

JUST: A neutrino fit software for JUNO's solar analysis and sensitivity

A. GÖTTEL^{(1)(2)(*)}, L. LUDHOVA⁽¹⁾⁽²⁾, A. MERAVIGLIA⁽³⁾⁽⁴⁾, L. PELICCI⁽¹⁾⁽²⁾
and G. SETTANTA⁽¹⁾

⁽¹⁾ *Forschungszentrum Jülich GmbH, Nuclear Physics Institute IKP-2 - Jülich, Germany*

⁽²⁾ *RWTH Aachen University Physics Institute 3.b. - Aachen, Germany*

⁽³⁾ *INFN, Sezione di Milano - Milano, Italy*

⁽⁴⁾ *Università degli Studi di Milano - Milano, Italy*

received 4 November 2021

Summary. — This paper briefly describes a software, currently being developed in the nuclear physics institute IKP-2 of the Forschungszentrum Jülich, used in the context of sensitivity studies for solar neutrino measurements with JUNO. As the main strategy of the analysis consists of performing spectral fits on many pseudo-experiments a fast, reliable, and flexible software is needed in order to be able to quickly adjust to progresses with the experiment design and analysis strategy. JUST is being developed using continuous integration tools to ensure stability throughout different analysis steps, and is made so flexible as to be easily usable also for other experiments.

1. – JUNO

The Jiangmen Underground Neutrino Observatory [1] (JUNO) is a next generation liquid scintillator reactor neutrino experiment being built in the Guangdong province in China. JUNO is a multi-purpose experiment with a wide range of applications in neutrino physics, ranging from a mass-hierarchy determination to solar, geo, and atmospheric neutrino measurements, to detecting supernovae. Moreover JUNO will measure oscillation parameters, namely $\sin^2 \theta_{12}$, Δm_{21}^2 , and Δm_{32}^2 , with a precision of less than one percent.

The over 50-meter wide experimental hall, which was recently successfully dug out, stands under more than 700 m of granite overburden (1800 *m.w.e.*). It contains a 35.4 m diameter acrylic vessel containing 20 kt of LAB-based liquid scintillator, making it the largest liquid scintillator detector in the world.

In total, with the high light-yield of the scintillator and the high PMT coverage, along with a rigorous calibration campaign, JUNO is expected to reach an energy resolution of 3% at 1 MeV. The detector construction is expected to be completed in 2022.

(*) On behalf of the JUNO Collaboration.

2. – Motivation

The measurement of the rates of different solar neutrino species can shed light on different nuclear processes that take place in the very core of the Sun. They can be used to test and verify the standard solar model, to test different models of solar metallicity, and to look at other neutrino-related effects *e.g.* the MSW-effect [2, 3]. In this analysis we focus on low-energy solar neutrinos *i.e.* pep and 7Be neutrinos. These are neutrinos from the pp-chain [4].

The combination of high light-yield and low energy threshold of liquid scintillator detectors have already made pioneering measurements of low-energy solar neutrinos possible [5-7]. Because of the unprecedented size of the JUNO detector, its measurements of solar neutrinos can be expected to significantly surpass the results of past experiments, should background contributions be kept under control.

3. – Analysis strategy

Solar neutrinos are exclusively detected through elastic scattering off electrons in the liquid scintillator. While this allows for a low-threshold real-time detection of solar neutrinos, the presence of a lot of background makes it impossible to detect solar neutrinos on an event-by-event basis. Instead, events are measured over a given period of time and for each event an energy estimator is calculated and added to a histogram. A spectral fit on this histogram is then used to untangle signal from background contributions.

In JUNO the background mainly consists of intrinsic radioactive decays, *e.g.* from ${}^{232}Th$ and ${}^{238}U$ in the liquid scintillator, and from cosmogenic isotopes like ${}^{11}C$. In theory external gammas reach deep into the detector, but given JUNO's immense target mass a stringent fiducial volume cut can be applied which completely eliminates that background. Figure 1(b) shows a toy-spectrum of how the signal and data could be expected to look like. As one can see - it is not easy to extract neutrino rates without shape information of the signals and backgrounds.

To estimate JUNO's precision when measuring solar neutrino rates, and to study the effects of different variables on the final results, sensitivity studies are performed. This is done based on different assumptions, on the signal and background rates, which are beyond the scope of this paper. In order to estimate JUNO's sensitivity, many pseudo-experiments are simulated using Monte-Carlo generated Probability Density Functions (PDFs) for each particle species contributing to the total spectrum. In each pseudo-experiment the number of events per particle species is sampled from a Poisson distribution, then all species contributions are added to form the "data" spectra. These spectra are then all individually fitted and the resulting rate of each species is stored. The precision to the different fit species can then be directly read from those histograms.

During the sensitivity studies we expect to run this process many times and under different configurations, each with large numbers of individual fits. This can not only be used to precisely estimate the sensitivity, but also to optimise the measurement strategy and to study possible biases and inter-parameter correlations.

