Detector Response impact on MH Sensitivity

Forschergruppe JUNO

Cheng Yaping

September 22, 2017

IKP-2, FZJ

Gefördert durch DFG Deutsche Forschungsgemeinschaft

Overview

Factors that will impact MH(Ref: JUNO Document 207-v1)

- multiple baselines
- spectral shape uncertainty
- Backgrounds

Its impact:

- multiple baselines from Taishan and Yanjing reactors change $\Delta\chi^2 = 16$ to $\Delta\chi^2 = 12$
- 1% Spectral shape uncertainty will reduce $\Delta\chi^2$ by $\sim$0.9
- 100% Background uncertainty will reduce $\Delta\chi^2$ less than 0.5

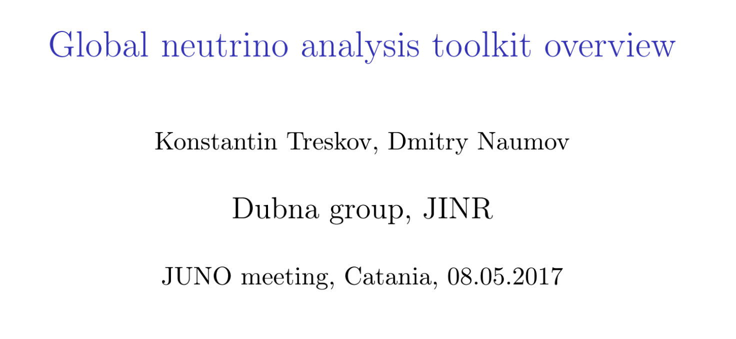
Other possible effects from detector that will impact MH Sensitivity

- energy resolution
- Residual Non-uniformity
- Residual Nonlinearity
- Charge reconstruction method

In principle, all these factors should be decoupled, as a start, I studied the mix-up effect first.

Analysis tools

χ^2 calculation is done by:



Global neutrino analysis toolkit overview

Konstantin Treskov, Dmitry Naumov

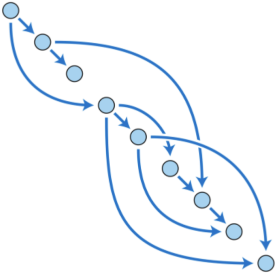
Dubna group, JINR

JUNO meeting, Catania, 08.05.2017

The detector reponse is from JUNO simulation framework SNiper.

few words about GNA

High modularity ,many Computation blocks in c++ ,Python for binding



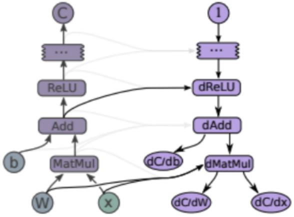


Figure 5: Gradients computed for graph in Figure 2

Using matrix to represent the detector response

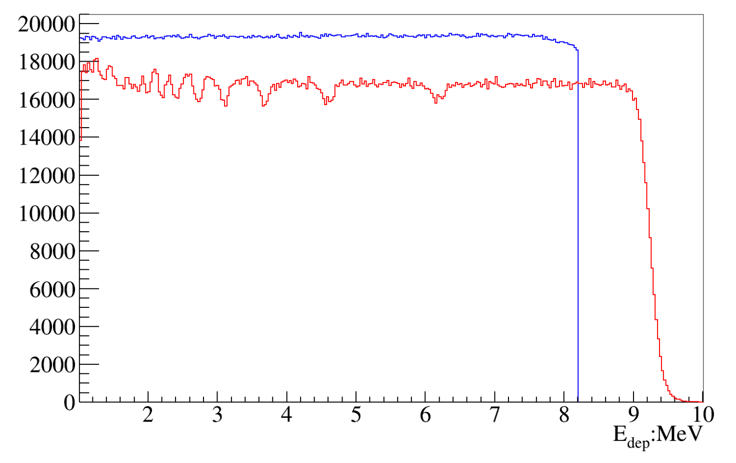
matrix generated from simulation using release version J16v2r1. Using the standard chain 1 production script, i.e. DetSim → "Calib" converter → Vtx+Ene reconstrution.

