Measurement of Z angular coefficients

FLORIAN HARZ
Institut für Physik, JGU Mainz

The measurement of angular coefficients in the Drell-Yan process provides a powerful probe of the underlying dynamics of Z boson production and decay, as well as a stringent test of perturbative Quantum Chromodynamics (QCD). This analysis focuses on the extraction of angular coefficients from Z boson decays into dilepton final states (electron or muon pairs) using proton-proton collision data collected with the ATLAS detector at the Large Hadron Collider (LHC) during Run 2, corresponding to an integrated luminosity of $140\,\mathrm{fb}^{-1}$ at a center-of-mass energy of $\sqrt{s} = 13\,\mathrm{TeV}$.

The angular distribution of the decay leptons in the Z boson rest frame can be described in terms of eight angular coefficients, A_0 through A_7, which are functions of the Z boson transverse momentum and rapidity. These coefficients are extracted by performing template fits to the observed angular distributions in the Collins-Soper frame, a specific rest frame that reduces polarization biases. The analysis is carried out in bins of the Z boson transverse momentum p_T^Z and rapidity y^Z, separately for electron and muon channels, and results are subsequently combined.

Rather than performing an unfolding to correct for detector effects, the analysis employs a forward-folding approach based on a binned maximum likelihood fit at detector level. Monte Carlo simulations are used to model the detector response, and are reweighted at truth level to construct templates for each angular coefficient. Specifically, generated events are reweighted to have a flat distribution in the Collins-Soper angular variables and are then further weighted by harmonic polynomials corresponding to each angular term in the cross-section. For each reconstructed (p_T^Z and y^Z) bin, contributions from neighboring truth bins-arising due to detector resolution and smearing-are included in the modeling by summing over the appropriate truth-level bins.

The fit determines simultaneously the cross section in each truth-level bin and the angular coefficients. This approach ensures a consistent treatment of detector effects and allows the measurement to remain directly comparable with theoretical predictions at particle level. It also includes and considers the effects of up to 1000 systematic uncertainties on the data by including them as nuisance factors.

My personal work focuses on retrofitting the currently used analysis framework: Create a new framework to fill histograms from NTuples ROOT files. There is an existing implementation for this. However, it is slow which increases the turn around time to the analysis and therefore slows down the speed of progress. One has to iterate over each event in the NTuples and writes the nominal weight into a histogram corresponding to the p_T^Z and y^Z detector and p_T^Z and y^Z truth bins of the event. This has to be done for each systematic variation as well as each systematic effect is associated with two separate histograms (one for the upwards variation and one for the downwards variation). The current implementation does this for each systematic variation in a separate process. This ends up in a huge amount of processes as we have to consider up to 1000 different systematic effects. However, most of this systematic effects are represented as weight variations. Instead of filling the nominal weight into the histogram, one has to use the weight associated to the corresponding systematic variation. Therefore, one would only need to look at each event once and fill the nominal histogram as well as the histograms of all weight variations in this step. This needs less resources, as we do not have to carry around the whole event content for each systematic variation separately. To optimize this even more, one should avoid the histograming (sorting the event into a bin) of the event for each systematics histogram because the bin into which the event will be sorted is identical, only the weight that is added to the bin content changes. This is achieved by multi weight histograms. These are histograms that keep track of multiple independent weights per bin. To fill an event into the histogram, one provides a list of weights, one weight for each weight sum that is stored in the histogram. The sorting into a bin has to be done only once and then all weights are treated at the same time. This approach greatly increases the speed of the histogram filling as now all systematic variation are treated at the same time.

A second framework that I am retrofitting is the fitting framework to perform the likelihood fit. Currently, we are using a custom made framework for this. However, the original author has left the analysis group and the code is hard to maintain. Instead, I explore the possibility to switch to the RooFit package of the ROOT team. Their package is well maintained and has (and still is) receiving many speed improvements over the past years. However, as of now, their package is unable to create a workspace for our fit because our fit is too large and exceeds ROOTs current 1 GB ROOT file limitation. Future improvements on the ROOT/RooFit side may mitigate this problem. Currently, it is a showstopper because you can not efficiently perform a fit without being able to prepare a workspace for it.