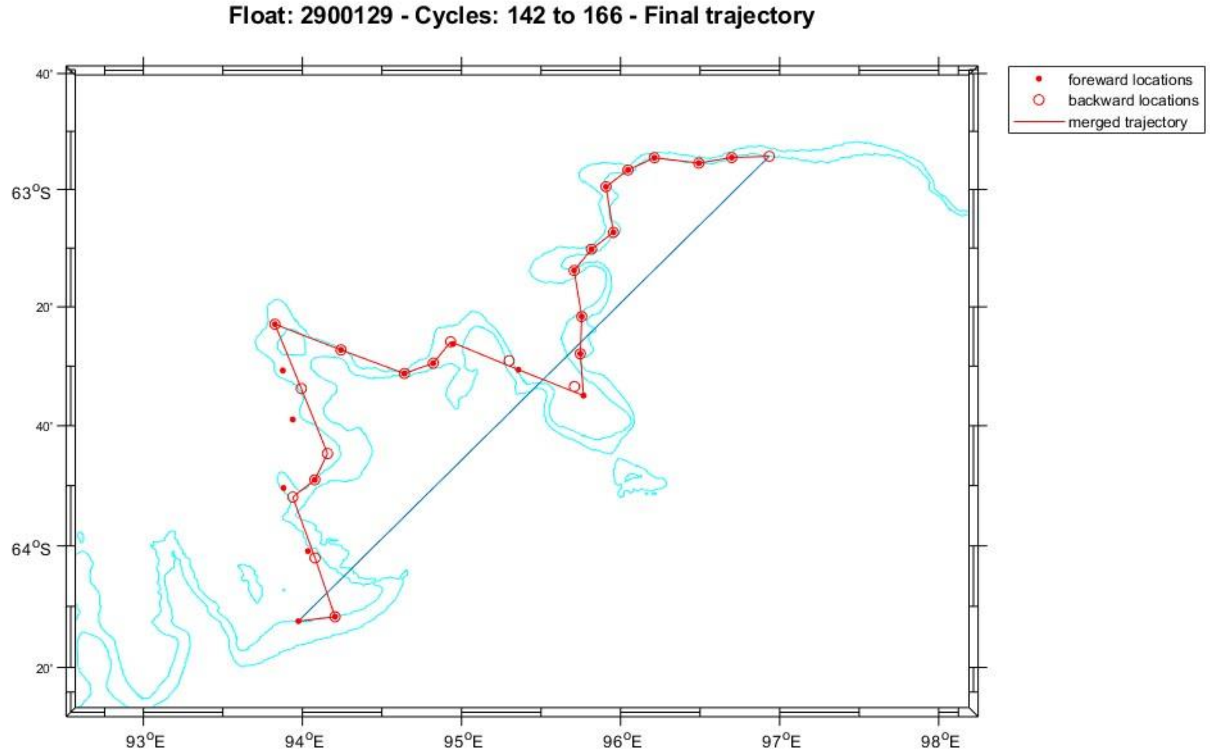
Red points/circles are estimated positions from forward/backward pass.
Red line is the final proposed trajectory.

