GSI
Online
Offline
Object
Oriented

Go4Fit package

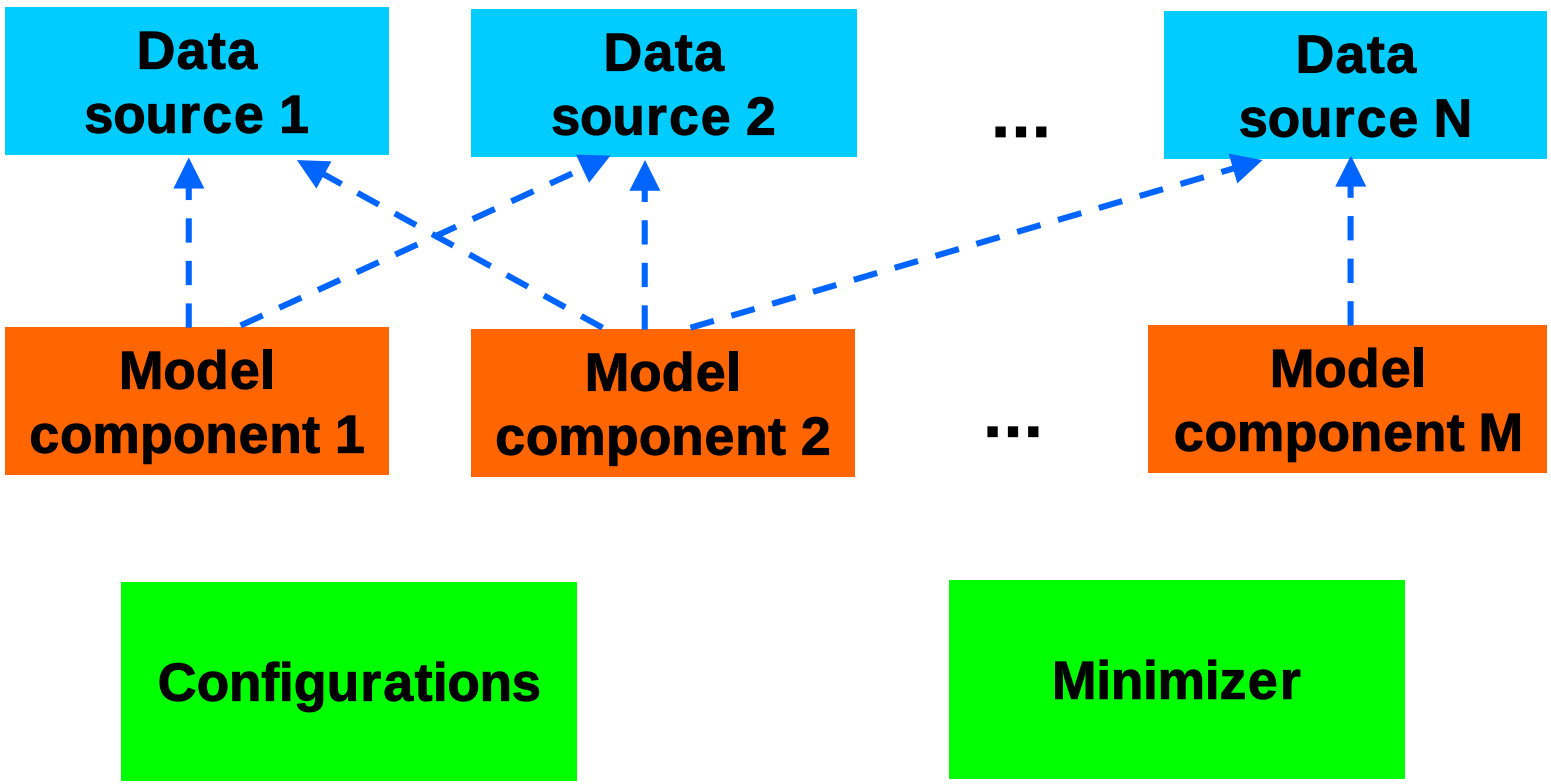
S.Linev, J. Adamczewski, M. Al-Turany, D. Bertini, H.G.Essel

ROOT 2002

Motivation

- Fit package should have clear and easy-to-use interface to setup fitting task
- Same fit setup should be easely reused for a number of similar data
- Should be possible to fit more then one data objects (histograms or other) simultaneously
- Should be possible to store full fit setup to reuse it in other program or reproduce all fit results later
- Should be possible to create GUI, where be able to setup and perform fit of any complexity

