Nathan Benton

Lene van Rootselaar

Aug 6, 2025

Summer Project Documentation

Overview:

My project this summer was to learn MCEq (a cascade equation solving python package for simulating air showers) and to generate energy spectrum plots of muons and the three neutrinos flavors for comparative analysis. Most members of the astro-particle team at TU Dortmund use MCEq to compare their measured results from experiments like ICECube to theory, so investing in someone like me to dig in and write code to create the graphs they need is worthwhile.

The first two weeks in Dortmund were spent reading and trying to get the example files on MCEq to run. I read a paper for background on astro-particle neutrino physics as well as skimmed a paper regarding the cascade equations (analytical equations used by MCEq to predict particle flux without manually simulating real events). Using the most recent update from GitHub, not one example file would run past the import cell. My main challenge was overcoming how intentionally difficult MCEq is to use. Some sections of the package have been left intentionally outdated or expunged so that, to generate useful plots for publication, you must reach out to the package's creator for permission and to guarantee credit. Many functions listed in the documentation simply do not exist in the program files, so I had to spend many days reading the source code to discover their replacements and how MCEq handles adding partial hadronic contributions. These contributions are from parent particles like the various kinds of D, K, and pi mesons as well as from Tau or other charm contain particles and did not have energy or decay tables in the base version of MCEq.

Thanks to people here who have worked personally with MCEq's creator, I got my hands on the non-public file which contains these tables ("mceq_db_fine_v150.h5"); others at TU had given up on trying to get this to work. The issue was that the file's structure was laid out differently than what MCEq was implicitly programmed to handle. I tried futilely to manually remove and alter layers of the H5 file to retrofit it into a form the package could read, but this ultimately would have made it so that, once again, no one besides the creator could use it. During my investigation, I found a branch on the github (new_build_system) that had a different way of configuring and looked like it would read in the file correctly. After figuring out an SSH connection between VSCode and github, I installed the branch and eventually got the desired particles added for MCEq to track. After this update, I was also able to access the newest simulation model, SIBYLL2.3d. I experimented with the CORSIKA python package, a more traditional simulator that kept track of each decay component (even photons), but realized the run-times were far too long (several days) and I was too close to the end of the summer.

Below, I will detail all of my final Jupyter notebooks, the plots they generate, necessary dependencies, and steps to alter the default MCEq/Python to get them to run.

Dependencies and Updating MCEq:

The default version of MCEq will output a limited number of flux contributions by default. These include all the particles given when you call .print_particle_tables(0) from MCEq's particle manager file (total, conventional, prompt, and parent particles like pi, kaons, and muons). It lacks neutrino or muon daughter tracking from parent particles like tau, d-mesons, Λ_c, or unflavored sources as well as access to the updated SIBYLL2.3 version.

For these reasons, it is valuable to update versions to generate these complete plots. The downside, however, is the run time of the updated MCEq. Prior to updating, fully computing the fluxes for 20 zenith angles would take upwards of 20-30 minutes; with the larger, more complete tables, it takes anywhere from 3-4 hours. I would recommend staying on the original version if partial hadronic contributions aren't necessary and older versions of SIBYLL are acceptable. If not, follow the steps below:

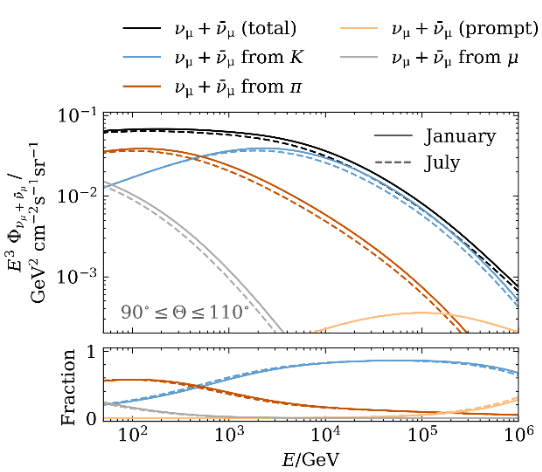
- Download the H5 file named "mceq_db_fine_v150.h5"

- Navigate to your scratch folder where python libraries are stored (on phobos, this is at "/scratch/{username}". You could also use my scratch folder if it still exists, change your python kernel to "/scratch/nbenton/project/bin/python/".
- Once you find where MCEq is installed and being run from, navigate to the "data" folder there. Upload the new H5 file to this folder. There should already be a file in there named something like "mceq_db_lext_dpm191_v12.h5," you can leave that one in the folder.
- All of my notebooks that make the upgraded graphs will have these cells, but look to "variable_angle+particle.ipynb" for an example of how to redirect MCEq to pull from your new file. The cells immediately above and below where we initialize MCEqRun() are the ones necessary to update versions.
- Now you should be able to run the notebooks straight through and access the aforementioned particles and new models.

