



Investigation the BATOR cy43t2bf06 for radar DA

report from RC LACE stay in Budapest, 20180820-20180909

Michal Nestiak, SHMU

in collaboration with Maté Mester

““

Our way to higher spatial and time resolution weather forecasts must go hand in hand with improvement of weather model initial state. In last years most of the RC-LACE member states focused their efforts on a one-hour rapid update cycle (RUC) of analyse. This gives a demand for high temporal and spatial resolution of meteorological observations. Data from meteorological radars are one of these types of measurements. On this stay in Budapest we were concentrating on using radar data in OPERA Data Information Model (ODIM) HDF5 format as input for BATOR.

In my last radar DA stay in Budapest (2014) we were also using ODIM HDF5 files, but the preprocessing method (CONRAD) used that time was much more CPU consuming compared to the new tool (HOOF) and was degrading the accuracy of the radar data due to involved interpolations. The CONRAD tool was converting the HDF5 files to the MF BUFR which was only supported format for data assimilation of radar data in that days. The change of format and geometry of the data lead to results which were far from the ideal. In 2016 Meteo France expanded the BATOR functionality to read directly the HDF5 files. The main part of the development was carried by F. Guillaume. Reading of HDF5 files is concentrated in the routine `bator_decodhdf5_mod.F90`. The new approach of handling radar data assimilation inside the ALADIN source code reinitiated the efforts to assimilate the radar data in RC-LACE countries. In 2018 the common radar works were starting from beginning. The idea was first understand the source code before switching on the assimilation to avoid problems from not proper validation of the code or not properly prepared data for the assimilation. By set of stay reports we share the knowledge between RC-LACE members:

Ljubljana - Alena Trojáková, 2018/03: [Radar data pre-processing](#)

Ljubljana - Máté Mester, 2018/03: [Towards homogenization of OPERA radar data](#)

Bratislava - Alena Trojáková, 2018/05: [Evaluation of OPERA data quality for NWP DA purposes](#)

Budapest - Michal Nestiak, 2018/08: [Radar DA and BATOR](#)

Installation of the back phased version of BATOR cy43t2_bf06 on cy40 at OMSZ

One of the outputs of A. Trojáková stay in Bratislava was technical run of Bator on cycle 43t2_bf06, she also provided backphased version on cycle 40t1.

As the first step at OMSZ we installed back phased version of BATOR cy43t2_bf06.

OMSZ:wfma: /home/mnestiak/pack/op/bin/BATOR

OMSZ:wfma: /home/mnestiak/wrk/cy40/bator_c432c40/scr/

File `ics_bator` must include `grib_api`, `hdf5` and `netcdf` library

```
# LOW-LEVEL LIBRARIES (from top to bottom) :
cat <<end_of_bator_sys> $GMKWRKDIR/.bator_sys
-L/home/aladin/lib/auxlibs.2.5-fPIC -lrgb -lbufr -lgribex
-L/opt/netcdf/lib -lnetcdff -lnetcdf
-L/opt/grib_api-1.17.0/lib -lgrib_api_f90 -lgrib_api
-L/home/aladin/lib/auxlibs.2.5-fPIC -libmdummy
-L/home/mnestiak/lib/hdf5-1.8.16/lib -lhdf5_fortran -lhdf5
```

end_of_bator_sys

I would like to be sure that we use the latest HDF5 library and testing required switches for BATOR. I installed version 1.8.16 (<https://www.hdfgroup.org/downloads/hdf5/>)

OMSZ:wfma:/home/mnestiak/lib/hdf5-1.8.16

OMSZ:wfma:07feb416b3f9c3dc82595bbe0f078669 /home/mnestiak/wrk/hdf5lib/hdf5-1.10.1.tar

```
./configure --prefix=/home/mnestiak/lib/hdf5-1.8.16 --enable-fortran --enable-fortran2003 --enable-parallel --enable-instrument
--enable-strict-format-checks FCFLAGS="-fPIC -O3 -g -xHost -ip -no-prec-div -fno-alias -fp-model precise" CFLAGS="-fPIC
-fPIC -O3 -g -xHost -ip -no-prec-div -fno-alias -fp-model precise" FC=mpiifort CC=mpiicc
```

make

make check

make install

NOTE: In 2018/09 we were also testing version 1.10.1, but nothing change for us.

