Precision measurement of neutrino oscillation with the JUNO experiment : Implementation of a new energy spectrum modelling

Internship Defense

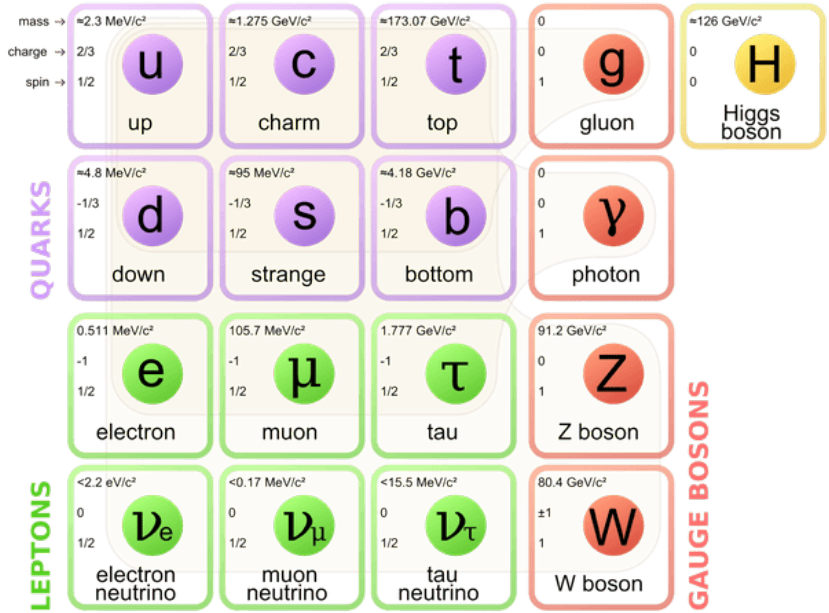
R. GUITTON

Laboratory of subatomic physics and associated technologies, CNRS
Supervised by Benoît VIAUD

19/06/2024

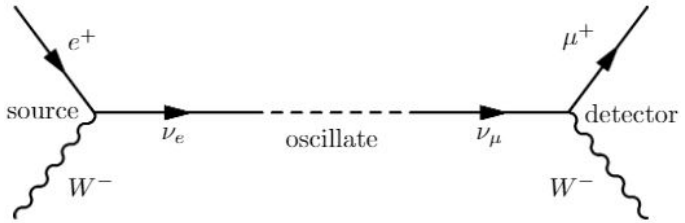
Neutrinos in a nutshell

- An hypothetical particle presented by Pauli (1931)
- Discovered by Clyde COWAN and Frederick REINES in 1956.
- Neutrino facts
  - 3 known flavours
  - Electrically neutral
  - Very weak interaction with matter (limits experimental knowledge)
  - Masses unknown, at most 10^{-5} lower than any other fermion.
  - Nature unknown : Dirac or Majorana



Neutrino oscillation

Neutrino can Change flavor spontaneously



- This oscillation between flavour is possible because
  - Neutrino flavor eigenstates $\neq$ mass eigenstates

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \underbrace{\begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix}}_{U_{PMNS}} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \quad (1)$$

with U_{PMNS} characterised by 3 mixing angles and 1 CP violation phases.

The oscillation is impossible if all neutrinos are massless : It implies that at least 2 neutrino have non zero mass.

Neutrino oscillation

Considering Dirac neutrinos, the 3×3 PMNS matrix U can be parameterized in terms of three flavor mixing angles and one CP-violating phase:

$$U_{PMNS} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{23}e^{-i\delta_{CP}} \\ -s_{12}c_{23} - c_{12}s_{13}s_{23}e^{i\delta_{CP}} & c_{12}c_{23} - s_{12}s_{13}s_{23}e^{i\delta_{CP}} & c_{13}s_{23} \\ s_{12}s_{23} - c_{12}s_{13}c_{23}e^{i\delta_{CP}} & -c_{12}s_{23} - s_{12}s_{13}c_{23}e^{i\delta_{CP}} & c_{13}c_{23} \end{pmatrix}, \quad (2)$$

where $c_{ij} \equiv \cos\theta_{ij}$ and $s_{ij} \equiv \sin\theta_{ij}$ (for $i,j = \nu, \mu, \tau$) are defined with the mixing angles, and δ_{CP} the CP-violating phase.

- The 3-flavor model is nowadays well established
  - Oscillation parameters measured up to a few percents
- To complete the 3-flavor description:
  - δ_{CP} still to be measured (do neutrinos violate CP)?
  - The Neutrino Mass Ordering must also be determined

Mass Ordering

- It informs us on the link between mass and flavor.
- It has significant implications in particle physics, nuclear physics, astrophysics, and cosmology, including neutrinoless double beta decay searches, Supernova neutrino flavor conversionn Nucleosynthesis, ...

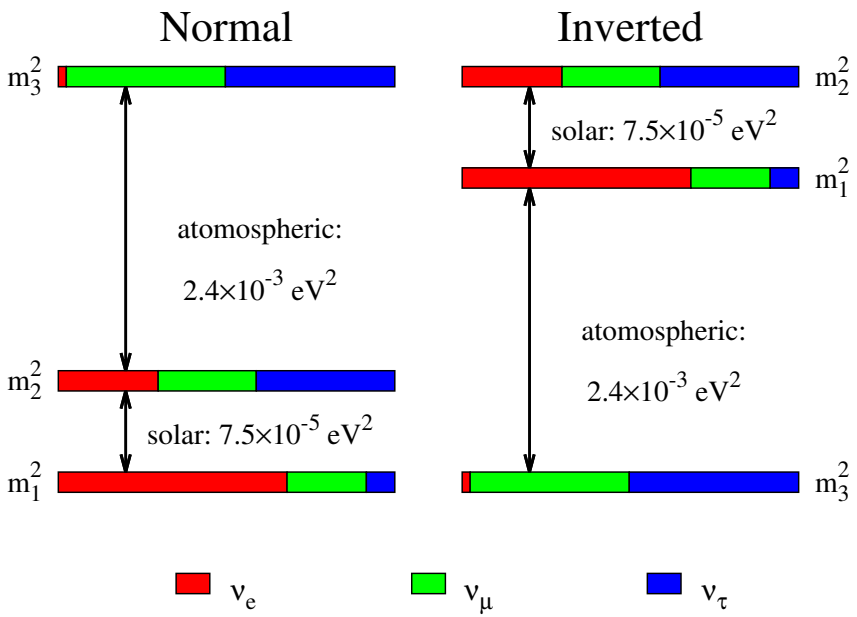


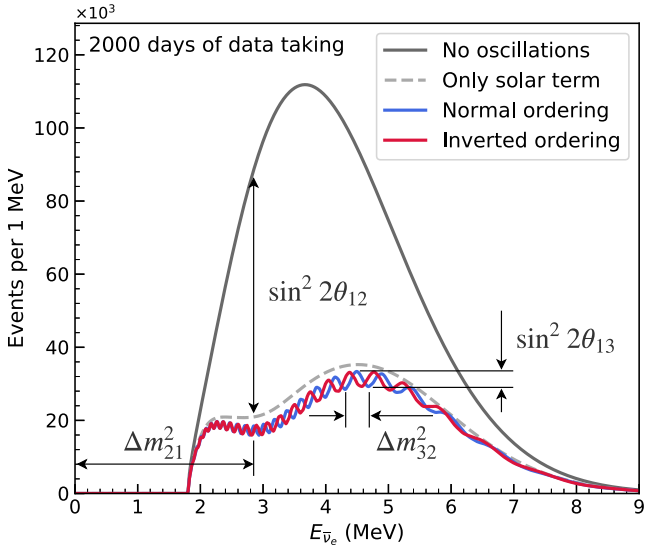
Figure 1: Illustration for the patterns of normal and inverted neutrino mass hierarchies.

