PROCEEDINGS OF SCIENCE

Recent Improvements of the ROOT Fitting and Minimization Classes

L. Moneta*
CERN, Geneva, Switzerland
E-mail: Lorenzo.Moneta@cern.ch

I. Antcheva
CERN, Geneva, Switzerland
E-mail: Ilka.Antcheva@cern.ch

D. Gonzalez Maline
CERN, Geneva, Switzerland
E-mail: David.Gonzalez.Maline@cern.ch

Advanced mathematical and statistical computational methods are required by the LHC experiments for analyzing their data. Some of these methods are provided by the Math work package of the ROOT project, a C++ Object Oriented framework for large scale data handling applications. We present in detail the recent developments of this work package, in particular the recent improvements in the fitting and minimization classes, which have been re-designed and re-implemented with an object-oriented approach. New minimization algorithms have been added recently in ROOT and they can be used consistently for fitting via a common interface. These algorithms include Minuit2, the new object-oriented version of Minuit, various minimization methods from the GNU Scientific libraries, stochastic and genetic algorithms. Furthermore, a new graphical user interface has been also developed for performing and monitoring fits on ROOT data objects such as histograms, graphs and threes in both one or multi-dimensions. We will describe in detail the new capabilities provided by the new fitting and minimization classes and the functionality of the new user interface.

XII Advanced Computing and Analysis Techniques in Physics Research
November 3-7 2008
Erice, Italy

*Speaker.

PoS(ACAT08)075

1. Introduction

The ROOT Math work package is responsible to provide and to support a coherent set of mathematical and statistical libraries required for simulation, reconstruction and analysis of high energy physics data. Existing libraries provided by ROOT are in the process of being re-organized with the aim to avoid duplication, increase modularity and to facilitate support in the long term.

The basic core mathematical functionality in ROOT has all been included in the MathCore library. This library provides the basic implementation of mathematical functions and algorithms and it defines as well the required interfaces to use and extend these algorithm in a coherent way. Through these interfaces, it is possible to plug-in different implementations of numerical algorithms, which can exist in separate libraries. For example, in the case of numerical minimization, various implementations exist in ROOT and they can all be used for fitting via a new common minimization interface class.

Classes and interfaces required for fitting the ROOT data objects have been re-designed and re-implemented for the latest ROOT release (version 5.22). These new modular code allows the fitting of all ROOT data objects using some common methods and it removes previously existing code duplications, facilitating the maintenance in the long term.

In this paper, after presenting the content of the MathCore library we describe in detail the design and implementation of the new fitting and minimization classes. We present as well the recently developed graphical user interface for fitting. A detailed description of other recent new developments of the ROOT mathematical library can be found elsewhere [1].

2. MathCore library

This library contains a self-consistent set of functions and C++ classes needed for basic numerical computing. It consists up to now of the following components:

- commonly used mathematical functions like special functions and statistical distribution functions. These functions are provided as free functions in the name-space `TMath` or `ROOT::Math`;
- classes for random number generations (`TRandom` classes);
- basic numerical algorithms, like integration, derivation or simple minimization and their interfaces classes;
- fitting classes and interfaces used for implementing the fitting of ROOT data objects;
- abstract interfaces and adapter classes for defining function evaluation in one or more dimensions.

2.1 MathCore Interfaces

In order to have a common entry point, interfaces classes for the ROOT numerical methods have been developed in MathCore. The classes implementing these interfaces can be located in different ROOT libraries and they can be loaded automatically using the ROOT plug-in manager

system. A common interface exists in MathCore for multi-dimensional numerical minimization, the class `ROOT::Math::Minimizer`. Various implementations of this class exist in ROOT and they are located in various libraries, like Minuit [2], Minuit2 (a new C++ version of Minuit [3]), Fumili [4] or minimizers based on the GNU Scientific Library (GSL) [5] located in the MathMore library. The ROOT plug-in manager system creates, according to the user option, the corresponding implementation class by loading automatically its shared library. Furthermore, a stochastic minimization method based on simulated annealing is provided via the class `ROOT::Math::GSLSimAnMinimizer` implementing the Minimizer interface and a new one is being developed using a genetic algorithm. These methods do not rely on the function gradient and they can be used for finding the global function minimum, while a gradient method, such as the MIGRAD algorithm from Minuit [2], is able to find only a local function minimum.