4. FEW EXAMPLES

4.1. 2900129 cycles 142-166

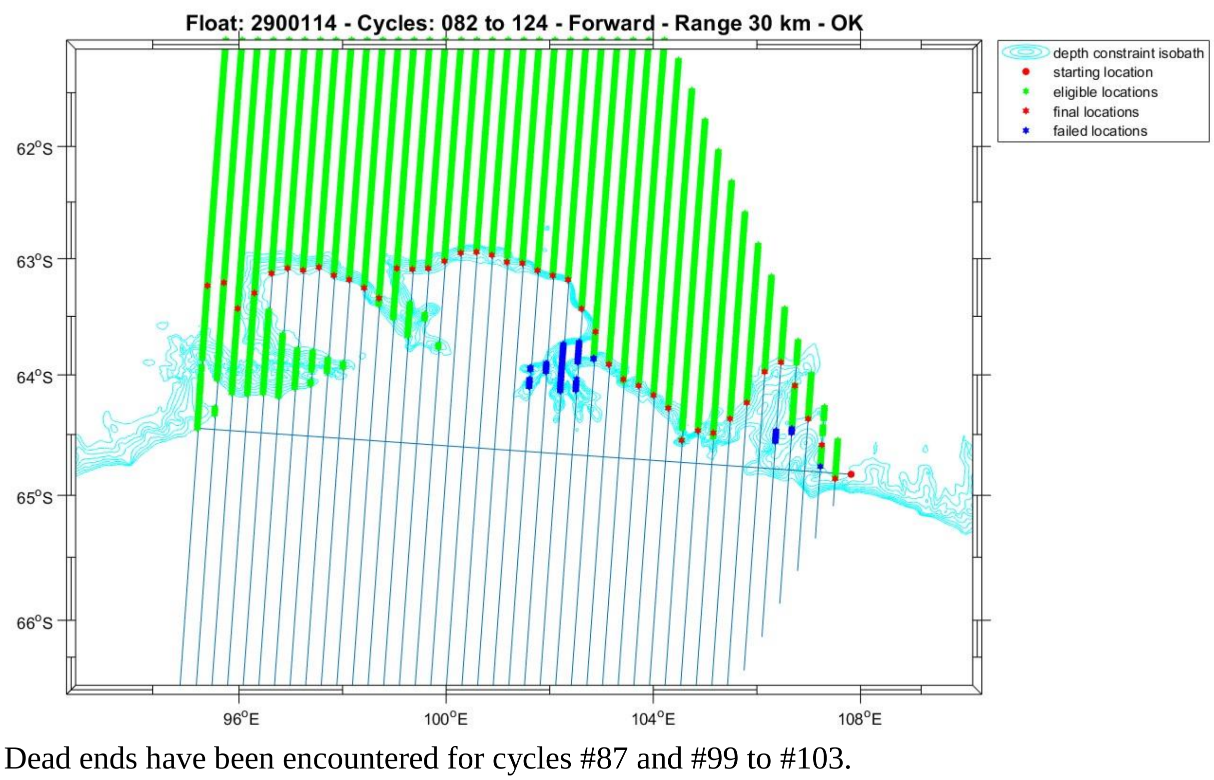




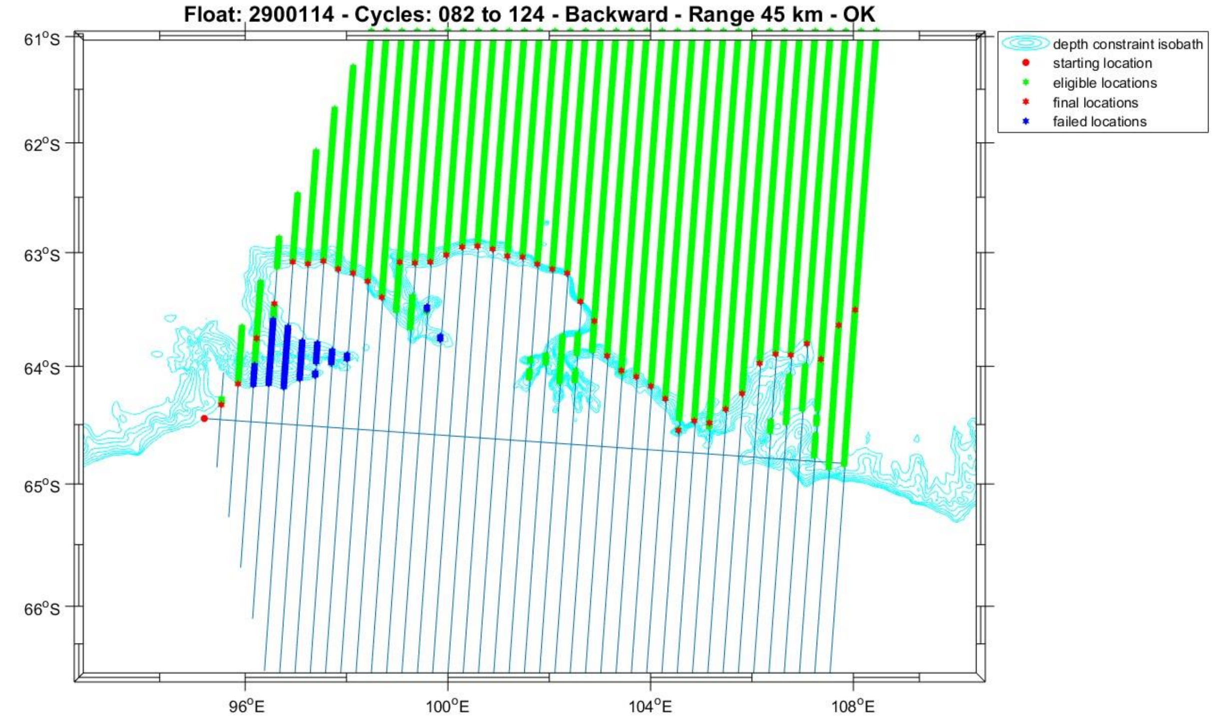


The result seems sensible, but it is mainly because of the well-conditioned geographic configuration.

