

Novel Modeling Framework

A First Glance

S. Pflüger

MOTIVATION

Requirements

- ⊙ construction of “any” (complex) model/function
example: luminosity fit model

$$\frac{dN(\theta)}{d\theta} = L \cdot \left[\frac{d\sigma(\theta)}{d\theta} \cdot \epsilon(\theta) \right] \otimes Res(\theta)$$

- ⊙ software requirements
 - ▷ modularity: components are replaceable
 - ▷ flexibility: applicable everywhere
 - ▷ OO and creation of composite models
 - ▷ easily extendible: creation of user defined functions
 - ▷ easy to use
 - ▷ few dependencies

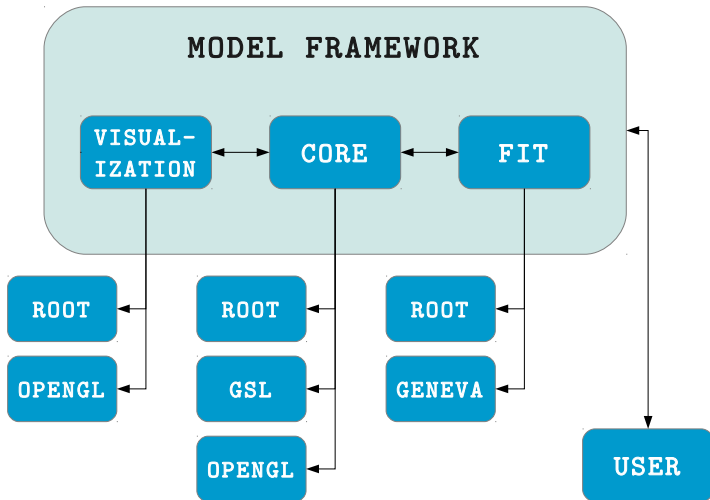
MOTIVATION

Possible Existing Solutions

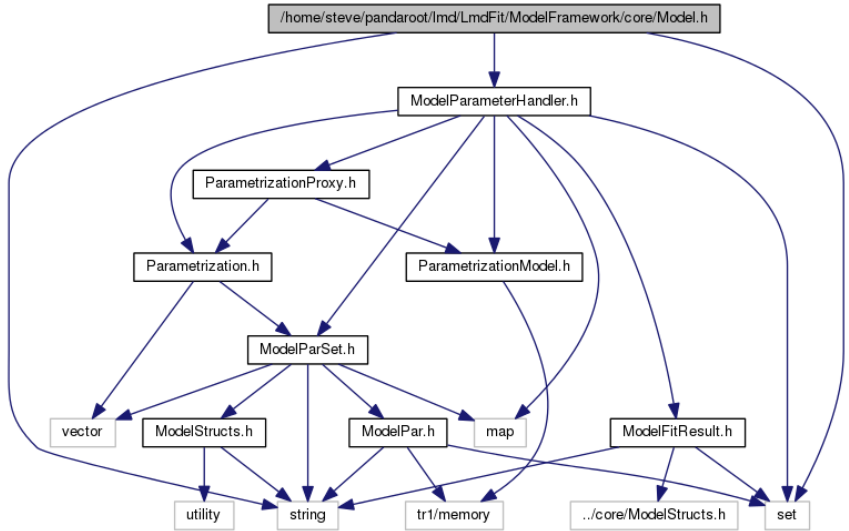
- ⊙ ROOT (TF1/2): TF1/2 lack OO and composite operations
- ⊙ RooFit: lacks modularity feature and flexibility (e.g. luminosity model)
- ⊙ others: found nothing suitable

→ new modeling framework!

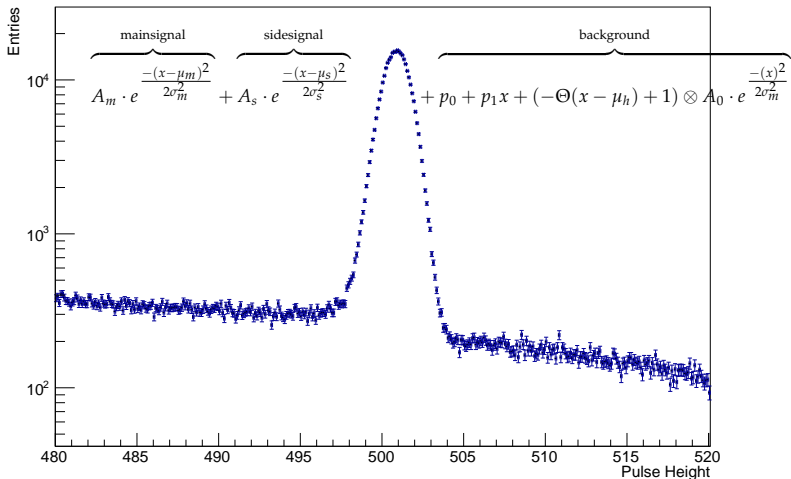
CONCEPT



CONCEPT: THE CORE



EXAMPLE



^{60}Co Spectrum (Courtesy of Marcell Steinen)

