**INSTRUCTIONS FOR CROSS SECTION EXTRACTION**

These instructions are meant for differential cross sections using files for Run1 – Run3 included

The cross section deals with 4 distributions in reco space (FC0p, FCNp, PC0p, PCNp) and it requires a few steps to fully achieve the final measurement

This is where all folders and scripts can be found: <https://github.com/giacomoscanavini>

Here it is possible to get the content of

*LEEana*: configuration files + wiener\_svd folder

*wcp-uboone-bdt*: wire-cell framework scripts updated with changes for this analysis

*NCpi0\_analysis*: jupyter notebook and python scripts for plotting, NUISANCE scripts

Some of the following instructions can be found in *LEEana/wiener\_svd/README*

1. In *LEEana* run the following command:

*./run\_xs\_full.pl NNN*

This command removes the old configuration files and replaces them with the ones associated to the *NNN* option passed to the command

It’s important to notice that it removes the existing files in */configurations* and replace them with other, the file names is taken care of so this should run with no problems

Each number corresponds to a specific cross section that needs to be run and identifies the reconstructed distributions, the true bins, and the true channels for the cross section script

The legend with the number can be found in *LEEana/run\_xs\_full.pl*

This script also runs DET sys, REWEIGHT sys, and all other systematics

This scripts generates multiple files: *merge\_xs.root*, *xs\_tot.log*, *DetVar* and *XsFlux* root files, and other root files with the reco distributions

1. In *LEEana/wiener\_svd* run the following command:

*./copy.pl*

This command copies the needed files from *LEEana* (generated after step (1)) into *LEEana/wiener\_svd*

First it removes the existing files in *LEEana/wiener\_svd* and replaces them with the new ones just generated

More info of the files handled here can be found in *LEEana/wiener\_svd/copy.pl*

1. In *LEEana/wiener\_svd* run the following command:

*./root -l --web=off 'convert\_wiener\_4ch.C(1)'*

This script prepares the reconstructed distribution for the cross section extraction:

* it subtracts the background and EXT from the selected data
* it imports/creates the covariance matrices for each uncertainty source
* it saves them into a root file called *wiener.root*

1. Based on the cross section to be extracted the options are slightly different

In *LEEana/wiener\_svd* run the following command:

*../bin/wiener\_example wiener.root output.root 2 0.5*

*or*

*../bin/wiener\_example wiener.root output.root 20 0.5*

This script takes the output of step (3) and applied the Wiener-SVD unfolding method. It generates a new root files with the output (unfolded covariance matrix, smear matrix, bias, and true distribution) called *output.root*

This script has two choices for the first option to pass to the command line: 2 and 20

* 2 is for Xp cross sections (second derivative)
* 20 is for 0p and Np cross sections which are extracted together (second derivative)

The different number decides which C matrix to use, in the second case a special C matrix is needed to decouple 0p and Np (removes differences created by the order chosen, 0p before Np or viceversa)

This information can be found in *wcp-uboone-bdt/src/WienerSVD.cxx*

There are more options to choose from in case of 2D/3D cross sections, London knows more about these as they are taken in part from his work

The second option is *Norm\_type* = 0.5 and defines a normalization in the Wiener filter formula

A C0 matrix is defined based on the C matrix chosen with the first option

C0 is multiplied by another diagonal matrix which contains *1/(signal(i))^(Norm\_type)*

This information can be found in *wcp-uboone-bdt/src/WienerSVD.cxx*

My experience is that 0.5 is a good choice, smaller values in the range [0.1,0.5] allow the final extracted cross section to be less smoothed and less stable

1. In LEEana/wiener\_svd run the following command:

*root -l --web=off 'plot\_diff\_xs.C(1,1000.,40.,1)'*

This script takes information in *wiener.root* and *output.root* and returns the final cross sections, the unfolded bin by bin uncertainties and the chi2/ndf values

The information is saved into a file called *import\_xsec.txt* that can be imported later in a Jupyter notebook for the final cross section plot

This command has many options:

* The first value is 1 and is important for cross section with reweighting

[potential choices: 0,1]

* The second value defines the order of magnitude of the final cross section, by default is 1e-36, by passing 1000 the output is divided by 1000 so it will be 1e-39 which is more appropriate in this case

