General Long Baseline Experiment Simulator (GLoBES)

Pavan Poot Pandey

Department of Physics and Astrophysics
University of Delhi, New Delhi
Delhi, 110007

August 08, 2012

Overview

- Introduction
- Terms of usage of GLoBES
- GLoBES installation
- Writing a program in GLoBES
- Compilation and Creation of Data
- Plots and Result

Introduction

- GLoBES (General Long Baseline Experiment Simulator) is a flexible software package to simulate neu- trino oscillation long baseline and reactor experiments.
- GLoBES allows to simulate experiments with stationary neutrino point sources, where each experiment is assumed to have only one neutrino source.
- Such experiments are neutrino beam experiments and reactor experiments.
- Geometrical effects of a source distribution, such as in the sun or the atmosphere, can not be described.

Introduction..

- It is, however, possible to simulate beams with bunch structure, since the time dependence of the neutrino source is physically only important to suppress backgrounds.
- On the experiment definition side, either built-in neutrino fluxes (e.g., neutrino factory, β-Beam) or arbitrary fluxes can be used.
- GLoBES includes the simulation of neutrino oscillations in matter with arbitrary matter density profiles, as well as it allows to simulate the matter density uncertainty.
- With the C-library, one can extract the 2 for all defined oscillation channels for an experiment or any combination of experiments.

Terms of usage of GLoBES

- Referencing the GLoBES software
  P. Huber, M. Lindner and W. Winter
  Simulation of long baseline neutrino oscillation experiments with GLoBES,
  Comput. Phys. Commun. 167 (2005) 195,
  arXiv:hep-ph/0407333,

  P. Huber, J. Kopp, M. Lindner, M. Rolinec, and W. Winter
  New features in the simulation of neutrino oscillation experiments with GLoBES 3.0, arXiv:hep-ph/0701187,

- Referencing data in the GLoBES GLoBES wouldnt be useful without having high quality input data. Much of these input data have been published elsewhere and the authors of those publications would appreciate to be cited whenever their work is used.

GLoBES Installation

GLoBES is free software, one can redistribute it and/or modify it under the terms of the GNU General Public License. The GNU General Public License does not permit this software to be redistributed in proprietary programs.

- Pre-requisites for GLoBES installation For GLoBES installation, besides the usual things like a working libc one needs to have

  gcc The GNU compiler collection
  gcc.gnu.org

  GSL The GNU Scientific Library
  www.gnu.org/software/gsl/

GLoBES Installation instructions

GLoBES follows the standard GNU installation procedure. To compile GLoBES one will need an ANSI C-compiler.

- After unpacking the distribution the Makefiles can be prepared using the configure command,

  './configure'

- One can then build the library by typing,

  'make'

  A shared version of the library will be compiled by default.

- The libraries and modules can be installed using the command,

  'make install'

GLoBES installation instructions..

- One can remove the program binaries and object files from the source code directory by typing

  ‘make clean’.

- After the installation, since a library has been installed, it is necessary to run

  ’ldconfig’

Writing a program in GLoBES

- Initialisation of GLoBES: Before one can use GLoBES, one has to initialize the GLoBES library :
- Function 1: void glbInit(char *name) initializes the library libglobes and has to be called in the beginning of each GLoBES program.
- Function 2: int glbInitExperiment(char *infile, glb_exp *ptr, int *counter) adds a single experiment with the filename infile to the list of currently loaded experiments. The counter is a pointer to the variable containing the number of experiments, and the experiment ptr points to the beginning of the experiment list. The function returns zero if it was successful.
- Function 3: void glbClearExperimentList() removes all experiments from the internal list and resets all counters.

Writing a GLoBES program

- Units in GLoBES and the integrated luminosity:
- Function 4: void glbSetSourcePower(int exp, int fluxno, double power) sets the source power of experiment number exp and flux number fluxno to power.
- Function 5: double glbGetSourcePower(int exp, int fluxno) returns the source power of experiment number exp and flux number fluxno.
- Function 6: void glbSetRunningTime(int exp, int fluxno, double time) sets the running time of experiment number exp and flux number fluxno to time years.

Writing a GLoBES program..

- Function 7: double glbGetRunningTime(int exp, int fluxno) returns the running time of experiment number exp and flux number fluxno.
- Function 8: void glbSetTargetMass(int exp, double mass) sets the fiducial detector mass of experiment number exp to mass tons or kilotons (depending on the experiment definition).
- Function 9: double glbGetTargetMass(int exp) returns the fiducial detector mass of experiment number exp.

Writing a GLoBES program..

Thus, these functions also demonstrate how to use the assigned experiment number and others. These numbers run from 0 to the number of experiments-1, fluxes-1, etc., where the individual elements are numbered in the order of their appearance. Note that the source power and running time are quantities defined together with the neutrino flux, whereas the target mass scales the whole experiment. Thus, if one has, for instance, a neutrino and an antineutrino running mode, one can scale them independently.

- Handling Oscillation parameters: Before we can set the simulated event rates or access any oscillation parameters, we need to become familiar with the concept GLoBES uses for oscillation parameters.
- Function 10: glb_params glbAllocParams() allocates the memory space needed for a parameter vector and returns a pointer to it. All parameter values are initially set to nan (not a number).

Writing globes program..