For decoupling the algorithm implementations from the function definition used by the method, abstract interfaces for function evaluation in one or more dimensions have been introduced. They are used by all the ROOT Math classes implementing numerical algorithms, such as integration, derivation or minimization. Specialized interfaces are as well present for functions providing analytical derivatives. In this way, algorithms using derivative information, like minimization, can profit in some cases from using analytical derivatives or customized calculations provided by the users. Interfaces for parametric functions, which are used for fitting and data modeling are also provided. Figure 1 shows the available interfaces of one dimensional functions. The basic interface, `ROOT::Math::IGenFunction`, provides the method for evaluating the function (`double operator()(double x)`), while the gradient interface, `ROOT::Math::IGradFunction`, gives in addition the possibility to evaluate the derivative with respect to the coordinate. The parametric function interface, `ROOT::Math::IParamFunction`, is used in modeling and fitting and it provides in addition the functionality to retrieve and set the parameter values and to evaluate the function passing both coordinate and parameter values.

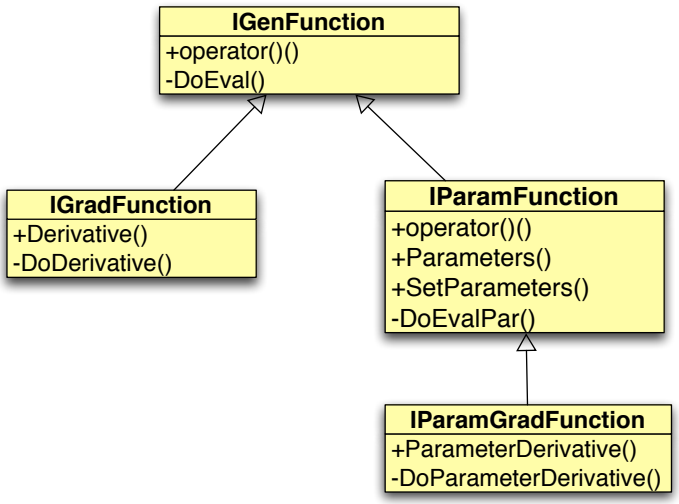

Figure 1: Interfaces for one-dimensional function evaluation present in the MathCore library and used by all the numerical algorithms including fitting and minimization

An interface exists as well for functions providing also the derivatives with respect to the parameters, `ROOT::Math::IParamGradFunction`.

For user convenience, adapter classes are available to wrap any free function, C++ callable object (functor) or member function, with the right signature in the desired interface. In this way, user function classes can be easily created and plugged in the required numerical algorithm class. For example, one-dimensional function classes can be created from global functions like `double f(double x)` , from C++ classes implementing the `double operator()(double x)` or any member function of a class returning a `double` and taking a `double` as argument. This approach is very powerful, giving the user the possibility to customize the function objects using its constructor, and it is also very easy to use.

The existing drawable `TF1` class in ROOT has also been also extended with the possibility to be created from function objects (functors) or class member functions with the right signature (`double F (double * x, double * p)`).

3. New ROOT Fitting Classes

Fitting in ROOT is possible directly via the `Fit` method of the data object classes, like histograms (`TH1::Fit`), graphs and trees. Various fitting techniques, such as least-square or binned and un-binned maximum likelihood, are supported depending on the type of data or user choice. The fitting solution is obtained by finding the minimum of the objective function, least-square or likelihood, using a numerical minimization algorithm. For the new ROOT release, version 5.22, new classes have been designed and implemented using an object-oriented approach and they are now used for fitting the data objects.

The minimization and fitting functionality is now separated in this new design. Separate classes describe as well the fit data and the model function which are provided as input to the fitter class. The fitter has the role to build the appropriate objective function to be minimized and it drives the minimization process, by creating the minimizer class and setting all the control parameters and options. As explained previously, it is possible to select at run-time different minimization algorithms depending on the complexity of the problem. For example, in case of linear least-square fitting, an implementation exists for finding directly the solution by solving a linear matrix system. The model function, which is provided by the user, must implement the parametric function interface. The fit data are instead described by separate classes for binned or un-binned data sets. Classes exist as well for defining objective functions used in typical fitting methods. A χ^2 function class is used for fitting binned data sets, a Poisson log-likelihood function is for fitting binned data sets described with Poisson statistics, like histograms and a generic log-likelihood function is used for fitting unbinned data sets with the maximum likelihood method.

