Flavour Symmetry Embedded - GLoBES (FaSE-GLoBES)

Jian Tang[1][*] and Tse-Chun Wang[1][†]

[1]*School of Physics, Sun Yat-Sen University, Guangzhou 510275, China*

Abstract

Neutrino models based on flavour symmetries provide the natural way to explain the origin of tiny neutrino masses. At the dawn of precision measurements of neutrino mixing parameters, neutrino mass models can be constrained and examined by on-going and up-coming neutrino experiments. We present a supplemental tool **Fla**vour **S**ymmetry **E**mbedded (**FaSE**) for **G**eneral **Lo**ng **B**aseline **E**xperiment **S**imulator (**GLoBES**). It can translate the neutrino mass model parameters to standard neutrino oscillation parameters, and offers prior functions in a user-friendly way. We demonstrate the robustness of **FaSE-GLoBE** with two examples on how the model parameters can be constrained and even whether the model is excluded by an experiment or not. We wish that this toolkit will facilitate the study of new neutrino mass models in an efficient and effective manner.

Keywords: Neutrino Oscillations, Leptonic Flavour Symmetry

---

[*] tangjian5@mail.sysu.edu.cn

[†] wangzejun@mail.sysu.edu.cn

I. INTRODUCTION

The discovery of neutrino oscillations points out the fact that neutrinos have mass, and provides an evidence beyond the Standard Model (BSM). This phenomenon is successfully described by a theoretical framework with the help of three neutrino mixing angles (θ_{12}, θ_{13}, θ_{23}), two mass-square splittings (Δm_{21}^2, Δm_{31}^2), and one Dirac CP phase (δ) [1–4]. Thanks to the great efforts in the past two decades, we almost have a complete understanding of such a neutrino oscillation framework. Nevertheless, more efforts in the neutrino oscillation experiments are needed to determine the sign of Δm_{31}^2, to measure the value of $\sin\theta_{23}$ more precisely, to discover the potential CP violation in the leptonic sector and even to constrain the size of δ [4]. For these purposes, the on-going long baseline experiments (LBLs), such as the NuMI Off-axis ν_e Appearance experiment (NOνA) [5] and the Tokai-to-Kamioka experiment (T2K) [6], can answer these questions with the statistical significance $\gtrsim 3\sigma$ in most of the parameter space. Based on the analysis in T2K and NOνA, the normal mass ordering ($\Delta m_{31}^2 > 0$), the higher θ_{23} octant ($\theta_{23} > 45°$), and $\delta \sim 270°$ are preferred so far [4]. The future LBLs, Deep Underground Neutrino Experiment (DUNE) [7], Tokai to Hyper-Kamiokande (T2HK) [8], and the medium baseline reactor experiment, the Jiangmen Underground Neutrino Observatory (JUNO) [9, 10] will further complete our knowledge of neutrino oscillations.

It is clear that neutrinos carry the peculiar mass smaller than $\mathcal{O}(1)$ eV among elementary particles, although the absolute value of each neutrino mass is unknown. In addition, one of the fundamental questions with regards to neutrino properties is how neutrino masses are generated naturally. The seesaw mechanism has been used to explain such a tiny mass in the neutrino sector. More degrees of freedom in a theoretical model, the less predictivity of observables in experiments. Furthermore, flavour symmetries are employed to reduce degrees of freedom in the neutrino mass model. These models can explain the origin of the neutrino mixing, and predict correlations of oscillation parameters (some of recent review articles are [11–17]). Before the discovery of non-zero θ_{13} by Daya Bay reactor neutrino experiment [18], the 'tri-bi-maximal' neutrino mixing (TBM) ansatz, which was proposed in

2002 by Horrison, Perkins, and Scott [19, 20], was in line with experimental data:

$$U_{\text{TBM}} = \begin{pmatrix} 2/\sqrt{6} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{3} & 1/\sqrt{2} \\ 1/\sqrt{6} & -1/\sqrt{3} & 1/\sqrt{2} \end{pmatrix}.$$

Given $\theta_{13} \approx 8°$ measured by several reactor neutrino experiments, we have to break this mixing pattern to to obtain such non-zero value of θ_{13}. One of popular proposals is to correct the tri-bi-maximal neutrino mixing such that

$$\sin\theta_{12} = (1+s)/\sqrt{3}, \ \sin\theta_{13} = r/\sqrt{2}, \ \text{and} \ \sin\theta_{23} = (1+a)/\sqrt{2}.$$

