GLoBES Course

Example-tour.c

Analysis of the example-tour.c file

```
int main(int argc, char *argv[])
{
  /* char* MYFILE=""; */
  char* MYFILE="gl-tour.dat"; /* if empty, write to screen, otherwise to file name given here */
  FILE* stream;
  if(strlen(MYFILE)>0) stream=fopen(MYFILE, "w");
   else stream = stdout;

  /* Initialize libglobes */
  glbInit(argv[0]);
```

```
/* Define my standard oscillation parameters */
double theta12 = asin(sqrt(0.8))/2;
double theta13 = asin(sqrt(0.001))/2;
double theta23 = M_PI/4;
double deltacp = M_PI/2;
double sdm = 7e-5;
double ldm = 2e-3;
```

```
/* Initialize one experiment NFstandard.glb */
glbInitExperiment("NFstandard.glb",&glb_experiment_list[0],&glb_num_of_exps);
```

```
/* Initialize a number of parameter vector(s) */
glb_params true_values = glbAllocParams();
glb_params fit_values = glbAllocParams();
glb_params central_values = glbAllocParams();
glb_params input_errors = glbAllocParams();
glb_params minimum = glbAllocParams();
```

```
/* Assign standard oscillation parameters */
glbDefineParams(true_values,theta12,theta13,theta23,deltacp,sdm,ldm);
glbSetDensityParams(true_values,1.0,GLB_ALL);
```

```
/* The simulated data are computed with "true_values" */
glbSetOscillationParameters(true_values);
glbSetRates();
```

```
/* Return some low level information */
int i;
fprintf(stream,"\nOscillation probabilities in vacuum: ");
for(i=1;i<4;i++) fprintf(stream,"1->%i: %g ",i,glbVacuumProbability(1,i,+1,50,3000));
fprintf(stream,"\nOscillation probabilities in matter: ");
for(i=1;i<4;i++) fprintf(stream,"1->%i: %g ",i,glbProfileProbability(0,1,i,+1,50));
fprintf(stream,"\n\n");
```

```
/* Set test/fit values slightly off true values at s22th13=0.0015 */
glbCopyParams(true_values,fit_values);
glbSetOscParams(fit_values,asin(sqrt(0.0015))/2,GLB_THETA_13);
```

```
/* Compute chi2 with systematics only for all experiments and rules */
double chi2,chi2b,chi2sum;
chi2 = glbChiSys(fit_values,GLB_ALL,GLB_ALL);
fprintf(stream,"chi2 with systematics only: %g\n\n",chi2);
```

```
/* What would be obtain from the appearance channels only? */
chi2 = glbChiSys(fit_values,0,0);
fprintf(stream,"This we would have from the CP-even appearance channel only: %g\n\n",chi2);
```

```
/* What would be obtain from the appearance channels only? */
chi2 = glbChiSys(fit_values,GLB_ALL,0)+
       glbChiSys(fit_values,GLB_ALL,1)+
       glbChiSys(fit_values,GLB_ALL,2)+
       glbChiSys(fit_values,GLB_ALL,3);          ;
fprintf(stream,"The sum over all rules gives again: %g\n\n",chi2);

/* Prepare minimizors: Set errors for external parameters: */
/* 10% for each of the solar parameters, 5% for the matter density */
glbDefineParams(input_errors,theta12*0.1,0,0,0,sdm*0.1,0);
glbSetDensityParams(input_errors,0.05,GLB_ALL);
glbSetCentralValues(true_values);
glbSetInputErrors(input_errors);
```

```c
/* Let's do the some with correlations included: */
chi2 = glbChiTheta13(fit_values,minimum,GLB_ALL);
fprintf(stream,"chi2 with correlations: %g \n",chi2);
fprintf(stream,"Position of minimum: theta12,theta13,theta23,delta,sdm,ldm,rho\n");
glbPrintParams(stream,minimum);
fprintf(stream,"Note that s22theta13 is unchanged/kept fixed: %g! \n\n",
  pow(sin(2*glbGetOscParams(minimum,GLB_THETA_13)),2));

/* Check what the difference is if we keep in addition deltacp fixed. */
/* This corresponds to projection onto theta13-deltacp-plane */
chi2 = glbChiTheta13Delta(fit_values,minimum,GLB_ALL);
fprintf(stream,"chi2 with correlations other than with deltacp: %g \n\n",chi2);

/* Similarly: what if we only take into account correlation with deltacp? */
/* For this, we need an arbitrary user-defined projection: */
glb_projection myprojection = glbAllocProjection();
glbDefineProjection(myprojection,GLB_FIXED,GLB_FIXED,GLB_FIXED,GLB_FREE,
  GLB_FIXED,GLB_FIXED);
glbSetDensityProjectionFlag(myprojection,GLB_FREE,GLB_ALL);
glbSetProjection(myprojection);
chi2 = glbChiNP(fit_values,minimum,GLB_ALL);
fprintf(stream,"chi2 with correlation only with deltacp: %g \n\n",chi2);
glbFreeProjection(myprojection);

/* Now: compare with statistics chi2 only */
glbSwitchSystematics(GLB_ALL,GLB_ALL,GLB_OFF);
chi2 = glbChiSys(fit_values,GLB_ALL,GLB_ALL);
glbSwitchSystematics(GLB_ALL,GLB_ALL,GLB_ON);
fprintf(stream,"chi2 with statistics only: %g\n\n",chi2);

/* Find position of sgn-degeneracy;  */
glbDefineParams(input_errors,theta12*0.1,0,0,0,sdm*0.1,ldm/3);
glbSetDensityParams(input_errors,0.05,GLB_ALL);
glbDefineParams(central_values,theta12,theta13,theta23,deltacp,sdm,-ldm);
glbSetDensityParams(central_values,1.0,GLB_ALL);
glbSetCentralValues(central_values);
glbSetInputErrors(input_errors);
chi2=glbChiAll(central_values,minimum,GLB_ALL);
fprintf(stream,"chi2 at minimum: %g \n",chi2);
fprintf(stream,"Position of minimum: theta12,theta13,theta23,delta,sdm,ldm,rho\n");
glbPrintParams(stream,minimum);
```

```
/* Now: multi-experiment setup */
/* Destroy parameter vectors, because they depend on no of exps */
glbFreeParams(true_values);
glbFreeParams(fit_values);
glbFreeParams(central_values);
glbFreeParams(input_errors);
glbFreeParams(minimum);

fprintf(stream,"\nNOW: TWO-EXPERIMENT SETUP NuFact@3000km+NuFact@7500km\n\n");
```

```
/* Initialize two experiments NFstandard.glb */
glbClearExperimentList();
glbInitExperiment("NFstandard.glb",&glb_experiment_list[0],&glb_num_of_exps);
glbInitExperiment("NFstandard.glb",&glb_experiment_list[0],&glb_num_of_exps);
```

```
/* Change baseline of second experiment to magic baseline */
double* lengths;
double* densities;
glbAverageDensityProfile(7500,&lengths,&densities);
fprintf(stream,"Magic baseline length: %g, Density: %g\n\n",lengths[0],densities[0]);
glbSetProfileDataInExperiment(1,1,lengths,densities);
free(lengths);
free(densities);
```

```
/* Initialize a number of parameter vector(s) again */
true_values = glbAllocParams();
fit_values = glbAllocParams();
central_values = glbAllocParams();
input_errors = glbAllocParams();
minimum = glbAllocParams();
```

```
/* Assign standard oscillation parameters */
glbDefineParams(true_values,theta12,theta13,theta23,deltacp,sdm,ldm);
glbSetDensityParams(true_values,1.0,GLB_ALL);
```

```
/* The simulated data are computed with "true_values" */
glbSetOscillationParameters(true_values);
glbSetRates();
```

```
/* Set test/fit values slightly off true values at s22th13=0.0015 */
glbCopyParams(true_values,fit_values);
glbSetOscParams(fit_values,asin(sqrt(0.0015))/2,GLB_THETA_13);
```

```c
/* Compute chi2 with systematics only for all experiments and rules */
chi2 = glbChiSys(fit_values,GLB_ALL,GLB_ALL);
fprintf(stream,"chi2 with systematics for all exps: %g\n",chi2);
```

```c
/* Compute chi2 with systematics only for each experiment */
chi2 = glbChiSys(fit_values,0,GLB_ALL);
fprintf(stream,"chi2 with systematics for 3000km: %g\n",chi2);
chi2b = glbChiSys(fit_values,1,GLB_ALL);
fprintf(stream,"chi2 with systematics for 7500km: %g\n",chi2b);
fprintf(stream,"The two add again to: %g\n\n",chi2+chi2b);
```