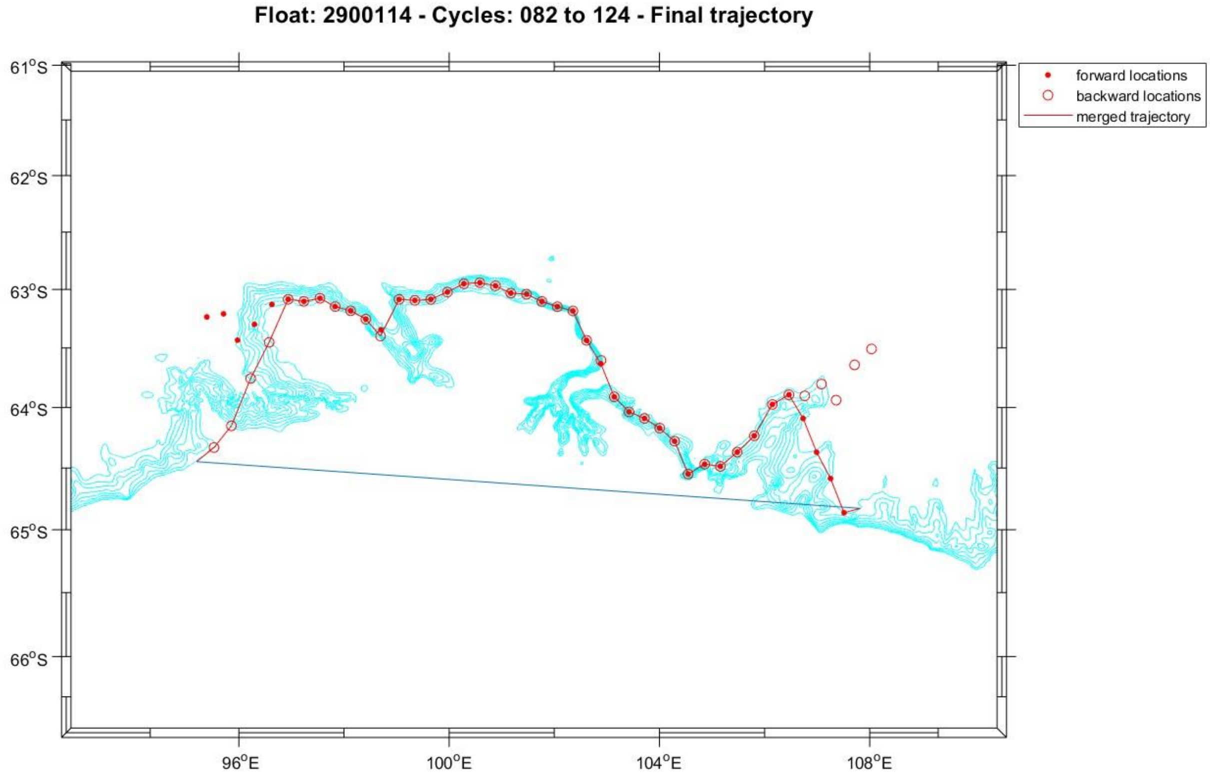
4.2. 2900114 cycles 082-124



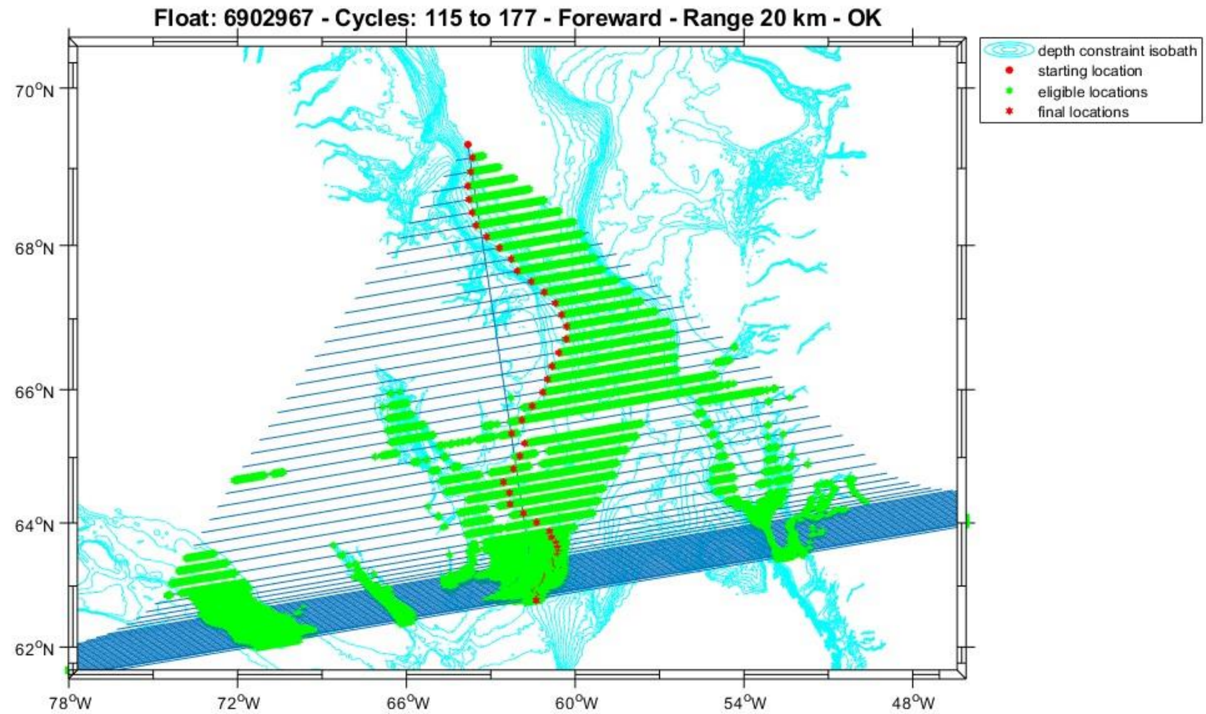
Dead ends have been encountered for cycles #87 and #99 to #103.

