

PostCabanaDimuons Renovation

Matt Wysocki
University of Colorado
10/28/2009

Overall Structure

- * **HistManager** is charged with loading, summing, and normalizing the CabanaBoy histograms.
- * **FitManager** initializes the fit functions and performs the fits.
- * All fit methods have their own class which inherits from **AbstractFitMethod**, which inherits TF1.
 - * Each fit class knows how to initialize itself given some basic input; this is probably one of the features I'm most pleased with, given our proliferation of fit methods.
- * **PlotMaker** has utility functions for plotting the histograms and fit results.

HistManager

* As mentioned, deals with the CB histograms. I tried to keep it simple... the constructor and `set_subtraction_method()` sets it up, then `load()` opens the file, grabs the necessary histograms, and `sum_histograms()` adds them across centrality, `zvtx,r.p.`, normalizes them, and takes the difference.

* Usage:

```
HistManager* hm = new HistManager(filename, A1ot);  
hm->set_subtraction_method(HemmickNorm);  
if( !hm->load(South, Pt, 4, 7) )  
    exit(1);  
hm->sum_histograms();
```

FitManager

- * FitManager keeps a list of the requested fits, but doesn't know about them individually (only AbstractFitMethod). It takes a pointer to the HistManager, and initializes the fits from it, then performs the fits.

- * Usage:

```
FitManager* fm = new FitManager(Alot);  
fm->set_hist_manager(hm);
```

```
GausExpFit* gfit = new GausExpFit;  
fm->add_fit_function(gfit);
```

```
fm->initialize_functions();  
fm->fit();
```

```
cout << "FitManager::get_signal(): " << fm->get_signal()  
      << " +- " << fm->get_stat_error() << endl;
```

AbstractFitMethod

- * Mostly-abstract base class to enforce the API with the fit functions. I found you can pass 0 as the version to ClassDef to tell it not to worry about I/O.

- * Some important methods:

- * Evaluate the fit function (this is what the TF1 base class is given):
`virtual double Eval(double *x, double *par) = 0;`

- * Return the signal/error from the fit:
`virtual double get_signal() = 0;`
`virtual double get_stat_error() = 0;`

- * Reset the fit parameters to their defaults:
`virtual void reset_parameters() = 0;`

- * These are called by the FitManager for initializations that need info about the current bin/arm/histogram/etc.

- `virtual void initialize_from_histogram(TH1*) = 0;`
`virtual void set_config(CommonConfig*) = 0;`

Summary

- * I think this division of labor is easier to maintain in the future. Remember, object-oriented is just to make our programming lives easier. This usually breaks down when we try to do too much with one class, which I think happened to CabanaboyDimuonSignal as we added new fit methods, new histograms, and reached >3000 lines of code. I know I was getting lost!
- * Also should allow things like
 - * Multiple HistManagers/FitManagers per macro
 - * Any combination of fit methods, even the same functions with different options enabled

Bits and Pieces

- * When trying to compare new and old PostCabana results, I found that the normalization was different between running on my laptop and RCF.
 - * Turns out calling `hist->FindBin(x)` doesn't give a stable result when `x` is a bin edge. `FindBin(1.7)` returned 18 on my laptop (correct) and 17 at RCF.
 - * I wrote a `find_bin_by_low_edge(x)` function that assumes `x` is a bin edge.
- * I also got different fit results with ROOT 5.24 on my laptop (after fixing the above problem), and found I can get the old result (within some tiny precision) if I go back to ROOT 5.20. What changed?

To Do

- * Add methods for getting the systematic error.
- * I've only implemented the fit functions that I use so far.
- * Working on DoubleGausAcceptWeightedFit.
- * Currently, writing a fit method to a file and closing ROOT, then reading it back in in a new session results in a crash when exiting ROOT (ye olde double-delete ROOT issues).
- * PlotMaker is currently empty. I'm thinking static functions that take the HistManager* and FitManager* and draw them to a TPad.