After preliminary testing we find that in automated generated file *.gmkfile/MPIIFORT.WFMA* is missing

```
# External include pathes (path1:path2:...)
```

```
# -----
```

```
INCLUDEPATH = /opt/netcdf/include:/opt/grib_api-1.17.0/include:/usr/include/libxml2:/home/mnestiak/lib/hdf5-1.8.16/include
```

NOTE:

Also we must be careful in source code of ALADIN where is HDF5 library controled by `h5get_libversion_f` in `bator_decodhdf5_mod.F90` which control installed version of HDF5 library when we switch to next major version.

```
call h5get_libversion_f(MajNum,MinNum,RelNum,Error)
```

```
if (Error == 0 .and. MajNum >=1 .and. ((MinNum == 8 .and. RelNum > 7) .or. MinNum >8)) then
```

```
if (Error == 0 .and. MajNum >=1 .and. MinNum > 7 .and. RelNum > 7) then
```

After this changes and cleaning of obsolete system environmental variables the first technical run of radar DA in OMSZ was performed.

ODBVIEW

Next step was installation of ODB viewer (odbview) on OMSZ:wfma to be able to control BATOR generated ODB content.

```
OMSZ:wfma:/vol3/mnestiak/tools/bin/  
0bf945e203bab3505efad598a2d52a5e /home/mnestiak/wrk/odb/odb-1.0.2-Source.tar.gz
```

Usage of odbviewer is very simple. We must just enter ECMA directory and run *use_odb.sh* from there and then odbsql selects.

```
./vol3/mnestiak/tools/bin/use_odb.sh  
odbsql -q "select statid@hdr,llddegrees(lon@hdr),llddegrees(lat@hdr),stalt@hdr,varno@body,obsvalue@body,report_status@hdr  
from hdr,body;"
```

Getting access to OPERA Internet File System (OIFS)

We contacted the Daniel Idziorek (Daniel.Idziorek@meteo.fr , cmr_odc@meteo.fr) to ask for access to OIFS from Hungary. We received many valuable advices from him. One of them was that agreed policy is only one account per country. He already checked that nobody from Hungary asked in that time for OIFS account creation. We ask OPERA delegate for Hungary Istvan Sebok and Roland Steib for creation and grant of access to OIFS.

Installation of Python 2.7

We installed python 2.7 under local user on wfma because it is required for HOOF and some visualisation scripts. After successful tests we asked local system administrator Lowinger Endre to install python2.7 on OMSZ:wfma.

```
OMSZ:wfma:/opt/rh/python27/root/usr/bin/python2.7
```

HOOF

OPERA hdf5 radar preprocessing program HOOF was prepared by Slovenian colleagues Peter Smerkol and Benedikt Strajnar was prepared OPERA hdf5 radar transforming program HOOF. It is written in python (required version 2.7). During my stay we reported a few bugs and consulted any appeared problem with development team of HOOF.

We start with version of HOOF from RC LACE forum

```
OMSZ:wfma:/home/mnestiak/wrk/cy40/bator_c43t2bf06c40/tools/RH_rlaceforum20180825.py
3e6ea24dbdb3b3c0dc56ffae822cb35d RH_rlaceforum20180825.py
```

Final version from September was

```
OMSZ:wfma:/home/mnestiak/wrk/cy40/bator_c43t2bf06c40/tools/RH.py
3f7182b887e22518612fcb59570cbbf9 RH.py
```

It is necessary to have the right version of python libraries in LD_LIBRARY_PATH. We were exporting LD_LIBRARY_PATH before each run of HOOF to avoid possible problems see the example.

```
OMSZ:wfma:/home/mnestiak/wrk/cy40/bator_c43t2bf06c40/tools/run_RH
```

```
export LD_LIBRARY_PATH=.. /opt/rh/python27/root/usr/lib64: ...
/opt/rh/python27/root/usr/bin/python2.7 /vol3/mnestiak/wrk/cy40/bator_c432c40/tools/RH.py RHNamelist.nam
/home/mnestiak/wrk/cy40/bator_c432c40/tools/in /home/mnestiak/wrk/cy40/bator_c432c40/tools/out
```

In the namelist of HOOF, it is important to setup right extensions of input files. When you try to use some locally produced HDF5 in ODIM format with different extension than the extension of files from OIFS, you will reach problems. In our case it was QRAD input to INCA3 with .hdf extension :

```
[FileExtensions]
{.h5 .hdf}
```

In the namelist you can also specify which quantities will be in the output files so e.g. when you don't need to have reflectivity or wind in your experiment. It is also very useful when TH is missing to temporarily replace TH by DBZ. But beware impact of this is unknown and must be carefully investigated in near future.

```
[SavedQuantities]
DBZ = {DBZ DBZH}
TH = {TH} #or DBZH
VRAD = {VRAD VRADH}
```

Section [DbzQualityGroups] describes quality indexes fields.