4. – Jülich Nusol Sensitivity Tool

4.1. Software and architecture. – The Jülich nUsol Sensitivity Tool (JUST) is designed to be as flexible as possible during any sensitivity studies. It is written in c++ using the c++14 standard, and compiled using a flexible CMake script.

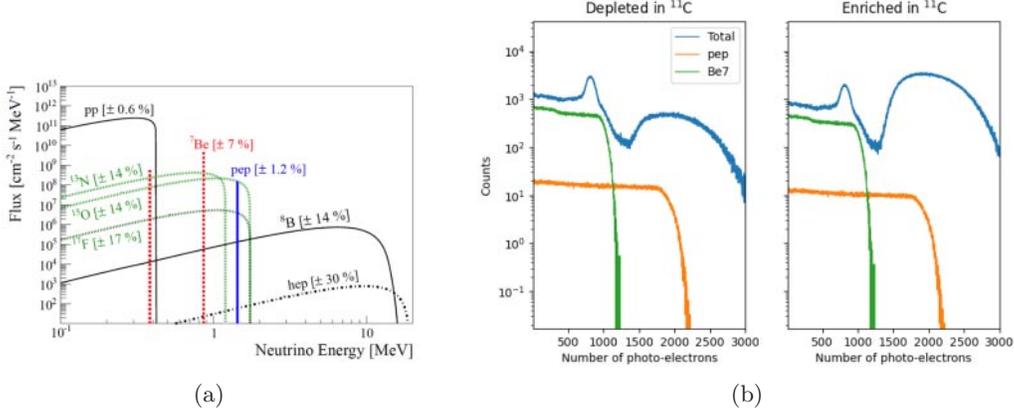


Fig. 1.: (a) Example solar neutrino spectrum as expected from the Sun [9] (b) Example dataset used in the analysis. The two plots denote a same dataset that was separated by a tag on ^{11}C -like and non- ^{11}C -like events. As one can see, the contribution from ^7Be neutrinos is quite high compared to the background, especially near the characteristic ^7Be shoulder near 1100 p.e. The *pep* contribution however is more difficult to measure as it is significantly lower than the background, with its most prominent differentiating feature quasi-hidden under ^{11}C events.

The main goal of JUST is to perform a binned likelihood fit on an arbitrary number of histograms. Concerning the data, there are two options for JUST. The first is to fit a pre-determined histogram from a ROOT [10] file. In this case JUST will read-in the data and the PDFs separately, make sure the binning is consistent, then perform a single fit. The second option is to perform so called "toy-data fits". With this option the user can decide to perform an arbitrary number of fits. Let N be the number of toy-data fits to be performed. JUST will then generate a poisson-fluctuated random number $N_{i,j}^{\text{poisson}}$ for each relevant PDF and each fit. Before each fit i , $N_{i,j}^{\text{poisson}}$ events will be sampled from each PDF j to create one data histogram to be subsequently fitted. The outputs from each individual fit are then saved for later study.

The architecture of JUST is fully modular, meaning that any module can in principle be removed and replaced if needed, making testing and development seamless. All modules are called by *main.cpp* that acts as an organiser. The modules are called in the following order:

- 1) *Parser* that reads from command line arguments and config files, thereby creating a global *NuFitConfig* object that is accessible from all other files
- 2) *DataReader* that opens different ROOT files according to *NuFitConfig* and converts ROOT histograms to c++ vectors (to gain speed in the fitting)
- 3) *ToyDataGenerator* that creates a generator for pseudo-data to be used in the fitter
- 4) *Fitter* that either carries out a fit on data, or iteratively calls the *ToyDataGenerator* and then fits each newly obtained dataset
- 5) *FitResults* that extracts the results from the fits, only keeps the relevant parameters, and converts them to more physically interpretable results
- 6) *OutputManager* that finally takes the *FitResult* objects to create plots and write all necessary output to file

4.2. *Binned template likelihood fit.* – Although extensions are planned in the future, the current version of the software strictly performs a template fit - in practice “only” finding scaling factors for each PDF. The current version of the software can fit an arbitrary number of histograms using a binned Poisson likelihood fit based on a likelihood ratio test as outlined in Baker&Cousins [11]. For a single histogram, the likelihood looks as follows:

$$(1) \quad \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=0}^{N_{bins}} y_i(\boldsymbol{\theta}) - n_i + n_i \cdot (\ln(n_i) - \ln(y_i(\boldsymbol{\theta})))$$

with

$$(2) \quad y_i = \sum_{j=0}^{N_{species}} \theta_j \cdot PDF_{i,j}$$

whereby \mathcal{L} is the log-likelihood, y_i is the fit function estimated by the parameters $\boldsymbol{\theta}$ in the bin i , $PDF_{i,j}$ is the value of the MC PDF of species j in bin i , and n_i is the number of entries in bin i . In case multiple histograms are included in the fit, the sum in eq. (1) will simply extend over the other histograms.