It is relatively easy for model builders to check the validity of the neutrino mass model and constrain model parameters by the public NuFit results [4]. However, there is no such a public toolkit to evaluate model predictions in future neutrino experiments. **G**eneral **Lo**ng **B**aseline **E**xperiment **S**imulator (**GLoBES**) [21, 22] is a convenient tool to simulate neutrino oscillation experiments via the Abstract Experiment Definition Language (AEDL) as one of the most popular and powerful tools in the community of neutrino oscillation physics. Some AEDL files to describe experiments are also available in **GLoBES** website, while the working group in the DUNE experiment also releases their neutrino flux information and detector descriptions in a AEDL file, provided in [23]. It is to be extended for the purpose of analysing flavour symmetry models in an universal way.

As we are entering the era of precision measurements in the neutrino oscillations, recent works pay more attentions to how the future neutrino experiments can be used to test these flavour-symmetry neutrino mass models, *e.g.* Ref. [24–28]. In this work, we will present our simulation toolkit **Fl**avour **S**ymmetry **E**mbedded - **GLoBES** (**FaSE-GloBES**) in a C-library to facilitate the study in the flavour symmetry neutrino models. **FaSE** is a supplemental tool for GLoBES, written in `c/c++` language, and allows users to assign any flavour symmetry model and analyze how a flavour symmetry model is constrained by the simulated neutrino oscillation experiments.

II. OVERVIEW FASE-GLOBES

FaSE-GLoBES is written in the `c` language, and consists of two source codes **FASE_GLoBES.c** and **model-input.c**. About these two c-codes, the user defines the

correlations between model inputs and standard neutrino mixing parameters in **model-input.c**, while **FASE_GLoBES.c** is an engine to perform analysis with user-specified experiments in a simulaton. Note that we define the standard neutrino mixing/oscillation parameters $\vec{\theta}_{OSC}$ (θ_{12}, θ_{13}, θ_{23}, Δm_{21}^2, Δm_{31}^2, δ) to separate from model parameters $\vec{\theta}_{Model}$ hereafter.
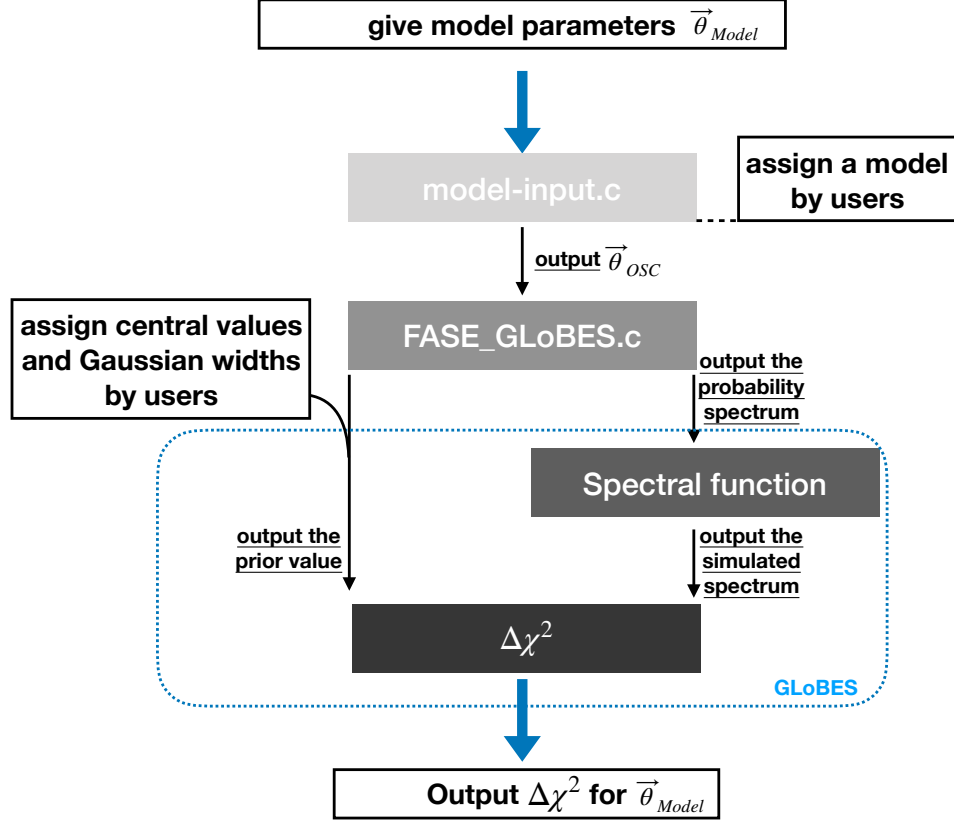

Figure 1: A scheme to correlate the model parameters with standard neutrino oscillation parameters. The error propagation is implemented in the simulation code up to the spectra analysis.

