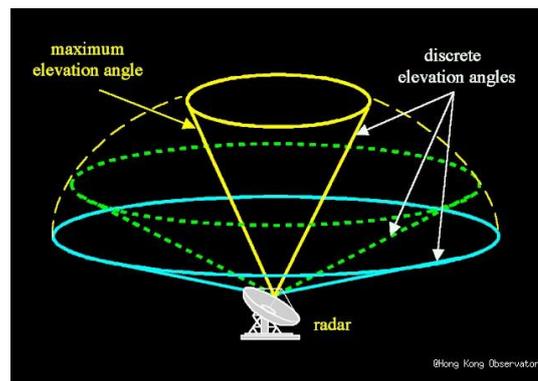


Radar data pre-processing



report from RC LACE stay in Ljubljana, 6/03/ - 13/03/2018
Alena Trojáková, CHMI, alena.trojakova@chmi.cz
in collaboration with Benedikt Strajnar and Peter Smerkol, ARSO

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 3 |
| 2 | Radar data pre-processing | 3 |
| 2.1 | Brief overview of OPERA data usage within RC LACE | 3 |
| 2.2 | Proposal for further work | 4 |
| 3 | Summary | 5 |
| A | Experiments | 8 |
| B | ODBVIEWER | 9 |
| C | HDF Viewer | 9 |

1 Introduction

Observation from meteorological radars are considered as very important for the initialization of mesoscale Numerical Weather Prediction (NWP) models. The assimilation methodology developed by Wattrelot *et al.* (2014) showed capabilities to improve the analysis and short-range weather forecast of AROME France. For many years there has been an interest in radar data assimilation (DA) in RC LACE countries, but advancements were rather slow. Coordinated efforts are planned for 2018 (Mile, 2018) to enhance the progress on radar DA implementation. This report summarizes the work done during a short stay dedicated to the review of available technical solutions for radar data pre-processing and analysis of essential parameters needed for assimilation of radar data.

2 Radar data pre-processing

The first implementation of radar data processing within ALADIN NWP used a specific MF-BUFR format developed at Meteo France. For an extended use of radar data from abroad the data from OPERA (Operational Programme for the Exchange of Weather Radar Information) have been progressively used. The OPERA data can be available either in OPERA Data Information Model (ODIM) BUFR (not a standard BUFR format) or ODIM HDF5 format. The CONRAD tool (Gronsløth *et al.*, 2012) developed by HIRLAM consortia was originally employed for the conversion from OPERA (and/or raw) radar data to the MF-BUFR format. Recently the ODIM HDF5 format has been promoted and at the time of writing there exist two BATOR solutions for processing ODIM HDF5 data. The first approach was developed by the HIRLAM and it was adopted by Florian Meier in Austria for CY40T1, see Figure 1 for an illustration. The second option has been developed by Meteo France and is available from CY43T2 (Guillaume, 2017).

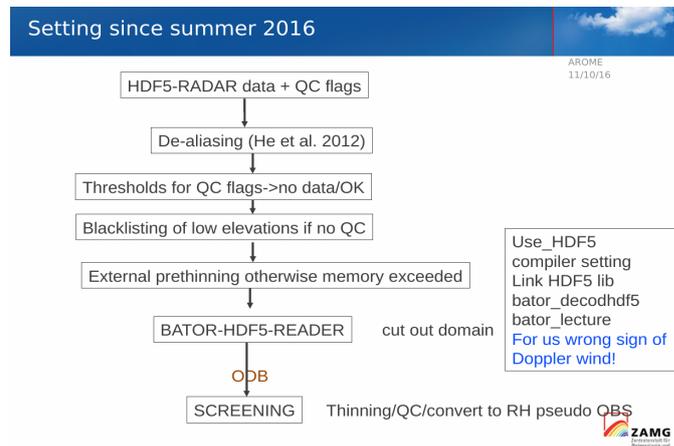


Figure 1: Radar data pre-processing at ZAMG, F. Meier: DAWD 2016 radar status presentation.

2.1 Brief overview of OPERA data usage within RC LACE

Previous experimentation's (Meier and Strajnar (2017), Kovacic (2018)) showed that there are many issues with ODIM HDF5 data usage, e.g.