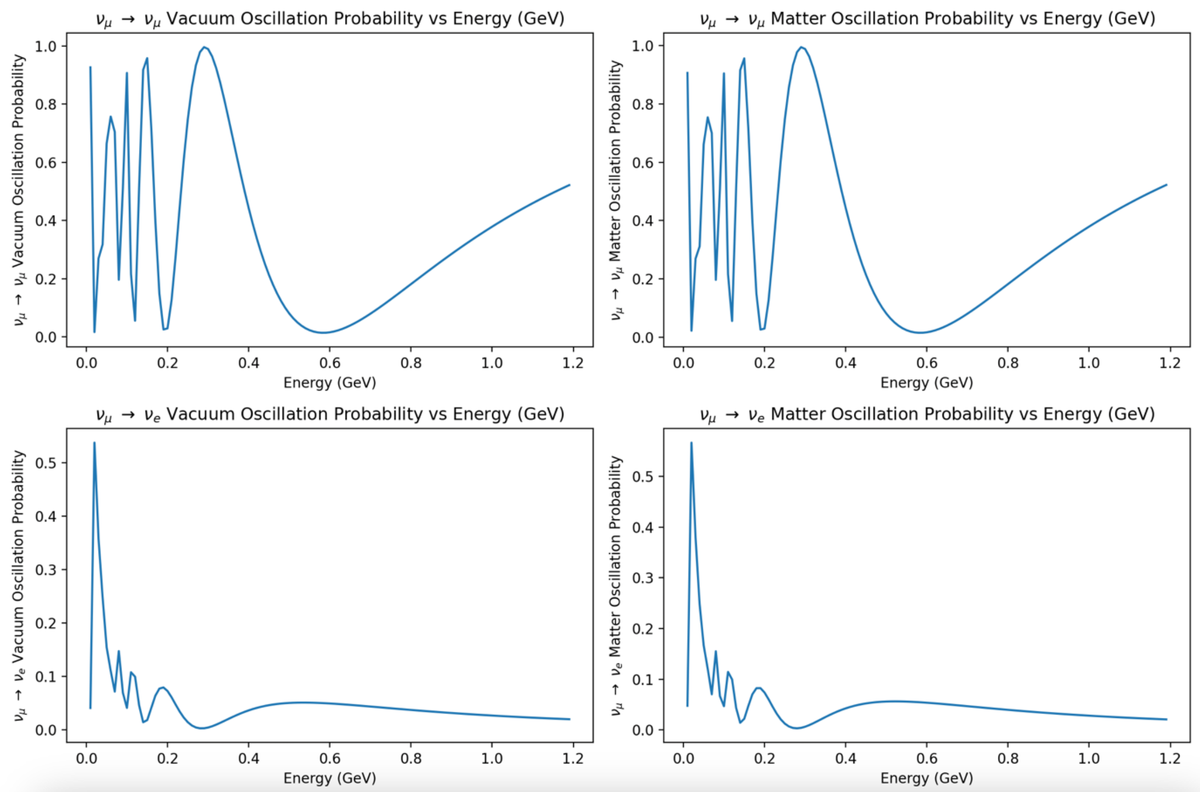
```c
/* Compute chi2 with correlations for each exp. and combination */
/* The sum of the two chi2 is not equal to the chi2 of the combination anymore, *
 * since the minimum may have a different position and the priors are only added once */
/* Note that there are now two densities in the vectors! */
/* Note that a minimum at a negative value of theta13 is unphysical. However, if there
   can be none at a positive value found at small enough chi2, there is none (Magic Baseline) */
glbDefineParams(input_errors,theta12*0.1,0,0,0,sdm*0.1,0);
glbSetDensityParams(input_errors,0.05,GLB_ALL);
glbSetCentralValues(true_values);
glbSetInputErrors(input_errors);
chi2 = glbChiTheta13(fit_values,minimum,0);
fprintf(stream,"chi2 with correlations for 3000km: %g \n",chi2);
glbPrintParams(stream,minimum);
chi2b = glbChiTheta13(fit_values,minimum,1);
fprintf(stream,"\nchi2 with correlations for 7500km: %g \n",chi2b);
glbPrintParams(stream,minimum);
chi2sum = glbChiTheta13(fit_values,minimum,GLB_ALL);
fprintf(stream,"\nchi2 with correlations for combination: %g \n",chi2sum);
glbPrintParams(stream,minimum);
fprintf(stream,"\nThe sum of the two chi2s is %g, whereas the total chi2 is %g !\n\n",chi2+chi2b,chi2sum);
```

```c
/* Find sgn-degeneracy for 3000km experiments and test if it is still there for comb. */
glbDefineParams(input_errors,theta12*0.1,theta13,theta23,deltacp,sdm*0.1,ldm/3);
glbSetDensityParams(input_errors,0.05,GLB_ALL);
glbDefineParams(central_values,theta12,theta13,theta23,deltacp,sdm,-ldm);
glbSetDensityParams(central_values,1.0,GLB_ALL);
glbSetInputErrors(input_errors);
glbSetCentralValues(central_values);
chi2=glbChiAll(central_values,minimum,0);
fprintf(stream,"chi2 at minimum, L=3000km: %g \n",chi2);
glbPrintParams(stream,minimum);
chi2=glbChiAll(minimum,minimum,GLB_ALL);
fprintf(stream,"\nchi2 for combination at minimum of L=3000km: %g \n",chi2);
glbPrintParams(stream,minimum);
```

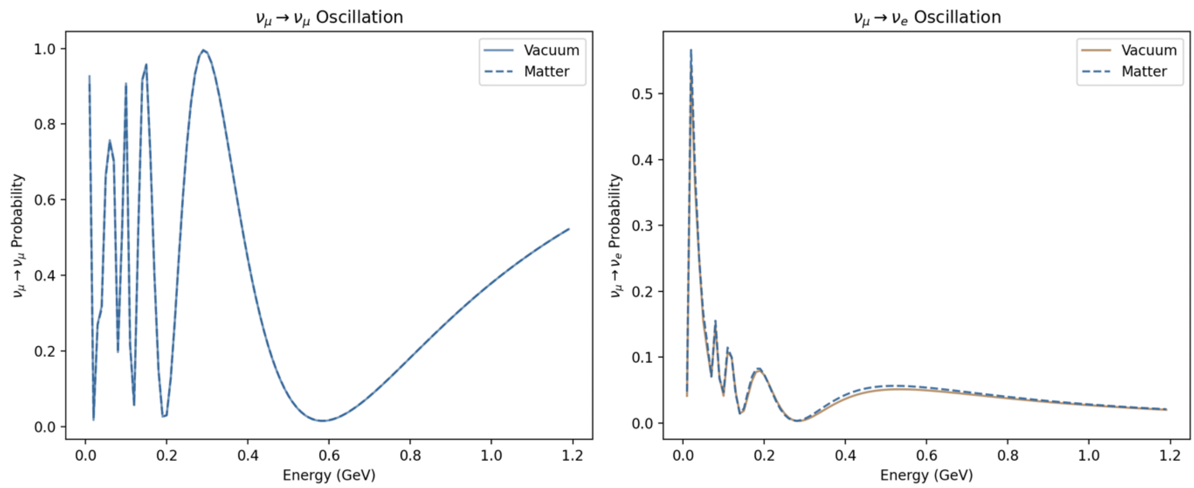
```
/* Destroy parameter vector(s) */
glbFreeParams(true_values);
glbFreeParams(fit_values);
glbFreeParams(central_values);
glbFreeParams(input_errors);
glbFreeParams(minimum);
if(strlen(MYFILE)>0) fclose(stream);

exit(0);
```

Task 1

This is to complete the T2K_simples.glb file with the presented auxiliary files in the folder exercise_1. Make graphs of oscillation probability against energy for v_e -> v_mu and all variations of this. Make one in python to compare. Will have to use the formula provided for oscillation probability from the notes.



A screenshot of the graphs showing the probabilities calculated by GloBES.



