Radar data assimilation In Croatia

Tomislav Kovačić and Antonio Stanešić

Outlook

- Developement of CONRAD_RC and getting data in ODB
- CONRAD-RC
- Assimilation of radar data on one case.
-

Developement of CONRAD-RC and saving data in ODB.

- Developement of CONRAD-RC
  - C++ calsses were restructured to make them easier to use.
  - Radial velocity was added.
  - Bug in usage of flags was corrected.
  - For the time being only polar data can be used.
  - It shouldn't take more than two days to make local data format readable to CONRAD if you enter it through CONRAD-RC.
  - CONRAD-RC is a part of CONRAD software and you can find it in HIRLAM repository.
- Saving data in ODB.
  - The radar data format used in Croatia (and Slovenia) is IRIS RAW.
  - Data are converted by CONRAD (through CONRAD-RC) into MF BUFR format.
  - BATOR cy36t1bf8 is used ot read MF BUFR and to store data in ODB.

CONRAD-RC

- CONRAD-RC uses C++ radar classes
- There is a script *new_format* that prepares environment for using CONRAD-RC with a new format.
- Configuration file must be modified when new format is added.
- Conrad must be recompiled when new format is added.
- User must have own methods for reading format that is added to CONRAD-RC.
- Quality flags: no data, clear sky and rain
- Plan: graund clutter and bean blocking flaging or correcting.

CONRAD-RC, radar classes

What do the radar classes know about radar measurement?

Kinds of radar data:

enum RadarData_t { DBZ, VEL, WIDTH, ZDR, DBT, NODATA };

Kinds of radar sweeps:

enum RadarSweep_t { PPI, RHI, CAPPI, …, HProj_PPI, NONE};

Kinds of coordinates:

enum Coordinate_t { AZIMUTH, ELEVATION, R, X, Y, Z, H };

... and much more.

CONRAD-RC, radar classes

How to make a volume scan?

```
clssRadarVolScan< int >  vols;
```

We need these:

```
radar_ t                      radar;
clssRadarDataType< int >  rdt;
radartime_t                   start_time,   end_time;
clssRadarSweep< int>          rdsw;
```

Building volume scan:

```
vols = radar;
vols .AddRadarDatType(rdt);
```

(Add more data types if needed.)

```
vols.PutTimes( start_time, end_time );
vols = rdsw;
```

(Add as many sweeps as you have.)

CONRAD-RC, radar classes

How to make *radar*?

Fill in these information:

radar.iwmo = ...;

radar.longitude = ...;

radar.latitude = ...;

radar.ground_hieght = ...;

radar.tower_hieght = ...;

radar.w_lengh = ...;

radar.beamwidth = ...;

radar.sensibility = ...;

radar.constant = ...;

radar.PRF = ...;

radar.puls_width = ...;

CONRAD-RC, radar classes

How to make *rdt(clssRadarDataType)*?

rdt.Code_params_1(DBZ, slope, offset, mincode, maxcode, no_data_val);

or

rdt.Code_params_2(VEL, minval, maxval, mincode, maxcode, no_data_val);

CONRAD-RC, radar classes

How to make *rdsw (clssRadarSweep)*?

We need these:

```
clssRadarDataType< int >     rdt;
clssRadarSweepGeometry   rsg;
clssRadarData< int >          rd;
```

Building radar sweep:

```
rdsw.AddRadarDatType(rdt);
```

(Add more data types if needed.)

```
rdsw  = rsg;
rdsw  = rd;
```

(Add as many radar data of different kind as you have.)

Only those data that have type that was added to sweep will be accepted.

CONRAD-RC, radar classes

How to make rsg *(clssRadarSweepGeometry)*?

```
rsg.swt(PPI);
rsg.Q( ELEVATION,    elevation);
rsg.Q( R, R_min, R_max, R_step,  number_of_bins);
rsg.Q( AZIMUTH, azimuth_min, azimuth_max, azimuth_step, number_of_rays);
```

CONRAD-RC, radar classes

How to make *rd (clssRadarData)*?

We need these:

```
clssRadarDataType< int >  rdt;
int                       *data; // Pointer to data buffer.
int                       data_size;
```

