demoMM: a quick analysis of ANTARES public data

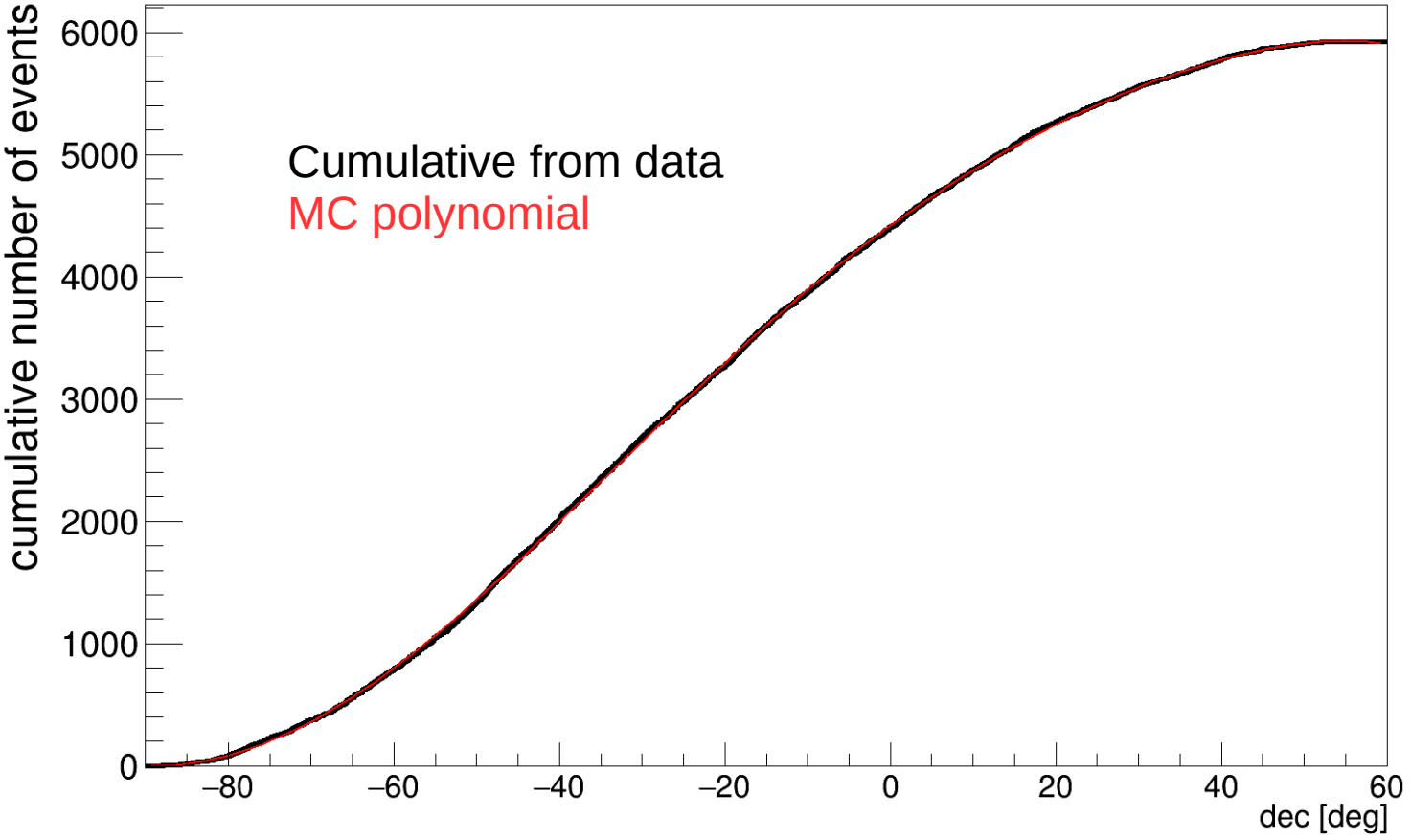
- Small utility to quickly give an estimation of the ANTARES result from a certain direction of the sky
- Use ANTARES public data set http://antares.in2p3.fr/publicdata2012.html for a demostrator
  - Approved by the ANTARES Institution Board in Caserta