- inconsistencies in radar meta-data (missing WMO id, wavelength, pulse width, antenna gain, sensitivity);

- German and French data are coded in real values with offset 0 instead of the 0-255 integers, "nodata" and "undetected" values are out of the 0-255 range;
- one elevation data from several countries (German, Czech Republic, Slovakia) are spread to several datasets instead of having one dataset containing several moments /data1 /data2.

The missing meta-data has been added and specific data encoding issues has been treated by the special tool (Python prepopera.py) originally developed by HIRLAM colleagues. Furthermore, many specific issues are handled within the HIRLAM version of BATOR routine of the radar data processing. This approach does not seem optimal for further developments. Therefore a following proposal was drafted for further stays organized in the RC LACE work-plan 2018.

2.2 Proposal for further work

We propose to avoid a hard-coding and to use a modular approach with namelist kind of flexibility. It seems highly impractical to develop and maintain hard-coded specifics for several tenths of radar sites available in RC LACE domain of interest. Moreover, it would be beneficial to group/merge related aspects on a single place (module/subroutine). We can hopefully expect improvements in the ODIM HDF5 data content and the modular design can simplify future maintenance and developments. The idea is not to rewrite existing tools from scratch, but to reuse already existing tools (prepopera.py, BATOR HDF5 routines, etc) as much as possible. The aim is to provide a modular framework for further developments, see Figure 2.

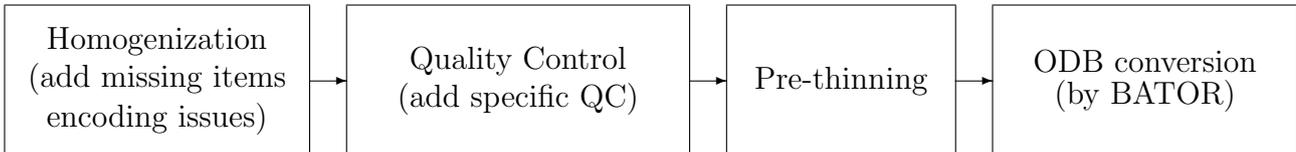


Figure 2: The scheme of the radar data pre-processing.

There are several type of issues to be handled by the data pre-processing: missing or specifically encoded meta-data and/or data, Quality Control (QC) issues and pre-thinning to avoid crashes in the 3DVAR minimization.

Data issues can be eliminated by a homogenization. There has been work on this subject already by Meier and Strajnar (2017) and Kovacic (2018). To continue this part we propose to elaborate a tool for an automatic checkup of missing parameters and/or data issues. A very very first draft of the homogenization tool was prepared by Peter Smerkol and Anja Fettich, but further work is needed. An overview of relevant parameters needed for ODB conversion by BATOR and subsequent data assimilation is lacking. The first code analysis started during this stay. Figure 3 shows the schema of ODB radar related tables and Table 1 contains ODB column names, IFS pointers and brief descriptions.

An inspection of available technical solutions for radar data conversion to ODB started. The BATOR CY43T1 (including Meteo France version of ODIM HDF5 processing) was back-phased to CY40T1 installed at ARSO to facilitate further evaluation of radar data processing. Due to lack of time only a limited number of tests were performed. Tests were intended as a kind of feasibility study and no detailed evaluation was done, for more details see Appendix A.