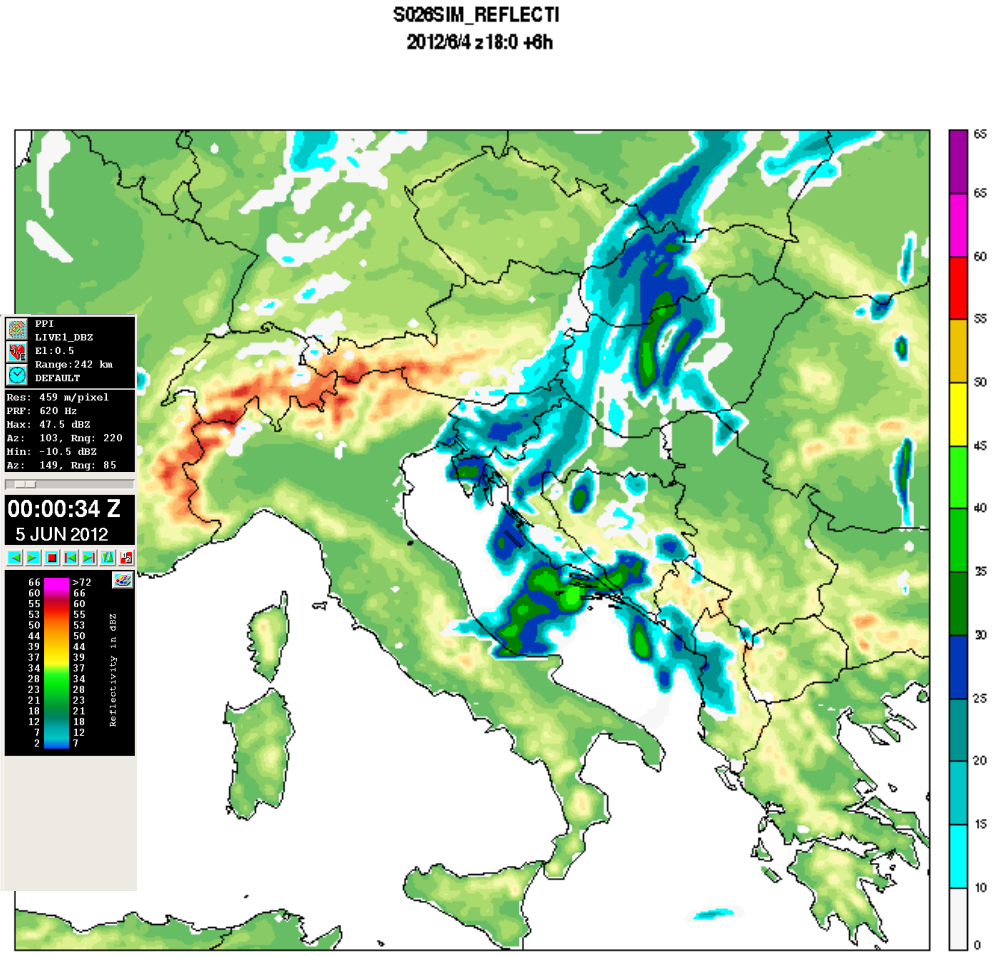
Building:

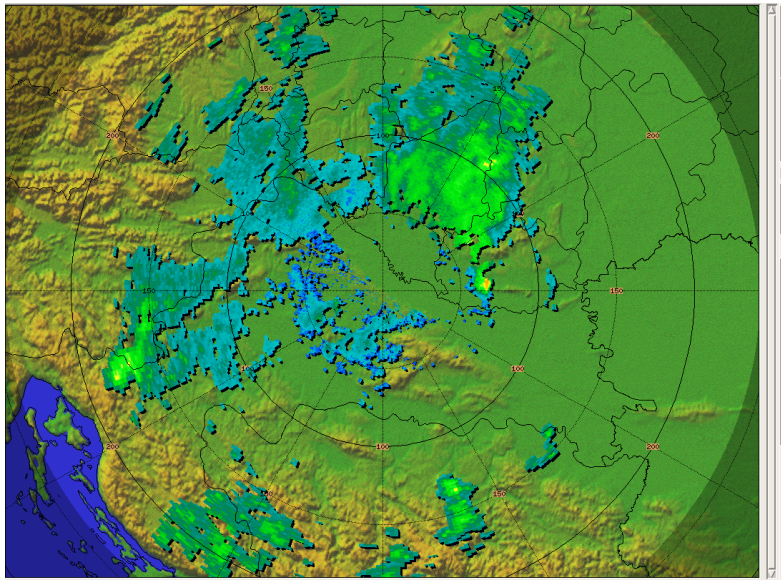
```
rd.RadarDataType = rdt;
rd.PutData(  data, data_size);
```

Assimilation of radar data on one case – observations and model

Model: ALADIN
8 km horizontal resolution
37 levels
hydrostatic
ALARO physics
Analysis time:
5. June 2012 00 UTC