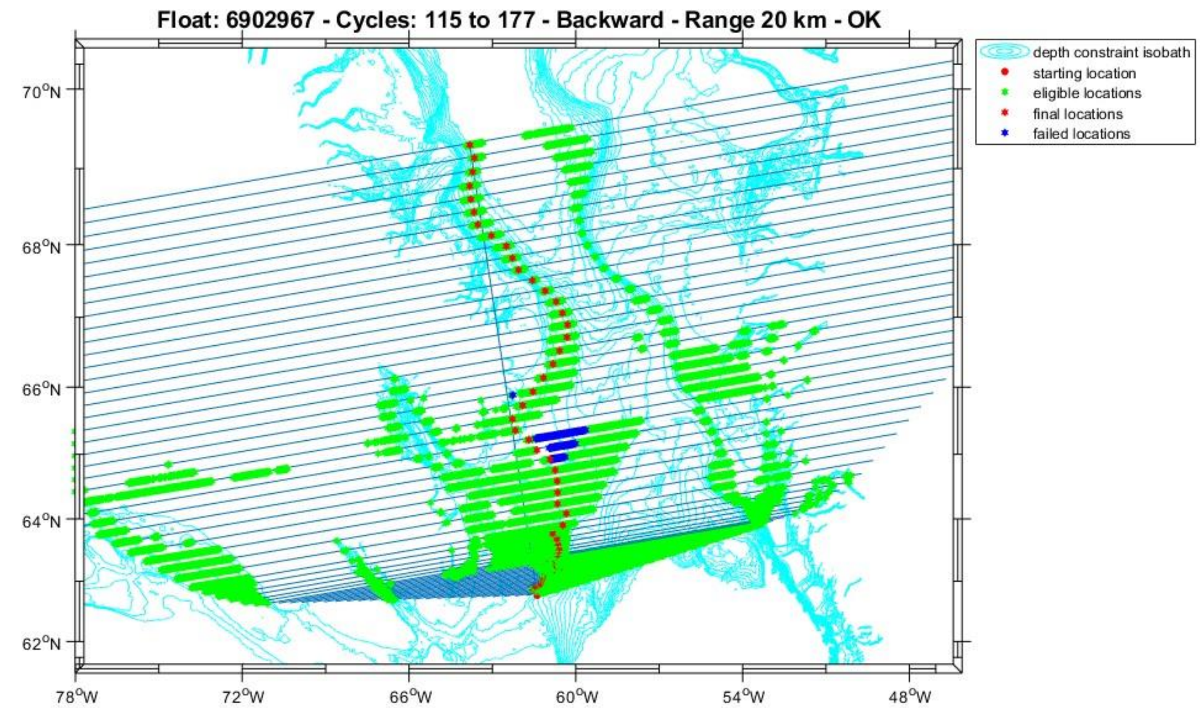


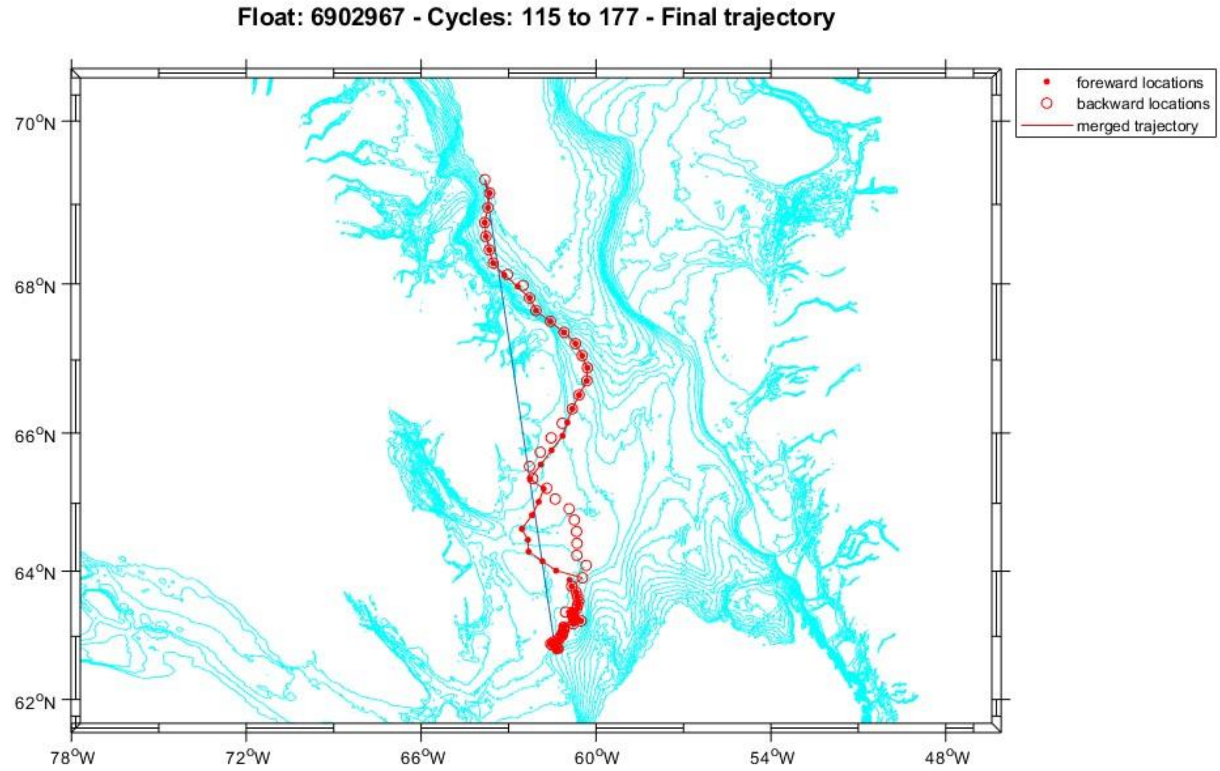
Dead ends have been encountered for cycles #109, #110 and #115 to #121.