There are differences in how the two files are laid out, code to check the structure of the keys (which includes all of the models each of them contain) can be found in the file InvestigateH5.ipynb in my MCEqProject folder.

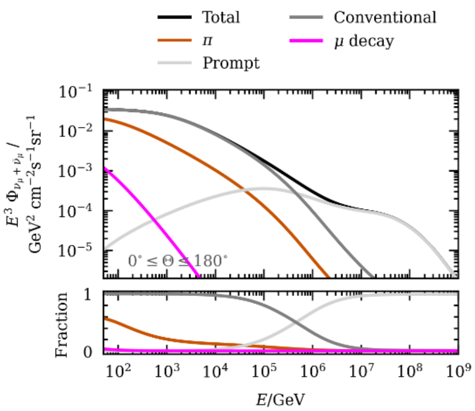
<u>Breakdown of Each Notebook:</u>

"Example Code.ipynb"



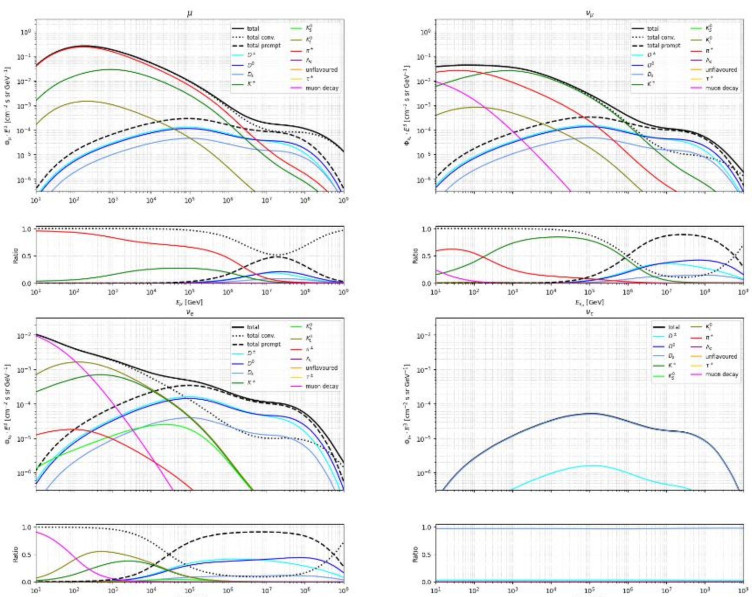
- This code file was given to me by Lene from the person who previously worked here and was better versed in MCEq, Caroline. RUNS ON OLDER VERSION, DEFAULT MCEQ
- Can choose two days of the year to simulate with, much of it is hard coded though, so changes to the legend would be necessary.
- Averages over 5 hardcoded angular bins and plots muon neutrino fluxes from K, pi, and muon
- Calculates and plots the ratio on the bottom

“angavg_single_particle.ipynb”



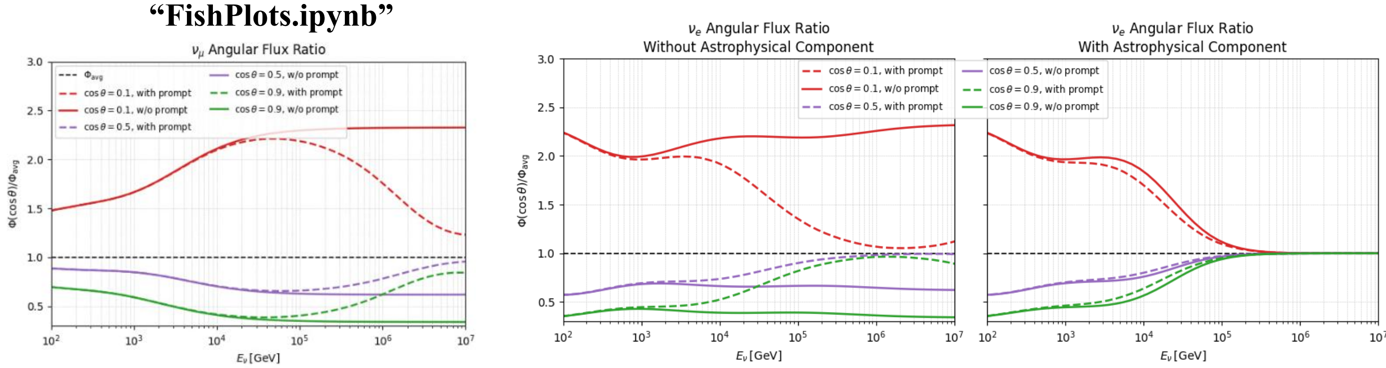
- RUNS ON OLDER VERSION, MCEq Version 1.3
- Essentially just the previous file except the simulation is wrapped in a function, so now you can choose how many bins and what angular range you want. No longer plots two days of the year.