Task 2

```
/* ####### Beam flux ####### */

nuflux(#JHFplus)<
    @flux_file="JHFplus.dat"
    @time = 2   /* years */
    @power = 0.77   /* MW (proton intensity) */
    @norm = 6.93185
>

/* ####### Detector settings ###### */

$target_mass =  22.5    /* kt (fiducial mass)*/
/* ####### Number of energy bins in the simulation ###### */

$sampling_points = 20

/* ####### Number of bins in the analysis, and analysis energy window ####### */

$bins =     10
$emin =     0.4 /* GeV */
$emax =     1.2 /* GeV */


/* ####### Baseline setting ####### */

$profiletype =      1       /* 1 = constant Earth matter density */
$baseline =     295.0   /* km */
```

```
/* ####### Technical information ####### */

$filter_state = 0
$filter_value = 1000000


/* ####### Energy resolution ####### */

energy(#ERES)<                  /* Energy resolution function is Gaussian, with */
    @type = 1               /* sigma = sigma_e[0] * E  +  sigma_e[1] * sqrt(e) + sigma_e[3] */
    @sigma_e = {0.0,0.0,0.085}
>
```

```
/* ####### Cross sections ####### */

cross(#CC)<
    @cross_file = "XCC.dat"    /* Charged current */
>

cross(#NC)<
    @cross_file = "XNC.dat"    /* Neutral current */
>

cross(#QE)<
    @cross_file = "XQE.dat"    /* Quasi-elastic */
>
```

```
/* ####### Channel definitions ####### */

/* The format is
 *      @channel =        <Name>: CP sign:initial: final : cross   : energy resolution
 *                              :        :flavor : flavor: sections: function
 */

/* 0 */
channel(#nu_mu_disappearance_CC)<
    @channel =  #JHFplus:   +:  m:  m:  #CC:    #ERES
>

/* 1 */
channel(#NC_bckg)<
    @channel =  #JHFplus:   +:  NOSC_m: NOSC_m: #NC:    #ERES
>

/* 2 */
channel(#nu_e_beam)<
    @channel =  #JHFplus:   +:  e:  e:  #CC:    #ERES
>

/* 3 */
channel(#nu_e_bar_beam)<
    @channel =  #JHFplus:   -:  e:  e:  #CC:    #ERES
>

/* 4 */
channel(#nu_e_appearance_QE)<
    @channel =  #JHFplus:   +:  m:  e:  #QE:    #ERES
>

/* 5 */
channel(#nu_mu_disappearance_QE)<
    @channel =  #JHFplus:   +:  m:  m:  #QE:    #ERES
>

/* 6 */
channel(#nu_e_appearance_CC)<
    @channel =  #JHFplus:   +:  m:  e:  #CC:    #ERES
>
```

```
/* ####### Setting the rules ####### */


rule(#NU_E_Appearance_QE)<
        /* Signal channels and associated systematical errors */
    @signal = 0.50498@#nu_e_appearance_QE
    @signalerror = 10. : 0.0001       /* Format is <normalization error> : <energy "tilt" error> */

        /* Background channels and associated systematical errors */
    @background =  0.00032671@#nu_mu_disappearance_CC  : 0.0056373@#NC_bckg  : 0.50498@#nu_e_beam : 0.50498@#nu_e_bar_beam
    @backgrounderror = 0.05 : 0.05    /* follow hep-ph/0504026: 5 per cent background tilt error */

        /* Analysis method (chi^2 formula) */
    @sys_on_function = "chiTotalRatesTilt"    /* chi^2 formula for systematics switched ON  */
    @sys_off_function = "chiSpectrumTilt"     /* chi^2 formula for systematics switched OFF */
>

rule(#NU_MU_Disapperance_QE)<
    @signal = 0.9@#nu_mu_disappearance_QE
    @signalerror = 0.025 : 0.0001

    @background = 0.0056373@#NC_bckg
    @backgrounderror = 0.2 : 0.0001

    @sys_on_function = "chiTotalRatesTilt"
    @sys_off_function = "chiSpectrumTilt"
>

rule(#NU_E_Appearance_CC)<
    @signal = 0.50498@#nu_e_appearance_CC
    @signalerror = 0.05 : 0.0001     /* follow hep-ph/0504026: 5 per cent norm errors for appearance */

    @background =  0.00032671@#nu_mu_disappearance_CC  : 0.0056373@#NC_bckg  : 0.50498@#nu_e_beam : 0.50498@#nu_e_bar_beam
    @backgrounderror = 0.05 : 0.0001    /*  do not use 0.05 for BG tilt here, since introduced twice uncorrelated then! */

    @sys_on_function = "chiTotalRatesTilt"
    @sys_off_function = "chiNoSysTotalRates"
>
```

The Task:

Implementing a neutrino experiment

Imagine that you want to build a neutrino experiment and want to explore its capabilities through a simulation in GLoBES. The detector will consist of 500 kT of water, where a neutrino beam will deliver either only electron neutrinos or only electron antineutrinos, depending on the configuration used. Information of flux and cross-sections are contained in the files "NUEplus.dat", "NUEminus.dat", "XCC.dat" and "XNC.dat".

The neutrinos will travel through a medium with a density of 2.95 g/cm^3 until they reach the detector, located 350 km from the source that produces the neutrinos. Are you interested in exploring oscillation channels $P(\nu_e \to \nu_\mu)$ and $P(\bar{\nu}_e \to \bar{\nu}_\mu)$ Assume that the error in reconstructing the energy of muon and electron neutrinos is σ_e = 0.085 GeV.

The energy bins are wide and a range from 0.2 to 2 GeV. The efficiencies in the reconstruction of muon neutrinos and antineutrinos for each of these 18 energy bins are contained in the files NUMUeff.dat and NUMUBAReff.dat. For background, assume that it is possible to reject 99.9% of neutral current (NC) events and 99.9% of charged current (CC) events from the electron neutrinos intrinsic to the beam (which have not oscillated). Finally, assume that the normalization error of the signal is 2.5% and of the background is 20%.

Questions:

a. Assuming that you do not know what the units the flux are, but you know that in 1 year, you would expect to have 13503.5 non-oscillated electronic neutrino events and 4612.18 electronic antineutrinos, calculate the normalization factor of your experiment;
b. Calculate, given the current oscillation parameters for normal mass hierarchy, the spectrum of electron neutrino events;
c. Fixing the other oscillation parameters (for simplicity, we generally have errors associated with them, or we do not know their values), calculate the ability of your experiment to measure theta_13 and delta_CP at 95% CL and 99% CL. Assume 2 years of exposure to the electron neutrino beam and 6 years of exposure to the electron antineutrino beam.

Concerning T2K:

- L = 295 km
- E = [0.4, 1.2] GeV

A:

Take the normalisation constants to be one and set the exposure time to 1 as well. Now running the GLoBES initial file "TOYDfraft", the resulting data can be used to find the expected normalisation constants:

```
/* ####################### Definição do Fluxo >

nuflux(#NUE_FLUX)<
    @flux_file = "NUEplus.dat"
    @time = 1   /* 2 */
    @norm = 1   /* 2e7 */

nuflux(#ANUE_FLUX)<
    @flux_file = "NUEminus.dat"
    @time = 1   /* 6 */
    @norm = 1   /* 2e7 */
>
```

It is visible that the provided values are commented out. When running the file using the "th13delta.c" file, the following is outputted:

Issues between MacOs and Linux:

```
(base) paolominhas@Paolos-MacBook-Air-4 exercicio_2 % file th13delta
th13delta: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=5a3d2f6fb888ff75fea3e1a30f6f41e26c067e5a, fo
r GNU/Linux 3.2.0, with debug_info, not stripped
```

The makefile uses ELF which works for linux, but in fact macOS uses Mach-O, and macOS cannot execute Linux binaries natively. `CC = clang` Inserting CC = clang rather than gcc fixes this hopefully.

```
O número total de eventos de nu_e é de:  0.000499962
 O número total de eventos de anu_e é de:  0.000173579
```

And therefore we need the following normalisation factors:

```
nuflux(#nue_plus)<
    @flux_file = "NUEplus.dat"
    @time = 1
    @norm = 2.4966e7
>
```

```
nuflux(#nue_minus)<
    @flux_file = "NUEminus.dat"
    @time = 1
    @norm = 2.49513e7
>
```

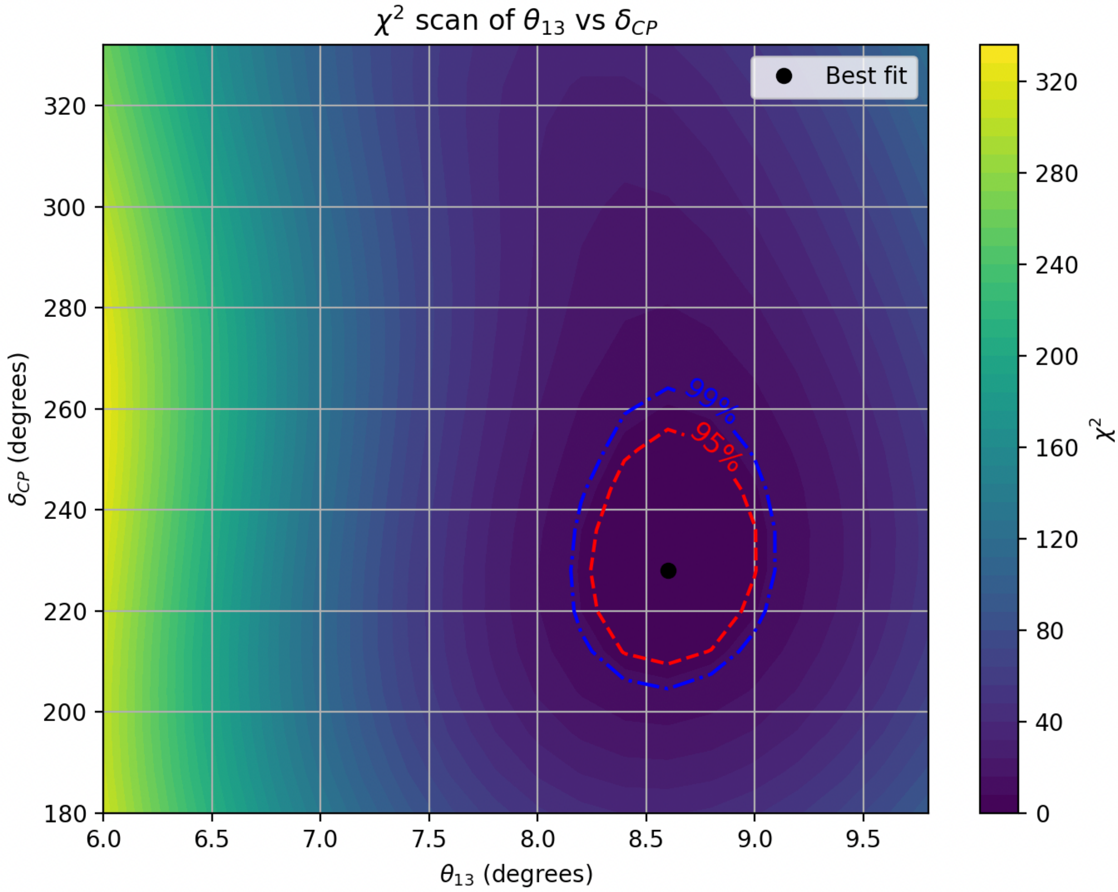
Giving the required output:

```
O número total de eventos de nu_e é de:  12482
 O número total de eventos de anu_e é de:  4331.02
```