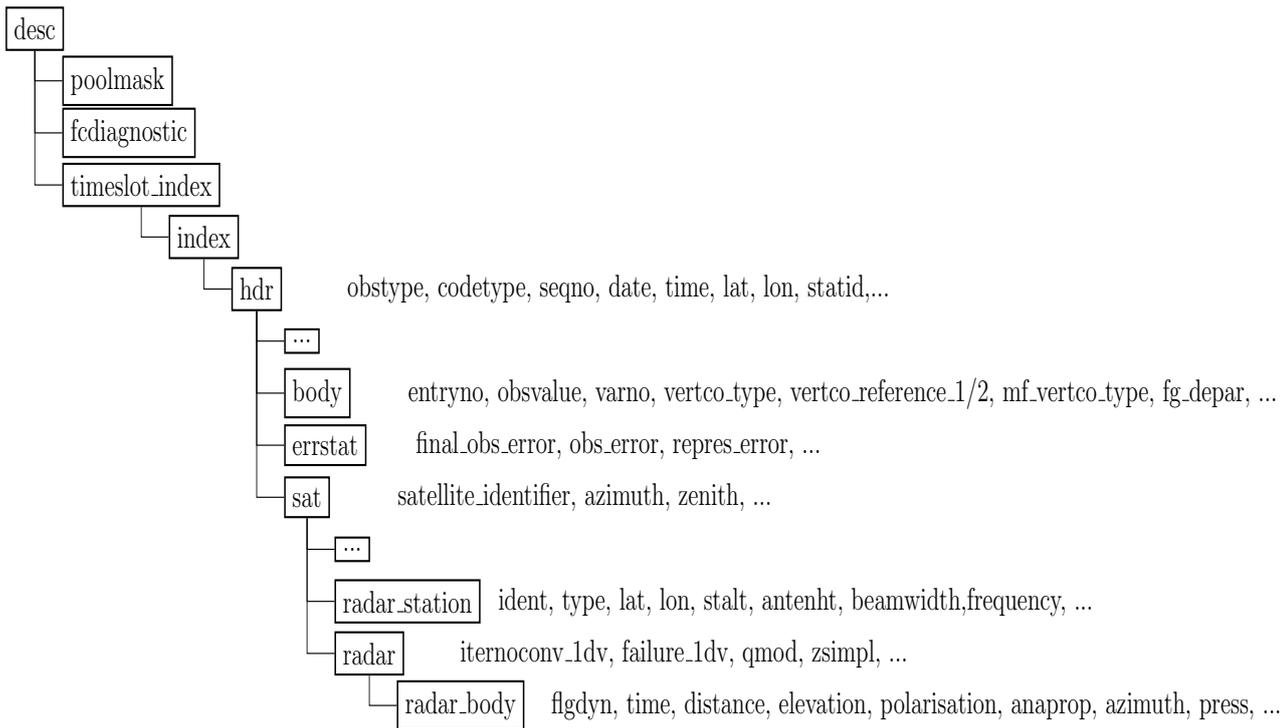


Figure 3: Schema of the ODB tables related to radar observations.

3 Summary

Previous section provided details on items tackled during the stay and here follows a brief summary and an outlook for the future work.

Radar data homogenization is essential for radar DA. We propose to elaborate a tool for an automatic checkup of missing parameters and/or data issues with a namelist kind of flexibility. Furthermore, it would be beneficial to group/merge related aspects on a single place (module/subroutine). The ODIM HDF5 reading available in BATOR CY43T2 seems more promising for further developments, but detailed cross-check is still needed (e.g. pre-thinning, QC handling and any other issues currently available in preopera.py).

The stay enabled valuable discussions, but due to a lack of time several items were not finalized, e.g. table of ODB parameters needed to be completed and QC indices processing need a deeper understanding. The report will be updated with relevant details as soon as available.

Acknowledgment

The author wishes to thank Benedikt Strajnar for the support and cooperation and all NWP group for their warm welcome and hospitality. This stay was made possible by the financial support of the LACE (Limited Area model for Central Europe) project.