Structure of TGo4Fitter



TGo4FitData class

- provides methods to get bins from data source
- supports N-dimensional case
- defines range(s) where data should be fitted
- can has special calibrations object for each axis

Current implementations:

- TGo4FitDataHistogram — TH1, TH2, TH3
- TGo4FitDataGraph — TGraph
- TGo4FitDataProfile — TProfile

TGo4FitModel class

- Handles a list of model parameters
- Each parameter has so-called “full name”, combined from the parameter name and owner (model component) name
- Has special amplitude parameter, which is considered separately from other so-called “shape” parameters
- Has method to build shape of this component according to given parameters values
- Can be assign to several data objects
- Defines range(s) where model is build

Current implementations

- TGo4FitPolynom – component of N-dim polynomial function like $x^{Px} \cdot y^{Py} \cdot z^{Pz} \cdot \ldots$
- TGo4FitGauss1 – 1-dim gaussian
- TGo4FitGauss2 – 2-dim gaussian
- TGo4FitGaussN – N-dim gaussian
- TGo4FitFormula – component, uses TFormula facility
- TGo4FitUserFunction – user function taken either from program code or from shared libraray
- TGo4FitModelFromData – model bins are taking from data object (for instance, TGo4FitDataHistogram)

Fitter configuration

- Sets up dependency between models parameters, using TFormula facilities like Par2 = Par0 + Par1/2
- Sets up for some parameters initial values, which also can be caclulated as expression from other parameters
- Defines some parameters as fixed during minimization
- Redefines minimization setup (valid range or epsilon) for some parameters
- Introduces new parameters, which can be used in dependency calculations and minimized instead of some of model parameters

Minimizer

- TMinuit class is used
- Any list of Minuit commands can be executed
- Additional “result”command introduced, which can store status and Minuit results like covariation matrice, parameters errors estimations, contour plot. Several “result” commands can be inserted to command list.
- Any other minimization methods can be implemented

