# Technical note :

## 2D MinMax Quality Control method
## Practical informations for implementation
## in CMS NRT operations at CORIOLIS

| | |
|---|---|
| Name of document | Technical Note MinMax 2D OPER |
| Date | February 13th 2024 |
| Authors | J. Gourrion & D. Leroy |
| Project | CMS In Situ TAC - MinMax |

# 1. Introduction

The MinMax method is implemented in the CORIOLIS NRT operations as an automatic Quality Control test that raises alarms for joint temperature and salinity observations lying outside of a predefined validity domain.
Presently, for the CMS In Situ TAC NRT operations, the dissemination of the data corresponding to the detected observations is interrupted.
In a second step, an operator visualizes the corresponding subset of data and makes a decision on the alarm veracity. The corresponding data are assigned a quality flag in agreement with the NRT operator decision, and finally disseminated.

Since 2019, a one-dimensional version of the MinMax approach has been running at CORIOLIS. Temperature and Salinity data are controlled separately, assuming that the data are not correlated. In this case, the test implementation consists in testing a double inequality involving the observed values on the one side, and reference minimum and maximum values on the other side. The reference values are estimated from an historical dataset and provided 1) independently of time and 2) over a discrete horizontal and vertical mesh.

During 2023, POKaPOK has been working on a two-dimensional version of the MinMax approach. The method now takes advantage of a potential correlation between the Temperature and Salinity observations to improve the method performance and robustness. Compared with the 1-D test currently running in operations, **the formalism has been generalized : the implementation consists in testing whether the couple of observed temperature and salinity values lies inside a reference polygon; this generalized approach allows running the 1D or any 2D configuration with a single program, only the reference fields being different**. The polygons are provided 1) independently of time and 2) over a discrete horizontal and vertical mesh.

With this formalism, the one-dimensional version of the method is the particular case where the polygon is rectangular i.e. 4 vertices and sides aligned with the T/S axes.
In a simplified 2-D version, the polygon is still rectangular i.e. 4 vertices but its sides are now aligned with the two principal axes defined by the ratio of S to T variances.
In the full 2-D configuration, the polygon is defined as the convex hull surrounding the local distribution of historical values.
A future N-dimensional approach will lie on the same implementation where the 2-D hull is replaced by a N-D one.

In the present technical note, we describe the 2d version functioning and provide the necessary inputs for operational implementation at CORIOLIS.

# 2. Reference File Description

The polygon coordinates are provided in a NetCDF file for each individual grid cell and depth layer. The file dimensions are the following ones:

- depth: number of depth layers - 135
- Depthvector: number of depth layer bounds - depth + 1 = 136
- Index: number of ISEA-4H6 grid cells - 40962

- Num_vertices_max: maximum number of polygon vertices (first one repeated at end)

The variables are the following ones:

- Depth (dims = depthvector) - list of pressures at the vertical layer bounds
- Index (dims = index) - list of ISEA 4H6 grid cell indices
- TEMP_POLY (dims = [index, depth, num_vertices_max]) - Polygon temperature coordinates
- PSAL_POLY (dims = [index, depth, num_vertices_max]) - Polygon salinity coordinates
- TEMP_MED (dims = [index, depth]) - Temperature coordinate of polygon center
- PSAL_MED (dims = [index, depth]) - Salinity coordinate of polygon center

The polygons are defined with a variable number of vertices, and the first vertex coordinates are repeated in last position in order to close the contour - something which may be necessary either for detection or visualization.
As the coordinates are stored into fixed-size matrices, the lines corresponding to polygons with a number of vertices Num_vertices lower than Num_vertices_max are filled with NaNs for vertex numbers larger than Num_vertices.
NaN values are also used for grid cells and depth layers for which the polygon is undefined.

# 3. Test description

## 3.1. Input data selection

The selection criteria for input data is identical to the current 1-D test and summarized below (from the informations available at POKaPOK) :

| | **Condition** |
|---|---|
| Profile type (ARGO case) | All (primary + others) |
| Position QC | POSITION_QC different from (0, 3, 4, 6, 9) |
| Date QC | JULD_QC different from (0, 3, 4, 6, 9) |
| Vertical variable (Z) | By priority order: PRES_ADJUSTED / PRES / DEPH |
| QC on vertical variable (Z) | {Z}_QC different from (0, 3, 4, 6, 9) |
| Variable to be tested (PARAM{k}) | By priority order: {PARAM}_ADJUSTED / {PARAM} |
| Measurement QC | QC different from (0, 3, 4, 6, 9) |
| Criterion for alert raising | Whether the point defined by (PARAM1 value, PARAM2_value) lies outside of the validity domain defined by the polygon. A point lying on the domain contour i.e. on the polygon sides is considered as valid and should not raise an alert. |