The concept of **FaSE-GLoBES** is shown in Fig. 1, in which three parts are shown: 1. **the parameter translation**, 2. **giving oscillation-parameter values**, and 3. **the χ^2-value calculation**. The idea behind this flow chart in Fig. 1 is that given a set of model parameters, the corresponding values in standard oscillation parameters are obtained by a

translation function, which is assigned by the user in **model-input.c**. And then, through **FASE_GLoBES.c** these oscillation-parameter values are passed into **GLoBES** library to simulate the event spectra so that we can perform physics analysis in the newly-defined χ^2.

Application Programming Interface (API) functions in **FaSE-GLoBES** are listed:

1. `MODEL_init`(N_{para}),

2. `FASE_glb_probability_matrix`,

3. `FASE_glb_set_oscillation_parameters`,

4. `FASE_glb_get_oscillation_parameters`,

5. `FASE_prior_OSC`,

6. `FASE_prior_model`.

The first one is to initialise **FaSE** with the number of input parameters N_{para}. The next three functions need to be included to replace the default **GLoBES** probability engine by the one that can read the output from **model-input.c**, as follows.

```
glbRegisterProbabilityEngine(6,
                           &FASE_glb_probability_matrix,
                           &FASE_glb_set_oscillation_parameters,
                           &FASE_glb_get_oscillation_parameters,
                           NULL);
```

This probability engine can work with oscillation or model parameters. It can be set by the user with the parameter `PARA` in the main code. If `PARA=STAN` (`PARA=MODEL`) the probability engine works with oscillation (model) parameters. The final two items on the API list are prior functions. Once the user gives the prior in oscillation (model) parameters, the user needs to call `FASE_prior_OSC` (`FASE_prior_model`) as follows.

```
glbRegisterPriorFunction(FASE_prior_OSC,NULL,NULL,NULL);
```

or

```
glbRegisterPriorFunction(FASE_prior_model,NULL,NULL,NULL);
```

We note that except for setting the probability engine and the prior function, the other parts in the main code should follow with the GLoBES manual.

III. MODEL SETTING

The function `MtoS` can do the translation from model parameters $\vec{\theta}_{Model}$ to oscillation parameters $\vec{\theta}_{OSC}$. After the user gives the array $\vec{\theta}_{Model}$ to the function `MtoS`, the output is the corresponding oscillation parameter $\vec{\theta}_{OSC}$, of which components are θ_{12}, θ_{13}, θ_{23}, δ, Δm_{21}^2, and Δm_{31}^2. For the first four components, values are given in the unit of **rad**, while the other two are in **eV^2**. These values will be passed into **FaSE_GLoBES** to simulate experimental spectra and compute prior values.

There are three methods to translate from $\vec{\theta}_{Model}$ to $\vec{\theta}_{OSC}$ in **FaSE-GLoBES**: the user can assign the correlation between the standard oscillation and model parameter sets, give the mixing matrix in model parameter U, or define the mass matrix in model parameters, which will be diagonalised by the function `STAN_OSC` to obtain the corresponding oscillation-parameter values. In the way of directly giving the relation between oscillation and model parameter sets, the user needs to provide

$$\vec{\theta}_{Model} = \vec{f}(\vec{\theta}_{OSC}), \tag{1}$$

in the function `MtoS`.

When the user give the mixing matrix U in model parameters, the corresponding mixing angles can be obtained easily through,

$$\tan\theta_{12} = \left|\frac{U_{e2}}{U_{e1}}\right|, \ \sin\theta_{13} = |U_{e3}|, \ \tan\theta_{23} = \left|\frac{U_{\mu3}}{U_{\tau3}}\right|. \tag{2}$$

After getting these mixing angles, we can easily get the Dirac CP phase δ with Jarlskog invariant J_{CP},

$$J_{CP} = c_{12}s_{12}c_{23}s_{23}c_{13}^2 s_{13}\sin\delta, \tag{3}$$

where c_{ij} and s_{ij} are $\cos\theta_{ij}$ and $\sin\theta_{ij}$, respectively. In `MtoS`, the user can pass the mixing matrix U to the function `STAN_OSC_U`, and can obtain the corresponding mixing matrix and the CP phase.