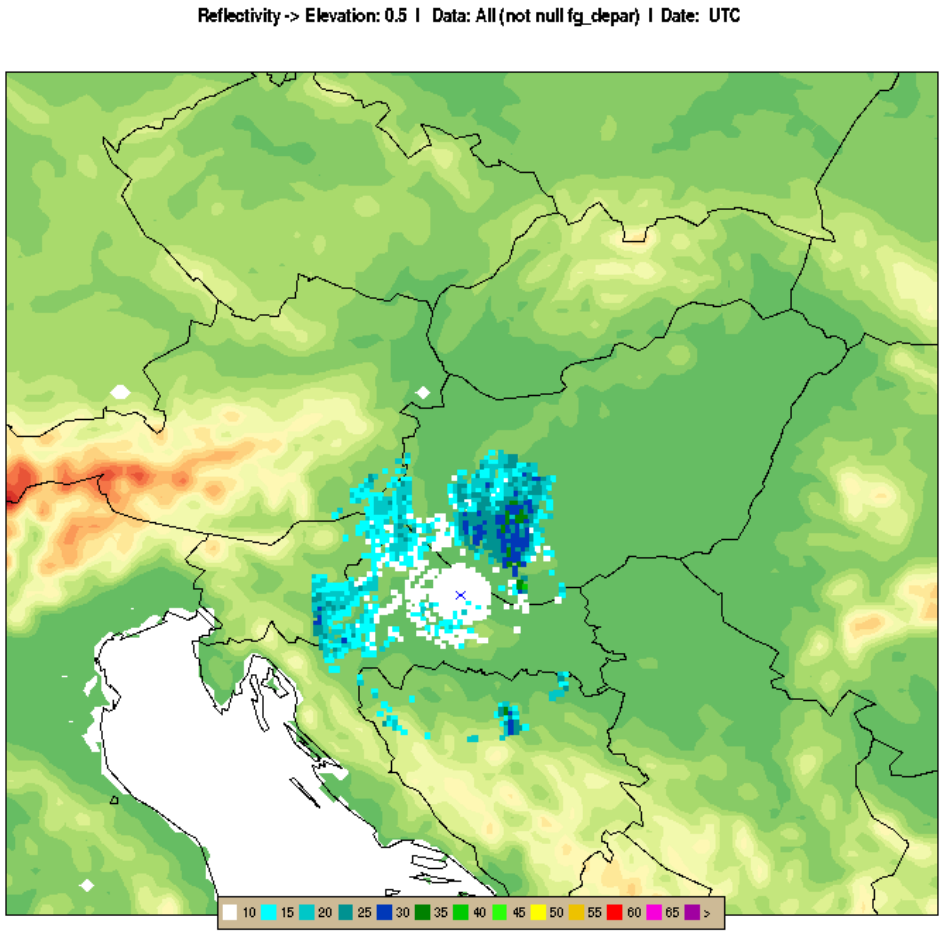
Simulated reflectivity at model level 26 (~2km) from ALARO run using fullpos from the first guess 4. June 2012 at 18 + 6 UTC.





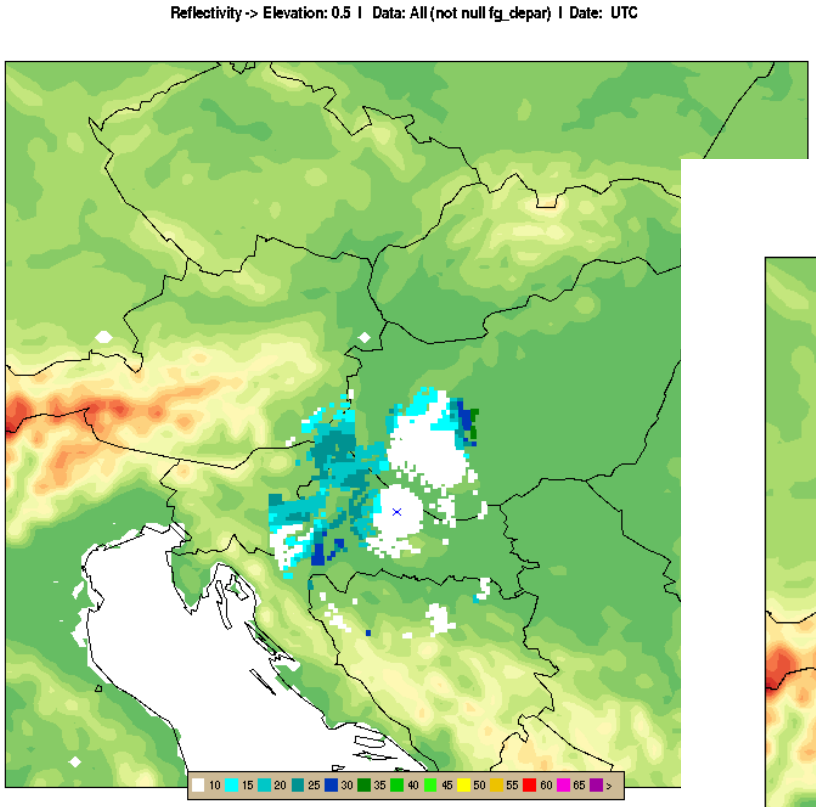
Observed radar reflectivity at 0.5° elevation on 5. June 2012 at 00 UTC.

