Follow this tutorial only if you do not have access to the LIP machines and are using Docker and the ROOT container in your own computer

Introduction

The purpose of this tutorial is to write a ROOT macro that receives a ROOT file with all the particles of an event and returns another ROOT file with only the Z boson that will be reconstructed. Then, you can write yet another ROOT macro to read the file you created, plot the Z boson mass peak and fit it to estimate the Z boson mass.

Download the data file for this exercise (`zjet_unrec.root` -- **note that this file is 3 GB**) and save it inside the Docker shared folder on your system. It's probably a good idea to create a new directory for this exercise inside that shared folder (e.g. `Tutorial_ROOT_advanced`)

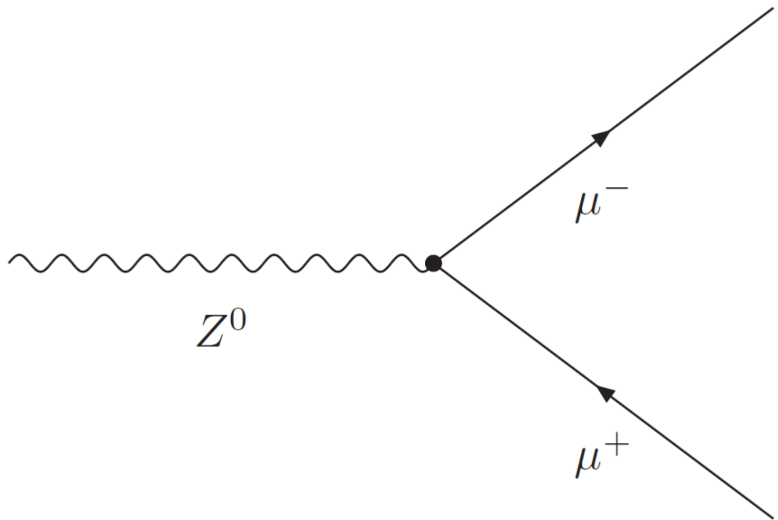
Open a new terminal and start the ROOT docker container (you should have already followed the instructions here to set-up Docker and start the ROOT container)

- Go to the shared folder inside the container: `cd /shared`
- Go into the directory you created in step 1: `cd Tutorial_ROOT_advanced`

Reconstruct the Z boson

- Create a new file (with whatever name you want) with a function with the same name as the file (so that you can run it as a standard ROOT macro) and save it in the same directory.
- Declare the file object that we are going use to access our input file (`zjet_unrec.root`)
- Declare the tree object as in previous tutorials
- Declare a histogram object (TH1 class reference), with the limits, number of bins, and title you think best for the Z mass
- Declare the variables that we will use to access the particles of each event, saved in the tree (as in the RootTutorial2.cc example macro)
- Link these variables to the tree with the SetBranchAddress function (as in the example macro)
- Loop over the entries in tree with a for loop, for example (to get the number of entries in the tree: `t->GetEntries()` and to get the values of entry i loaded on the linked variables make `t->GetEntry(i)`)

- Loop over the particles in each event
- The Z boson in this Monte Carlo ”data” was forced to decay to two muons
- You must therefore find the muons in the particle list. Tip: The ids of the different particles used can be found here (on the second page)



- When you find the muons (there should be exactly two per event), calculate their invariant mass and fill a histogram with it
- Verify that the spectrum makes sense and then save the events that pass your selection in a new tree, in a new file. You can choose what variables you want to save but as a starting point the four vector (momentum plus energy) and the mass should be enough.
- To do this you should declare a file, more or less in the same place where you had declared the input one and the same for the tree
- You must declare new output variables (i.e. px, etc.) and link them to the new tree with the Branch function (TBranch class reference)
- Every time an event passes you should fill the output variables with the values you want to save and then use t−>Fill(), in order to save them to your output tree
- You have now created your own data file with a TTree that has information about a Z boson reconstructed from a pair of muons!

Analyze the Z boson mass peak

- Write a new ROOT macro that opens your data file.
- Draw the Z boson mass peak and fit it using a Gaussian function. The fitted value of the Gaussian's mean should approximately correspond to the Z boson measured mass. How good is the agreement? (Compare, for example, with the value here)
- In this case, because we are not taking into account detector resolution effects, the Z boson mass peak should be better described by a Breit-Wigner function. This function is not predefined in ROOT, therefore you need to define it yourself (Fitting histograms reference, check section 7.2). Once you have defined your own TF1 function you can use it to fit the mass peak as before. Does your estimation of the Z boson mass gets close to the measured value? What about the goodness of fit, is it improved?