B:

```
double theta12 = asin(sqrt(0.303));
double theta13 = asin(sqrt(0.02246));
double theta23 = asin(sqrt(0.450));
double deltacp = 230*M_PI/180;
double sdm = 7.42e-5;
double ldm = 2.51e-3;
```

C:



Example 6

Examining how extra parameters (new physics) can be added in Example 6

Define a new parameter, in this case it is σ_E.

```
#define GLB_SIGMA_E 6          /* Index of non-standard parameter sigma_E */
```

Then when setting the oscillation parameters set and get as follows. The first line stores the oscillation parameters in internal data structures. The second line is part of a different function that writes oscillation parameters from internal data structures into p.

```
sigma_E = glbGetOscParams(p, GLB_SIGMA_E);
glbSetOscParams(p, sigma_E, GLB_SIGMA_E);
```

Now you calculate the probability using this extra parameter:

$$P_{\bar{e}\bar{e}} = c_{13}^4 \left\{1 - \frac{1}{2}\sin^2(2\theta_{12})[1 - \quad \cos(2\Delta_{21})]\right\}$$
$$+\frac{1}{2}\sin^2(2\theta_{13})[D_3\ c_{12}^2\cos(2\Delta_{31}) + D_3\ s_{12}^2\cos(2\Delta_{32})] + s_{13}^4$$

Which is not equal to this file as only uses D_3 whereas we use D_31 ,21 & 32 but the same principles apply.

```
/* Compute P_ee */
s12 = sin(th12);
c12 = cos(th12);
s13 = sin(th13);
c13 = cos(th13);
t = L / (4.0 * E);
Delta21 = sdm * t;
Delta31 = ldm * t;
Delta32 = Delta31 - Delta21;
t = M_SQRT2 * sigma_E / E;
D21 = exp(-square( Delta21 * t ));
D31 = exp(-square( Delta31 * t ));
D32 = exp(-square( Delta32 * t ));
P[0][0] = square(square(c13)) * ( 1 - 2.0*square(s12*c12)*(1 - D21*cos(2.0*Delta21)) )
          + 2.0*square(s13*c13) * ( D31*square(c12)*cos(2.0*Delta31)
                                  + D32*square(s12)*cos(2.0*Delta32) )
          + square(square(s13));

return 0;
```

Task 3:

The folder exercise_3 contains the implementation of the LSND (Liquid Scintillator Neutrino Detector) experiment. LSND ran from 1993 to 1998 and searched for sterile neutrino signatures. The experiment reported an unexpected excess of events that could be related to electron neutrino candidates oscillating at a short distance.

The simplest model that includes sterile neutrinos is the so-called “3+1” model, where only one sterile neutrino is included in the mix and modifies the neutrino oscillation pattern. Signatures of this modification can be sought in neutrino experiments. Assuming that we also introduce a fourth mass state, where we will have $\Delta m^2_{41} \approx 1\mathrm{eV}^2$, the oscillation probability, considering the baseline and energy range of the LSND, will be simplified into:

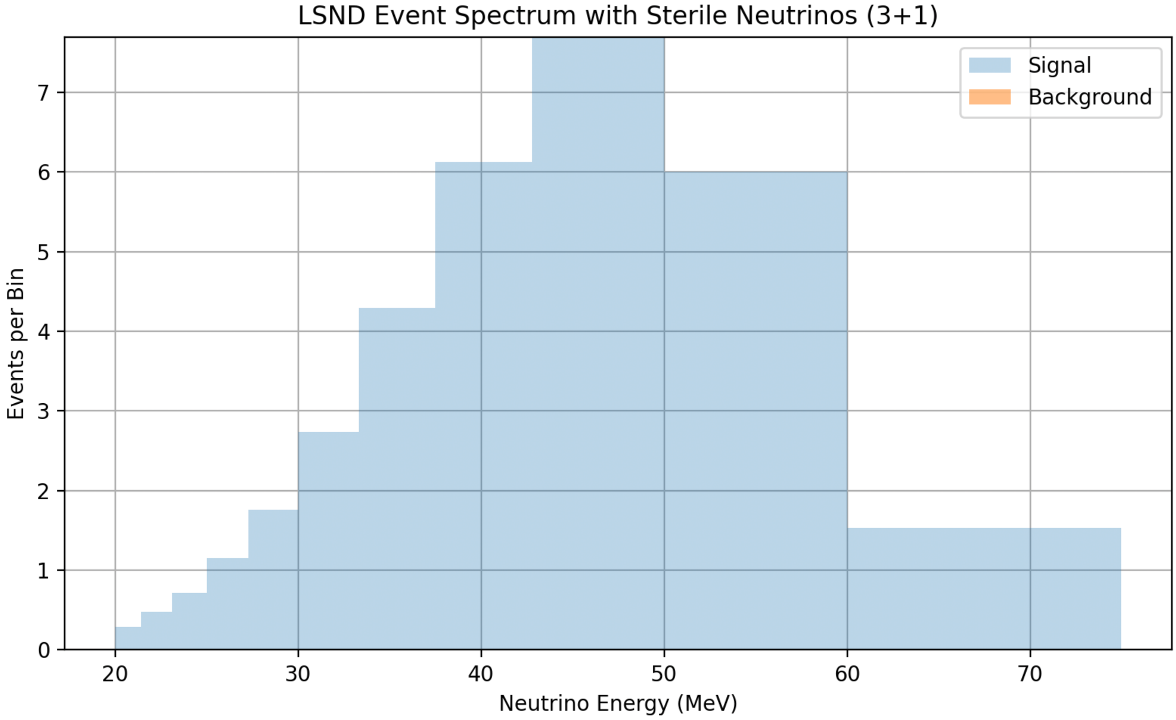
$$P^{SB}_{\overset{(-)}{\nu}_\mu \to \overset{(-)}{\nu}_e}(E, L) = \underbrace{\boxed{4|U'_{\mu 4}|^2 |U'_{e4}|^2}}_{\sin^2 2\theta_{\mu e}} \sin^2\left(1.27 \Delta m^2_{41} L/E\right)$$

As from the power point. Answer the questions:

a. Implement the oscillation probability of muon antineutrinos into electron antineutrinos for LSND
b. Plot the spectrum of events for the values
c. Calculate the LSND chi^2 considering 3+1 neutrino scenario and plot the parameter values that best fit the data at 95% and 99% confidence level. Are the minimum chi^2 for $\sin^2 2\theta_{\mu e}$ & Δm^2_{41} the same as these values:
    a. $\sin^2 2\theta_{\mu e} = 0.00316228$
    b. $\Delta m^2_{41} = 1.12\mathrm{eV}^2$