All data storage, which will be exported, is set up in [RadarAttributes common] section.

There should be possibility to change set up of attributes for individual radar in section [RadarAttributes {CCNOD}] (CC-country NOD-3 radar character from OPERA DB) but so far we were not very successful when we tried to use this feature for hunap radar. We tried to change '/what/source' because in Hungarian source is some additional text (Full name of

the radar with local Hungarian characters). We were afraid that the hungarian characters might cause problems in the BATOR processing. The issue with changing radar attributes should be solved in newer versions of HOOF, not yet tested.

Experiments

The environment and data was prepared so there was right time for experiments. First testing candidate for “new” BATOR (cy43t2bf06) was ODIM HDF5 from Lisca radar.

```
md5sum: 3761b7b647b6c0cd1d53fcbcb79dd2c silis
source: "WMO:14024,RAD:SI41,PLC:Lisca,NOD:silis"
date: 20180306
time: 004500
```

Decoding of HDF5 files inside BATOR is done in `bator_decodhdf5_mod.F90`
Each input file must include:

```
TYPE Fichier
  character(len=5)           :: Nod   = '?'
  integer(kind=jpim),dimension(6) :: DateOpt = nabsi
  real(kind=jprb)           :: Lat   = rabsi
  real(kind=jprb)           :: Lon   = rabsi
  real(kind=jprb)           :: Alt   = rabsi
  real(kind=jprb)           :: BeamWidth = 1._jprb
  integer(kind=jpim)        :: NPoints = nabsi
  integer(kind=jpim)        :: NRayons = nabsi
  TYPE(GAttributes)         :: Attrib
  TYPE(FilteredEAngle),dimension(:),pointer :: FinalElev
END TYPE Fichier

TYPE(Fichier)               :: Radar
```

```
Radar%Nod = String1Buf(1)(index(String1Buf(1),'NOD')+4:index(String1Buf(1),'NOD')+8)
```

If the NOD name is not set in the BATOR namelist (`HODIM%NodeNames`) this radar data are not processed! All attributes of “TYPE Fichier” are necessary to continue filling of data to ODB.

For testing of all input and output data we use external h5 utilities like

```
h5ls -fv
```

```
In h5get_libversion_f
```

Next phase of stay was devoted to understand the reading of individual rays data. For attributes of individual rays BATOR uses TYPE GAttributes which is subtype of TYPE GrpData and GrpQual.

Join of these two parts together is TYPE GrpDataset. All GrpDataset-s are capsuled under FullDatasetList as is see in following part of the source code.

```
TYPE GAttributes
  character(len=6)           :: StartTime = '?'
  character(len=8)           :: StartDate = '?'
  character(len=LabelLength) :: Quantity = '?'
  real(kind=jprb)            :: Gain      = rabsi
  real(kind=jprb)            :: Offset    = rabsi
  real(kind=jprb)            :: NoData    = rabsi
  real(kind=jprb)            :: NoDetect  = rabsi
  real(kind=jprb)            :: Elangle   = rabsi
  integer(kind=jpim)         :: NBins     = nabsi
  integer(kind=jpim)         :: NRays     = nabsi
  real(kind=jprb)            :: RScale    = rabsi
  real(kind=jprb)            :: Rstart    = rabsi
  character(len=LabelLength) :: Task      = '?'
END TYPE GAttributes

TYPE GrpData
  character(len=LabelLength*2+1) :: Label
  TYPE(GAttributes)               :: Attrib
  real(kind=jprb),dimension(:,,:),pointer :: Values => NULL()
END TYPE GrpData

TYPE GrpQual
  character(len=LabelLength*2+1) :: Label
  TYPE(GAttributes)               :: Attrib
  real(kind=jprb),dimension(:,,:),pointer :: Values => NULL()
END TYPE GrpQual

TYPE GrpDataset
  character(len=LabelLength)       :: Label
  TYPE(GrpData),dimension(:,),pointer :: GData => NULL()
  TYPE(GrpQual),dimension(:,),pointer :: GQuality => NULL()
  TYPE(GAttributes)                 :: Attrib
END TYPE GrpDataset

TYPE(GrpDataset),dimension(:,),allocatable :: FullDatasetList
```