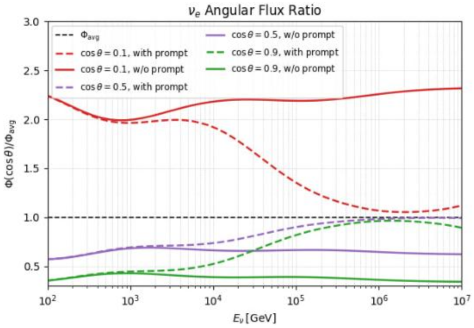
“MesonPlots_PartialHadronContrib.ipynb”



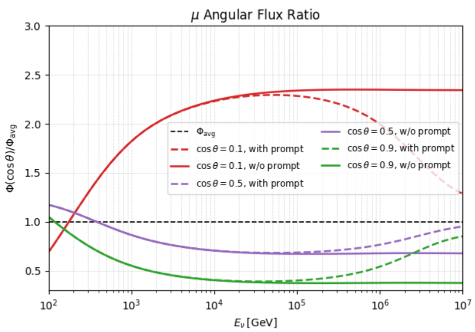
- RUNS ON OLDER VERSION, MCEq 1.3
- Makes 4-way spectrums + ratio plots for muons and all three flavors of neutrino.
- Lacks particles like tau and charm containing like Λ_c, D-mesons, and unflavored; still included in the legends, should be removed
- Useful if you want to publish without those particles (without need for Anatoli’s permission)
- Should run as is with all the proper imports, MCEq 1.3, and with the old H5 file selected by default

“FishPlots.ipynb”



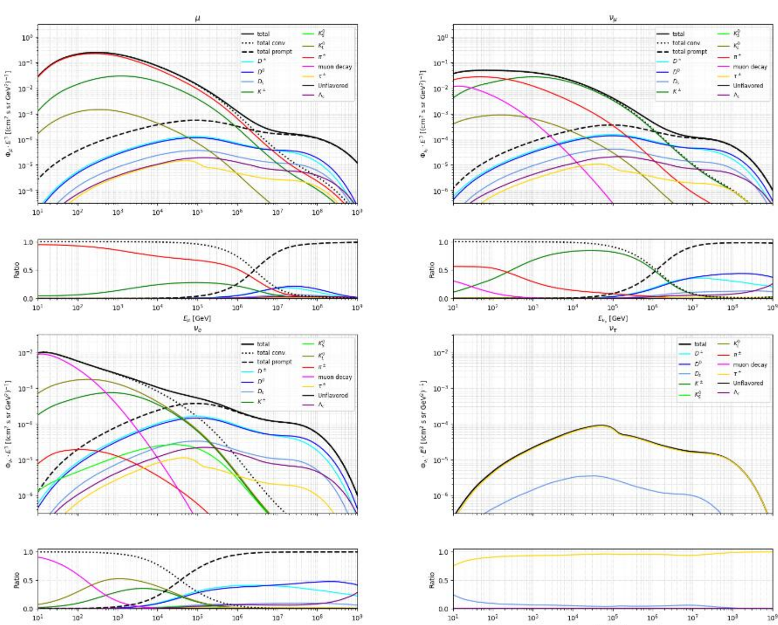


- Made for new build of MCEq, MCEq 1.6.0 (new_build_system branch on github). Should run as is once this is set up.
- Simulates & saves data for 21 angles. You can select which three angles to compare which then get divided by their respective average (for angles equally spaced in cos(theta)) to form ratios. This is done for conventional (without prompt) and total channels for all three angles.