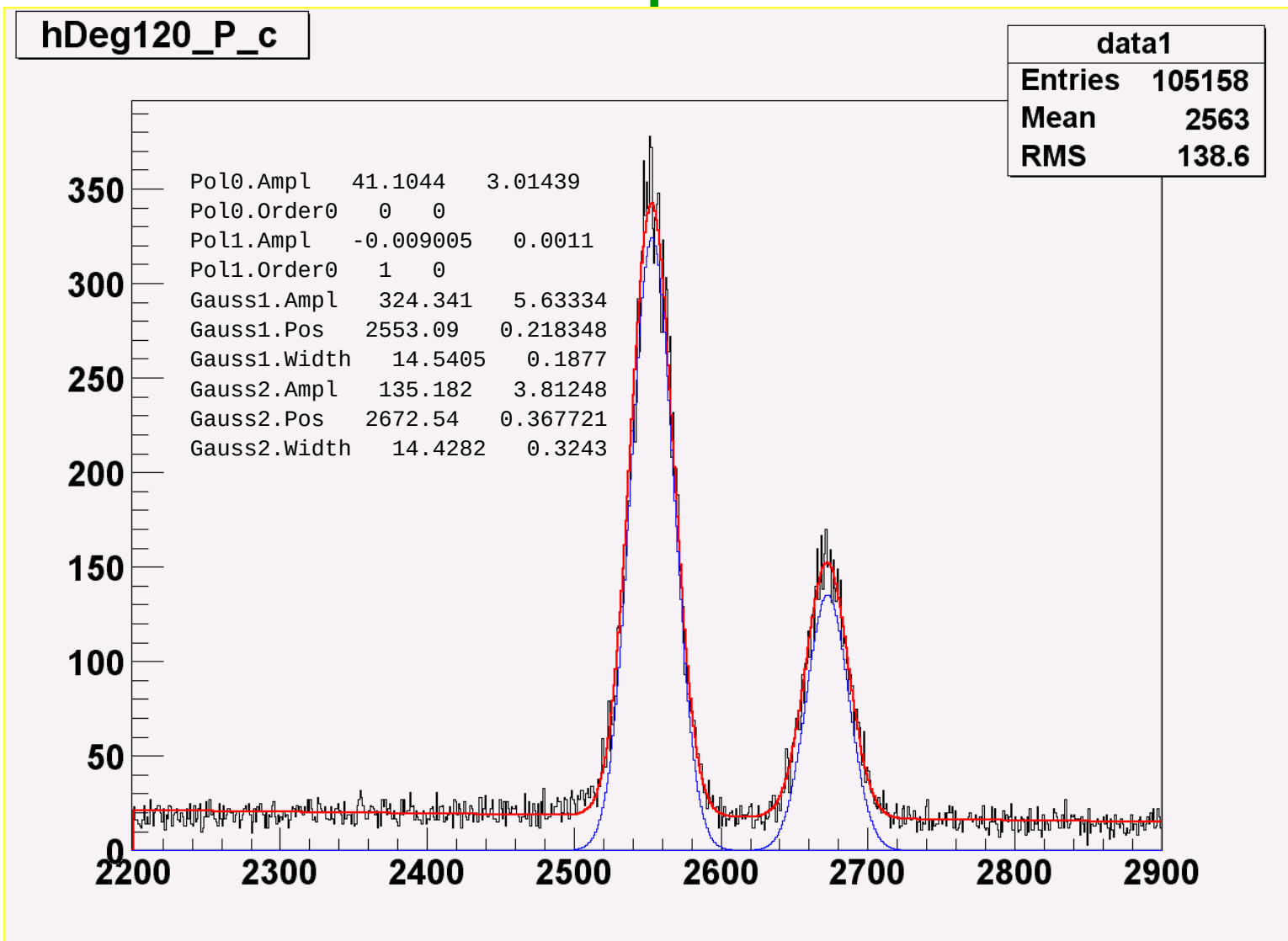
TGo4Fitter class

- Contain full setup, necessary to perform one fit
- Provides access to all objects via names or via full names
- Has a fast method for amlitude estimations
- Fitter can set and reset histograms or calibrations or other objects, situated in model and data objects, directly
- Can be stored with or without supplied data
- Has several fitting functions like χ^2, some of modified χ^2, maximum likelihood or user specified function

Current status of Go4Fit

- Design of base classes almost finish
- Implementation for specific data source objects and basic model components
- Independent part of Go4
- Can be used in compiled code or with CINT
- Has a set of simple examples
- GUI development underway