EXAMPLE: THE CODE

Listing 1: Constructing Models

```
1 shared_ptr<ModelID> maingauss(new GaussianModelID("gauss_main"));
2 ModelParSet &maingauss_parset = maingauss->getModelParameterSet();
3 shared_ptr<ModelPar> amplitude = maingauss_parset.getModelParameter("gauss_amplitude");
4 amplitude->setParameterFixed(false);
5 amplitude->setValue(15000);
6 shared_ptr<ModelPar> mean = maingauss_parset.getModelParameter("gauss_mean");
7 mean->setParameterFixed(false);
8 mean->setValue(501.0);
9 shared_ptr<ModelPar> sigma = maingauss_parset.getModelParameter("gauss_sigma");
10 sigma->setParameterFixed(false);
11 sigma->setValue(0.8);
12
13 shared_ptr<ModelID> sidegauss(new GaussianModelID("gauss_side"));
14 ModelParSet &sidegauss_parset = sidegauss->getModelParameterSet();
15 amplitude = sidegauss_parset.getModelParameter("gauss_amplitude");
16 amplitude->setParameterFixed(false);
17 amplitude->setValue(700);
18 ...
19
20 shared_ptr<ModelID> doublegauss(new AdditionModelID("doublegauss", maingauss, sidegauss));
```

EXAMPLE: THE CODE

Listing 2: Creating an Estimator and Fitting

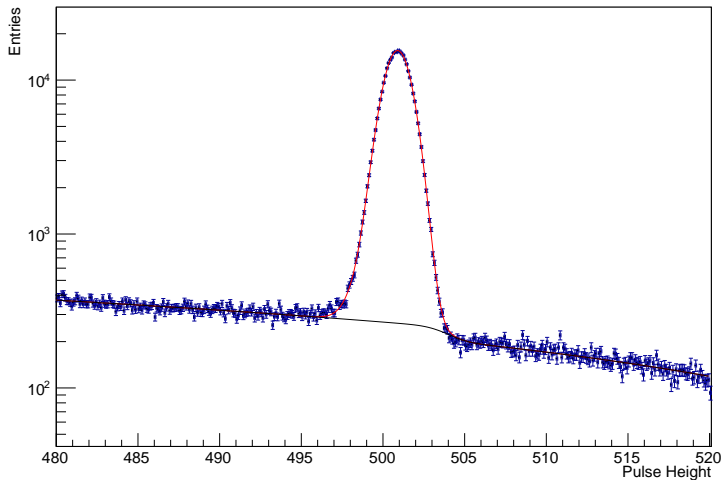
```
1 full->init(); // would initialize Model with Parametrizations etc (not required here)
2 full->getModelParameterSet().printInfo(); // prints info of the model parameters in the set
3
4 TFile *f = new TFile("CO60_spec.root", "OPEN");
5 TH1D *hist;
6 f->GetObject("Energy", hist);
7
8 Chi2Estimator chi2_est;
9 chi2_est.setModel(full);
10
11 ROOTDataHelper data_helper;
12 chi2_est.setData(data_helper.createBinnedData(hist));
13
14 EstimatorOptions est_opt;
15 est_opt.setFitRangeX(std::make_pair(480.0, 520.0));
16 est_opt.setWithIntegralScaling(true);
17 chi2_est.applyEstimatorOptions(est_opt);
18
19 ROOTMinimizer fitter(chi2_est);
20 int fit_status = fitter.doMinimization();
```


EXAMPLE: THE CODE

Listing 3 : Plotting the Results

```
1 // create plotter that uses the ROOT framework
2 ROOTPlotter plotter;
3 // create visualization prop with standard 500 point evaluation and binning factor from data
4 ModelVisualizationProperties1D vis_prop(chi2_est.getData());
5 vis_prop.setPlotRange(fit_range);
6
7 TGraphErrors *graph = plotter.createGraphFromModel1D(full, vis_prop);
```

EXAMPLE: FIT RESULTS



Fitted ^{60}Co Spectrum (Courtesy of Marcell Steinen)

CONCLUSIONS

Summary

- ⊙ well working and tested for complex 1D models
- ⊙ external libraries: ROOT (but more to come)
- ⊙ common functions and many operators built-in
- ⊙ however: still under development (report bugs please)

Request

- ⊙ try it out (currently in PandaRoot under lmd/LmdFit)
- ⊙ important: give feedback! (email: s.pflueger(at)gsi.de)

END

Thanks for Your Attention!