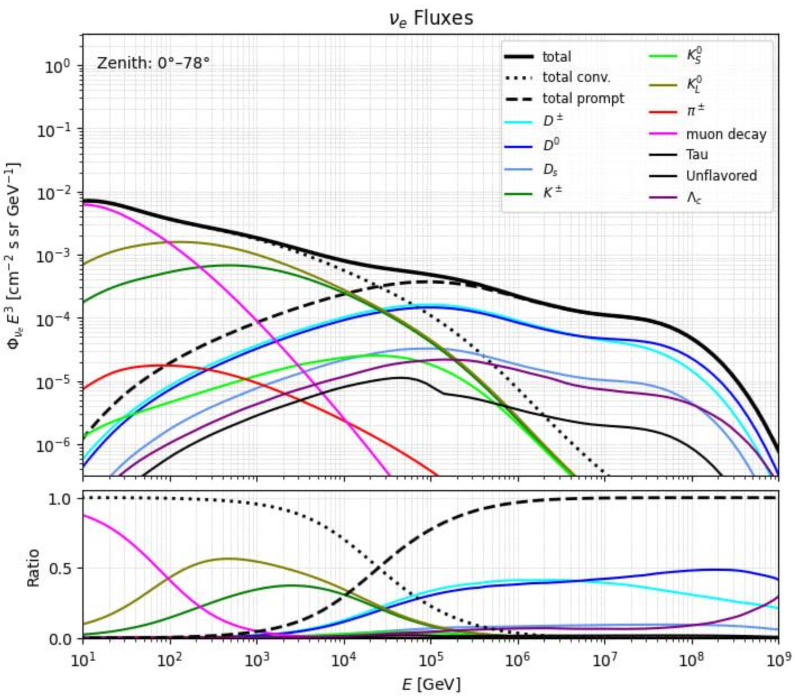


- One of several astrophysical flux models is added to the data, altering it at high energies. This is compared for all three (technically four, but tau is not very useful) particles in order to compare the ratios for simulations with and without astrophysical contributions.
- There is a .py file named “FishPlot_overnight.py” which essentially contains all of the simulation code used in this file. Intended use is for running overnight on a screen on the computing cluster so your local client doesn’t need to stay connected. Saves results to a file named “FishPlot_data{run number}.pkl.gz” These should be stored in MCEqDatafiles.

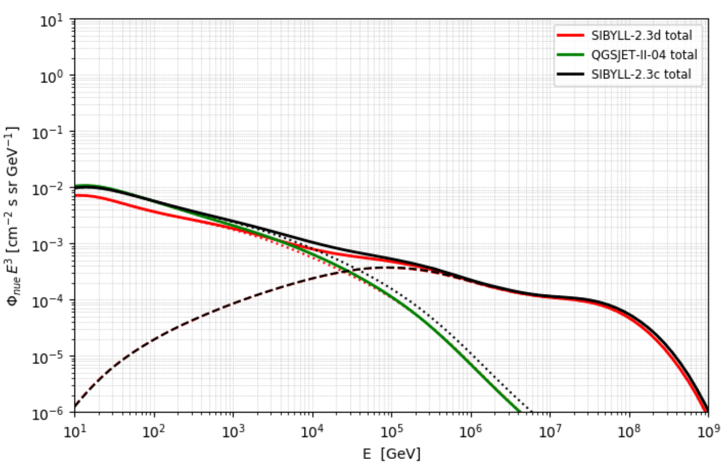
“variable_angle+particle.ipynb”



- Made for new build of MCEq, MCEq 1.6.0 (new_build_system branch on github). Should run as is once this is set up.
- Makes the same plot as seen in the MesonPlots file, but now with SIBYLL2.3d and all of the desired particle contribiutions.
- Averages angles over desired range and results can be saved to files called “angleAvg_{run}.pkl.gz”



- Allows you to select and manually plot one at a time if you so choose. Displays the particle selected in the title as well as the angular range.
- There is also an overnight (.py) version of the simulation code found in this file. It is called “angleAvg_overnight.py” and saves the data for later.
- There are already several simulations run with this file saved in MCEqDatafiles, if you want to use them, the parameters they were run with is contained in said folder in a markdown file called “data_explained”



- At the bottom of the file, there is code to make comparison plots between models. Just run overnight file with an available model (inputted in MCEqRun()) found in “InvestigateH5.ipynb” and add its number and model label to “run_ids” dictionary. Should also add descriptions for this new data file in MCEqDatafiles