- Function 11: void glbFreeParams(glb_params stale) frees the memory needed for a parameter vector stale and sets the pointer to NULL.
- Function 12: glb_params glbDefineParams(glb_params in, double theta12, double theta13,double theta23, double delta, double dm21, double dm31) as- signs the complete set of oscillation parameters to the vector in, which has to be allocated before. The return value is the pointer to in if the assignment was successful, and NULL otherwise.

Writing Globes program..

- Function 13: glb_params glbCopyParams(const glb_params source, glb_params dest) copies the vector source to the vector destination. The return value is NULL if the assignment was not successful.
- Function 14: void glbPrintParams(FILE *stream, const glb_params in) prints the parameters in in to the file stream. The oscillation parameters, all density values, and the number of iterations are printed as pretty output. Use stdout for stream if you want to print to the screen.

Compilation, Creation of data and plot

- The program "th13delta.c" can be compiled by "make th13delta"
- It can be run by "./th13delta"
- Before that care should be taken to modify the "make" file according to the program.
- "th13delta.gnuplot" file should also be modified according to the program.

Plot and Result

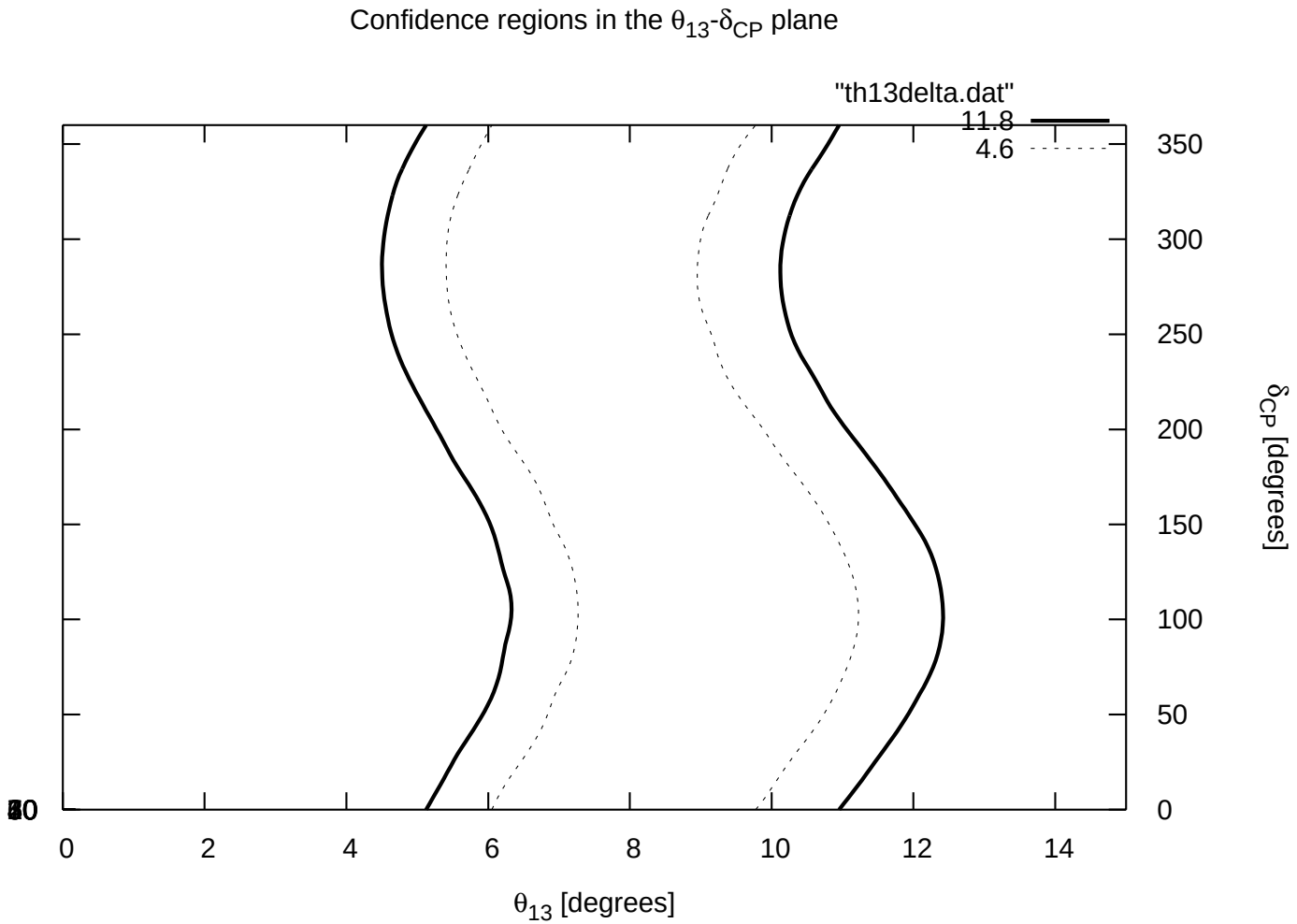

Fig1:

Correlation between θ_{13} and δ_{CP}.

- Basics of GLoBES have been studied and its installation is understood.
- A program in C is written to understand the features of GLoBES and it was found that GLoBES is a nice tool to understand neutrino experiments and its physics.
- Correlation between θ_{13} and δ_{CP} was studied and it was found that there is correlation between them.