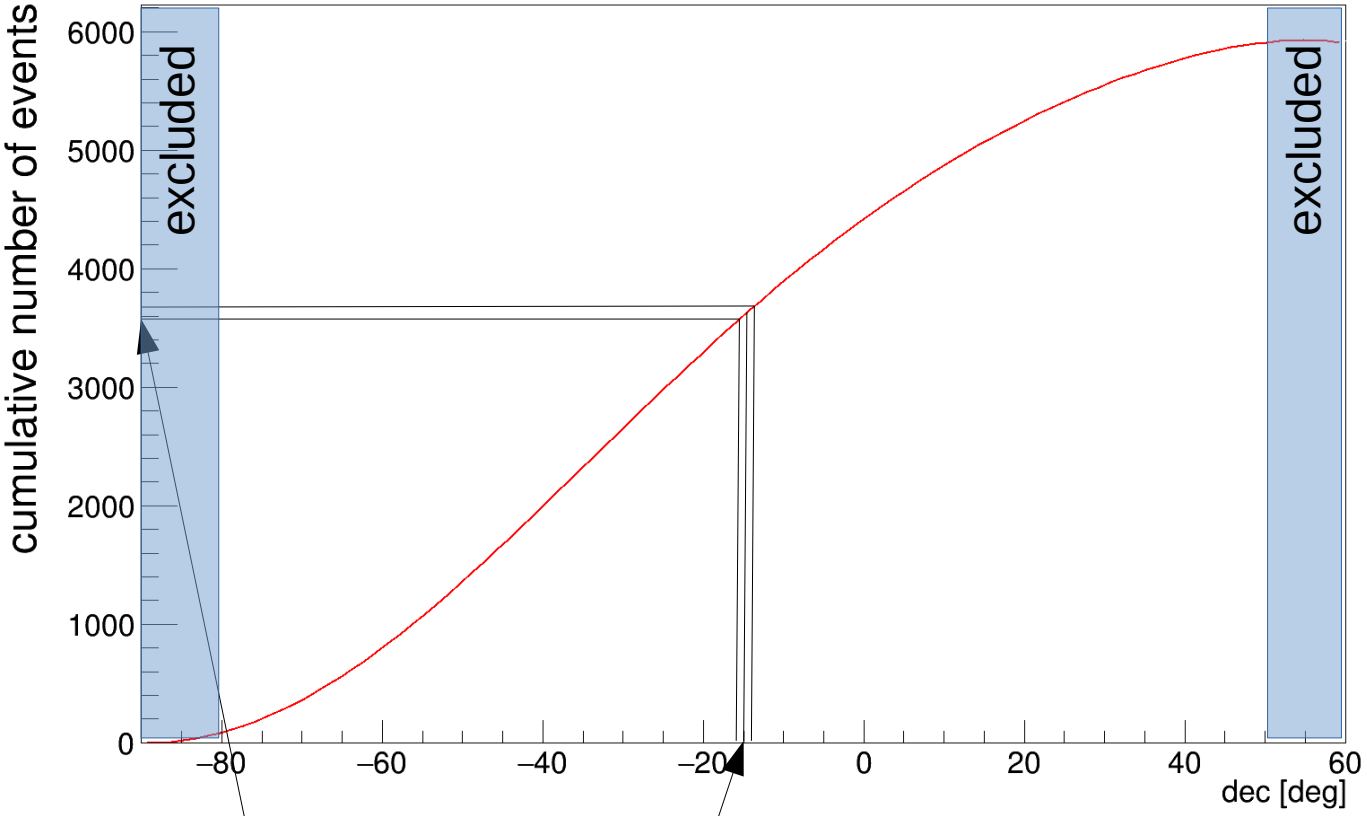
ANTARES demonstrator (v0.1)

- Take the ANTARES 2007-2012 public data
- Perform a simple PS analysis (no time-dependent at the moment)
- Give the program a (dec, RA), a spectral index for the source flux, a region of interest around it:
  - For ANTARES we limit:
    - Dec in]-80, 50[[deg]
    - Gamma in [1.5, 3.0]
    - RoI in [0.1, 2.5] [deg]
- Return: expected bkg in the RoI, observed signal, UL in terms of PS flux

Background estimation

- Get the bkg in the RoI declination band
- Scale it to bkg/sr → convert to number of events taken the RoI solid angle





Directly using the polynomial
No need to open the data multiple times

HARDCODED in the main
Needs to be recomputed if a different data-set is used

Input declination + RoI

Total number of events in the declination band → Scale by solid angle of the band → Multiply by the solid angle of the RoI

[events/livetime] [events/livetime/sr] [events/livetime/RoI]

Signal and U.L.

- Take the (dec, RA) given as input
- Count the events around the position within the RoI
- Compare this to the expected bkg
- Return: expected bkg in the RoI, observed signal in the RoI, UL (F&C, 90% C.L.) in terms of a counting number of events
  - If underfluctuation, returns the sensitivity ad average F&C(bkg,bkg) U.L. (from Model Rejection Factor estimation)
  - Also returns the Lower Limit 90% C.L.

UL in terms of PS flux

- From number of events to signal flux via acceptance
  - Request to the ANTARES IB to produce them (accepted in Caserta)
  - Acceptances produced for the full spectral index range (1.5, 3.0)
  - $\Phi_{UL(LL)} = n_{UL(LL)}/Acc(\Gamma)$
  - Flux normalisation reported at 1 GeV and 100 TeV

demoMM in a Docker container

- docker container
  - ROOT and C++ only required
  - Dockerfile available, simple build
- In the container one can:
  - Compile the C++ code
  - Run it
- Submit.sh script takes in input from command line whatever needed and runs the container producing the output in a txt file

Code availability

- Code at https://github.com/lfusco/demoMM