Oscillation probability

JUNO will measure the oscillation with great precisions. It has been shown that the **optimal oscillation distance between the antineutrino emission source and the detector location to differentiate the mass hierarchy is 53 km** from the neutrino source [JUNO Collaboration (2015)]. At such distance, the oscillation probability between electronic flavor state is given by [JUNO collaboration (2024)]:

$$\begin{aligned} \mathscr{P}_{\overline{\nu}_e \to \overline{\nu}_e}(E,L) \quad = \quad & 1 - \sin^2 2\theta_{12} c_{13}^4 \sin^2 \Delta_{21} - \frac{1}{2} \sin^2 2\theta_{13} \left(\sin^2 \Delta_{31} + \sin^2 \Delta_{32} \right) \\ & - \frac{1}{2} \cos 2\theta_{12} \sin^2 2\theta_{13} \sin \Delta_{21} \sin(\Delta_{31} + \Delta_{32}), \end{aligned} \tag{3}$$

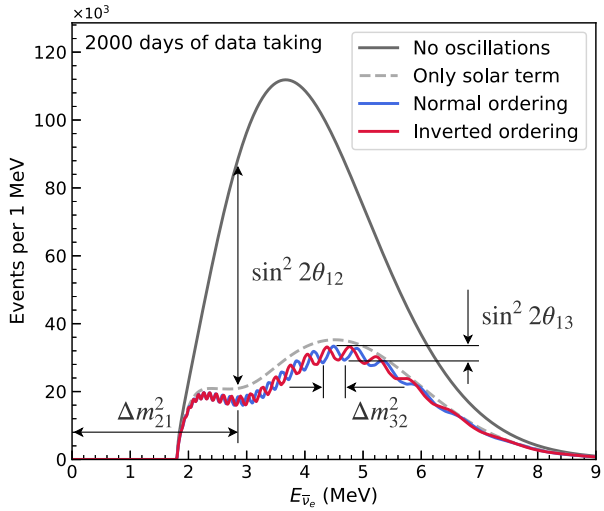
where $\Delta_{ij} \equiv \Delta m^2_{ij} L/(2E)$ with $\Delta m^2_{ij} \equiv m^2_i - m^2_j$.



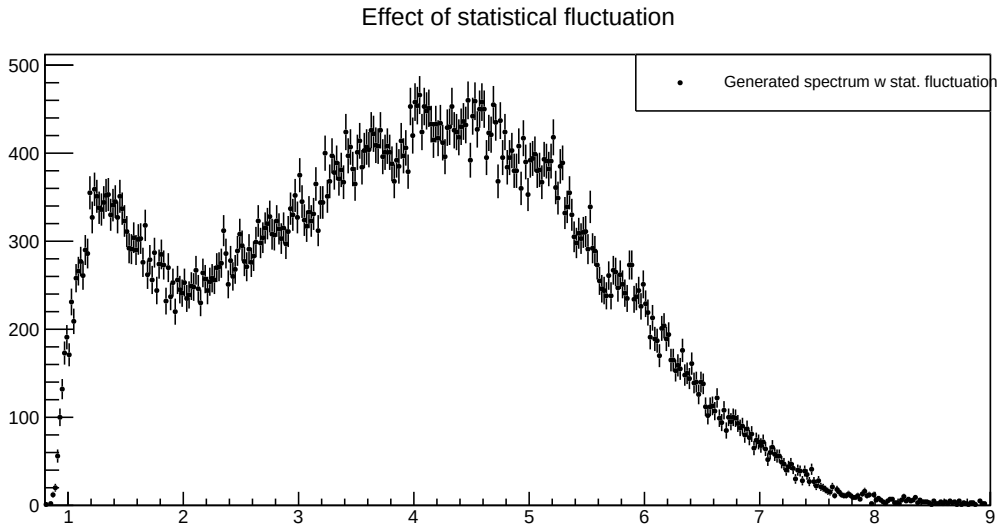
One of the main goals of JUNO: distinguish between the red and the blue spectra to determine the NMO

A difficult measurement

Theory



Practice



These issues have guided the design of the JUNO detector

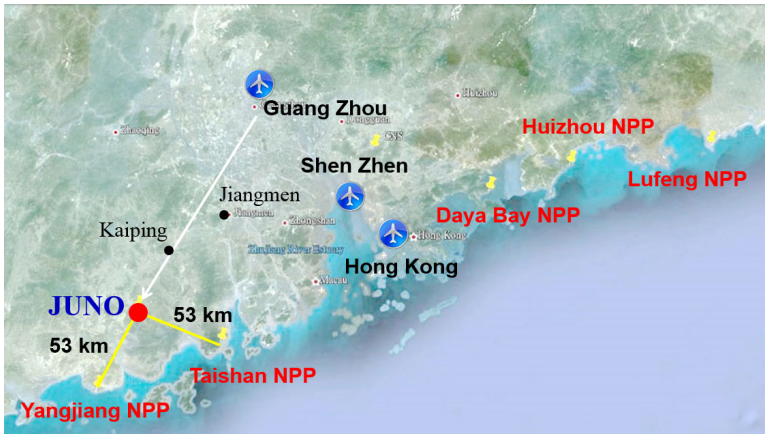
(a) The number of detected antineutrinos is finite: statistical fluctuations

(b) Energy reconstruction effect

(c) Presence of backgrounds

The JUNO experiment

- JUNO main goal is to constrain the NMO.
- It also aims to measure $\Delta m^2_{13}, \Delta m^2_{12}$, and θ_{12} with an unprecedented precision (below the percent).
- JUNO's physics case is large (supernovae, geoneutrino, nucleon decays, ...)
- The Neutrino oscillation program is mainly based on reactor antineutrinos.



Reactor	Power (GW_{th})	Baseline (km)	Relative Flux (%)
Taishan	9.2	52.71	32.1
Core 1	4.6	52.77	16.0
Core 2	4.6	52.64	16.1
Yangjiang	17.4	52.46	61.5
Core 1	2.9	52.74	10.1
Core 2	2.9	52.82	10.1
Core 3	2.9	52.41	10.3
Core 4	2.9	52.49	10.2
Core 5	2.9	52.11	10.4
Core 6	2.9	52.19	10.4
Daya Bay	17.4	215	6.4

(a) The number of detected antineutrinos is finite (stat. fluct.)

- JUNO uses a powerful source of antineutrinos: 8 reactors on 2 sites at 53 km, total thermal power: 26.6 GW
- JUNO is a LARGE detector
  - The detector is also the target for the antineutrino interaction
  - Neutrinos interact very weakly ⇒ need a large volume.
  - **35m diameter** & built **700-m-deep underground**
- JUNO's main component good at catching neutrinos
  - 20 kton of Scintillating Liquid (LS)
  - Well adapted to a kind of interaction that can be detected efficiently (time and space coincidence) : **Inverse beta decay** (IBD)