[potential choices: 1, 100, 1000, 10000]

* The third value is the number of nucleons in the argon nucleus, this provides a final cross section in 1/nucleon instead of 1/Ar

[potential choices: 1,40]

* The forth value defines the choice of Xp or 0pNp: 0 is for Xp, 1 is for 0p followed by Np

[potential options: 0,1]

1. Once everything else run correctly, in *LEEana/wiener\_svd* run the following command:

*python save\_results.py ABC*

This script generates a new folder in *LEEana/wiener\_svd* that saves a lot of information used in the cross section extracted

The new folder has the name provided as option *ABC*

It saves the configuration files currently in use, the different outputs from the previous stages and the txt file used for plotting the cross section

1. This is an extra script that can be useful at times, to use it run the following command:

*python pull\_results.py ABC*

*ABC* is the name of an existing folder that has been generated using command (6)

This script is useful as it helps saving times when you want to investigate a saved cross section

It overwrites the configuration files with the ones used when the cross section was generated

This means that commands (2), (3), (4), (5) can be re-run with different options and saved with command (6) in a different folder without the need to re-run (1)

**FILES USED:**

The files used are found in the folder called *LEEana/processed\_checkout\_rootfiles*, which is a created in *LEEana* via symbolic link to another folder

The name and location are important and should be maintained consistent, so either all names are changed in the configuration files or all files are saved in such folder or another symbolic link is created in *LEEana* to the folder containing all your files

*LEEana/processed\_checkout\_rootfiles/CV\_full* contains the following files:

273M checkout\_data\_bnb\_run1\_PF.root

493M checkout\_data\_bnb\_run2\_PF.root

463M checkout\_data\_bnb\_run3\_PF.root

193M checkout\_data\_extbnb\_run1\_PF.root

557M checkout\_data\_extbnb\_run2\_PF.root

632M checkout\_data\_extbnb\_run3\_PF.root

572M checkout\_prodgenie\_bnb\_dirt\_overlay\_run1\_PF.root

423M checkout\_prodgenie\_bnb\_dirt\_overlay\_run2\_PF.root

613M checkout\_prodgenie\_bnb\_dirt\_overlay\_run3\_PF.root

2.4G checkout\_prodgenie\_bnb\_nu\_overlay\_run1\_PF.root

4.2G checkout\_prodgenie\_bnb\_nu\_overlay\_run2\_PF.root

2.3G checkout\_prodgenie\_bnb\_nu\_overlay\_run3\_PF.root

302M checkout\_prodgenie\_nc\_pi0\_overlay\_run1\_PF.root

1.1G checkout\_prodgenie\_nc\_pi0\_overlay\_run2\_PF.root

255M checkout\_prodgenie\_nc\_pi0\_overlay\_run3\_PF.root

Other DetVar and XsFlux files can be found in *LEEana/processed\_checkout\_rootfiles/DetVar\_full* and *LEEana/processed\_checkout\_rootfiles/XsFlux\_full*

**PLOTTING:**

Some plots are generated as you run the chain of scripts previously mentioned but the following instructions provide access to more tools for plotting in python

The following refers to the info in *NCpi0\_analysis/CrossSections\_Visualization.ipynb*

After command (6) is run and a new folder is created it is possible to access that folder from the Jupyter notebook and import *import\_xsec.txt* and *output.root* with all the information needed to plot the cross section

A lot of functions are defined in *NCpi0\_analysis/custom\_functions.py* and can be imported in the jupyter notebook using the command:

*import custom\_functions as cf*

Generally speaking the function *print\_xs\_info(folder)* provides all info that can be copied and pasted in a new cell for plotting the cross section

*folder* defines the name of the folder the information is extracted from, and therefore the cross section in exam

Other functions like *plot\_single\_xs* and *plot\_single\_xs\_err* can be used to plot Xp cross sections

0p and Np cross sections are handled in the same way but a small trick is used in the notebook to still allow the use of *plot\_single\_xs* and *plot\_single\_xs\_err*

**CHANGES THAT CAN BE DONE TO THE INSTRUCTIONS:**

1. REMOVE DETVAR SYST
2. REMOVE REWEIGHTING FUNCTION WEIGHTS
3. FAKE DATASET RUNS