Assimilation of radar data on one case –after BATOR



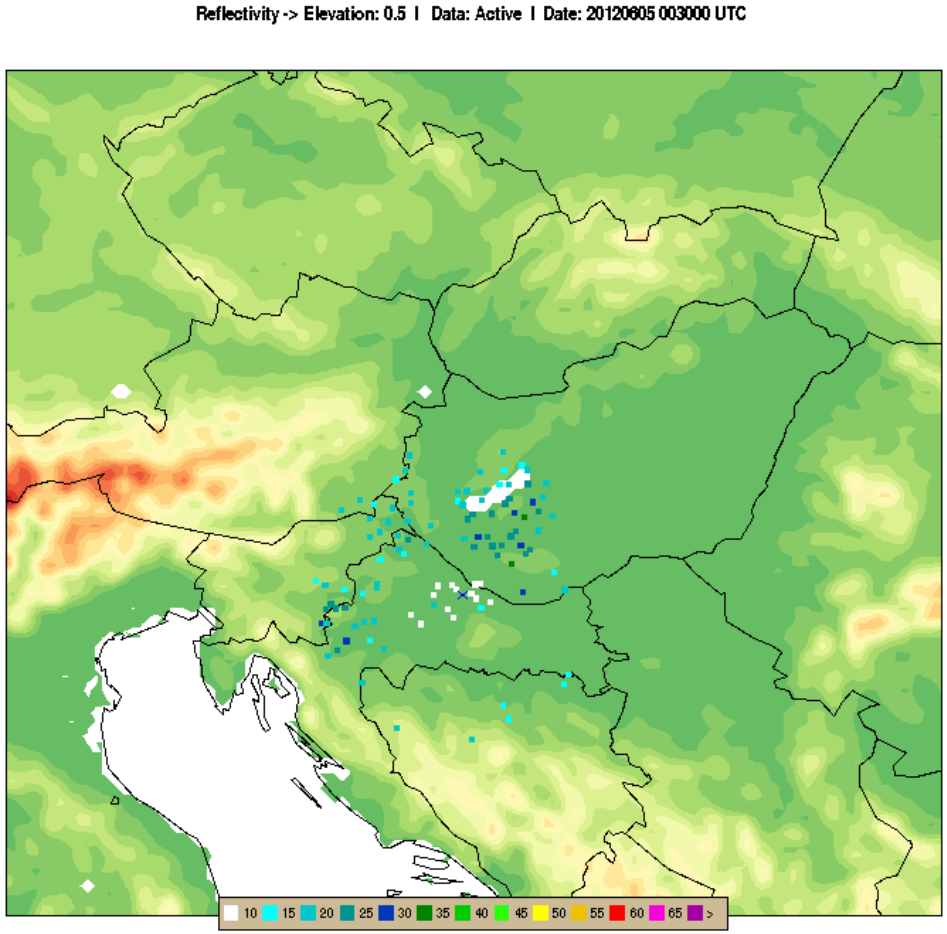
Using unmodified BATOR (cy36t1bf8) MF bufer file was written to the ECMA database. After SCREENING, observational data for 0.5°elevation was extracted from ECMA database using MANDALAY
(just for points where first guess departure was not NULL).
Comparison with first figure shows that observational data was correctly written in ECMA database.

Assimilation of radar data on one case - screening



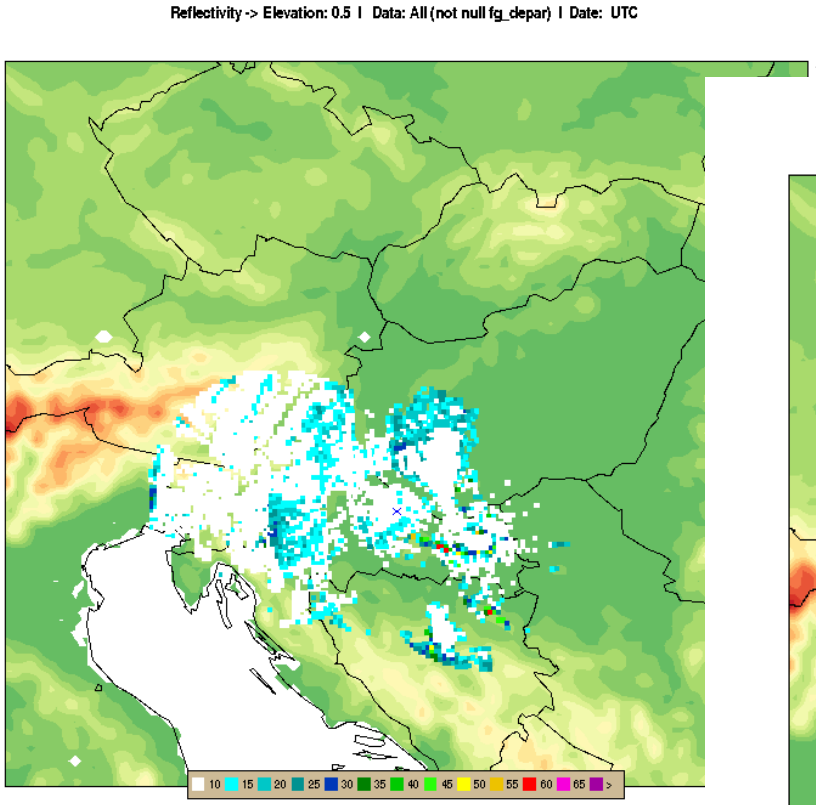
After SCREENING using first guess departure and observed value, model reflectivity of 0.5° elevation (for points where fg_depar/=0; calculated with observation operator) was extracted and shown at figure below.