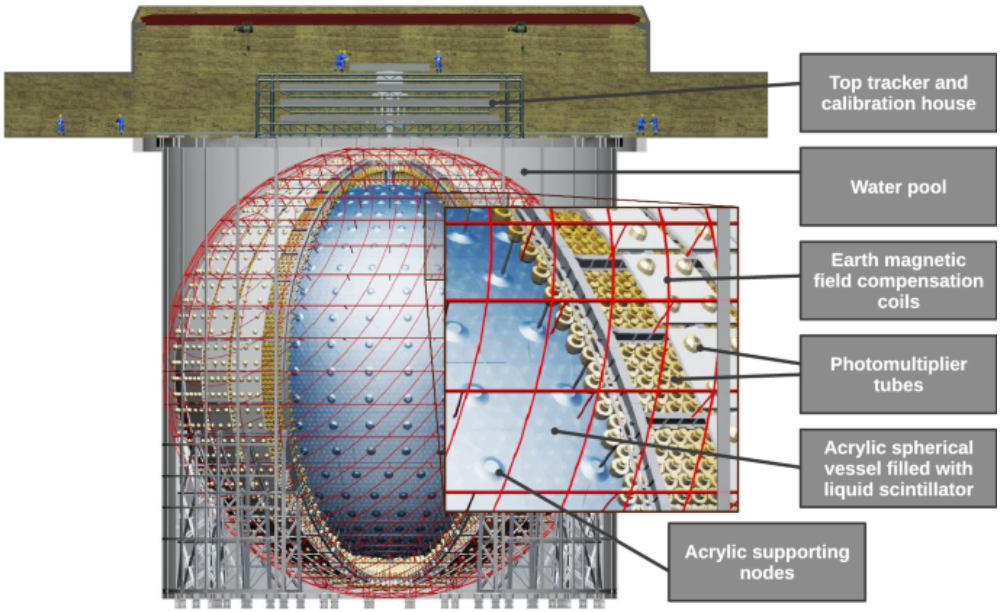


Figure 2: A schematic view of the JUNO detector

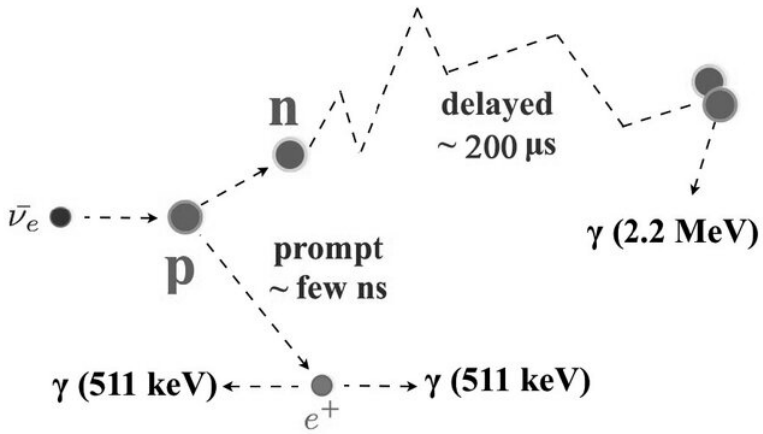


Figure 3: IBD process ($\overline{\nu}_e + p \rightarrow e^+ + n$)

(b) Energy Reconstruction effects

- JUNO needs a *very* precise and reliable E reconstruction to distinguish between NO and IO.
  - Resolution : $\leq$ 3% @ 1 MeV
  - Non linearities in $E_{\text{rec}} = f(E_{\text{true}})$ known better than 1%.
- LS: produces a lot of scintillating photons when positron and gammas deposit E.
  - But need to detect them as efficiently as possible.
  - **17612** LARGE PMT (high coverage) of 20 inches (50cm).

The reliability of the reconstruction enhances by using two PMT systems: in addition to LPMTs: **25600** Small PMTs of 3 inches (8cm).

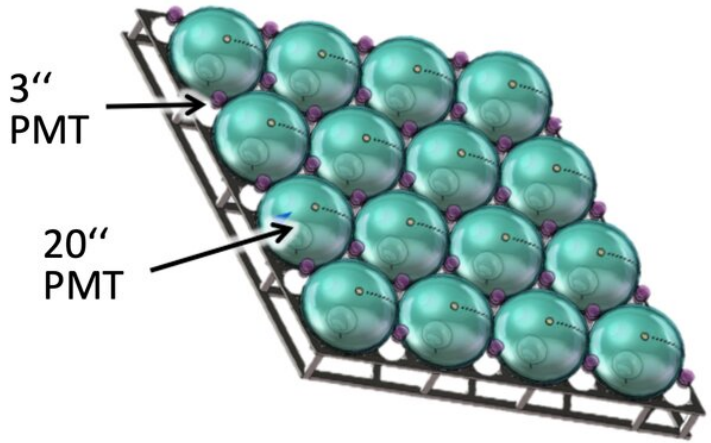


Figure 4: LPMT picture

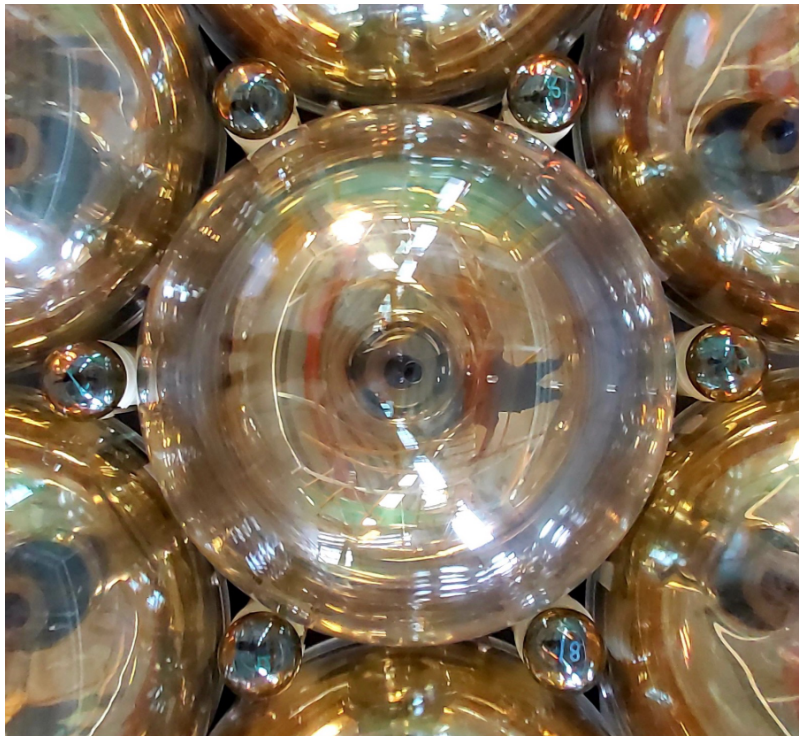
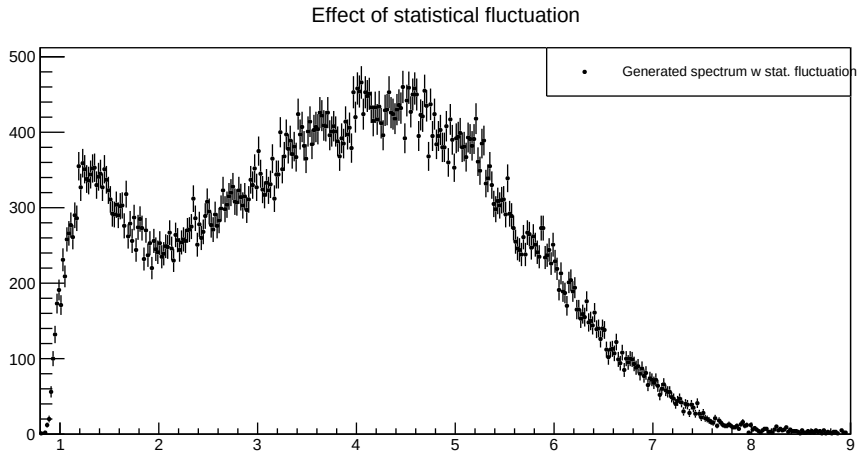