The oscillation parameters can also be obtained in the way based on

$$U^\dagger \mathcal{M}\mathcal{M}^\dagger U = \mathbf{M}^2, \ \text{where} \ \mathbf{M}_{\alpha\beta}^2 = m_\alpha^2 \delta_{\alpha\beta}, \tag{4}$$

where $\mathcal{M}$ ($\mathbf{M}$) is the neutrino mass matrix in the flavour (mass) state. The matrix $\mathcal{M}$ is given by the user with model parameters $\vec{\theta}_{Model}$. The mixing matrix U can be used to get

mixing angles and the CP phase, as Eqs. (2) and (3). The difference between any two diagonal elements of $\mathbf{M}$ ($\mathbf{M}_{ii} - \mathbf{M}_{jj} = m_i^2 - m_j^2$) is the mass-squared difference (Δm_{ij}^2). This diagnolisation in Eq. (4) can be done by the function `STAN_OSC`, which needs to be called in `MtoS` with outputs of the vector $\vec{\theta}_{OSC}$. We note that with non-diagonal mass matrix for charged leptons, this translation method is not suggested.

IV.   PRIOR SETTING

Given a set of values for model parameters, **FASE_GLoBES.c** will obtain the corresponding oscillation-parameter values from **model-input.c**, and will pass these values to simulate event spectra and to compute the prior values. Two gaussian prior functions are provided in **FaSE**: `FASE_prior_OSC` and `FASE_prior_model`. These two functions are constructed for different purposes. If the user gives the prior in oscillation (model) parameters, the user should register `FASE_prior_OSC` (`FASE_prior_model`) with the **GLoBES** function `glbRegisterPriorFunction`, as we introduced in Sec. II. The user also needs to assign the parameters `PARA=STAN` (`PARA=Model`), when the user prefers to give the prior in oscillation (model) parameters. The Gaussian prior is

$$\chi_{prior}^2 = \sum_i \frac{(\theta_i - \theta_i^c)^2}{\sigma_i^2}, \tag{5}$$

where θ_i is one of parameters constrained by prior, θ_i^c (σ_i) is the central value (Gaussian width) of the prior for θ_i. We note that θ_i can be either model ($\vec{\theta}_{Model}$) or oscillation parameters ($\vec{\theta}_{OSC}$). The values of θ_i^c and σ_i need to be given by the user in the main code through three arrays: `Central_prior`, `UPPER_prior`, and `LOWER_prior`, in which there are six components. To treat asymmetry of width for upper ($\theta_i > \theta_i^c$) and lower ($\theta_i < \theta_i^c$) Gaussian widths, we give values in two arrays `UPPER_prior`, and `LOWER_prior`, respectively. If the user gives the prior in model parameters, the order of each component follows with the setup of input of the probability engine. While the user gives the prior in oscillation parameters, the six components of these three arrays in order are θ_{12}, θ_{13}, θ_{23}, δ, Δm_{21}^2, and Δm_{31}^2. The first four parameter are in **rad**, and the final two are in **eV^2**.

Finally, some restrictions are imposed by the chosen flavour symmetry model. We set up these restrictions in the function `model_restriction` in `model-input.c`. In the function `model_restriction`, the user needs to *return* 1 once the restriction is broken. For

example, if the normal ordering is imposed, we give " `if (DMS31<0) { return 1;}` " in `model_restriction`, where `DMS31` is the variable for Δm_{31}^2. And, if there is no restriction, we simply *return* 0 in `model_restriction` as follows.

```
double model_restriction(double model []){ return 0;};
```