Active data for 0.5° elevation after SCREENING.



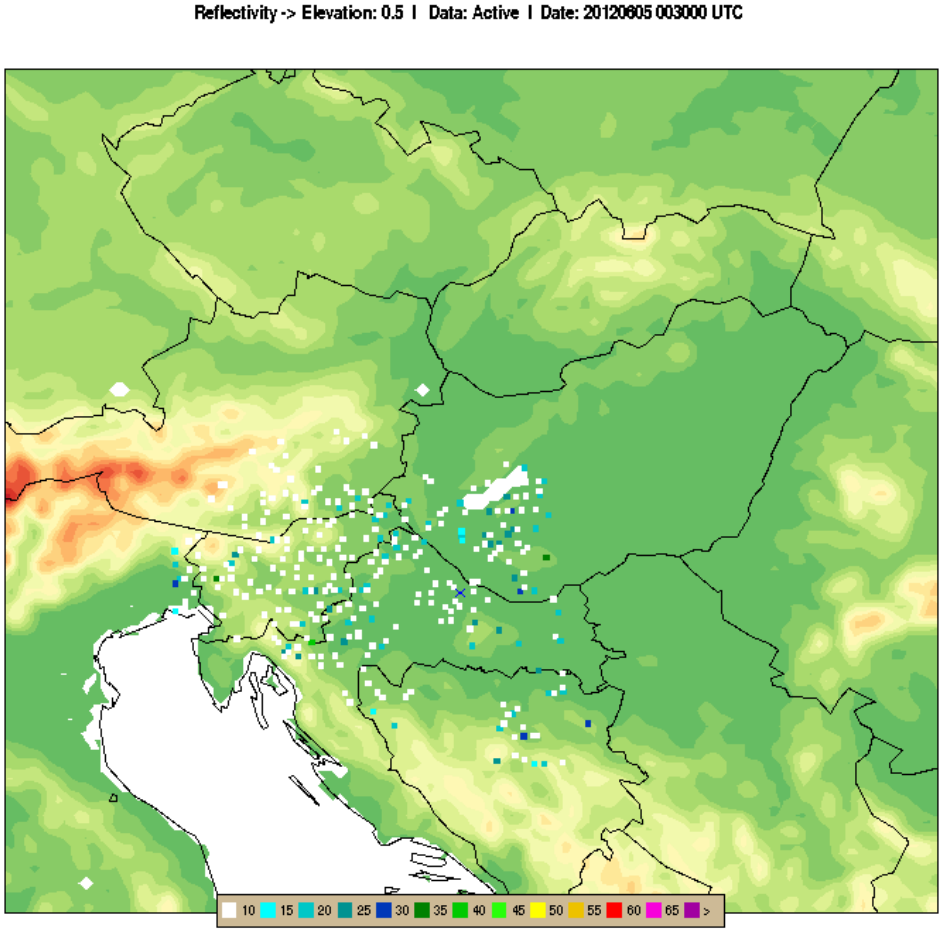
Assimilation of radar data on one case -assimilation

Model: ALADIN 8 km, 37 levels, hydrostatic, ALARO physics
Analysis time: 5. June 2012 00 UTC.
Radars: Lisca (SI), Bilogora (HR) and Osijek (HR).