Figure 5: PMT view

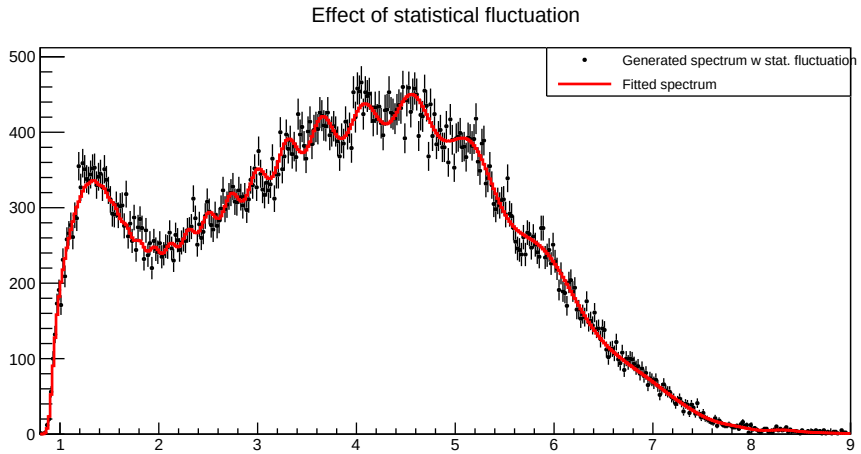
Development and Work at Subatech

- NMO and oscillation parameters inferred by analysing the antineutrino Energy spectrum
- In practice : fit a modelled spectrum to data



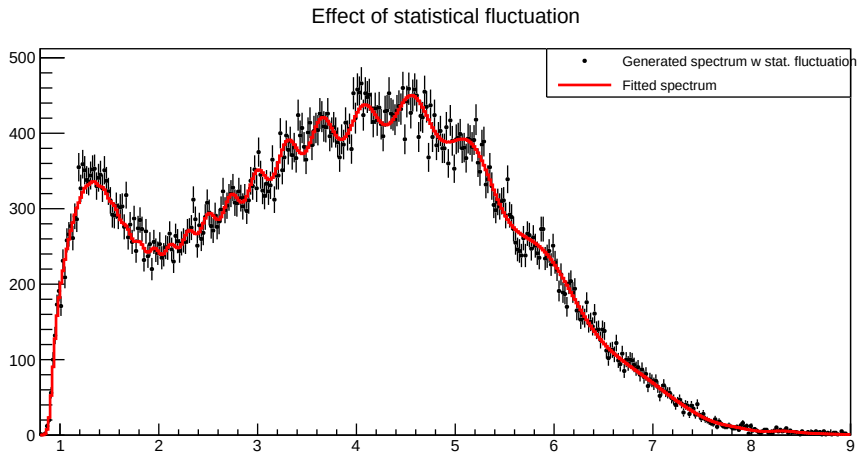
Development and Work at Subatech

- NMO and oscillation parameters inferred by analysing the antineutrino Energy spectrum
- In practice : fit a modelled spectrum to data



Development and Work at Subatech

- NMO and oscillation parameters inferred by analysing the antineutrino Energy spectrum
- In practice : fit a modelled spectrum to data



- Subatech develops a statistical tool called AveNue
  - Produce modelled spectra
  - Fit them to data $\Rightarrow$ NMO, Δm^2_{12}, Δm^2_{13}, θ_{12}, ...
  - Also use these models to simulate fake data samples, to study the performance of statistical procedures.
- My Work: add to AveNue a new faster way to produce modelled spectra
  - $\Rightarrow$ Phase 1: understand the spectrum (physics)
  - $\Rightarrow$ Phase 2: understand AveNue
  - $\Rightarrow$ Phase 3: design and implement a new way to produce modelled spectra.

Spectrum Model Computation

We divide the spectrum in 410 bins, from 1 to 9 MeV.

$$N_i = \sum_j C_{ij}^{\text{Eres}} \int_{E_j^{\text{vis}}}^{E_{j+1}^{\text{vis}}} dE^{\text{vis}} \int_{-1}^{1} d\cos\theta\, \Phi(E^\nu)\, \mathscr{P}_{\overline{\nu}_e \to \overline{\nu}_e}(E^\nu)\, \frac{d\sigma}{d\cos\theta}(E^\nu, \cos\theta)\, \frac{dE^\nu}{dE^{\text{dep}}}\, \frac{dE^{\text{dep}}}{dE^{\text{vis}}} \tag{4}$$

- $\Phi(E^\nu)$ is the original differential antineutrino flux hitting the detector
- $\mathscr{P}_{\overline{\nu}_e \to \overline{\nu}_e}(E^\nu)$ is the oscillation probability expressed eq 3
- $\frac{d\sigma}{d\cos\theta}(E^\nu, \cos\theta)$ is the IBD differential cross section, which depend on the antineutrino energy and on the incidence angle.
- $\frac{dE^\nu}{dE^{\text{dep}}}$ is the differential relation linking E^ν to E^{dep}
- $\frac{dE^{\text{dep}}}{dE^{\text{vis}}}$ is the differential relation linking E^{dep} to E^{vis}
- The sum $\sum_j C_{ij}^{\text{Eres}}$ is a way to consider the resolution effect over the spectra, and act as a convolution.

In the above equation, 4 kinds of energies are involved. Indeed, the detector response that maps the antineutrino energy to the reconstructed energy can be decomposed through several processes (D.Dolzhikov, M. Gonchar, V. Zavadskyi (2024)):

$$E_{\bar{\nu}_e} \xrightarrow[\text{kinematics}]{} E_e(E_{\bar{\nu}_e}, \cos\theta) \xrightarrow[\text{annihilation}]{} E_{dep} \xrightarrow[\text{LSNL}]{} E_{vis}(E_{dep}) \xrightarrow[\text{resolution}]{} E_{rec}(E_{vis})$$

Expected reactor flux

The reactor $\bar{\nu}$ flux car be represented by [JUNO collaboration (2024)] :

$$\Phi(E^{\nu}) = \sum_{r} \frac{\mathscr{P}_{\bar{\nu}_e \to \bar{\nu}_e}(E^{\nu}, L_r)}{4\pi L_r^2} \frac{W_r}{\sum_i f_{ir} e_i} \sum_i f_{ir} s_i(E^{\nu}), \tag{5}$$

where

- $\mathscr{P}_{\bar{\nu}_e \to \bar{\nu}_e}(E^{\nu}, L_r)$ is the antineutrino survival probability at a distance of L_r from reactor r the reactor core index
- e_i is the mean energy per fission for isotope i
- W_r is the reactor thermal power
- f_{ir} is the fraction of isotope i
- $s_i(E^{\nu})$ is the antineutrino energy spectrum per fission for each isotope

At first order the differential cross section from equation 4 can be developed as (P. Vogel, J. F. Beacom (1999)) :

$$\left(\frac{\mathrm{d}\sigma(E_e,\cos\theta)}{\mathrm{d}\cos\theta}\right)^{(1)} = \frac{\sigma_0}{2}\left[(f^2+3g^2)+(f^2-g^2)v_e^{(1)}\cos\theta\right]E_e^{(1)}p_e^{(1)} - \frac{\sigma_0}{2}\left[\frac{\Gamma}{M}\right]E_e^{(0)}p_e^{(0)}, \tag{6}$$

where

$$\Gamma = 2(f+f_2)g\left[(2E_e^{(0)}+\Delta)(1-v_e^{(0)}\cos\theta)-\frac{m_e^2}{E_e^{(0)}}\right]+(f^2+g^2)\left[\Delta(1+v_e^{(0)}\cos\theta)+\frac{m_e^2}{E_e^{(0)}}\right]$$
$$+(f^2+3g^2)\left[(E_e^{(0)}+\Delta)\left(1-\frac{1}{v_e^{(0)}}\cos\theta\right)-\Delta\right]+(f^2-g^2)\left[(E_e^{(0)}+\Delta)\left(1-\frac{1}{v_e^{(0)}}\cos\theta\right)-\Delta\right]v_e^{(0}$$