V.   THE DEFINITION FOR χ^2 (BASED ON GLOBES)

The user can use **FaSE-GLoBES** to constrain model parameters. Suppose we have the measurement $\vec{x}$ and the likelihood function $L(\vec{\theta}) = P(\vec{x}|\vec{\theta})$ for a set of parameters $\vec{\theta} = (\theta_1, ..., \theta_N)$, where $P(\vec{x}|\vec{\theta})$ is the probability function for data $\vec{x}$ in favour of the hypothesis $\vec{\theta}$. The constraint of model parameters can be obtained with the statistis parameter $\chi^2 \equiv \ln L(\vec{\theta})$. The expression χ^2 is used as the default **GLoBES** setting. In more detail, the χ^2 function, following the Poisson distribution, is constructed based on a log-likelihood ratio,

$$\chi^2(\vec{\theta}^{hyp}, \xi_s, \xi_b) = 2 \sum_i \left(\eta_i(\vec{\theta}^{hyp}, \xi_s, \xi_b) - n_i + n_i \ln \frac{n_i}{\eta_i(\vec{\theta}^{hyp}, \xi_s, \xi_b)} \right)$$
$$+ p(\xi_s, \sigma_s) + p(\xi_b, \sigma_b) + \chi^2_{prior}, \tag{6}$$

where i runs over the number of bins, $\eta_i(\vec{\theta}, \xi_s, \xi_b)$ is the assumed event rate in the ith bin and E_i is the central value in this energy bin. The vector $\vec{\theta}$ consists of model or oscillation parameters. The parameters ξ_s and ξ_b are introduced to account for the systematic uncertainties in the normalisation for the signal (subscript $_s$) and background (subscript $_b$) components of the event rate, and are allowed to vary in the fit as nuisance parameters. For a given set of parameters $\vec{\theta}$, the event rate in the ith energy bin is calculated as

$$\eta_i(\vec{\theta}^{hyp}, \xi_s, \xi_b) = (1 + \xi_s) \times s_i + (1 + \xi_b) \times b_i, \tag{7}$$

where s_i and b_i are the expected number of signal and background events in ith energy bin, respectively. The nuisance parameters are constrained by the Gaussian prior $p(\xi, \sigma) = \xi^2/\sigma^2$ with corresponding uncertainties σ_s and σ_b for the signal and background, respectively. Finally, χ^2_{prior} is a set of Gaussian priors for hypothesis, and is expressed as Eq. (5). After doing all minimisations, the user obtains the χ^2 value for a specific hypothesis $\vec{\theta}^{hyp}$, $\chi^2(\vec{\theta}^{hyp})$.

Based on the χ^2 function Eq. (6), we can study how model parameters can be constrained and whether a flavour-symmetry neutrino model is excluded by simulated experiments. In

the following we will demonstrate with typical examples how it works, before presenting some demonstrations in next two sections.

Applications

The user of **FaSE-GLoBES** is able to study how model parameters can be constrained by the simulated experiments. To do so, the user needs to simulate the true event spectrum n_i with a set of model ($\vec{\theta}_{Model}^{true}$) or oscillation parameters ($\vec{\theta}_{OSC}^{true}$), *i.e.* set up $n_i(\vec{\theta}_{Model}^{true})$ or $n_i(\vec{\theta}_{OSC}^{true})$. The hypothesis $\vec{\theta}_{Model}^{hyp}$ predicts the tested event spectrum $\eta_i(\vec{\theta}_{Model}^{hyp}, \xi_s, \xi_b)$. With the default settings for χ^2 function as Eq. (5) in **FaSE-GLoBES**, the user computes the statistical quantity,

$$\chi^2(\vec{\theta}_{Model}^{hyp}), \text{ with } n_i(\vec{\theta}_{Model}^{true}) \text{ or } n_i(\vec{\theta}_{OSC}^{true}). \tag{8}$$

We note that the minimum of χ^2 in the whole parameter space ($\chi^2_{min.}$) may not be 0. Therefore, to get the precision of model parameters, the user should use the value $\Delta\chi^2(\vec{\theta}_{Model}^{hyp}) \equiv \chi^2(\vec{\theta}_{Model}^{hyp}) - \chi^2_{min.}$, instead of $\chi^2(\vec{\theta}_{Model}^{hyp})$ itself. By varying different hypotheses $\vec{\theta}_{Model}^{hyp}$, we will obtain the allowed region of model parameters with the statistical quantity $\Delta\chi^2(\vec{\theta}_{Model}^{hyp})$.

The user can also study how well a flavour symmetry model explains the computed data, or predict whether the simulated experiment can exclude this model or not. In other words, the user studies the minimum of χ^2 for the flavour symmetry model $\vec{\theta}_{Model}$ as a hypothesis, by assuming different true oscillation values, *i.e.* different $\vec{\theta}_{OSC}^{true}$. To do so, one can compute the same statistical quantity in Eq. (8), while the true spectrum is varied with different true values $\vec{\theta}_{OSC}^{true}$. All model parameters are allowed to be varied with the user-defined prior. Finally, the user might adopt Wilk's theorem to interpret results [29]. When we compare nested models, the $\Delta\chi^2$ test statistics is a random variable asymptotically distributed as a χ^2-distribution with the number of degrees of freedom, which is equal to the difference in the number of free model parameters.