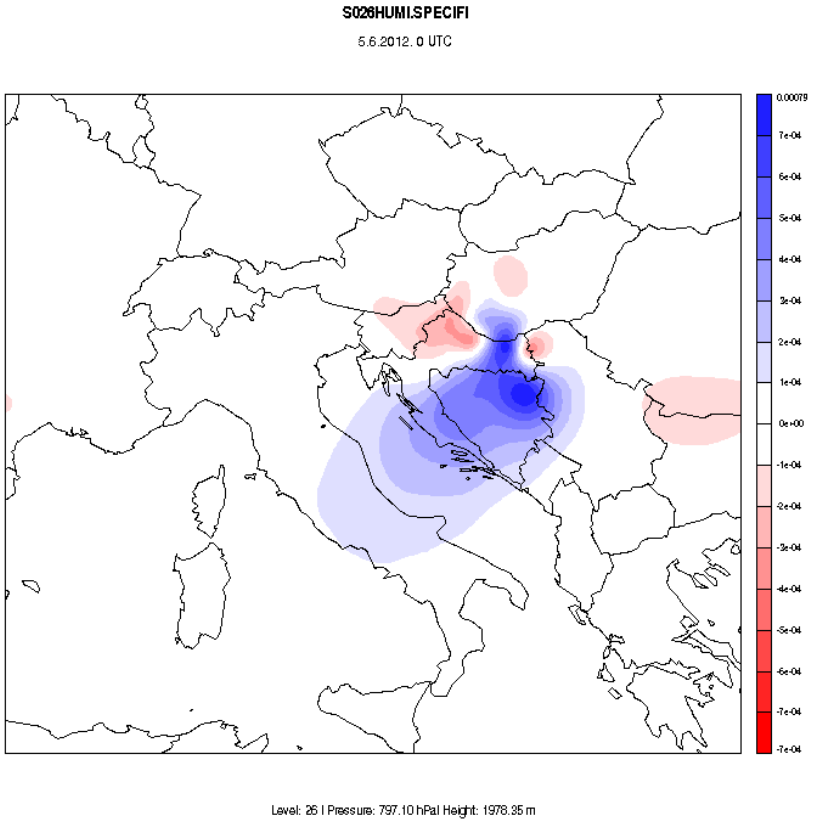
Observed data extracted from ECMA database for points where fg_depar /=0 and for 0.5° elevation.

Active data for 0.5° elevation after SCREENING.