- single energy positron were generated. Why not IBD-plus? Even without electronics simulation, from det2calib, a dummyplitebytime was done, then we lost alignment.
- from 1 MeV to 8.2 MeV 0.03MeV per step 240 steps 20K events uniform

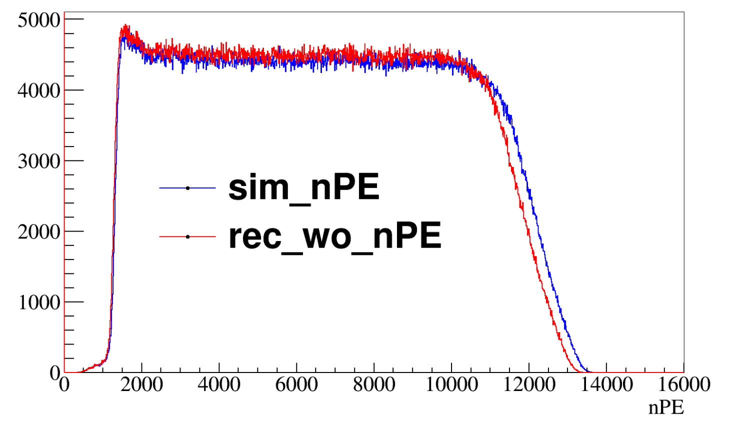
Ene+Vtx rec: start with RecTimeLikeAlg

Blue:energy deposited from simulation

Red:reconstructed energy

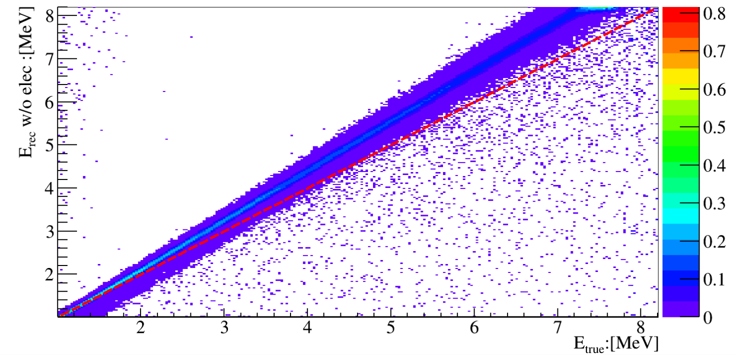


overtuned Erec?



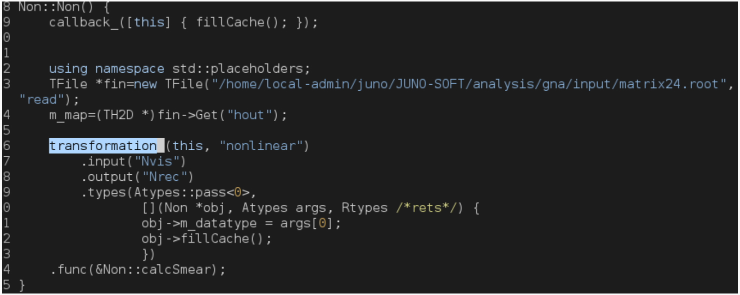
Based on the simulation results from J16v2r1

The detector response matrix is:



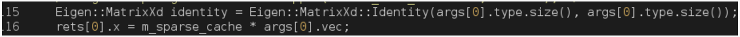
Convolute the detector response with the spectra