In following two sections, we will present examples to demonstrate how the user can do with **FaSE-GLoBES** to constrain the model parameter and to exclude a model by the simulated experiment configuration.

VI. CONSTRAINT OF MODEL PARAMETERS

FaSE-GloBES can be used to study how model parameters are constrained by simulated neutrino oscillation experiments as we introduced in Sec. V. We take *the tri-direct littlest seesaw* (TDLS) [30–32] as an example. In this model, the light left-handed Majorana neutrino mass matrix in the flavor basis is given by

$$
m_\nu = m_a \begin{pmatrix} 1 & \omega & \omega^2 \\ \omega & \omega^2 & 1 \\ \omega^2 & 1 & \omega \end{pmatrix} + e^{i\eta} m_s \begin{pmatrix} 1 & x & x \\ x & x^2 & x^2 \\ x & x^2 & x^2 \end{pmatrix} , \tag{9}
$$

where x, η, m_a, and the ratio $r \equiv m_s/m_a$ are four parameters to be constrained by simulated data. We note that from Eq. (9), $m_1 = 0$ and the normal mass ordering are imposed, and will need to be imposed in **FaSE-GLoBES**. Therefore, the restrictions in this model are $m_a > 0$ and $r > 0$.
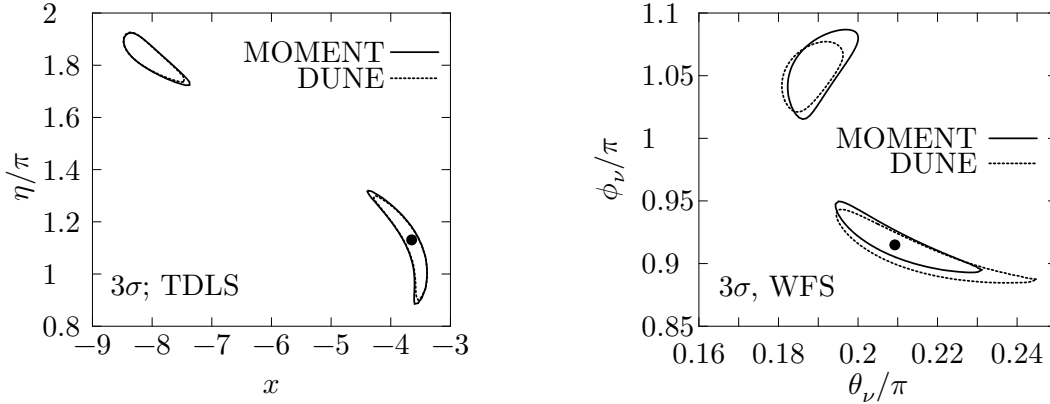

Figure 2: Examples for using **FaSE-GLoBES** to obtain the constraints of model parameters for tri-direct littlest seesaw (left) and the warped flavor symmetry (right), with simulated DUNE and MOMENT data. Two results are assumed the normal ordering. And, the dot denotes the model prediction with with NuFit4.0 results.

In left panel of Fig. 2, we study how parameters x and η can be constrained at 3σ by the MuOn-decay MEdium baseline NeuTrino beam experiment (MOMENT) [33] and DUNE experiment. Parameters r and m_a are varied with the prior that is given in standard oscillation parameters, according to the global-fit result NuFit4.0.

To show the generality of **FaSE-GLoBES**, we also present the similar result for another model – *the warped flavor symmetry* (WFS) [34]. This model predicts further simplified correlations that the standard oscillation parameters including mixing angles and the CP phase are functions of only two model parameters θ_ν and ϕ_ν,

$$
\begin{aligned}
\sin^2\theta_{12} &= \frac{1}{2 - \sin 2\theta_\nu \cos\phi_\nu}, \\
\sin^2\theta_{13} &= \frac{1}{3}(1 + \sin 2\theta_\nu \cos\phi_\nu), \\
\sin^2\theta_{23} &= \frac{1 - \sin 2\theta_\nu \sin(\pi/6 - \phi_\nu)}{2 - \sin 2\theta_\nu \cos\phi_\nu}, \\
J_{CP} &= -\frac{1}{6\sqrt{3}}\cos 2\theta_\nu.
\end{aligned}
\tag{10}
$$