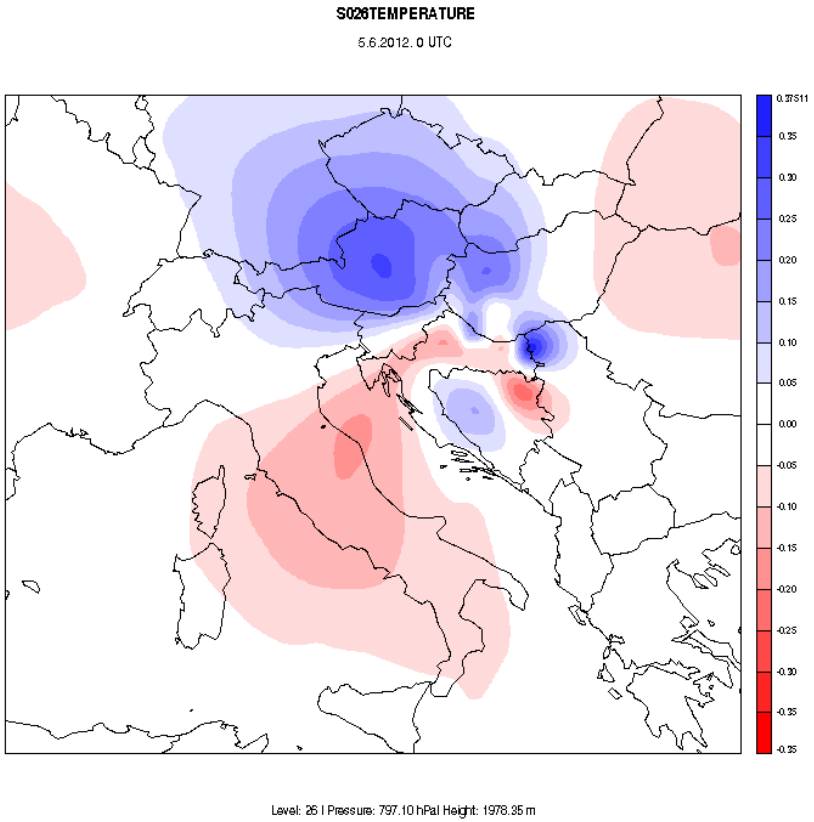


Assimilation of radar data on one case - analysis

Analysis increments



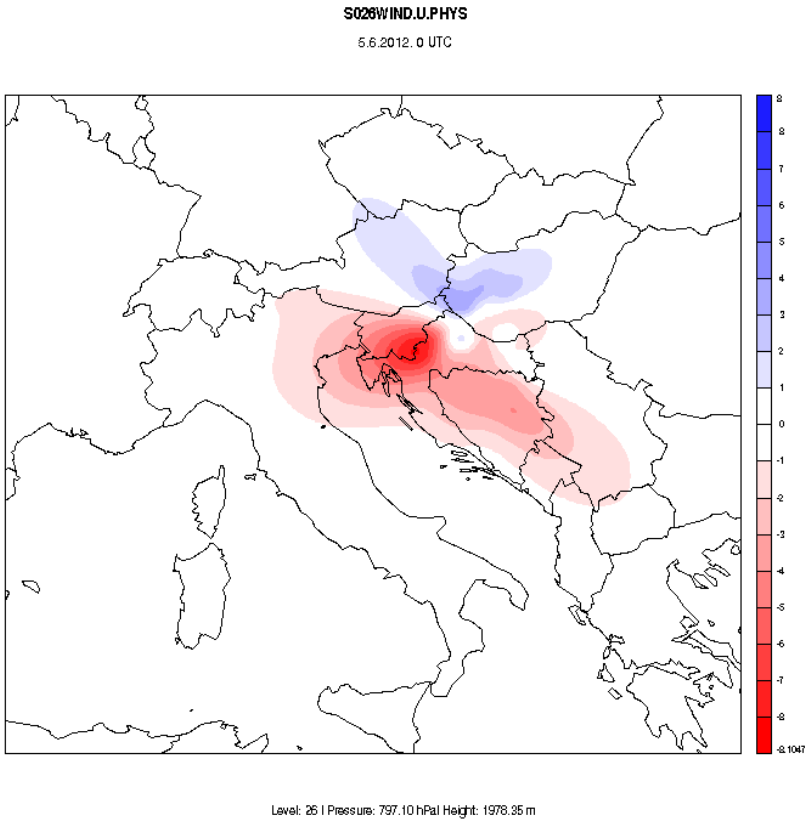
Specific humidity at level 26 (~2km height).



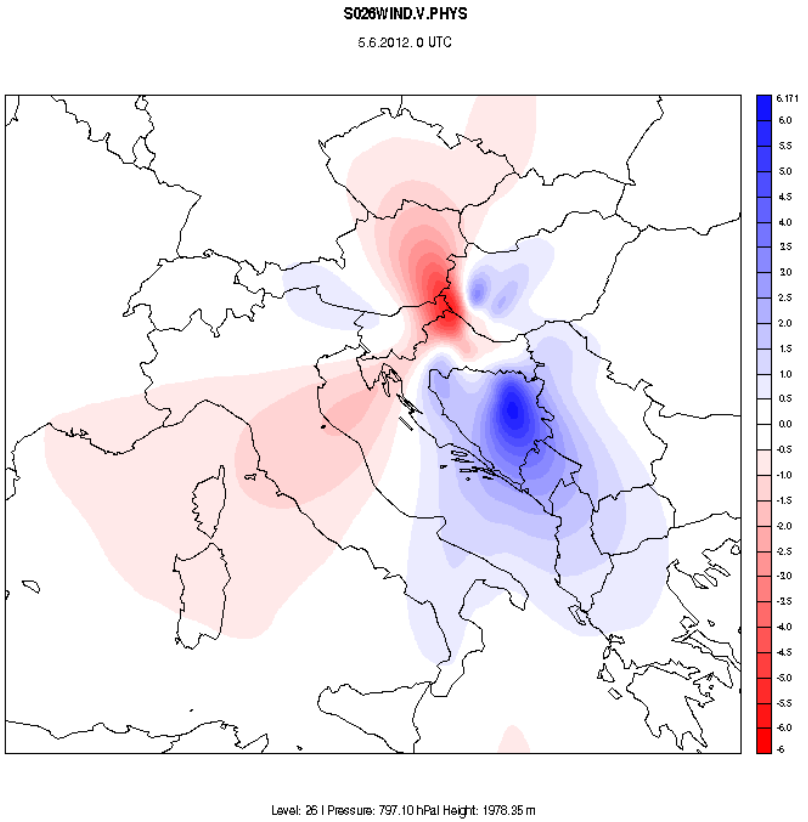
Temperatureat level 26 (~2km height).

Assimilation of radar data on one case – analysis

Analysis increments



U - wind at level 26 (~2km height).