4.3. 6902967 cycles 115-177

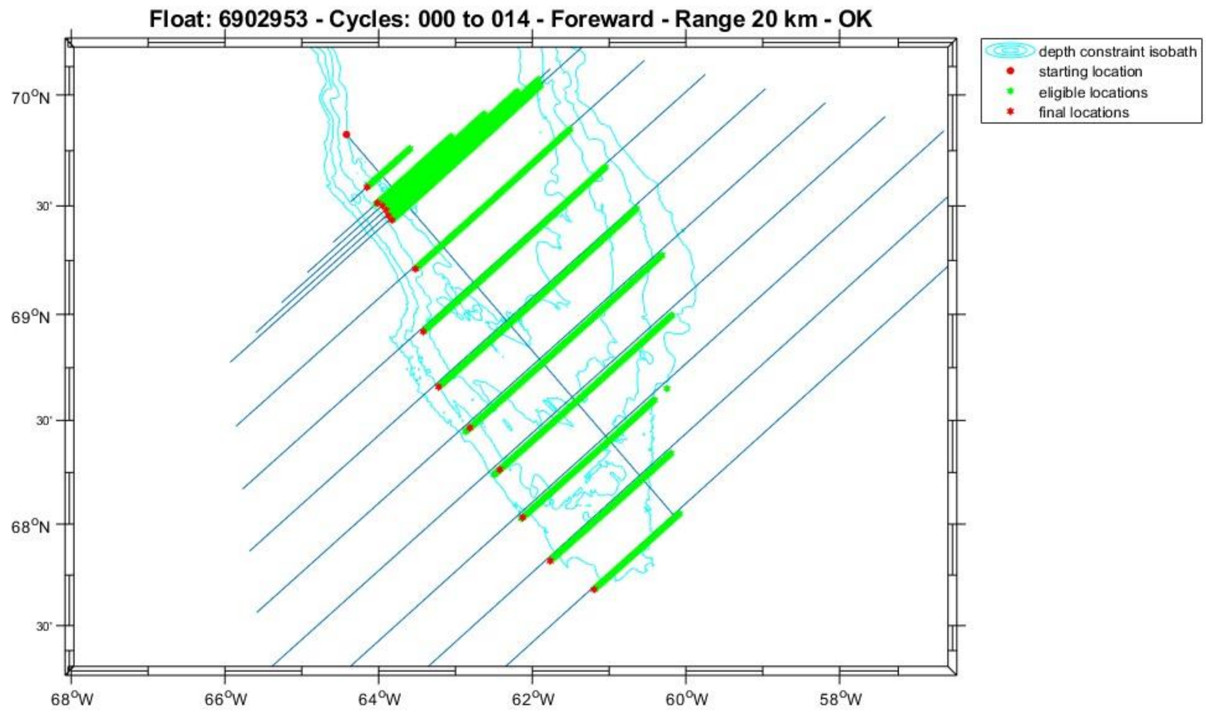


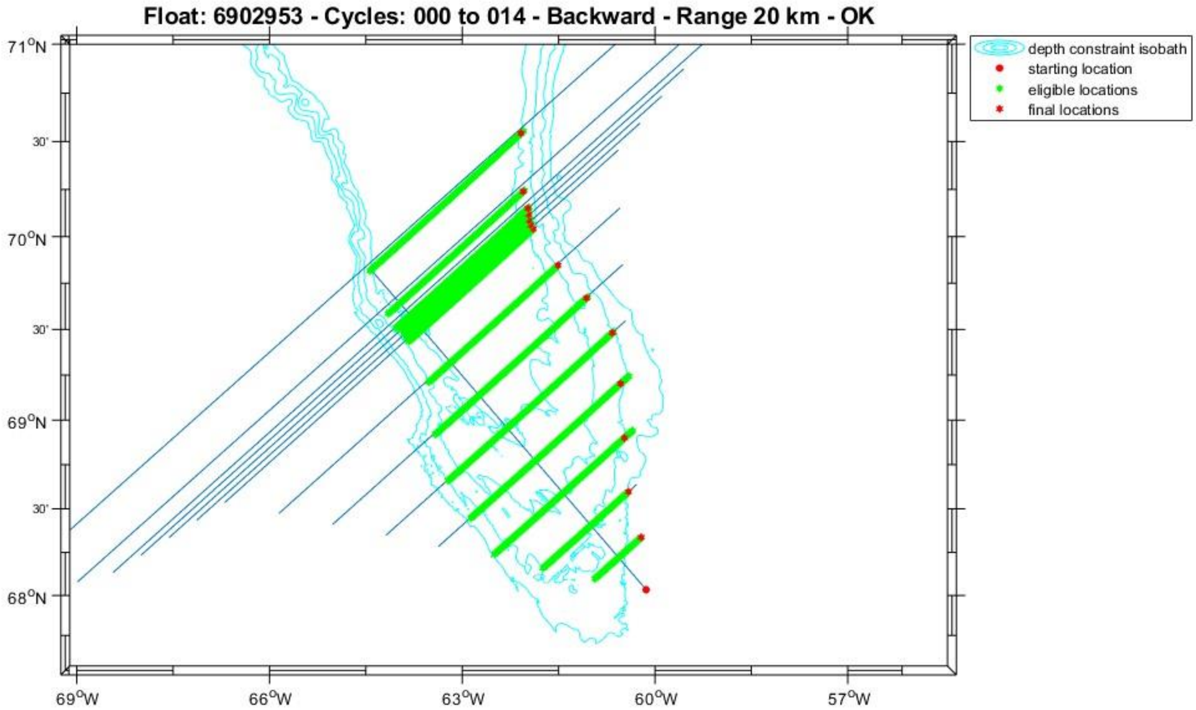


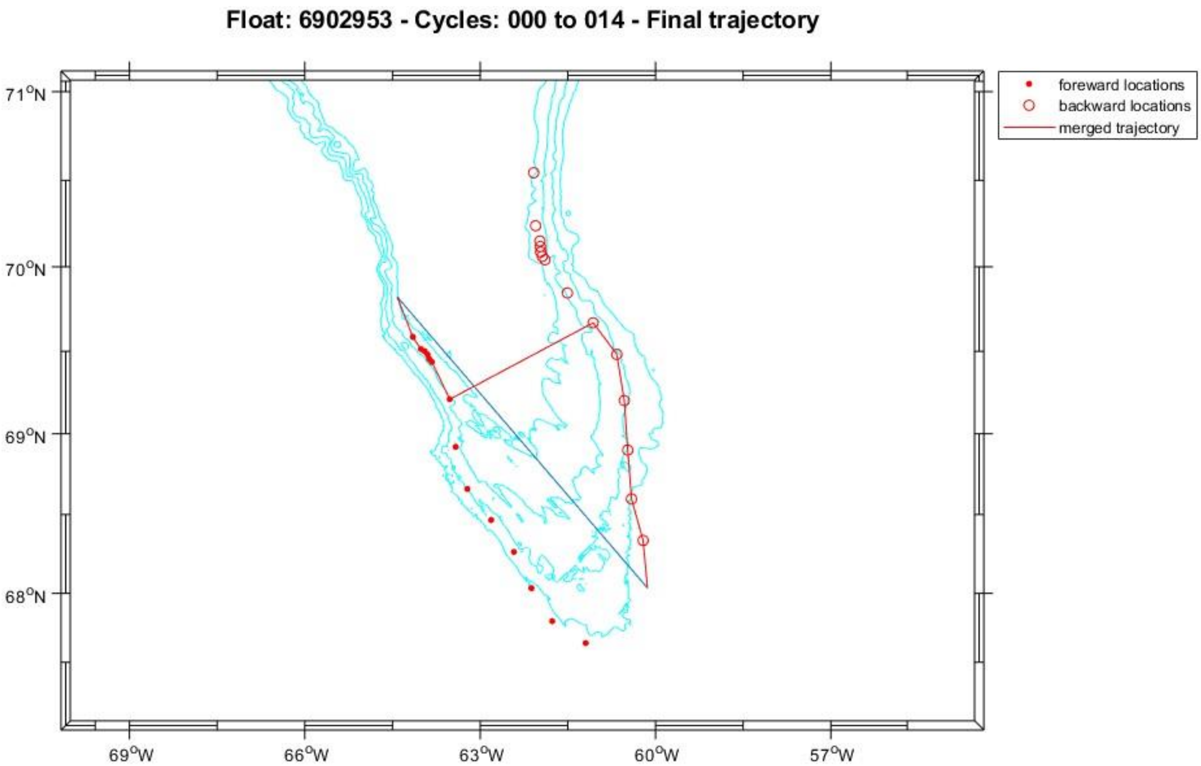


Both passes seem consistent but the DMQC operator will have to choose between 2 ways for few cycles.

4.4. 6902953 cycles 000-014







Each pass has is own set of estimated positions, merged trajectory cannot be used.