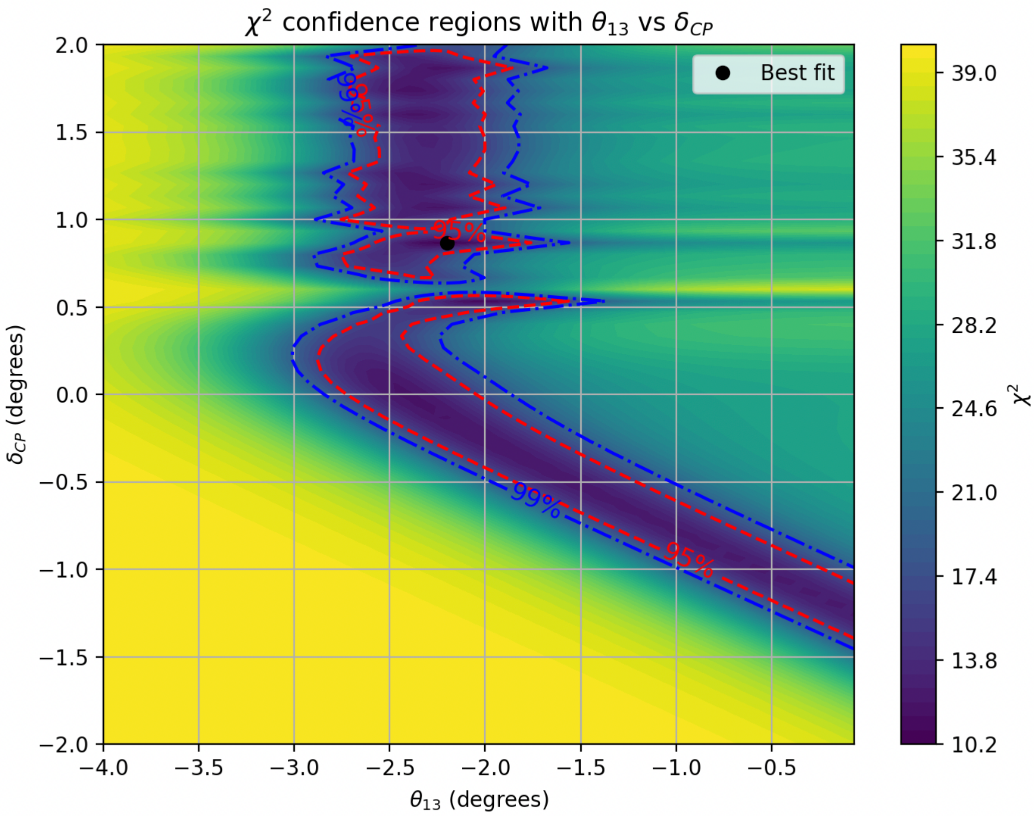
A:

B:



LSND Event Spectrum with Sterile Neutrinos (3+1)
Signal
Background
Events per Bin
Neutrino Energy (MeV)
0
1
2
3
4
5
6
7
20
30
40
50
60
70

C:



χ^2 confidence regions with θ_{13} vs δ_{CP}
Best fit
δ_{CP} (degrees)
θ_{13} (degrees)
χ^2
99%
95%
39.0
35.4
31.8
28.2
24.6
21.0
17.4
13.8
10.2

Task 4:

$$\chi^2 = \sum_{i=1}^{N} \frac{\left(n^i_{\text{true}} - S^i_{\text{test}}(a,b) - B^i_{\text{test}}(c,d)\right)^2}{n^i_{\text{true}}} + \frac{a^2}{\sigma_a^2} + \frac{b^2}{\sigma_b^2} + \frac{c^2}{\sigma_c^2} + \frac{d^2}{\sigma_d^2}$$

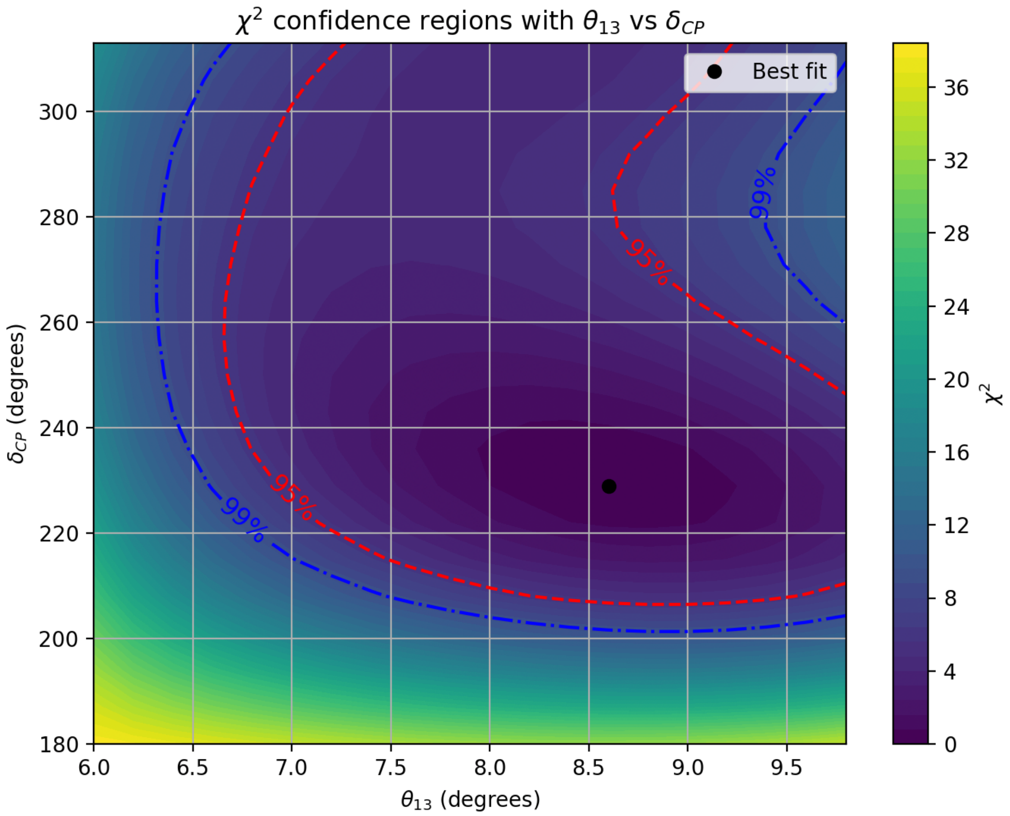
$$S^{\text{total}}_{\text{test}} = \sum_{i=1}^{N} S^i_{\text{test}}$$

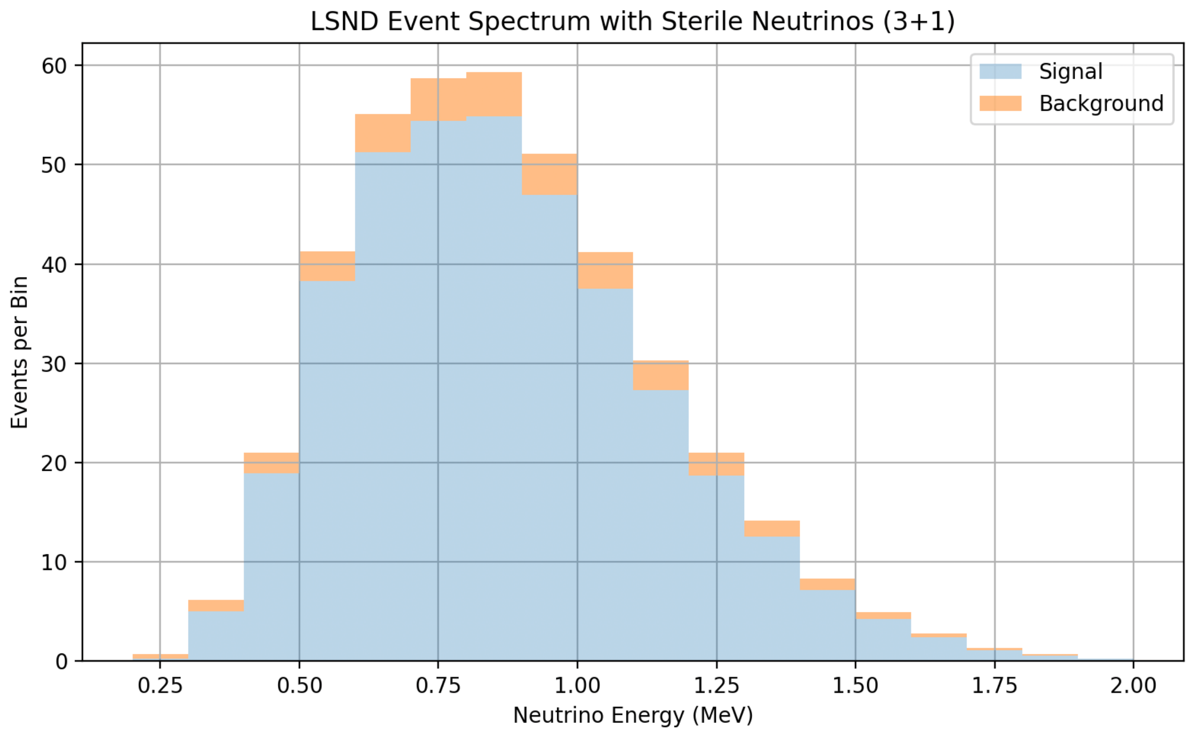
$$B^{\text{total}}_{\text{test}} = \sum_{i=1}^{N} B^i_{\text{test}}$$

$$n^{\text{total}}_{\text{true}} = \sum_{i=1}^{N} n^i_{\text{true}}$$

$$S^{\text{total}}_{\text{test}}(a,b) = (1+a) S^{\text{total}}_{\text{test}} + b\, S^{\text{total}}_{\text{test}} \frac{\left[E^i_{\text{rec}} - 0.5(E^{\max}_{\text{rec}} + E^{\min}_{\text{rec}})\right]}{\left[E^{\max}_{\text{rec}} - E^{\min}_{\text{rec}}\right]}$$

$$B^{\text{total}}_{\text{test}}(c,d) = (1+c) B^{\text{total}}_{\text{test}} + d\, B^{\text{total}}_{\text{test}} \frac{\left[E^i_{\text{rec}} - 0.5(E^{\max}_{\text{rec}} + E^{\min}_{\text{rec}})\right]}{\left[E^{\max}_{\text{rec}} - E^{\min}_{\text{rec}}\right]}$$