The constraint of θ_ν and ϕ_ν for DUNE and MOMENT is presented in the right panel of Fig 2, in which we use the best fit of NuFit 4.0 result as the true values $(\theta_{12},\ \theta_{13},\ \theta_{23},\ \delta,\ \Delta m_{21}^2,\ \Delta m_{31}^2) = (33.82°,\ 8.61°,\ 49.6°,\ 215°,\ 7.39 \times 10^{-5}\ \mathrm{eV}^2,\ 2.525 \times 10^{-3}\ \mathrm{eV}^2)$. To reproduce results shown in [25], we do not include any priors. More details about these codes are presented in the user manual.
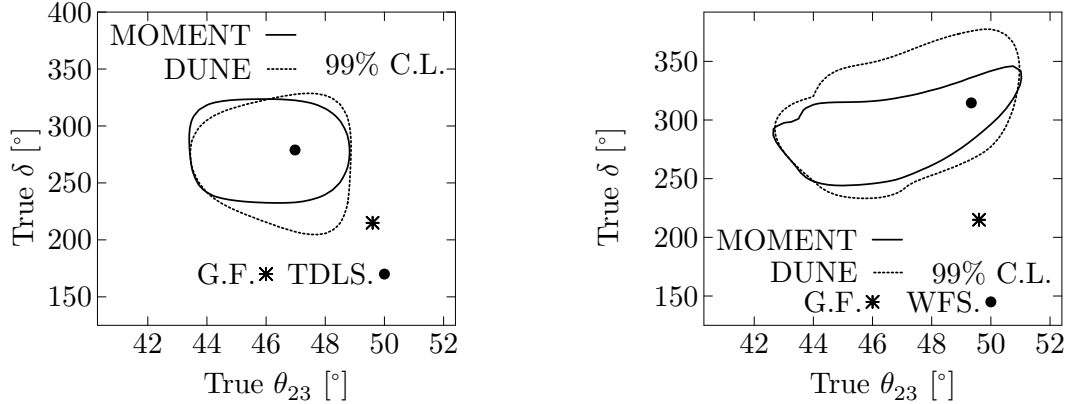
VII. MODEL TESTING

Figure 3: Examples for using **FaSE-GLoBES** to obtain the 2-D exclusion contour at 99% C.L. on the plane of true values of θ_{23} and δ for sub-models of TDLS (left) and of WFS (right) with DUNE and MOMENT. Two results are assumed the normal ordering. And, the dot (star) denotes the model prediction with NuFit4.0 results (best fit of NuFit4.0).

We can also study on whether a neutrino mass model is excluded or not, assuming different true values for oscillation parameters. In this example, we present testing submodels of TDLS in various true values of θ_{23} and δ. We compute the minimal χ^2 value for the model with r and m_a, varied with the prior in Eq. (5) according to NuFit4.0 results. Two parameters x and η are fixed at the best fit. In addition, the studied statistics function is exactly given by Eq. (6), but the true event rate n_i is predicted by a set of oscillation parameters, which will be varied in the code. We present this result in the left panel of Fig. 3. We further show the generality of **FaSE-GLoBES** by demonstrating the similar result for another model – WFS in Eq. (10) for DUNE and MOMENT, in right panel of Fig. 3. In this result, two mass-squared differences are fixed at the true values. More details are presented in the user manual.

VIII. SUMMARY AND CONCLUSIONS

With the progress of precision measurements in the neutrino experiments, and the success of numerous flavour symmetry theories to explain tiny neutrino masses, there are strong motivations to test and discriminate theoretical models by the next-generation neutrino oscillation experiments. We have presented a simulation toolkit **FaSE-GLoBES** to study the leptonic flavour symmetry models with neutrino oscillation experiments in a user-friendly way. **FaSE-GLoBES** contains two c-codes: **model-input.c** and **FASE_GLoBES.c**. While **FASE_GLoBES.c** works as a bridge between models and standard neutrino mixings, all inputs from the user need to be given in **model-input.c**. With the help of two main functions provided by **FaSE**, it is convenient to assign a flavour symmetry model and include Gaussian priors associated with oscillation or model parameters. Users are able to study how a flavour model can be studied by the simulated experimental configuration in various perspectives, *e.g.* model parameter constraints, hypothesis testing. **FaSE-GLoBES** will contribute to the selection and screening of underlying neutrino mass models by oscillation experiments. Further improvements and extensions can be envisioned as more requests are made by users.