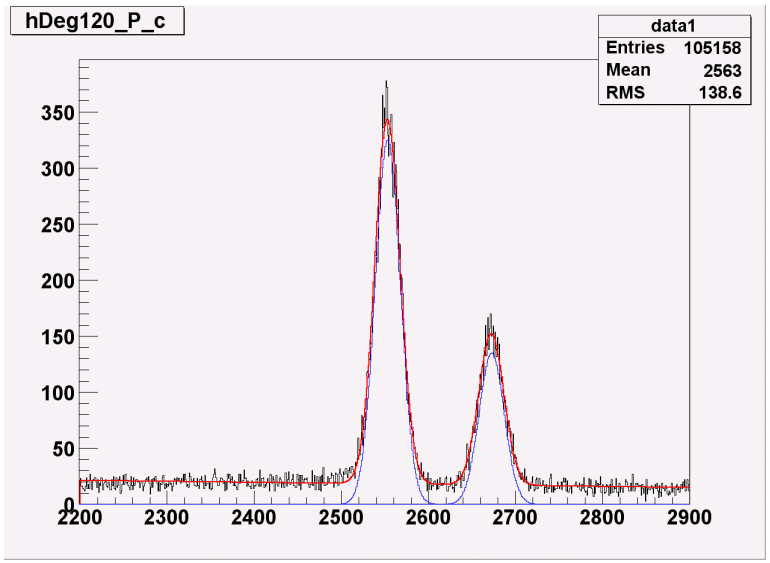
Example 1

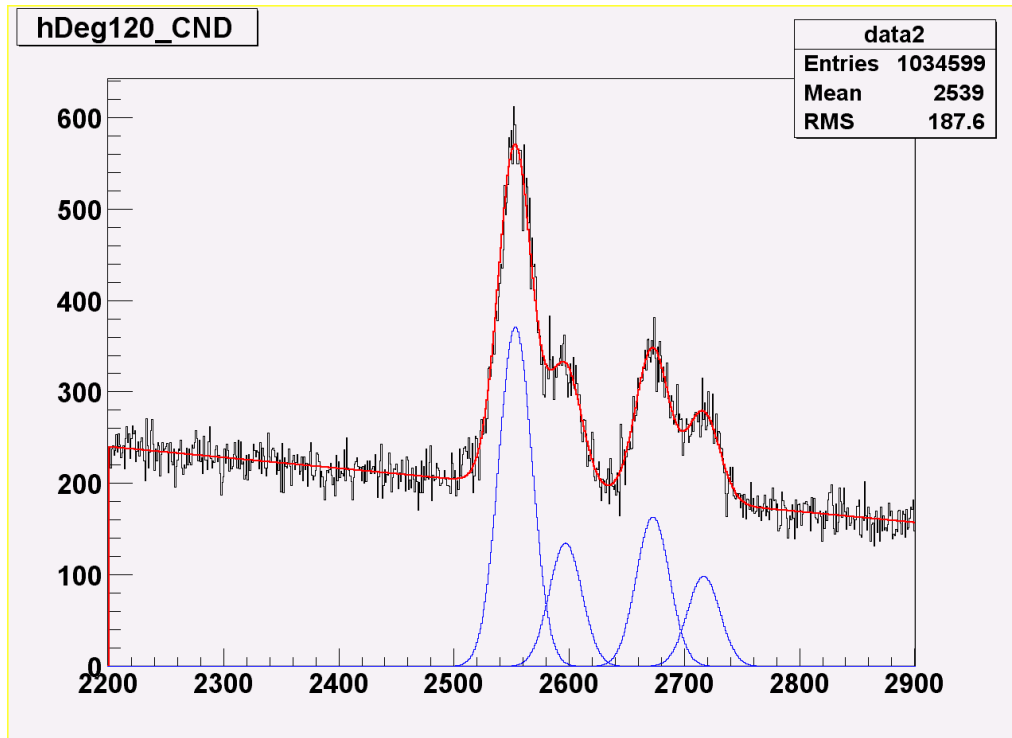

hDeg120_P_c
data1
Entries 105158
Mean 2563
RMS 138.6
Pol0.Ampl 41.1044 3.01439
Pol0.Order0 0 0
Pol1.Ampl -0.009005 0.0011
Pol1.Order0 1 0
Gauss1.Ampl 324.341 5.63334
Gauss1.Pos 2553.09 0.218348
Gauss1.Width 14.5405 0.1877
Gauss2.Ampl 135.182 3.81248
Gauss2.Pos 2672.54 0.367721
Gauss2.Width 14.4282 0.3243
350
300
250
200
150
100
50
0
2200
2300
2400
2500
2600
2700
2800
2900

Example1.C

```
void Example1() {

   gROOT->Reset();
   TH1::AddDirectory(kFALSE);

// create fitter together with minimizer
   TGo4Fitter fitter("Fitter",TGo4Fitter::ff_ML_Poisson);

// create object to fit
   TH1* histo = LoadHisto("hDeg120_P_c");
   TGo4FitDataHistogram *data = new TGo4FitDataHistogram("data1",histo,kTRUE);
   data->SetUseBinScale(kTRUE);
   data->SetRange(0,2200.,2900.);
   fitter.AddData(data);

// create four models and add them to fitter
   fitter.AddModel( "data1", new TGo4FitPolynom("Pol0",0.) );
   fitter.AddModel( "data1", new TGo4FitPolynom("Pol1",1.) );
   fitter.AddModel( "data1", new TGo4FitGauss1("Gauss1",2553.,15.) );
   fitter.AddModel( "data1", new TGo4FitGauss1("Gauss2",2672.,15.) );

// do fit and show results
   fitter.Initialize();

   fitter.EstimateAmplitudes();
   fitter.DoMinimization();

   new TCanvas("Canvas","Example 1 histogram");
   fitter.Draw("data1,Gauss1,Gauss2");
   fitter.PrintPars();

   fitter.Finalize();

}
```

Example 2


hDeg120_P_c
data1
Entries 105158
Mean 2563
RMS 138.6
350
300
250
200
150
100
50
0
2200
2300
2400
2500
2600
2700
2800
2900


hDeg120_CND
data2
Entries 1034599
Mean 2539
RMS 187.6
600
500
400
300
200
100
0
2200
2300
2400
2500
2600
2700
2800
2900