Task 5:

Task 6:

Final Project:

Final Project:

You must download the folder “final_project”. In it, you will find the official implementation of the future DUNE experiment at GLoBES (https://arxiv.org/src/2103.04797v2/anc) with all dependent files. Use the file dune-proj.c (it is practically empty!) and do the following tasks, using the following as true values:

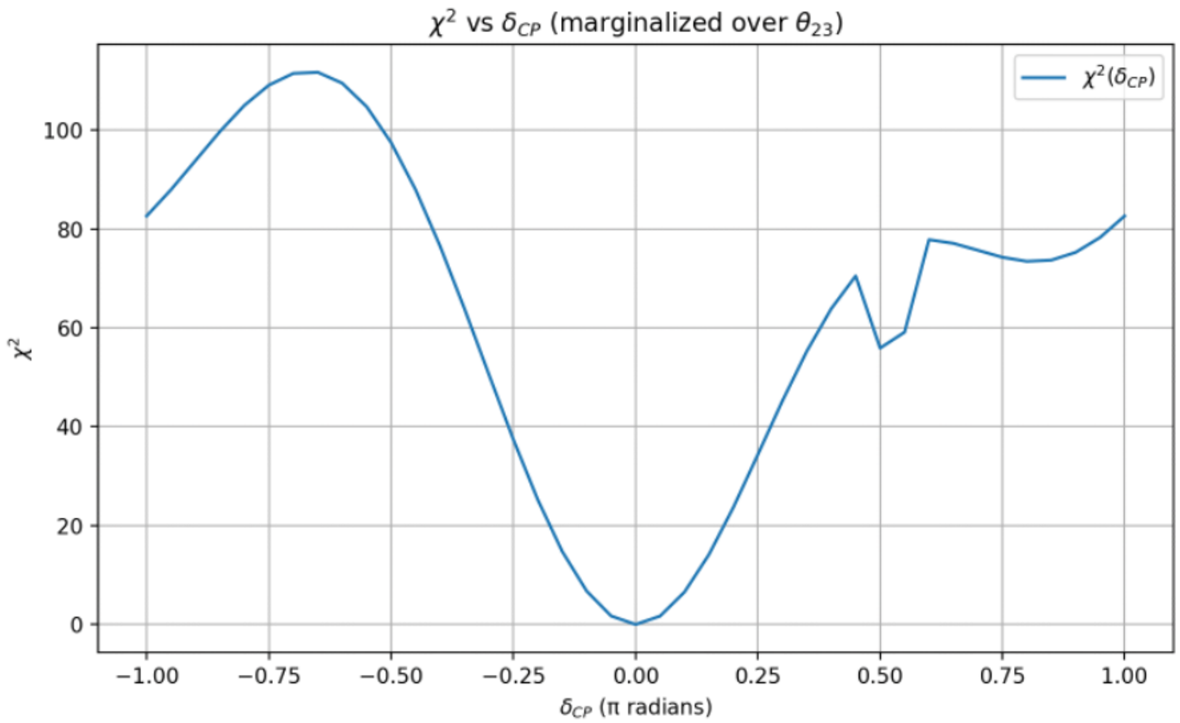
$$\sin^2\theta_{12} = 0.303 \qquad \delta_{\rm CP} = 0$$

$$\sin^2\theta_{13} = 0.02246 \qquad \Delta m^2_{21} = 7.42\times 10^{-5}$$

$$\sin^2\theta_{23} = 0.450 \qquad \Delta m^2_{31} = 2.51\times 10^{-3}$$

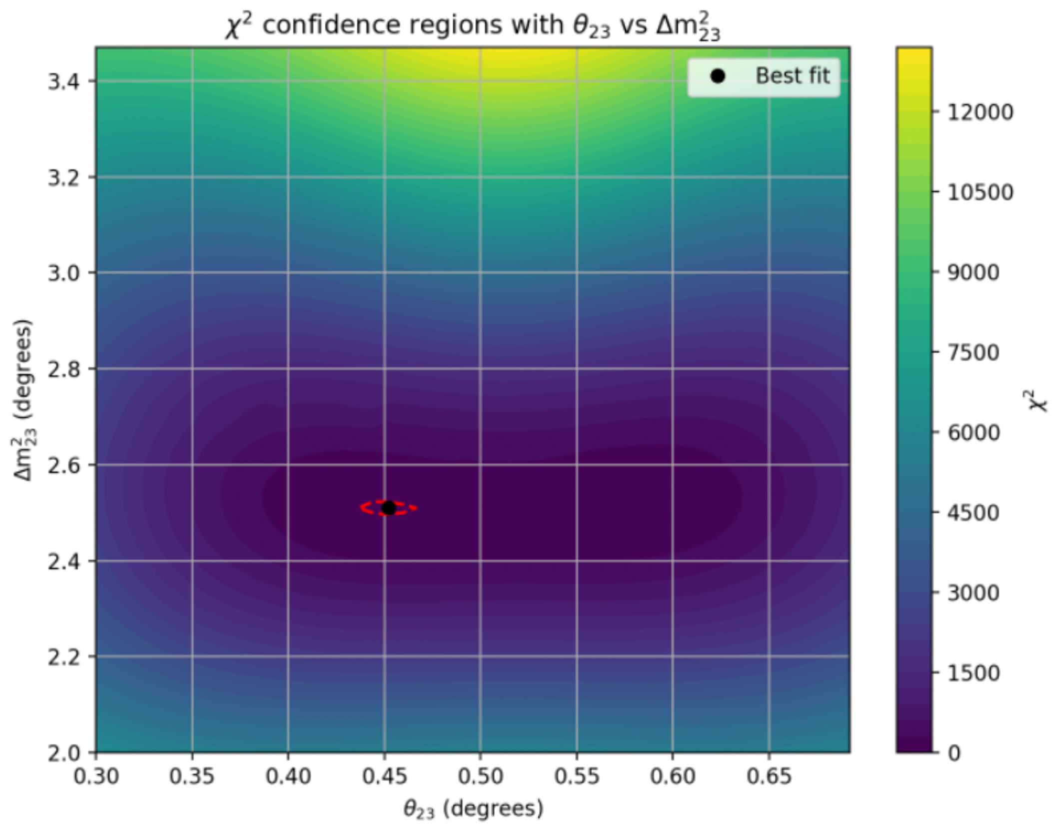
a. Calculate the chi-square as a function of in the interval and plot it. Marginalise
b. Find the allowed region in the plane at 99% confidence level. Put a 3% prior on
c. Combine the DUNE with the T2K and repeat b. above

A:

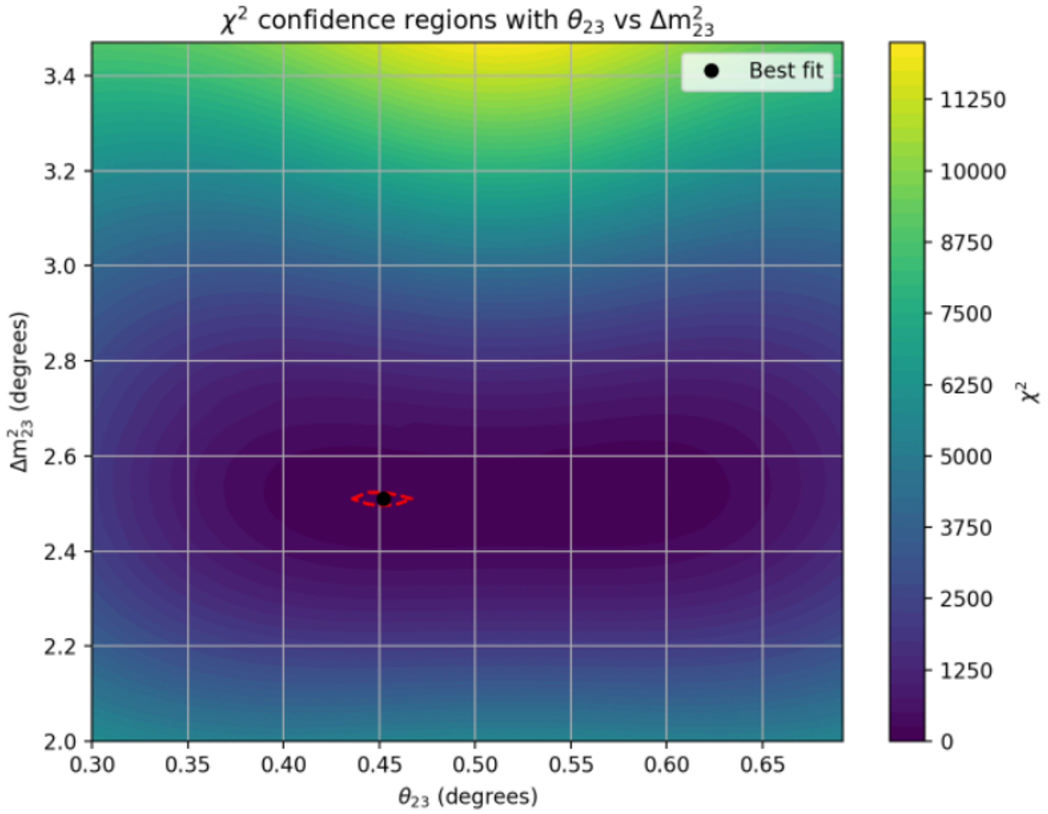


B & C:

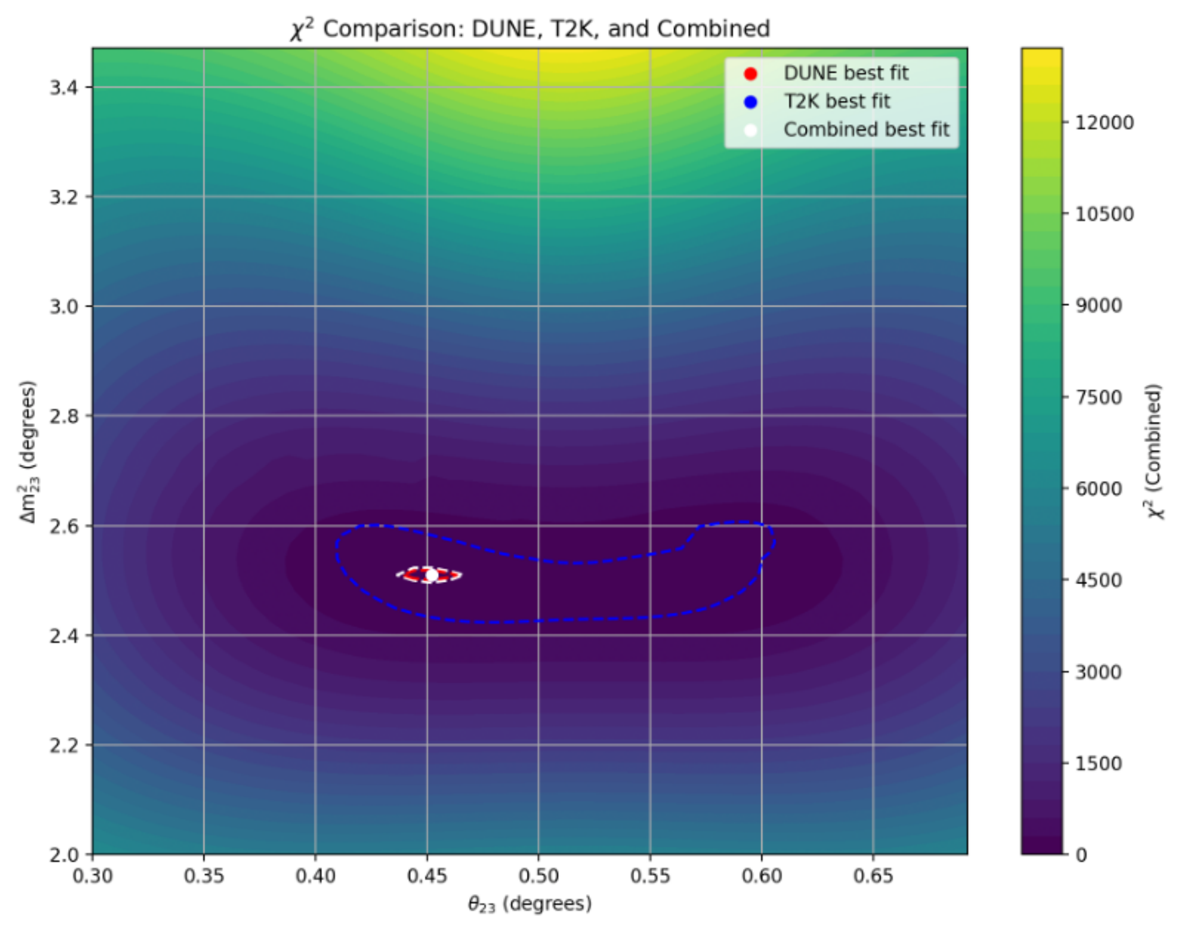
This graph shows delta CP varying with chi^2.



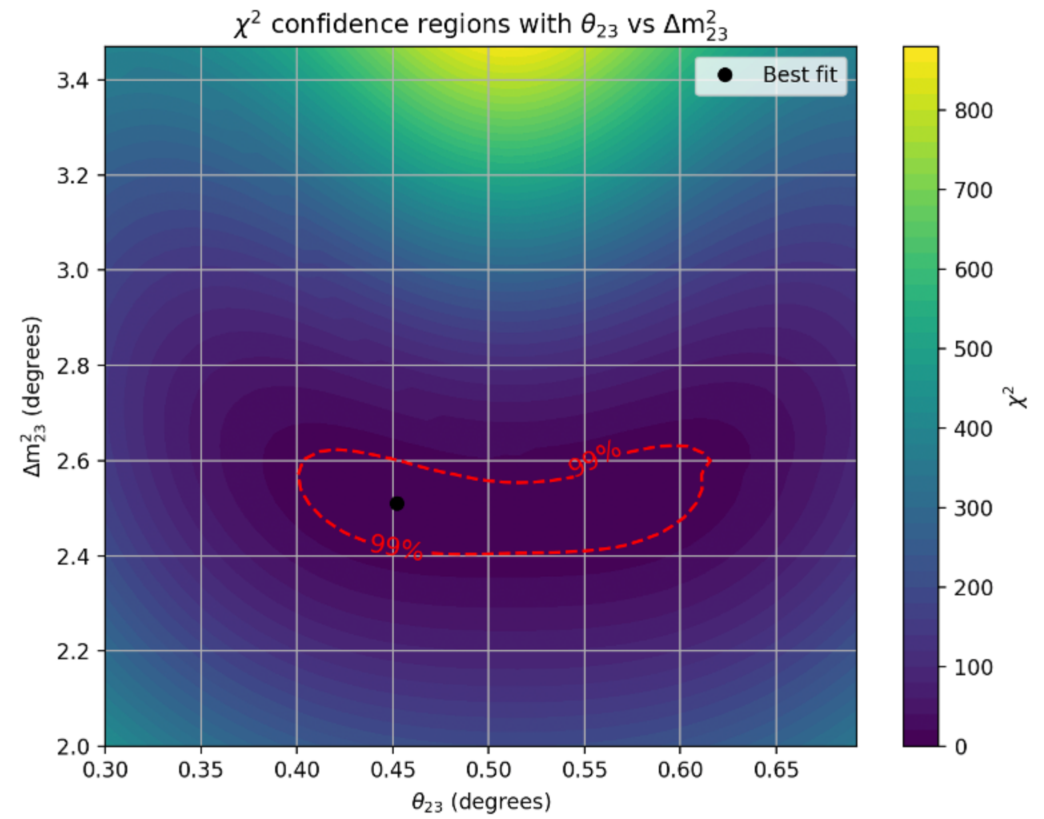
The graph below shows the chi^2 of the DUNE collaboration.



This graph shows the chi^2 for DUNE and T2K. The region is visibly similar to the one above where only DUNE is considered, as the sensitivity is much higher than T2K.



Finally we have the colour map of the combined tests with the individail texts overlayed. The 99% confidence regions show that for combined and T2K it is similar, but alone T2K is far bigger than DUNE. The graph for T2K alone is below:



Proposed analysis in TDR: https://arxiv.org/pdf/2002.03005

Page: 5–161
5.8.2 DUNE Sensitivity Studies

```
glbDefineProjection(delta_projection,GLB_FIXED,GLB_FIXED,GLB_FREE,GLB_FIXED,GLB_FIXED,GLB_FREE);
```

We have fixed specific parameters and left the others free. In this case only ldm and theta23 were left free, and the rest were fixed. The simulation was done with delta cp at 0 and pi, the two values known to be zero on the cyclic graph.

```
double theta12 = asin(sqrt(0.303));
double theta13 = asin(sqrt(0.02246));
double theta23 = asin(sqrt(0.450));
double deltacp = 0;
double sdm = 7.42e-5;
double ldm = 2.51e-3;
```

```
double theta12 = asin(sqrt(0.303));
double theta13 = asin(sqrt(0.02246));
double theta23 = asin(sqrt(0.450));
double deltacp = M_PI;
double sdm = 7.42e-5;
double ldm = 2.51e-3;
```