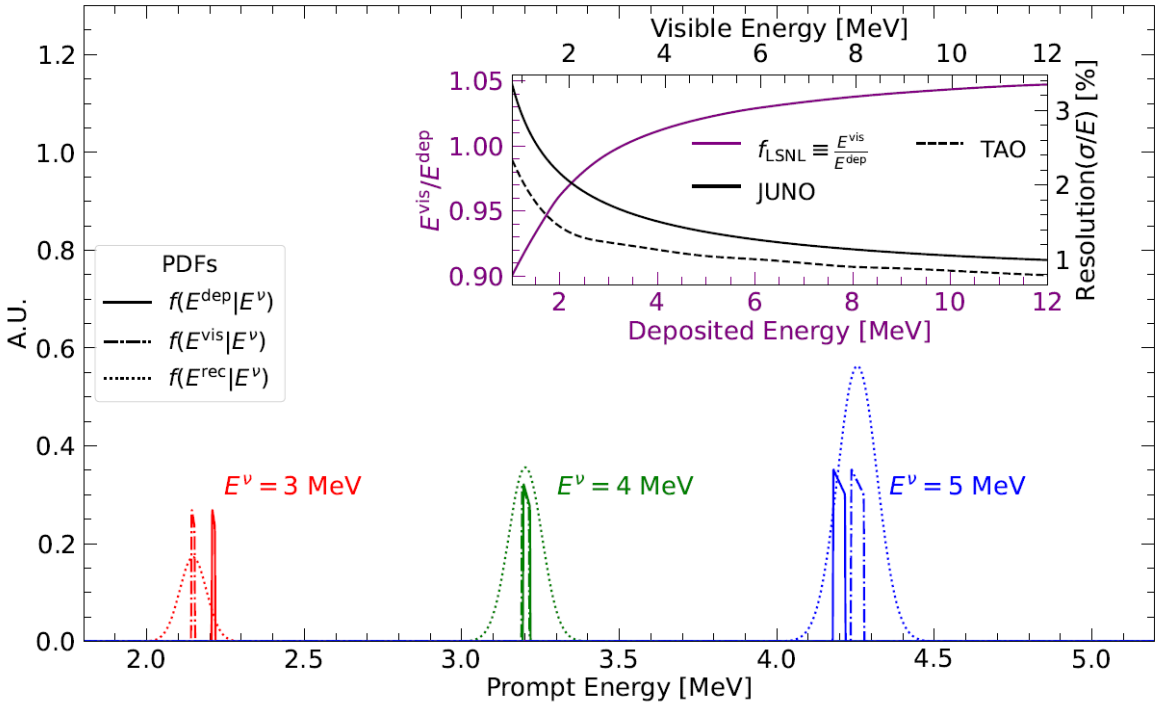
with

$$\sigma_0 = \frac{G_F^2\cos^2\theta_C}{\pi}(1+\Delta^R_{\text{inner}}), \tag{7}$$

$$E_e^{(1)} = E_e^{(0)}\left[1-\frac{E_\nu}{M}(1-v_e^{(0)}\cos\theta)\right]-\frac{y^2}{M}, \tag{8}$$

where $\Delta^R_{\text{inner}} \simeq 0.024$, $\cos\theta_C = 0.974$, $f = 1$, $g = 1.26$, $v_e = p_e/E_e$, G_f the Fermi constant and $y^2 = (\Delta^2 - m_e^2)/2$.

Energy reconstruction effect



$$\frac{\sigma_{E_{\text{rec}}}}{E_{\text{vis}}} = \sqrt{\left(\frac{a}{\sqrt{E_{\text{vis}}}}\right)^2 + b^2 + \left(\frac{c}{E_{\text{vis}}}\right)^2} \tag{9}$$

- The **a term** is the statistical term driven by photostatistics.
- The **b term** is dominated by the position non-uniformity.
- The **c term** represents the contribution of background noises.

AveNue Framework

Set of C++ class and ROOT files developed at SUBATECH in order to generate antineutrino modelled spectrum and fit to the spectrum measured in data. It uses **MINUIT**.

```
├── CUfits.cc
├── CUfits.o
├── CUfits_rdict.pcm
├── dat
│   ├── LPMT
│   └── SPMT
├── docs
│   ├── CodingStandards.txt
│   └── doxygen_config
├── include
│   ├── CUBayesianInterface.hh
│   ├── CUchiSquare.hh
│   ├── CUeventSet.hh
│   ├── CUphysics.hh
│   ├── CUpullInput.hh
│   ├── CUreader.hh
│   ├── CUspectrum.hh
│   └── CUutils.hh
├── JunoFitExport.sh
├── lib
│   └── libCUfits.so
├── LinkDef.h
├── mac
│   ├── converter
│   ├── fitter
│   ├── Generator
│   └── inputs
├── Makefile
├── README.md
└── src
    ├── CUBayesianInterface.cc
    ├── CUBayesianInterface.hh
    ├── CUBayesianInterface.o
    ├── CUchiSquare.cc
    ├── CUchiSquare.hh
    ├── CUchiSquare.o
    ├── CUeventSet.cc
    ├── CUeventSet.hh
    ├── CUeventSet.o
    ├── CUphysics.cc
    ├── CUphysics.hh
    ├── CUphysics.o
    ├── CUpullInput.cc
    ├── CUpullInput.hh
    ├── CUpullInput.o
    ├── CUreader.cc
    ├── CUreader.hh
    ├── CUreader.o
    ├── CUspectrum.cc
    ├── CUspectrum.hh
    ├── CUspectrum.o
    ├── CUutils.cc
    ├── CUutils.hh
    └── CUutils.o
```

AveNue Framework

- AveNue previously generated a spectrum by simulating 10^7 events per Monte Carlo run, which is very time-consuming.
- The fit changes values for NMO, Δm^2_{12}, and other parameters like a, b, c.
- Changing parameters like a, b, c requires **complete regeneration of the spectrum**, making the process extremely slow and often impractical.

$$\frac{\sigma_{E_{\text{rec}}}}{E_{\text{vis}}} = \sqrt{\left(\frac{a}{\sqrt{E_{\text{vis}}}}\right)^2 + b^2 + \left(\frac{c}{E_{\text{vis}}}\right)^2} \tag{10}$$

An initial draft version already contained all the elements from Eq. 4.

- It was separate from AveNue and lacked integration.
- The integration method was basic.

The objectives of this internship were:

- To **improve** the code of this draft version.
- To **optimize** the integration method.
- To **implement** it in AveNue.
- To **measure performance** gains by conducting realistic fits on simulated samples.

Looking for the best integration method

Several methods have been tested:

- Gauss
- Gauss-Legendre
- Adaptive
- Adaptive singular
- Non adaptive

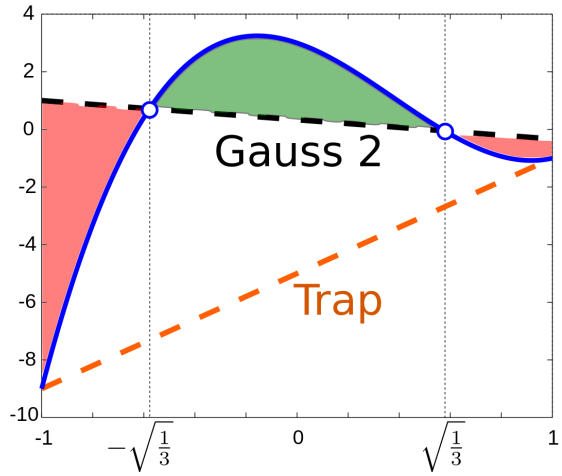


Figure 6: Gaussian quadrature

The comparison criteria are as follows:

- Integration time.
- Reliability of the obtained result :

$$\chi^2 = \frac{\sum_{i=1}^{410}(h_2^i - h_1^i)^2}{h_1^i} \tag{11}$$

h_2 represents the number of events predicted in this bin by the model (AveNue), and h_1 represents the number produced by our method.