The next step was devoted to understand how BATOR selects rays and elevations, but shortage of time didn't allowed us to finished this.

```

TYPE FilteredElangle
  real(kind=jprb)           :: Elangle = rabsi
  TYPE(GrpData),pointer    :: DBZH => NULL()
  integer(kind=jpib)       :: DbZHDeltaTime = 86400
  TYPE(GrpData),pointer    :: TH  => NULL()
  integer(kind=jpib)       :: THDeltaTime = 86400
  TYPE(GrpData),pointer    :: VRAD => NULL()
  integer(kind=jpib)       :: VRadDeltaTime = 86400
  TYPE(GrpQual),pointer    :: FLAG => NULL()
END TYPE FilteredElangle

TYPE(FilteredElangle),dimension(:),allocatable :: SelectedElangles
integer(kind=jpim),dimension(2)              :: SelectedNrays
integer(kind=jpim),dimension(:,:),allocatable :: NRaysPopulation
integer(kind=jpim)                            :: NbSelectedElangles

```

Our aim was to go through whole BATOR, but only in bator_decodhdf5_mod was quantum of routines to understand.

```

SUBROUTINE InitHdf5(Fct)
SUBROUTINE PrefetchHdf5(Kfic)
SUBROUTINE ValidOdim(IdIn,NomMembre,MaxRays,MaxLength,ElevationsList)
SUBROUTINE ExpandHdf5File(Filename,Kobs,Kw,Kfic,TabSlots,AnalysisDate)
SUBROUTINE PrintMessages(sensor,iterr)
SUBROUTINE GetData(IdIn,DataName,WaitedRank,WaitedAtomic,Iret)
SUBROUTINE GetDatasetDataspace(IdIn,DatasetName,XValues,YValues,ZValues,Iret)
SUBROUTINE ReadDataInt1(IdIn,Inte1Buf,Dimensions,MaxDimensions,WaitedAtomic,Iret) ! scalar or 1D
SUBROUTINE ReadDataInt2(IdIn,Inte2Buf,Dimensions,MaxDimensions,WaitedAtomic,Iret) ! 2D
SUBROUTINE ReadDataInt3(IdIn,Inte3Buf,Dimensions,MaxDimensions,WaitedAtomic,Iret) ! 3D
SUBROUTINE ReadDataReal1(IdIn,Real1Buf,Dimensions,MaxDimensions,Iret) ! scalar or 1D
SUBROUTINE ReadDataReal2(IdIn,Real2Buf,Dimensions,MaxDimensions,Iret) ! 2D
SUBROUTINE ReadDataReal3(IdIn,Real3Buf,Dimensions,MaxDimensions,Iret) ! 3D
SUBROUTINE GetAttribute(IdIn,AttribName,WaitedRank,WaitedAtomic,Iret)
SUBROUTINE ReadAttributeStr1(IdIn,TypeId,String1Buf,Dimensions,MaxDimensions,Iret) ! scalar or 1D
SUBROUTINE ReadAttributeStr2(IdIn,TypeId,String2Buf,Dimensions,MaxDimensions,Iret) ! 2D
SUBROUTINE ReadAttributeInt1(IdIn,Inte1Buf,Dimensions,MaxDimensions,Iret) ! scalar or 1D
SUBROUTINE ReadAttributeInt2(IdIn,Inte2Buf,Dimensions,MaxDimensions,Iret) ! 2D
SUBROUTINE ReadAttributeReal1(IdIn,Real1Buf,Dimensions,MaxDimensions,Iret) ! scalar or 1D
SUBROUTINE ReadAttributeReal2(IdIn,Real2Buf,Dimensions,MaxDimensions,Iret) ! 2D
SUBROUTINE Mtvza(FileId,TabSlots,Kobs,Kw,iterr,Conformity)
SUBROUTINE Odim(FileId,TabSlots,AnalysisDate,Kobs,Kw,Kfic,Conformity)
SUBROUTINE GetDAttributes(IdIn,NomMembre,Attributes)
SUBROUTINE GetFAttributes(IdIn,NomMembre,Attributes)
SUBROUTINE BuildAttributesList(RootAttributes,DatasetAttributes,DataAttributes)
SUBROUTINE Maps(rscale,rstart,elangle,arrayin,elevation)