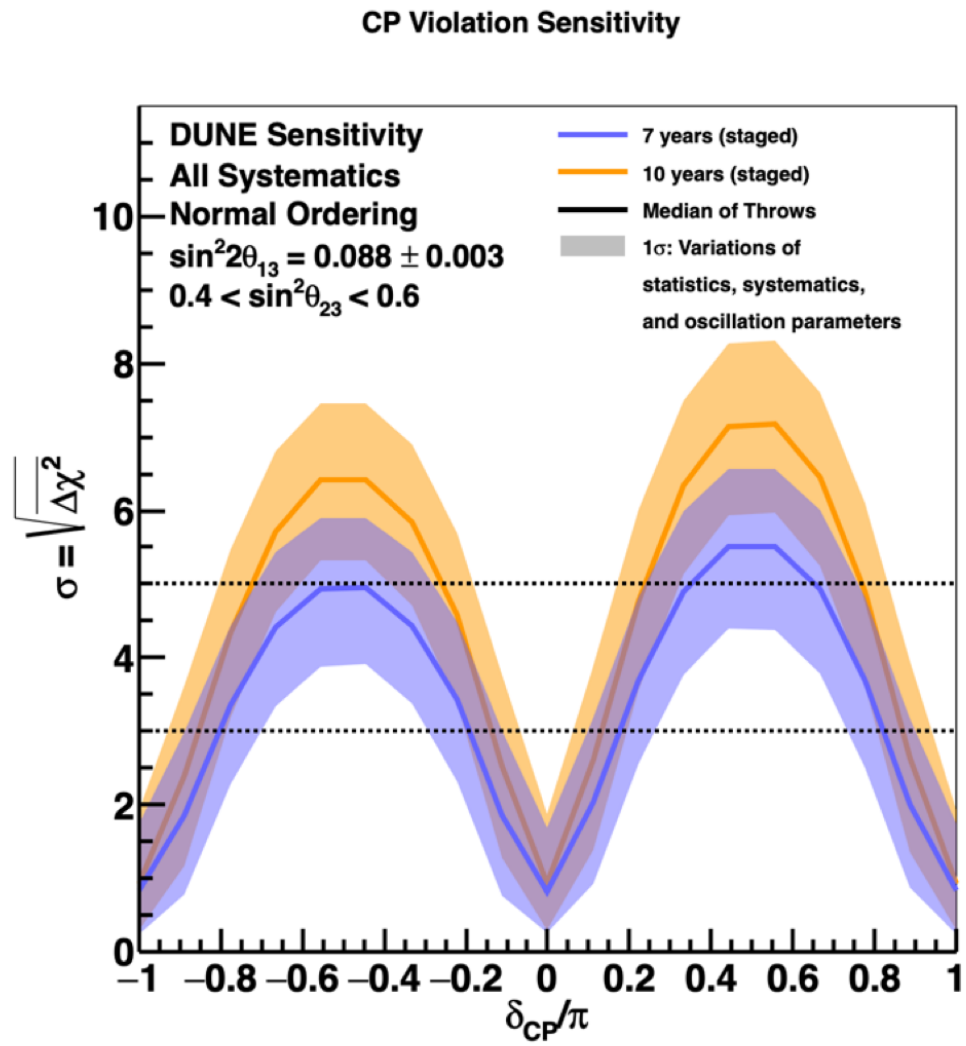
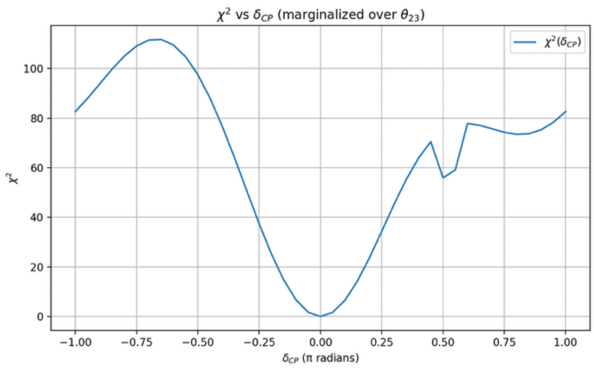
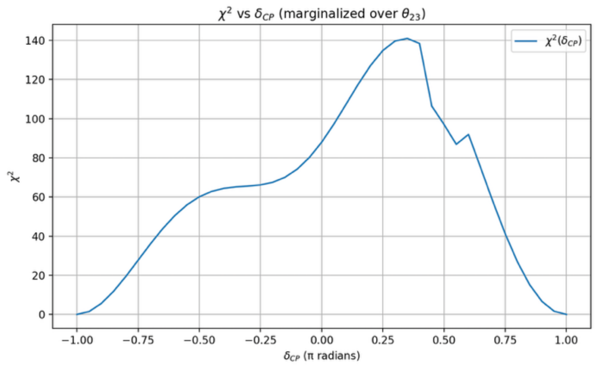
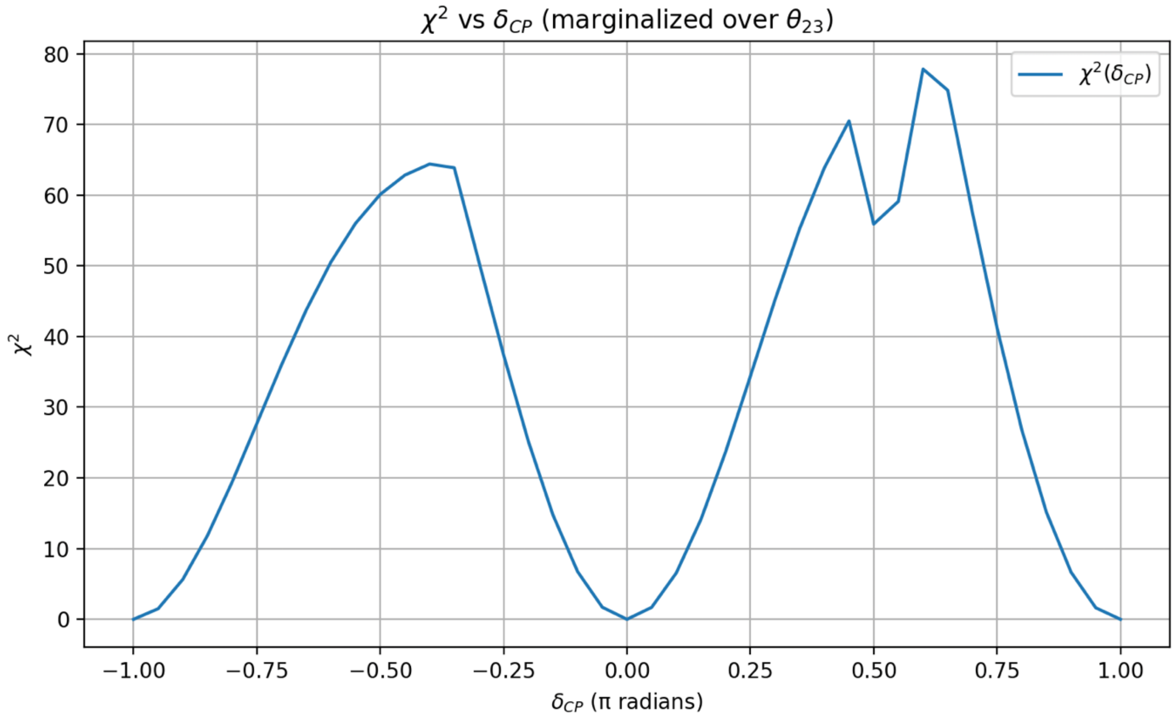


Figure 5.17: Significance of the DUNE determination of CP-violation (i.e.: $\delta_{\rm CP} \neq 0$ or π) as a function of the true value of $\delta_{\rm CP}$, for seven (blue) and ten (orange) years of exposure. True normal ordering is assumed. The width of the transparent bands cover 68% of fits in which random throws are used to simulate statistical variations and select true values of the oscillation and systematic uncertainty parameters, constrained by pre-fit uncertainties. The solid lines show the median sensitivity.





The variation in chi^2 with the values of deltaCP given the test value of deltaCP is 0 on the left and 1 on the right. Then by taking the minimum value of these functions we can see the sensitivity, as plotted below:



Note that the exposure time on this graph is 1 year.

where $\Delta\chi^2_{CP} = \chi^2_{\delta^{test}_{\text{CP}}} - \chi^2_{\delta^{true}_{\text{CP}}}$

$$\chi^2 = -2\log\mathcal{L} = 2\sum_i^{N_{\text{bins}}}\left[M_i - D_i + D_i \ln\left(\frac{D_i}{M_i}\right)\right]$$

$$M_i = \sum_\alpha^{e,\mu}\sum_\beta^{e,\mu,\tau}\sum_j P_{\alpha\beta}(E_j)M_{ij}^{\alpha\beta}$$

See page 5-160 onwards in TDR for more information on the calculations of the chi^2 function. (volume II)

$$\Delta\chi^2_{\text{ordering}} = \chi^2_{\text{opposite}} - \chi^2_{\text{true}} \tag{5.15}$$

$$\Delta\chi^2_{\text{octant}} = \chi^2_{\text{opposite}} - \chi^2_{\text{either}} \tag{5.16}$$

$$\Delta\chi^2_{\text{CPV}} = \text{Min}[\Delta\chi^2_{CP}(\delta^{\text{test}}_{\text{CP}} = 0), \Delta\chi^2_{CP}(\delta^{\text{test}}_{\text{CP}} = \pi)], \tag{5.17}$$

χ2(Δm2)=2i∑[Nexpi(Δm2)−Nobsi+NobsilnNexpi(Δm2)Nobsi]