| ODB column name | IFS column pointer | Description (BATOR variable) |
|--------------------------|--------------------------------|---|
| seqno@hdr | MDBONM | observation sequence number |
| obstype@hdr | MDBOTP | observation type (INT(ZENT(jobs,NCIOTP))) |
| date@hdr | MDBDAT | observation date (INT(ZENT(jobs,NCIDAT))) |
| time@hdr | MDBETM | observation time (INT(ZENT(jobs,NCIETM))) |
| statid@hdr | MDBSID | radar identifier (TREF_RADAR(j)%ident) |
| lon@hdr | MDBLON | (ZENT(jobs,NCILON) * RADIANS) |
| lat@hdr | MDBLAT | (ZENT(jobs,NCILAT) * RADIANS) |
| stalt@hdr | MDBALT | (ZENT(jobs,NCIALT)) |
| obstype@hdr | MDBOTP | 13 (INT(ZENT(jobs,NCIOTP))) |
| codetype@hdr | MDB.CODETYPE_AT_HDR | 2 (ZDATALOC(jobs,2)) |
| varno@body | MDBVNM | 192/195/29 (iovnm) |
| obsvalue@body | MDBOTP | 13 (INT(ZENT(jobs,NCIOTP))) |
| vertco_type@body | MDBRBVC | |
| vertco_reference_1@body | MDBPPP | |
| vertco_reference_2@body | MDBPRL | |
| flgdyn@radar_body | MDB.FLGDYN_AT_RADAR_BODY | dynamic quality flag (ZDATAWAG(jw0+jw,3)) |
| distance@radar_body | MDB.DISTANCE_AT_RADAR_BODY | distance o the radar [m] (ZDATAWAG(jw0+jw,4)) |
| elevation@radar_body | MDB.ELEVATION_AT_RADAR_BODY | elevation (ZDATAWAG(jw0+jw,5)) |
| polarisation@radar_body | MDB.POLARISATION_AT_RADAR_BODY | polarity (ZDATAWAG(jw0+jw,6)) |
| @radar_body | MDB.ANAPROP_AT_RADAR_BODY | ??? (ZDATAWAG(jw0+jw,7)) |
| time@radar_body | MDB.TIME_AT_RADAR_BODY | ??? (ZDATAWAG(jw0+jw,8)) |
| azimuth@radar_body | MDB.AZIMUTH_AT_RADAR_BODY | azimuth (of the measurement ?) (ZDATAWAG(jw0+jw,9)) |
| press@radar_body | MDB.PRESS_AT_RADAR_BODY | ??? |
| satellite_identifier@sat | MDB.SATID_AT_SAT | radar identifier (TREF_RADAR(j)%ident) |
| ident@radar_station | MDB.IDENT_AT_RADAR_STATION | radar identifier (TREF_RADAR(j)%ident) |
| type@radar_station | MDB.TYPE_AT_RADAR_STATION | radar type (TREF_RADAR(j)%type) |
| lat@radar_station | MDB.LAT_AT_RADAR_STATION | latitude of the radar (TREF_RADAR(j)%lat) |
| lon@radar_station | MDB.LON_AT_RADAR_STATION | longitude of the radar (TREF_RADAR(j)%lon) |
| stalt@radar_station | MDB.STALT_AT_RADAR_STATION | altitude of the radar (TREF_RADAR(j)%stalt) |
| antenht@radar_station | MDB.ANTENHT_AT_RADAR_STATION | height of the antenna (TREF_RADAR(j)%antenht) |
| beamwidth@radar_station | MDB.BEAMWIDTH_AT_RADAR_STATION | aperture at 3dbz (TREF_RADAR(j)%beamwidth) |
| frequency@radar_station | MDB.FREQUENCY_AT_RADAR_STATION | pulse frequency (TREF_RADAR(j)%frequency) |
| obs_error@errstat | MDBOER | (see note 1) |

Table 1: Explanation of ODB radar related items.

References

- Gronsløth M, Salomonsen M, Aspenes T, Elo C. 2012. Quality control of radar data for NWP assimilation using PRORAD and CONRAD. Technical report, MetNo, URL www.rclace.eu/File/Data_Assimilation/workshops/DAWD2012/proradconrad.pdf.
- Guillaume F. 2017. Pretraitement des données aux formats BUFR, NETCDF and HDF5 pour ARPEGE/ALADIN/AROME. Technical report, Meteo France, URL http://www.umr-cnrm.fr/gmapdoc/IMG/pdf/namel_bator_cy43.pdf.
- Kovacic T. 2018. An overview of ODIM HDF5 files from radars within Croatian ALARO NWP model domain. Technical report, RC LACE.
- Meier F, Strajnar B. 2017. Experiences from OPERA-OIFS data usage. Technical report, RC LACE, URL <http://www.rclace.eu/forum/download/file.php?id=465>.
- Mile M. 2018. RC LACE Work Plan for the area of data assimilation. Technical report, RC LACE, URL http://www.rclace.eu/File/Data_Assimilation/plans/DAplan2018_updated_v2.pdf.
- Wattrelot E, Caumont O, Mahfouf JF. 2014. Operational Implementation of the 1D+3D-Var Assimilation Method of Radar Reflectivity Data in the AROME Model.