```

Some temporal tests with Qrad (ODIM HDF5 format radar data with local quality control) prepared on SHMU

SHMU use for nowcasting purpose Qrad program, which was written by Ladislav Meri. Qrad provide quality control information for nowcasting precipitation analysis INCA3 based on single radar, satellites, At the beginning of the stay we didn't have access to SHMU and to OIFS hub from OMSZ. So in meantime we tried to do some preliminary tests with INCA3 data, but thanks to quick resolution of the connection problem we were not putting much effort to this topic. Maybe it will be a future local work on SHMU.

When we worked with local ODIM HDF5 format files without any quality control, BATOR drop down on line 1547 because it was trying to allocate zero field (NQuality was equal to zero). There is missing some control of this!

```
1547     allocate(FullDatasetList(NumGDataset)%GQuality(NQuality),STAT=Error)
1548     if (error /= 0) call Abort1('*** ERROR - BATOR : cannot allocate GQuality(:)')
```

When we have local quality indexes which don't have the same names and the right structure as BATOR expects we must change the name in BATOR namelist variable

```
HODIM%ChooosenTask      ='pl.imgw.quality.qi_total',
```

and then this part of bator_decodhdf5_mod

```
1535     if (ObjectType == H5G_GROUP_F .and. NomMembre1(1:len(trim(HODIM%GrpFlagName))) ==
trim(HODIM%GrpFlagName)) then
1536     NQuality = NQuality + 1
```

NOTE: In cy46t2 was already put condition "if (NQuality > 0) then" otherwise ended with error cannot allocate GQuality

Radar - FR - Toulouse (from OIFS)

bb7934e7725fd1831566fb8d33a73719 T_PAZZ50_C_EUOC_20180824120000_frtou.h5

Basic structure

```
/dataset1 Group
/dataset2 Group
...
/dataset17 Group
/dataset18 Group
/how Group
/what Group
/where Group
```



Also headers were read correctly

```
*** INFO - BATOR : File HDF5.site1   prefetched - NbOBS= 92520 NbWag= 3608280
*** INFO - BATOR : Type produit      : PVOL
*** INFO - BATOR : Source NOD        : frtou
*** INFO - BATOR : Date optimale     : 2018-08-24
*** INFO - BATOR : heure optimale    : 12:00:00
*** INFO - BATOR : Hauteur           : 160.00
*** INFO - BATOR : lat                : 43.574440
*** INFO - BATOR : lon                : 1.376110
*** INFO - BATOR : BeamWidth         : 1.00
*** INFO - BATOR : 18Dataset groups found.
```

```
*** INFO - BATOR : elevation = 0.800000
*** INFO - BATOR : selected dbzh = dataset8/data1
*** INFO - BATOR : selected th = dataset8/data2
*** INFO - BATOR : selected vrad = dataset8/data3
*** INFO - BATOR : selected flag = dataset8/quality4
```

...

```
*** INFO - BATOR : elevation = 10.480000
*** INFO - BATOR : selected dbzh = dataset18/data1
*** INFO - BATOR : selected th = dataset18/data2
*** INFO - BATOR : selected vrad = dataset18/data3
*** INFO - BATOR : selected flag = dataset18/quality4
NbElev : 12ilw : 12NbObs : 92520
```

Three different thinning distances were tested: 1000 m, 2000 m and 5000 m. I think there is a bad unit in output listing "Thinning every xx km." This should be in meters.

```
Thinning every    1000 km (each    1 pixels)
Number of obs left after sub-sampling    81584
la valeur de iobs vaut:    92520
    Selected Obs = 46264 --> 983028 datas.
Total selected Obs = 46264 --> 983028 datas.
```

Elimination de donnees

INBOBS = 46264 INBWAG = 983028

Nb de wagons en final : 983028

bator_pool_balance_mod: ISIMPLE_BALANCING method will be used

bator_pool_balance_mod: Total number of observations: 46264

bator_pool_balance_mod: Total number of wagon in bator arrays: 983028

bator_pool_balance_mod: Total number of wagon in ODB arrays: 983028

The results look promising that we decide go to testing RC LACE radars (first SK, HU, CZ, SL, CR ...)