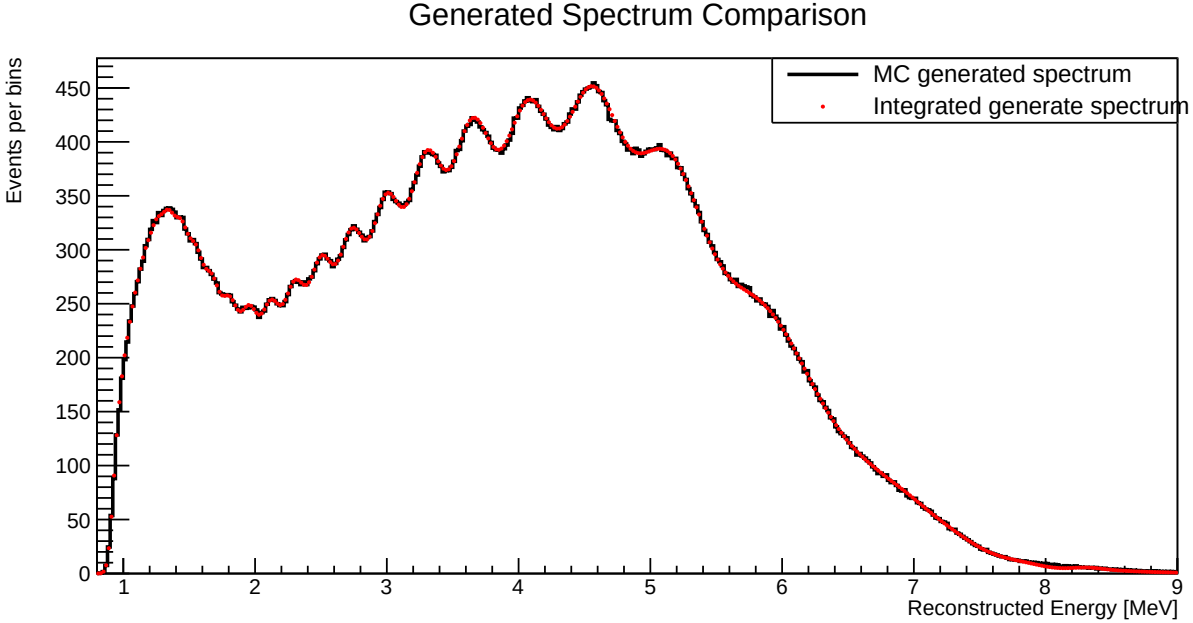
Results

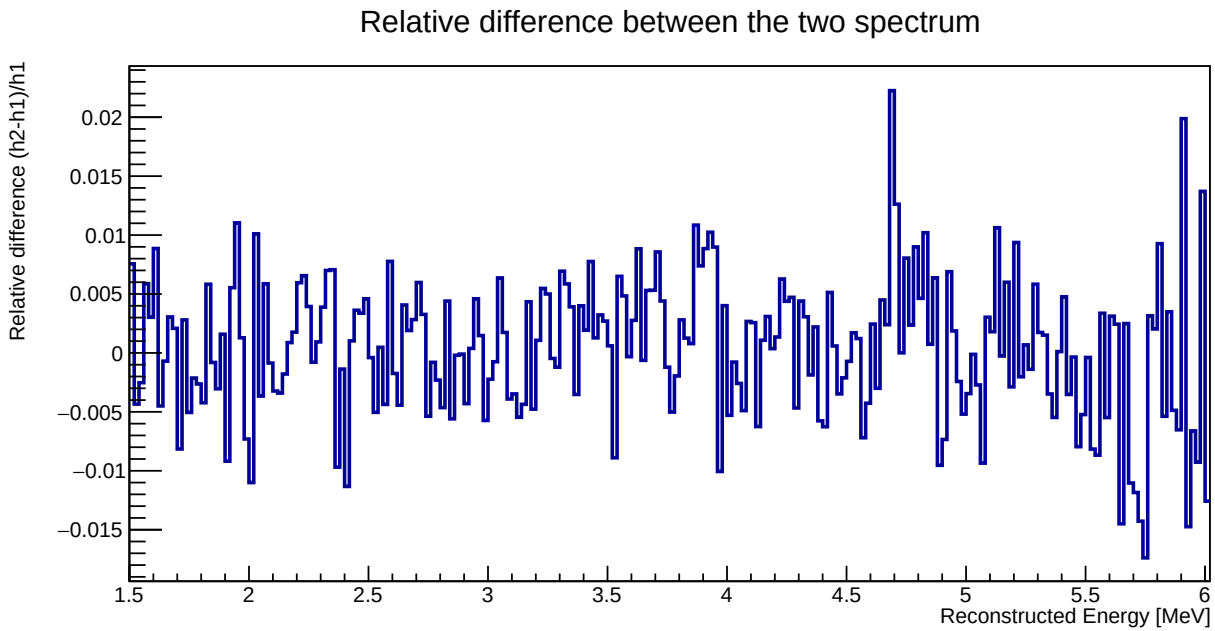
- Choice of method

Integration method	Integration time (sec)
GAUSS	154.55
GAUSS LEGENDRE	0.76
ADAPTIVE	1.43
ADAPTIVE SINGULAR	1.45
NON ADAPTIVE	1.42

- Optimization

Number of integration points	Computation time (sec)	χ^2
2	0.2	7.26
60	0.76	7.09

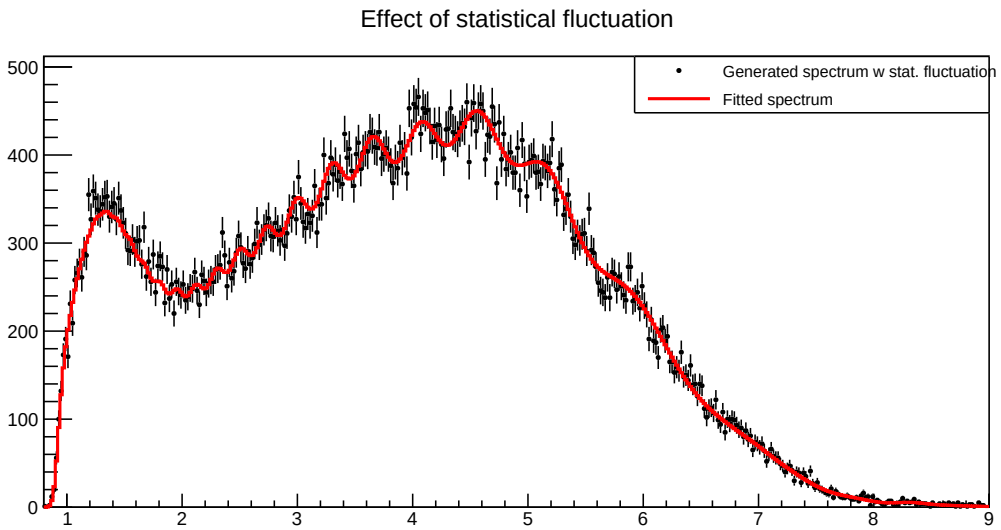

Generated Spectrum Comparison
MC generated spectrum
Integrated generate spectrum
Events per bins
Reconstructed Energy [MeV]


Relative difference between the two spectrum
Relative difference (h2-h1)/h1
Reconstructed Energy [MeV]

Implementation in AveNue

The spectrum is generated via the CUspectrum class in the CUspectrum.cc file. I added the previous method with the new one, adding new functions to the file. Moreover it was necessary to add root files used for the spectrum generation and containing predicted flux, correction bump, etc..

The CUspectrum class is used in other file, it is therefore necessary to respect a necessary format. Adapting the new spectrum function generation into the framework it can now be used to perform the fit:



χ^2 Test

We will compare the obtained uncertainty with the fit from the original MC method and from the integral method. The test statistics we chose is a full Pearson χ^2 :

$$\chi^2 = (\vec{d} - \vec{p})^T V^{-1} (\vec{d} - \vec{p}) + ln|V| \tag{12}$$

with

- $\vec{d}$: Vector containing the content of each bin in the simulated data sample
- $\vec{p}$: Prediction from the model
- V : matrix containing the statistical and systematic errors and their correlation

a, b, c can be promoted as pull parameters by adding term to the χ^2 expression :

$$\chi^2 = (\vec{d} - \vec{p}) V^{-1} (\vec{d} - \vec{p}) + ln|V| + \frac{(a - a_{calib})^2}{\sigma^2_{a_{calib}}} + \frac{(b - b_{calib})^2}{\sigma^2_{b_{calib}}} + \frac{(c - c_{calib})^2}{\sigma^2_{c_{calib}}} \tag{13}$$

with

- a,b,c : resolution parameters determined by the fit
- a_{calib},b_{calib}, c_{calib} : Values pre-evaluated by source calibration methods
- $\sigma^2_{a_{calib}}$,$\sigma^2_{b_{calib}}$, $\sigma^2_{c_{calib}}$: Uncertainties of the previous quantities

Results

The significance level has been set to $\mathbf{1}\sigma$, all the backgrounds have also been considered.