Monthly Weather Review **142**(5): 1852–1873, doi:10.1175/MWR-D-13-00230.1, URL
<https://doi.org/10.1175/MWR-D-13-00230.1>.

A Experiments

This appendix contains details of performed experiments. All experiments were run on ARSO server **ventus** based on cy40t1_bf07 compiled using Intel compiler.

Two user packs were considered:

- **BATOR cy40t1bf7** (reference)
 - BATOR modifications for OPERA HDF5 provided by F. Meier
 - local adaptation for (HRW, ASCAT, ...)
 - can be found on **ventus:/home/bstrajnar/aldpacks/cy40t1bf7**.
- **BATOR cy40t1bf7#bator43**
 - BATOR backphased from CY43T2_bf04 (including ODIM HDF5 processing)
 - can be found on **ventus:/home/atrojakova/packs/cy40t1bf7#bator43**.

A limited number of tests were performed. Tests were intended as a kind of feasibility study of various BATOR versions and no detailed evaluation was done due to lack of time. All experiments can be found on **ventus:/home/atrojakova/exp**. Considering that tests were mostly **TECHNICAL no outputs were stored permanently!** Here follows a summary of performed tests:

- **rad - reference experiment adapted by Beni**
- **r01 - trial for single radar site "silis"**
 - technical test of the reference BATOR cy40t1bf07
- **r02 - trail for single radar "uking" (listings *.1) and "silis" (listings *.2)**
 - technical test of the BATOR cy40t1bf7#bator43
 - preopera.py NOT employed !
 - due to CY43T2 further modifications needed in param.cfg, namel_bator, bator script
- **r03 - trail for single radar with preopera.py**
 - technical test of the BATOR cy40t1bf7#bator43
 - preopera.py employed
 - **no observations selected - further investigation is needed to understand why preopera.py breaks the pre-processing**

```
** ERROR - BATOR : PrefetchHdf5() file HDF5.site1 Attribute Conventions missing.  
*** INFO - BATOR : File HDF5.site1      prefetched - NbOBS=          0 NbWag=          0  
INBOBS =          0  INBWAG =          0
```

- **r05 - trail for all OPERA radars**
 - technical test of the BATOR cy40t1bf7#bator43
 - preopera.py NOT employed
 - **BATOR runs TECHNICALLY for 142 radar sites, but screening crashes :- (**
 - **further investigation is needed**

B ODBVIEWER

A stand-alone ODB package was installed and tested on ARSO server **ventus**. It allows to run a viewer to quickly check an ODB content. The core tool (odbsql) was compiled and tested successfully. Further details and basic usage information can be found in the corresponding README files.

```
atrojakova@ventus:/home/atrojakova/opt>
odb-1.0.3_README                # installation notes/guidelines
odb-1.0.3-Source.tar.gz        # original packages
odb_api_bundle-0.15.2-Source.tar.gz

# installation directory
/home/atrojakova/lib/gnu/v4.9.1/odb-1.0.3

# test data
/home/atrojakova/Trash/example
```

Following setting have to be exported (or set to a user .bashrc).

```
# setting for odbsql (stand-alone ODB)
export ODB_ROOT=/home/atrojakova/lib/gnu/v4.9.1/odb-1.0.3
alias use_odb="module load gcc/4.9.1 && source $ODB_ROOT/bin/use_odb.sh"
```

C HDF Viewer

HDF Viewer Version 3.0 for Linux was installed and tested on ARSO server **ventus**. It allows to run a viewer to quickly check content of HDF files.

```
atrojakova@ventus:/home/atrojakova/opt>
HDFView-3.0-centos7_README      # installation notes/guidelines
HDFView-3.0-centos7.tar.gz      # original packages
```

In order to use the HDF viewer

```
# execute
/home/atrojakova/opt/HDFView-3.0-centos7/HDFView-3.0.0-Linux/HDFView/3.0.0/hdfview.sh

# or make a corresponding symbolic link in your $HOME/bin e.g.
cd $HOME/bin
ln -s ~/atrojakova/opt/HDFView-3.0-centos7/HDFView-3.0.0-Linux/HDFView/3.0.0/hdfview.sh hdfview
# execute
hdfview
```