In GNA framework, computation is done by transformation



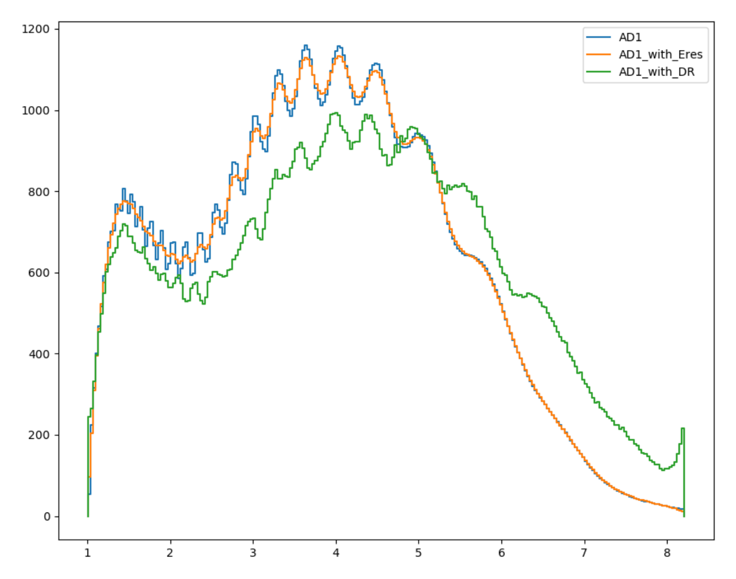
```
Non::Non() {
    callback_([this] { fillCache(); });


    using namespace std::placeholders;
    TFile *fin=new TFile("/home/local-admin/juno/JUNO-SOFT/analysis/gna/input/matrix24.root",
"read");
    m_map=(TH2D *)fin->Get("hout");

    transformation_(this, "nonlinear")
        .input("Nvis")
        .output("Nrec")
        .types(Atypes::pass<0>,
               [](Non *obj, Atypes args, Rtypes /*rets*/) {
               obj->m_datatype = args[0];
               obj->fillCache();
               })
    .func(&Non::calcSmear);
}
```

The element of the matrix means the probability of etrue reconstructed as erec. Probability is done by a 2D integration.

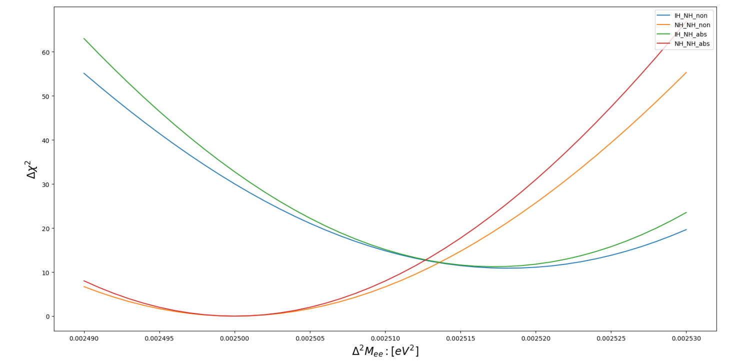


```
    Eigen::MatrixXd identity = Eigen::MatrixXd::Identity(args[0].type.size(), args[0].type.size());
    rets[0].x = m_sparse_cache * args[0].vec;
```

spectra before and after convolution with detector response



Impact on MH


IH_NH_non
NH_NH_non
IH_NH_abs
NH_NH_abs
$\Delta\chi^2$
0
10
20
30
40
50
60
0.002490
0.002495
0.002500
0.002505
0.002510
0.002515
0.002520
0.002525
0.002530
$\Delta^2 M_{ee} : [eV^2]$

Detector response

- only energy resolution considered:
  ```
  Namespace(cpu=0.0200000000000135, errors=array([
  4.04311830e-06]), fun=10.875204406771484, maxcv=0.01,
  nfev=20L, success=True, wall=0.010721921920776367,
  x=array([ 0.00251813]))
  ```
- all included:
  ```
  Namespace(cpu=0.0, errors=array([ 3.65067137e-06]),
  fun=11.22034055185152, maxcv=0.01, nfev=17L,
  success=True, wall=0.005696058273315430, x=array([
  0.00251726]))
  ```

Discussions and Todo

- 240 energy bins and 400K events per bin , $\sim$ 750 T
- How to reduce the disk space? Use some smart griding instead of uniform distribution?
- It's hard to evaluate the uncertainty on each matrix element. Try to parameterize each effect.
- Simply relies on simulation is not good. Since the reconstruction updates frequently, it's time consuming to generate new samples each time.