Parameter	Value	Obtained Uncertainty	
		Integral method	MC method
Δm^2_{31}	0.0025283	$\pm 4.7 \times 10^{-6}$	$\pm 4.7 \times 10^{-6}$
Δm^2_{21}	7.53×10^{-5}	$\pm 2.3 \times 10^{-7}$	$\pm 2.4 \times 10^{-7}$
$\sin^2 2\theta_{12}$	0.851004	± 0.0025	± 0.0025
$\sin^2 2\theta_{13}$	0.085299	± 0.010	± 0.010

Table 1: Obtained parameters value and uncertainties comparison

The fit of the integral method was performed with a computation time of **84.55** sec, while for the original Monte Carlo method the fit was performed with a computation time of **3768.01** sec.

Promoting a, b, c as pull parameters we obtain :

Parameter	Value	Obtained Uncertainty
Δm^2_{31}	0.0025283	$\pm 4.7 \times 10^{-6}$
Δm^2_{21}	7.53×10^{-5}	$\pm 2.3 \times 10^{-7}$
$\sin^2 2\theta_{12}$	0.851004	± 0.0024
$\sin^2 2\theta_{13}$	0.085299	± 0.010

Table 2: Obtained uncertainties using resolution as pull parameters

Conclusion

- I started by **reading some documentation** written by the JUNO international collaboration to gain a better overall understanding of the experiment, from its technical aspects to the theoretical part of neutrinos.

- Then I began **performing spectrum generation** using different methods to find the ideal configuration, and **adding some functionality** such as angular dependency and oscillation consideration.

- After that, I accessed the AveNue framework in which I **implemented the spectrum generation method** determined earlier. The new spectrum generation method showed significant performance improvements in the framework.

- Finally, **I promoted the resolution parameters as pull parameters** so they can be changed by the fit. This prevented the need to recreate a covariance matrix for the resolution with every parameter change, which would have taken a lot of computing time.

References

D.Dolzhikov, M. Gonchar, V. Zavadskyi (2024). Neutrino mass ordering sensitivity tech-note.

JUNO Collaboration (2015). Neutrino physics with juno.

JUNO collaboration (2022). Juno physics and detector.

JUNO collaboration (2024). Potential to identify the neutrino mass ordering with reactor antineutrinos in juno.

P. Vogel, J. F. Beacom (1999). The angular distribution of the reaction ve + p $\rightarrow$ e+ + n.

Thanks!

Integration method

The Gaussian quadrature method is a numerical integration technique that approximates the integral of a function $f(x)$ over a given interval $[a, b]$. This method involves selecting specific points x_i (called nodes) and corresponding weights w_i such that the integral can be approximated as:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

The nodes and weights are chosen to maximize the degree of exactness, typically based on the roots of orthogonal polynomials such as Legendre polynomials. This technique is particularly efficient for integrating polynomial functions and is widely used due to its accuracy and computational efficiency.

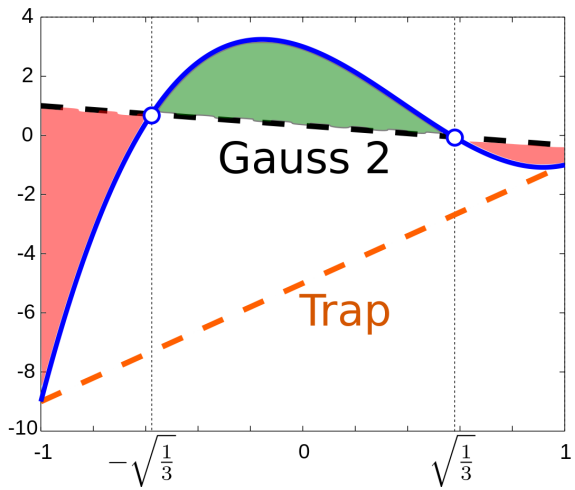


Figure 7: Gaussian quadrature

Integration method

Example: Integrate $f(x) = x^2$ over $[-1, 1]$ using two quadrature points.

$$\int_{-1}^{1} x^2\,dx \approx w_1 f(x_1) + w_2 f(x_2)$$

Points x_i: The points x_i are chosen as the roots of orthogonal polynomials (Legendre, Chebyshev, Hermite, etc.) within the integration interval.

Weights w_i: The weights w_i are often computed using the derivatives of the orthogonal polynomials and satisfy certain orthogonality conditions. For the n-point Gauss-Legendre quadrature, the weights w_i are given by: $w_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}$,

For $n = 2$, points are $x_1 = -\frac{1}{\sqrt{3}}$ and $x_2 = \frac{1}{\sqrt{3}}$ with weights $w_1 = w_2 = 1$. It gives :

$$\int_{-1}^{1} x^2\,dx \approx \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Exact value is $\int_{-1}^{1} x^2\,dx = \frac{2}{3}$.

Gaussian quadrature gives the exact result for this polynomial function with two points.

Integration method

Adaptive quadrature dynamically adjusts integration steps based on the local behavior of the integrand.

- Start with an initial interval.
- Evaluate the integrand and estimate an initial approximation.
- Check the error against a tolerance criterion.
- If error is acceptable, stop. If not, subdivide the interval and repeat.

```
1. procedure integrate ( f, a, b, τ )
2.     Q ≈ ∫_a^b f(x) dx
3.     ε ≈ |Q − ∫_a^b f(x) dx|
4.     if ε > τ then
5.         m = (a + b) / 2
6.         Q = integrate(f, a, m, τ/2) + integrate(f, m, b, τ/2)
7.     endif
8.     return Q
```

Parameter	Best fit	1σ range
Normal neutrino mass ordering ($m_1 < m_2 < m_3$)		
$\Delta m^2_{21}/10^{-5}$ eV^2	7.54	7.32 — 7.80
$\Delta m^2_{31}/10^{-3}$ eV^2	2.47	2.41 — 2.53
$\sin^2\theta_{12}/10^{-1}$	3.08	2.91 — 3.25
$\sin^2\theta_{13}/10^{-2}$	2.34	2.15 — 2.54
$\sin^2\theta_{23}/10^{-1}$	4.37	4.14 — 4.70
$\delta/180^\circ$	1.39	1.12 — 1.77
Inverted neutrino mass ordering ($m_3 < m_1 < m_2$)		
$\Delta m^2_{21}/10^{-5}$ eV^2	7.54	7.32 — 7.80
$\Delta m^2_{13}/10^{-3}$ eV^2	2.42	2.36 — 2.48
$\sin^2\theta_{12}/10^{-1}$	3.08	2.91 — 3.25
$\sin^2\theta_{13}/10^{-2}$	2.40	2.18 — 2.59
$\sin^2\theta_{23}/10^{-1}$	4.55	4.24 — 5.94
$\delta/180^\circ$	1.31	0.98 — 1.60

Table 3: The best-fit values, together with the 1σ intervals, for the six three-flavor neutrino oscillation parameters from a global analysis of the experimental data JUNO Collaboration (2015).