### 3.2. Test implementation

For each couple of measurements that went through the selection criteria, (see above section) :

- Determine the grid cell index corresponding to the measurement horizontal location
- Determine the depth layer corresponding to the measurement vertical location
- If a polygon for that grid cell and depth layer is defined, read the corresponding polygon coordinates
- Test if the point defined by (PARAM1_value, PARAM2_value) is located strictly outside the polygon.
- If the test is successful, raise an alert

N. B. : Under Matlab, Python or R, the function “InPolygon” can be used for the detection.

## 4. Polygon update by the NRT operator

As for the current 1-D test, the NRT operator might observe obvious imperfections in the polygon coordinates that raise erroneous detections. In such a case, he/she should be in capacity to modify some polygon coordinates through inclusion of the observations responsible for the erroneous detections.

If, by the end of evaluation period, the 2D version is retained for operations update, the MATLAB scripts historically developed at SISMER and currently used by the NRT operator shall be updated too.

## 5. Rationale for MinMax operational version update

Here we present quantitative results in terms of alert counts in order to provide estimates of detection improvement to be expected with an update to a 2D version of the MinMax QC test.

Ideally, such a prediction of good and bad alert counts should intend to mimic NRT conditions offline.
Unfortunately, the present operational configuration prevents from doing such simulations:

1. The CMS dataflow is permanently overwritten; any resubmission of the same data does erase the history. An incremental archive could be a solution, but …
2. The MinMax test runs on the CORIOLIS database, before CMS dissemination; rather than disseminating the profiles in alert with a flag suggesting its doubtfulness, it has been decided to suspend its dissemination into CMS: there is no guarantee that data, even from an incremental archive, are accessible in the state for which the alert has been raised.

The only reliable simulation can be done online, and SISMER will be in capacity to perform it during the march 2024 test under operational conditions.

Here, we provide numbers estimated from alerts raised on the CMS PR_PF history dataset where only PRIMARY profiles are considered.

The quality of the data flags may be far better than under NRT conditions, so bad alerts numbers are not illustrative of NRT conditions. Nevertheless, numbers gathered under such conditions allow a relative analysis from one version of the MinMax test to the other, something that allows a partial description of the future improvement. Results are presented in Figures 1 and 2.

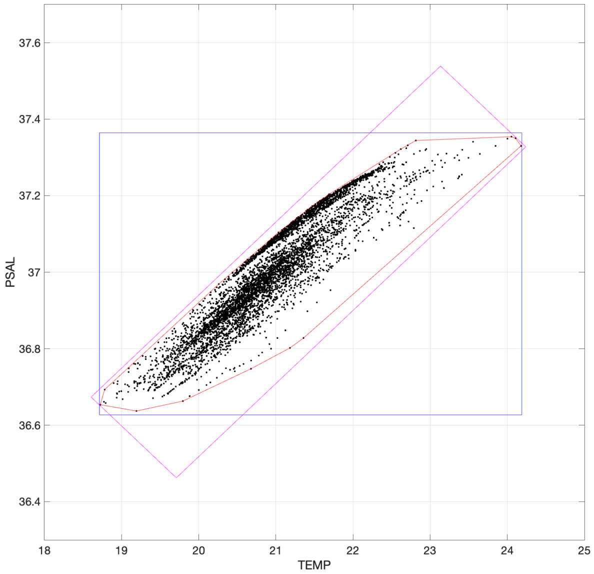

Figure 1: visualization of validity domains in the T/S space as defined for the 1D (blue), 2D-rectangle (pink) and 2D (red) versions of the MinMax test. Here, the validity domains correspond to the case without widening.