Bibliography

[1] B. Pontecorvo, Sov. Phys. JETP **26**, 984 (1968), [Zh. Eksp. Teor. Fiz.53,1717(1967)].

[2] Z. Maki, M. Nakagawa, and S. Sakata, Prog. Theor. Phys. **28**, 870 (1962), [,34(1962)].

[3] B. Pontecorvo, Sov. Phys. JETP **7**, 172 (1958), [Zh. Eksp. Teor. Fiz.34,247(1957)].

[4] I. Esteban, M. C. Gonzalez-Garcia, A. Hernandez-Cabezudo, M. Maltoni, and T. Schwetz, JHEP **01**, 106 (2019), 1811.05487.

[5] D. S. Ayres et al. (NOvA) (2007).

[6] K. Abe et al. (T2K), Nucl. Instrum. Meth. **A659**, 106 (2011), 1106.1238.

[7] R. Acciarri et al. (DUNE) (2015), 1512.06148.

[8] K. Abe et al. (Hyper-Kamiokande Working Group) (2014), 1412.4673.

[9] Z. Djurcic et al. (JUNO) (2015), 1508.07166.

[10] F. An et al. (JUNO), J. Phys. **G43**, 030401 (2016), 1507.05613.

[11] G. Altarelli and F. Feruglio, Rev. Mod. Phys. **82**, 2701 (2010), 1002.0211.

[12] H. Ishimori, T. Kobayashi, H. Ohki, Y. Shimizu, H. Okada, and M. Tanimoto, Prog. Theor. Phys. Suppl. **183**, 1 (2010), 1003.3552.

[13] S. F. King and C. Luhn, Rept. Prog. Phys. **76**, 056201 (2013), 1301.1340.

[14] S. F. King, A. Merle, S. Morisi, Y. Shimizu, and M. Tanimoto, New J. Phys. **16**, 045018 (2014), 1402.4271.

[15] S. F. King, J. Phys. **G42**, 123001 (2015), 1510.02091.

[16] S. F. King, Nucl. Part. Phys. Proc. **265-266**, 288 (2015).

[17] S. F. King, Prog. Part. Nucl. Phys. **94**, 217 (2017), 1701.04413.

[18] F. P. An et al. (Daya Bay), Phys. Rev. Lett. **112**, 061801 (2014), 1310.6732.

[19] P. F. Harrison, D. H. Perkins, and W. G. Scott, Phys. Lett. **B530**, 167 (2002), hep-ph/0202074.

[20] Z.-z. Xing, Phys. Lett. B **533**, 85 (2002), hep-ph/0204049.

[21] P. Huber, M. Lindner, and W. Winter, Comput. Phys. Commun. **167**, 195 (2005), hep-ph/0407333.

[22] P. Huber, J. Kopp, M. Lindner, M. Rolinec, and W. Winter, Comput. Phys. Commun. **177**, 432 (2007), hep-ph/0701187.

[23] T. Alion et al. (DUNE) (2016), 1606.09550.

[24] P. Ballett, S. F. King, S. Pascoli, N. W. Prouse, and T. Wang, JHEP **03**, 110 (2017), 1612.01999.

[25] S. S. Chatterjee, P. Pasquini, and J. Valle, Phys. Lett. B **771**, 524 (2017), 1702.03160.

[26] G.-J. Ding, Y.-F. Li, J. Tang, and T.-C. Wang, Phys. Rev. **D100**, 055022 (2019), 1905.12939.

[27] J. Tang and T. Wang, Nucl. Phys. B **952**, 114915 (2020), 1907.01371.

[28] M. Blennow, M. Ghosh, T. Ohlsson, and A. Titov (2020), 2004.00017.

[29] S. S. Wilks, Annals Math. Statist. **9**, 60 (1938).

[30] S. F. King, JHEP **07**, 137 (2013), 1304.6264.

[31] S. F. King, JHEP **02**, 085 (2016), 1512.07531.

[32] S. F. King and C. Luhn, JHEP **09**, 023 (2016), 1607.05276.

[33] J. Cao et al., Phys. Rev. ST Accel. Beams **17**, 090101 (2014), 1401.8125.

[34] P. Chen, G.-J. Ding, A. D. Rojas, C. Vaquera-Araujo, and J. Valle, JHEP **01**, 007 (2016), 1509.06683.