To answer issue (c), JUNO is built in an deep underground site (700 m deep). Also, two veto systems detect a part of the remaining muon events

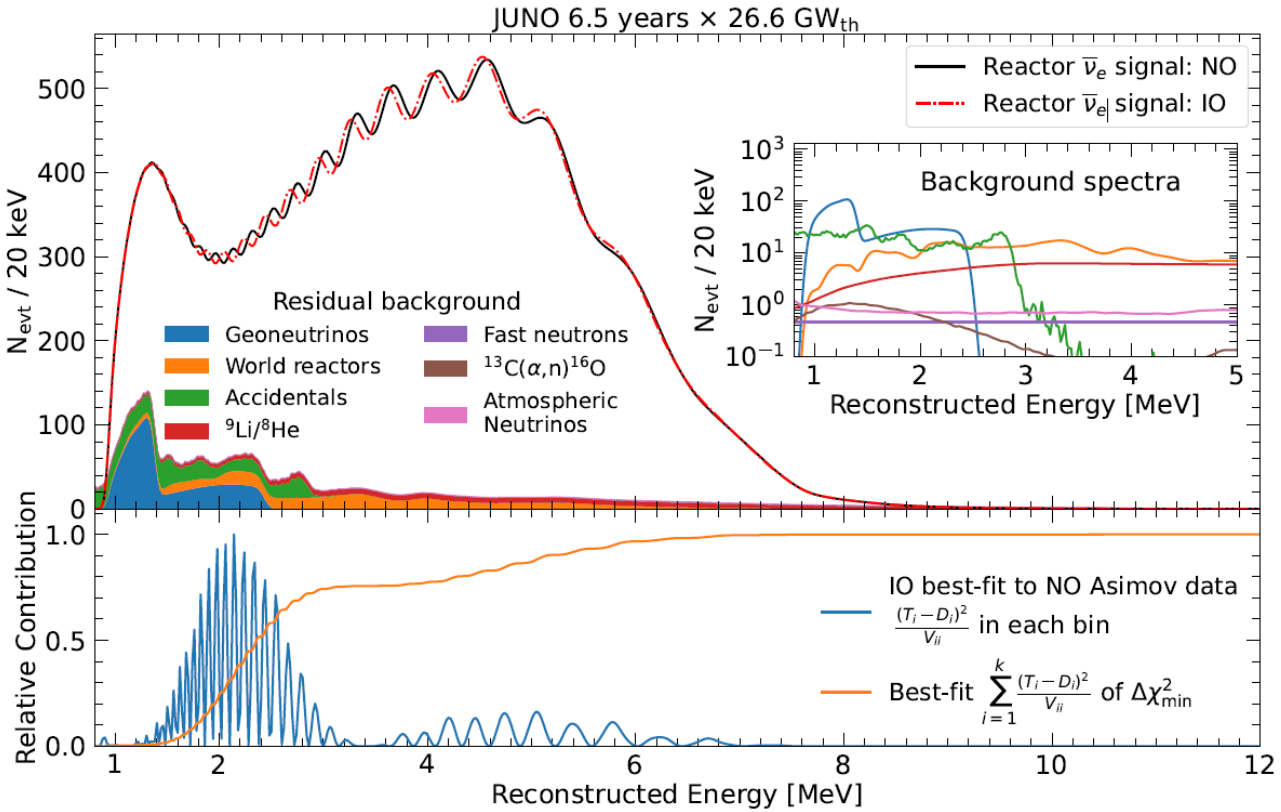


Figure 8: Background contribution to the spectrum (JUNO Collaboration (2015))

Reactor	Power (GW_{th})	Baseline (km)	Relative Flux (%)
Taishan	9.2	52.71	32.1
Core 1	4.6	52.77	16.0
Core 2	4.6	52.64	16.1
Yangjiang	17.4	52.46	61.5
Core 1	2.9	52.74	10.1
Core 2	2.9	52.82	10.1
Core 3	2.9	52.41	10.3
Core 4	2.9	52.49	10.2
Core 5	2.9	52.11	10.4
Core 6	2.9	52.19	10.4
Daya Bay	17.4	215	6.4

Table 4: Reactor properties and contribution to the JUNO experiment **?**

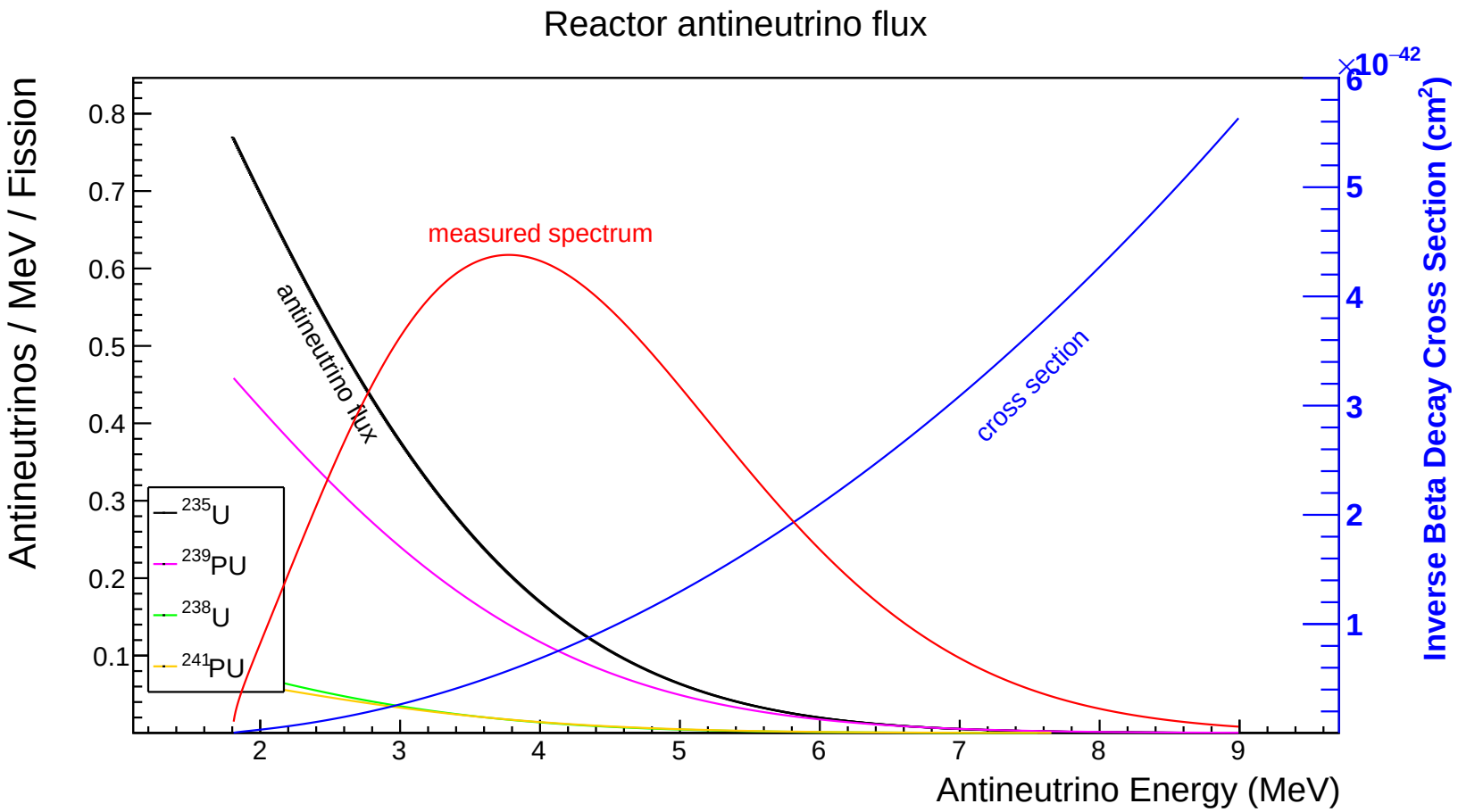


Figure 9: antineutrino reactor flux & cross section