

PELoRI - Probability Engine for Long-Range Interactions

PELoRI (Probability Engine for Long-Range Interactions) is a probability engine written for use with the General Long-Baseline Experiment Simulator (GLOBES) [1, 2]. If you use the files `lri_probability.h` and `lri_probability.c`, please cite the following paper:

Hooman Davoudiasl, Hye-Sung Lee, William J. Marciano, Sushant K. Raut

Long-range interactions for long-baseline neutrino experiments
arXiv:1804.0xxxx [hep-ph]

Theory of long-range interactions

We consider an anomaly-free combination of the gauge groups $U(1)_{B-L}$, $U(1)_{L_e-L_\mu}$, $U(1)_{L_\mu-L_\tau}$ and $U(1)_{L_\tau-L_e}$. The most general quantum number in this case can be written as $\eta(B-L) + \beta(L_e - L_\mu) + \gamma(L_\mu - L_\tau) + \delta(L_\tau - L_e)$ or equivalently $p_0 B + p_1 L_e + p_2 L_\mu + p_3 L_\tau$ where the p_i s are linear combinations of the real coefficients η , β , γ and δ . We call the fine structure constant of this new interaction α' . The protons, neutrons and electrons in the sun and earth generate a long-range potential V_{LR} due to this new gauge interaction that affects the oscillation of neutrinos at long-baseline experiments.

In the presence of the long-range potential, the propagation Hamiltonian of neutrinos becomes

$$\begin{bmatrix} V_{SM} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} p_1 V_{LR} & 0 & 0 \\ 0 & p_2 V_{LR} & 0 \\ 0 & 0 & p_3 V_{LR} \end{bmatrix},$$

where the first term is simply the usual Wolfenstein matter term. Both potentials V_{SM} and V_{LR} appear with a negative sign when we are dealing with antineutrinos. PELoRI uses this new potential to calculate neutrino oscillation probabilities within the GLOBES framework.

Parameters

Apart from the six standard oscillation parameters defined within GLoBES (indexed from 0 to 5), we introduce five more parameters as tabulated below. Each of the indices can be accessed within functions like `glbGetOscParams(glb_params,int)` by using the convenient macros, exactly like in standard GLoBES.

Parameter	Defined macro	Index number
α'	LRI_ALPHAprime	6
η	LRI_ETA	7
β	LRI_BET	8
γ	LRI_GAM	9
δ	LRI_DEL	10

Typically, a user will be interested in a specific new physics model defined by the values of η , β , γ and δ . Therefore these parameters will usually be kept fixed throughout the simulation. On the other hand, the value of the coupling strength α' will be subject to variation and marginalization, just like the standard oscillation parameters. (An exception would be if the user is interested in scanning over various possible models, in which case they may want to vary η , β , γ and δ as well.) For the purpose of this probability engine, no distinction has been made between α' and the remaining four parameters. The functions that have been provided for accessing them are completely general.

Functions

```
int lri_set_oscillation_parameters(glb_params p, void
*user_data)
```

This function provides the values of the oscillation parameters given in the `glb_params` parameter vector `p` to the probability engine. The argument `user_data` is unused, provided for compatibility with GLoBES. Returns 0 on succesful execution.

```
int lri_get_oscillation_parameters(glb_params p, void
*user_data)
```

This function stores the values of the oscillation parameters used by the

probability engine in the `glb_params` parameter vector `p`. The argument `user_data` is unused, provided for compatibility with GLoBES. Returns 0 on succesful execution.

```
int lri_probability_matrix(double P[3][3], int cp_sign, double
E, int psteps, const double *length, const double *density,
double filter_sigma, void *user_data)
```

This function calculates the oscillation probabilities for neutrinos of polarity given by `cp_sign` and energy `E`. The path of the neutrino is divided into `psteps` steps, whose lengths and matter densities are given by the arrays `length` and `density`, respectively. The results are stored in the array `P`. The arguments `filter_sigma` and `user_data` are currently unused. This function returns 0 on successful execution.

The user must register the PELoRI probability engine in the GLoBES `.c` file in order to calculate the probabilities. This is achieved using the GLoBES function `glbRegisterProbabilityEngine(11, &lri_probability_matrix, &lri_set_oscillation_parameters, &lri_get_oscillation_parameters, userdata);`, where 11 is the total number of oscillation parameters in PELoRI. The pointer `userdata` is unused and has been provided for compatibility with GLoBES. It can be set to `NULL`.

```
double lri_prior(const glb_params in, void *user_data)
```

Given the test values of the oscillation parameters in `in`, this function computes the corresponding prior. (The central values and errors are taken from those declared for the projection. An error of zero indicates no prior.) The argument `user_data` is unused. Returns the value of the total prior on succesful execution.

```
void lri_fixAllLRI(glb_projection p)
```

This function sets the projection flag for all the LRI parameters in the projection `p` to `GLB_FIXED`.

```
void lri_freeAllLRI(glb_projection p)
```

This function sets the projection flag for all the LRI parameters in the projection `p` to `GLB_FREE`.

`void lri_fixLRI(glb_projection p, int n)`

This function sets the projection flag for the LRI parameter denoted by the integer/macro `n` in the projection `p` to `GLB_FIXED`.

`void lri_freeLRI(glb_projection p, int n)`

This function sets the projection flag for the LRI parameter denoted by the integer/macro `n` in the projection `p` to `GLB_FREE`.

`void lri_setAllLRI(glb_params p, double v)`

This function sets the value of all the LRI parameters in the parameter vector `p` to `v`.

References

- [1] P. Huber, M. Lindner and W. Winter, *Comput. Phys. Commun.* **167**, 195 (2005).
- [2] P. Huber, J. Kopp, M. Lindner, M. Rolinec and W. Winter, *Comput. Phys. Commun.* **177**, 432 (2007).