Header for SHMU radar Maly Javornik looks correct

```
*** INFO - BATOR : Type produit      : PVOL
*** INFO - BATOR : Source NOD       : skjav
*** INFO - BATOR : Date optimale    : 2018-08-24
*** INFO - BATOR : heure optimale   : 12:00:00
*** INFO - BATOR : Hauteur          : 600.00
*** INFO - BATOR : lat              : 48.256100
*** INFO - BATOR : lon              : 17.153100
*** INFO - BATOR : BeamWidth        : 0.92
*** INFO - BATOR : 24Dataset groups found.
```

also selection of observations

```
*** INFO - BATOR : elevation = 0.000000
*** INFO - BATOR : selected th = dataset1/data2
*** INFO - BATOR : selected vrad = dataset13/data1
*** INFO - BATOR : elevation = 0.500000
...
*** INFO - BATOR : selected th = dataset6/data2
*** INFO - BATOR : selected vrad = dataset18/data1
*** INFO - BATOR : elevation = 3.400000
```

NbElev : 12ilw : 12NbObs : 86400

Also filtering of DOPW was performed, but we don't have a time to explore this aspect of radar DA and we concentrate only on reflectivity.

- FILTER CLEANER for elevation check

Begin of Cleaner. N of obs : 12626

End of Cleaner. N of obs : 9892

78.3% observations left

- MEDIAN FILTER FOR DOPW

Thinning every 5000 km

Number of obs left after sub-sampling 7236

la valeur de iobs vaut: 86400

Selected Obs = 819 --> 4352 datas.

Total selected Obs = 819 --> 4352 datas.

Elimination de donnees

INBOBS = 819 INBWAG = 4352

Nb de wagons en final : 4352

bator_pool_balance_mod: ISIMPLE_BALANCING method will be used

bator_pool_balance_mod: Total number of observations: 819

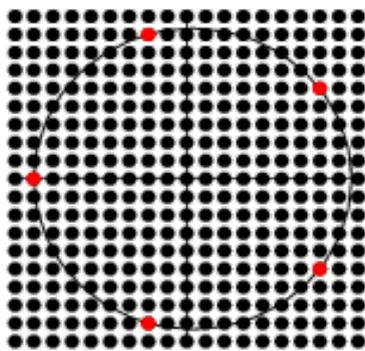
bator_pool_balance_mod: Total number of wagon in bator arrays: 4352

bator_pool_balance_mod: Total number of wagon in ODB arrays: 4352

Thinning every 1000 km : 14528

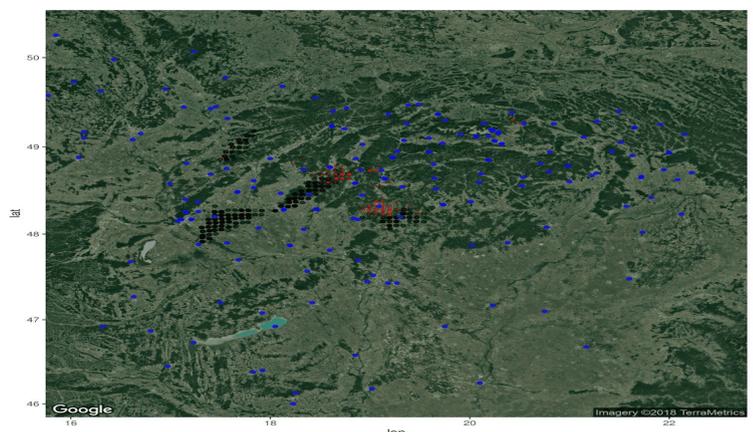
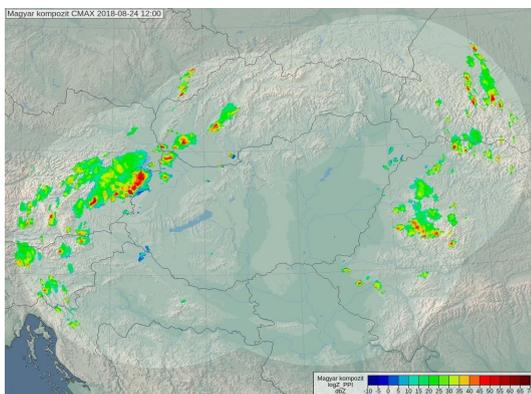
Impact of different thinning distance for SHMU radar Maly Javornik is in table below. The biggest shortcoming of the current method is that precipitation is not **consistent** field like temperature and from that reason is not the best choice to choose reflectivity like each n point from the dataset. It is important to make some higher statistic on this region because is possible to see in many cases like thinning just “jump small cell region”.