4.3. *Software usage.* – JUST is launched as a simple c++ executable with different configuration files whose paths must be specified as command line arguments. The first configuration file, commonly referred to as “general options” contains all possible details concerning the fitting. This includes, but is not limited to, the paths of the data and PDF files, the *fit range* in which to evaluate the likelihood, the exposure, the number of toy-data fits to perform, and the names of the histograms in the ROOT files. The second, “*specieslist*”, contains information about all of the PDFs to fit, including initial guesses for the fitter. Finally, “*toy_rates*” is very similar to *specieslist* but contains information about the creation of the toy-data spectra, including all of the species to include in the toy-data and their corresponding interaction rates. Separating *specieslist* and *toy_rates* in different files allows to fit the same data under different model assumptions, and vice-versa.

It is to be noted that the fit parameters used in JUST are entirely determined from the *specieslist* file. For each PDF, the user must specify a parameter name and a histogram in which this PDF/parameter should be considered. Should two PDFs share the same parameter name (for example when fitting the same particle species over several histograms), JUST will automatically detect it and treat both PDFs as a single fit variable. Similarly, JUST will detect the usage of different histograms. Thus, if a histogram is present in the data files but no parameter/PDF is set to use this histogram, JUST will ignore this histogram when constructing the likelihood. It is also possible to use *specieslist* to indicate that a parameter be fixed in the fit, or constrained with a Gaussian pull-term.

5. – Results and outlook

The result of an example fit can be seen in fig. 2. As one can see, the fitter is able to produce good agreements between model and data in this example. The results were

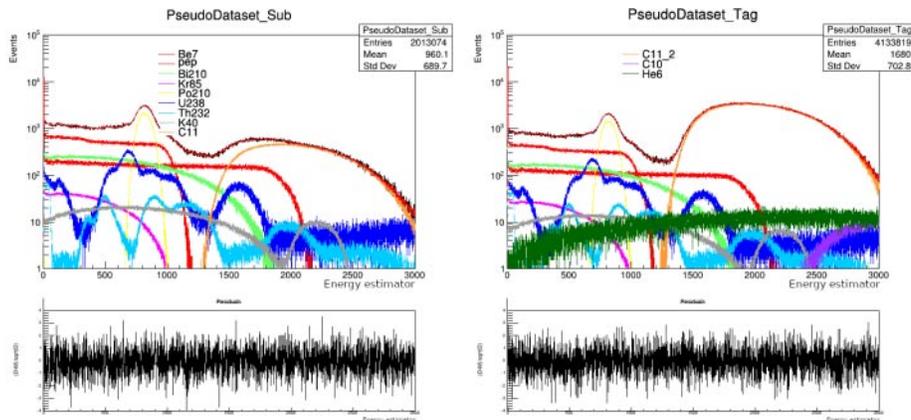


Fig. 2.: Results of a toy-data fit. This fit was performed on a dataset corresponding to (approximately) one year of exposure in the JUNO detector, given some background assumptions. The parameters labeled in the legend on the left plot are used in both histograms, and the parameters in the legend on the right plot as used solely in the right plot. As one can see, the residuals fluctuate around zero and the fit is a good match to the data.

verified down to a level of 10^{-4} by an independent team and software, and the biases to injected values in the toy-data fits were verified to a few percent for all neutrino species.

Overall, JUST was also shown to perform well on large scales ($\mathcal{O}(10^4)$ fits in one run) thanks to a strict memory management. The internal usage of c++ vectors instead of ROOT histograms lead to a speed-up of about a factor three in the fitting. The development of JUST was set-up with continuous integration tools to allow for seamless additions of new features. This allows the software to naturally grow as the analysis progresses. In the future we plan to simultaneously fit histograms of different kinematic variables, and to add features to facilitate systematic studies for JUNO's solar analysis. Due to JUST's easily customisable and fully modular structure it is also possible to use it for other current or future experiments with no modifications to the code itself.

REFERENCES

- [1] ABUSLEME A. *et al.*, arXiv:2104.02565 (2021).
- [2] McDONALDS A. B., ANTONELLI V., MIRAMONTI L., PENA GARAY C. and SERENELLI A., *Adv. High Energy Phys.*, **2013** (2013) 351926.
- [3] WOLFENSTEIN L., *Phys. Rev. D*, **17** (9) 2369.
- [4] SALARIS M. and CASSISI S., *Evolution of Stars and Stellar Populations* (John Wiley & Sons, Hoboken, USA).
- [5] KAMLAND COLLABORATION (ABE S. *et al.*), *Phys. Rev. C*, **81** (2010) 025807.
- [6] BOREXINO COLLABORATION, *Nature*, **562** (2018) 505.
- [7] BOREXINO COLLABORATION, *Nature*, **587** (2020) 577.
- [8] ABUSLEME A. *et al.*, *Chin. Phys. C*, **45** (2021) 1.
- [9] D'ANGELO D., *EPJ Web of Conferences*, **126** (2014).
- [10] "ROOT Data Analysis Framework", <https://root.cern.ch/>.
- [11] BAKER S. and COUSINS R. D., *Nucl. Instrum. Methods Phys. Res.*, **221** (1984) 437.