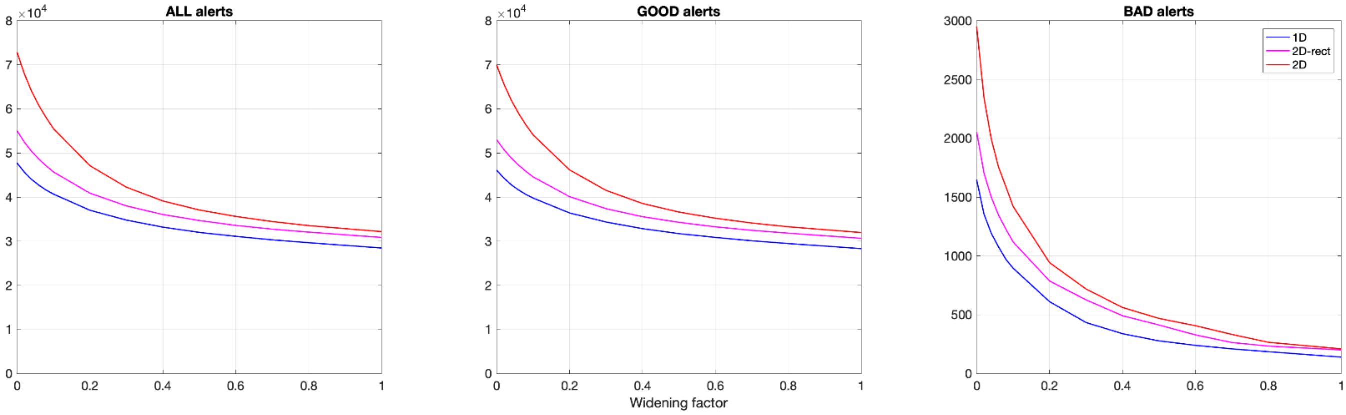

Figure 2: comparing 1D (blue), 2D-rectangle (pink) and 2D (red) versions of the MinMax test through evaluation with the CMS PR_PF History dataset. Number of profile alerts are presented as a function of the widening factor (MinMax parameter). All alerts (left), good ones (middle) and bad ones (right).

From Figure 1, we understand that the proposed updates from the 1D version act through a refined adjustment of the validity domain to the cloud of historical observations, which should allow an increased performance or detection capability.
This is confirmed in Figure 2: the simplified and full 2D versions provide more alerts: the detection capability is increased.
Nevertheless this does not tell anything about the robustness: increased number of good alerts is accompanied by an increased number of bad alerts.

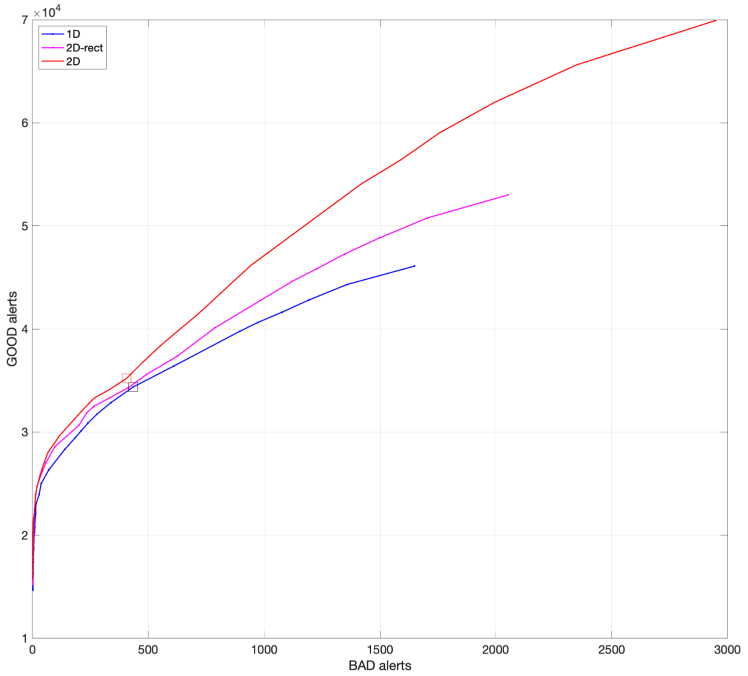

Figure 3: comparing 1D (blue), 2D-rectangle (pink) and 2D (red) versions of the MinMax test through evaluation with the CMS PR_PF History dataset. Number of good profile alerts are presented against the number of bad ones. The optimal point is the upper left corner. The grey square locates the present 1D operational configuration while the red square located the proposed configuration for the 2D update (widening factor = 0.6).

Figure 3 evidences that the 2D implementations provide better ratios of good to bad alerts (remembering that the optimal point is the upper left corner); for a given amount of false alerts, the amount of good ones is increased and for similar amount of good alerts the number of false ones is reduced.

Following the results of Figure 3, we propose to use a widening value of 0.6 (see the red square in Figure 3); with such a value, we expect to roughly maintain the operator workload, while rather reducing (increasing) the amount of false (good) alerts.