Thinning every	after sub-sampling	Total selected Obs
1000	75464	14528
2000	39896	5061
3000	19759	2264
5000	7236	819



To explain difference in sub-sampling and total selected obs I borrow image from <http://mathworld.wolfram.com/CircleLatticePoints.html> . When we imagine the points in circle as radar points. The so many "data" on the start of the ray is so close together this is not allowed. And also when you choose 2 elevations which is selected with same number of rays is usually the lowest ones that means the are so close each other.

On the right image, it possible to see selected observation from radar with high reflectivity denoted by red points and low reflectivity region denoted by black points. Blue points denote weather station position.



The position and the intensity seem to be in a good agreement with the reflectivity image on the left, but we think that it would be better to investigate more cases.

Bator bug: Not clearing memory at the end

In cy43t2bf06 there is a problem with freeing memory (deallocation), because BATOR deallocates only memory used for selected Elangles not for all angle Radar%FinalElev. See this lines in bator_decodhdf5_mod.F90:

2147 !free memory

2158 do i=1,NbSelectedElangles

replaced by

2158 do i=1,size(Radar%FinalElev)

```
if (associated(Radar%FinalElev(i)%DBZH)) deallocate(Radar%FinalElev(i)%DBZH%Values)
nullify(Radar%FinalElev(i)%DBZH)
if (associated(Radar%FinalElev(i)%TH)) deallocate(Radar%FinalElev(i)%TH%Values)
nullify(Radar%FinalElev(i)%TH)
if (associated(Radar%FinalElev(i)%VRAD)) deallocate(Radar%FinalElev(i)%VRAD%Values)
nullify(Radar%FinalElev(i)%VRAD)
if (associated(Radar%FinalElev(i)%FLAG)) deallocate(Radar%FinalElev(i)%FLAG%Values)
nullify(Radar%FinalElev(i)%FLAG)
enddo
```

In new 43t2bf09 was this problem already fixed.

Conclusion

During my stay at OMSZ I was dealing with technical installation and preparations of tools and programs for radar data preprocessing and processing. The backphased version of BATOR 43t1 to cycle 40t2_bf06 was compiled, we installed new HDF5 libraries, we installed odbviewer, new version of python (2.7), we also installed the HOOF tool. We created scripts for plotting in R with google maps plugin.

We prepared connection to OPERA file system.

Regarding validation of the BATOR processing of the radar data we were trying to understand the data flow and did several experiments that were helping us to understand the way BATOR handles the radar observations. We were checking if data are read correctly, if the thinning works as expected. The simple way the thinnings works is not the best since it might skip small cells so we should think of more complex way of thinning.

We reached several technical issues:

- We had troubles to change radar attributes with HOOF, this should be fixed in newer version of HOOF.
- BATOR issues
 - If you forget to put your radar to bator namelist it is not processed!
 - If you use local HDF5 files without QC bator crashes - see temporal tests
 - Cy43t2bf06 bug bator is not clearing memory at the end

Acknowledgement

I would like to thank Hungarian colleagues for perfect working condition, consultations and hospitality mainly Maté Mester and Szűcs Mihály. Special thanks to Endre Lowinger for IT support on OMSZ wfma side. Also Endre with Roman Zehnal for reestablish computer connection between OMSZ and SHMU. Big help also to ARSO Slovenian colleagues Peter Smerkol and Benedikt Strajnar for support and advices around HOOF and for creation HOOF, as usual to CHMU in the Czech Republic Alenka Trojáková and mainly Antonín Bučánek for many valuable advices and make my report readable.