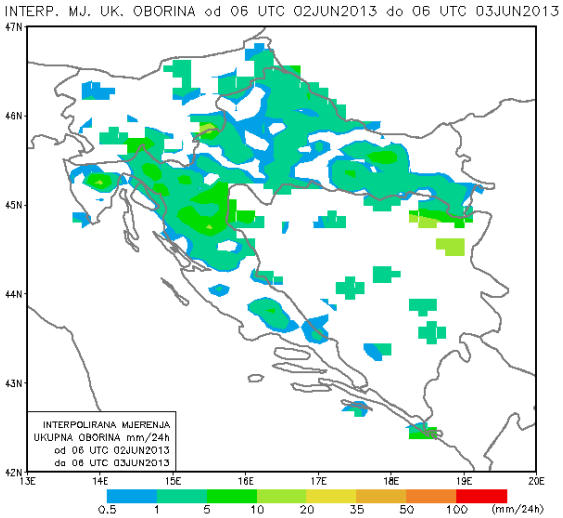


V - wind at level 26 (~2km height).

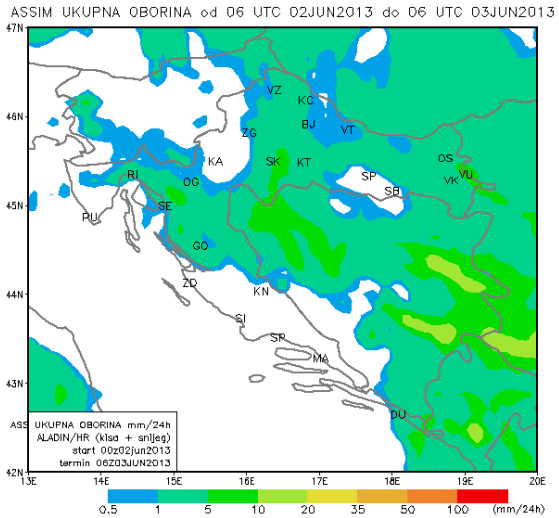
Assimilation of radar data - forecast

20130602 – only one analysis

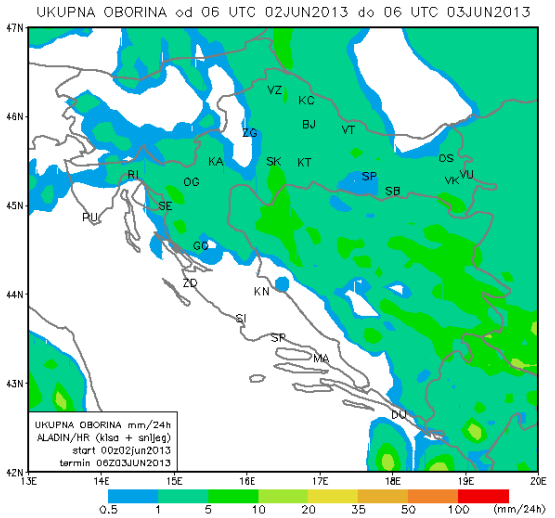
Rain gauges



Radar assim



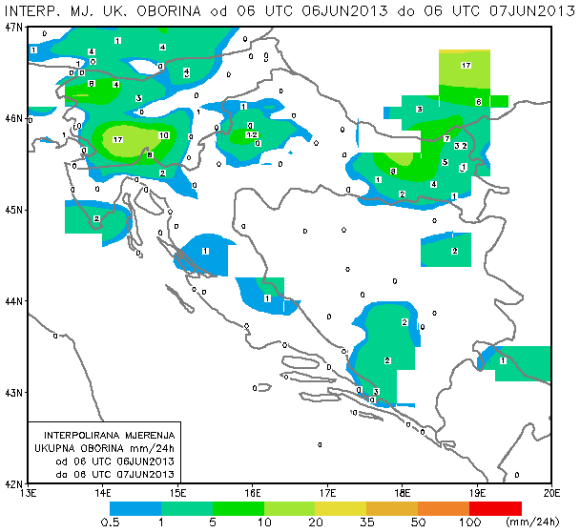
Oper



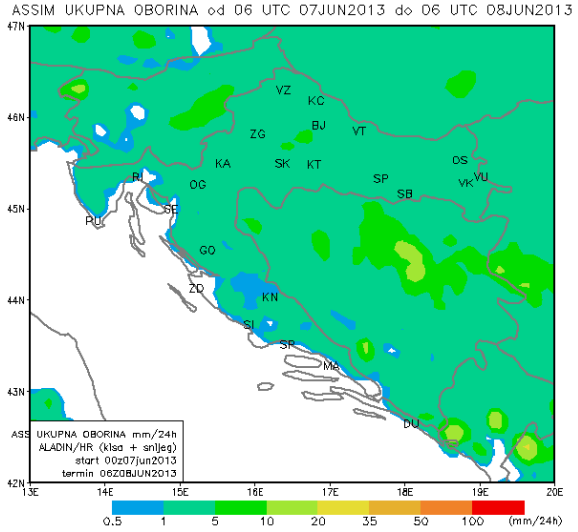
Assimilation of radar data - forecast

20130607 – 6h cycling from 20130602

Rain gauges



Radar assim



Oper