Example2.C

```
TObject* BuildFitter() {
   TGo4Fitter *fitter = new TGo4Fitter("Fitter",TGo4Fitter::ff_ML_Poisson,kFALSE);

   TGo4FitMinuit* fMinuit = new TGo4FitMinuit("Minuit","Minimization object");
   fMinuit->AddCommand("MIGRAD 5000 1");
   fMinuit->AddCommand("MIGRAD 5000 1");
   fMinuit->AddCommand("MINOS 100");
   fitter->SetMinimizer(fMinuit,kTRUE);

   TGo4FitDataHistogram *data1 = new TGo4FitDataHistogram("data1",0);
   data1->SetUseBinScale(kTRUE);
   data1->SetRange(0,2200,2900);
   fitter->AddData(data1);

   TGo4FitDataHistogram *data2 = new TGo4FitDataHistogram("data2",0);
   data1->SetUseBinScale(kTRUE);
   data2->SetRange(0,2200,2900);
   fitter->AddData(data2);

   TGo4FitModel* gauss1 = new TGo4FitGauss1("Gauss1",2553,15);
   gauss1->AssignToData("data1"); gauss1->AssignToData("data2",1.2);

   TGo4FitModel* gauss2 = new TGo4FitGauss1("Gauss2",2672,15);
   gauss2->AssignToData("data1"); gauss2->AssignToData("data2",1.2);

   fitter->AddModel( "data1", new TGo4FitPolynom("Pol0_1",0.) );
   fitter->AddModel( "data1", new TGo4FitPolynom("Pol1_1",1.) );
   fitter->AddModel( "data2", new TGo4FitPolynom("Pol0_2",0.) );
   fitter->AddModel( "data2", new TGo4FitPolynom("Pol1_2",1.) );

   fitter->AddModel( gauss1 );
   fitter->AddModel( gauss2 );
   fitter->AddModel( "data2", new TGo4FitGauss1("Gauss3",2597,14) );
   fitter->AddModel( "data2", new TGo4FitGauss1("Gauss4",2717,14) );

   return fitter;
}
```

Example2.C

```
void Example2() {

   gROOT->Reset();
   TH1::AddDirectory(kFALSE);

// creating fitter and store it to file
   TGo4Fitter* fitter = (TGo4Fitter*) BuildFitter();
   StoreFitter(fitter,"Stage1.root");
   delete fitter;

// read back fitter, supply data, do fit and store in other file
   fitter = (TGo4Fitter*) LoadFitter("Stage1.root");

   fitter->SetObject(LoadHisto("hDeg120_P_c"),"data1",kTRUE);
   fitter->SetObject(LoadHisto("hDeg120_CND"),"data2",kTRUE);

   fitter->Initialize();
   fitter->EstimateAmplitudes();
   fitter->DoMinimization();
   fitter->Finalize();

   StoreFitter(fitter,"Stage2.root");
   delete fitter;

// read fitter from file and show results of fit
   fitter = (TGo4Fitter*) LoadFitter("Stage2.root");

   fitter->Initialize();
   new TCanvas("Canvas1","First histogram");
   fitter->Draw("data1,Gauss1,Gauss2");
   new TCanvas("Canvas2","Second histogram");
   fitter->Draw("data2,Gauss1,Gauss2,Gauss3,Gauss4");
   fitter->Finalize();

   fitter->PrintPars();
   delete fitter;
}
```

Example2. Result parameters

```
*** LIST OF PARAMETERS VALUE ***
    Pol0_1.Ampl    41.1835      1.42716
    Pol0_1.Order0   0           0
    Pol1_1.Ampl    -0.00903     0.000556
    Pol1_1.Order0   1           0
    Pol0_2.Ampl   500.058       6.88113
    Pol0_2.Order0   0           0
    Pol1_2.Ampl    -0.1182      0.00269
    Pol1_2.Order0   1           0
    Gauss1.Ampl   325.238       5.47593
    Gauss1.Pos   2553.27        0.192105
    Gauss1.Width   14.4906      0.168595
    Gauss1.Ratio1   1.14139     0.0272282
    Gauss2.Ampl   134.996       3.74136
    Gauss2.Pos   2672.64        0.334977
    Gauss2.Width   14.4522      0.298319
    Gauss2.Ratio1   1.20911     0.0510514
    Gauss3.Ampl   134.858       6.44543
    Gauss3.Pos   2596.79        0.823768
    Gauss3.Width   14.2116      0.872011
    Gauss4.Ampl   98.5731       5.86399
    Gauss4.Pos   2716.74        1.00646
    Gauss4.Width   14.153       1.03049
```