All this classes reside in the MathCore library and they are independent of other ROOT libraries. For fitting ROOT data objects, like the `TH1` using `TF1` function objects, common methods exist in the histogram library for creating the fit data set from the histogram or graph class and for making the required calls on the fitter class according to the user specified options. Adapter classes exist for converting automatically the `TF1` function objects to the needed MathCore function interface.

This design in addition to increase the modularity of the fitting system allows also the re-usability with different types of data sets or functions, even defined outside the ROOT framework. It has also reduced the overall code, by using now the same implementation of the χ^2 function for fitting all the ROOT histograms and graph classes (`TH1`, `TGraph`, `TGraph2D`, `TMultiGraph`).

Another important capability of this new fitting system, is the multi-thread support for parallel fits. This allows the achievement of optimal scalable performances on multi-core CPU's when fitting time consuming tasks. In the latest release, parallelization has been made possible, by spawning the derivative calculation in the Minuit2 minimization algorithm on multi-threads using OpenMP. Studies have been performed as well by using MPI [6]. The new fitting classes allow as well to parallelize the calculation of the objective function using multi-thread with shared memory. An implementation of a parallel likelihood or χ^2 calculation using OpenMP is foreseen to be released in the next ROOT version.

4. The Fit Panel

The fits in ROOT can be performed by using a new Graphical User Interface, which has been re-designed recently and improved by adding new functionality. The fit panel interface became part of the ROOT version 5.14 and it has been re-designed and enhanced by new functionality recently for the 5.22 version of ROOT. Its goal is to provide a more user friendly way for fitting directly the ROOT data objects with the various available options. It allows an easy selection of a data set, a model function, a fit method and fitting options.

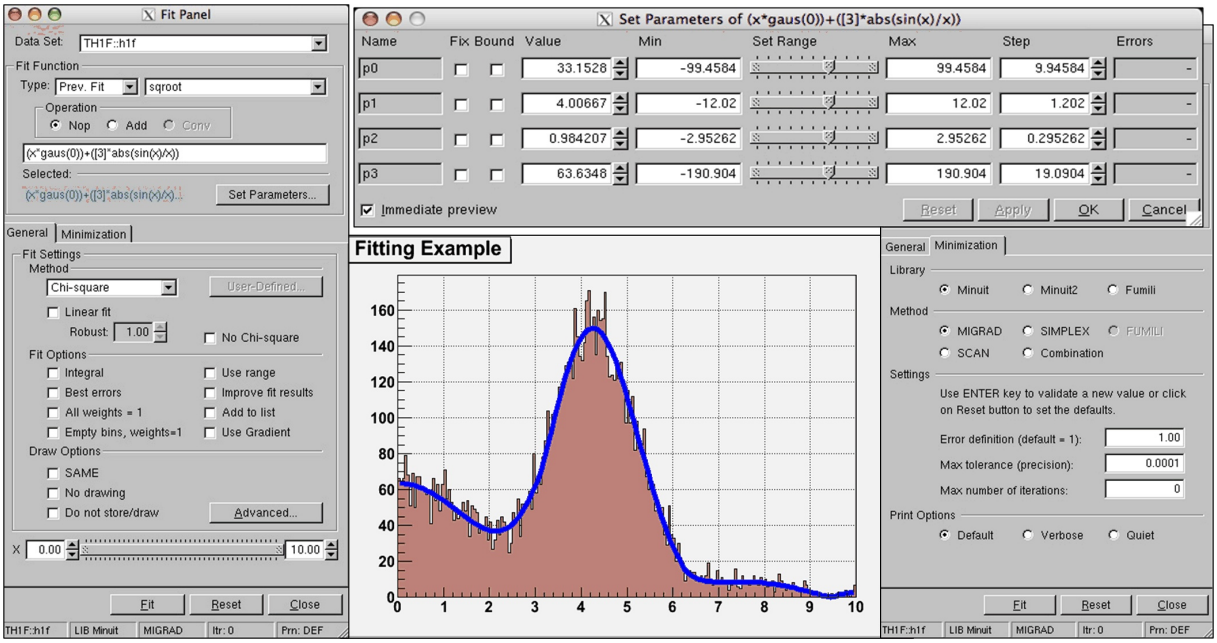

Figure 2: The General tab (left), the Set Parameter dialog (up right) and the Minimization tab (bottom right) of the fit panel.

The fit panel can be opened directly via the *Tool* menu of the ROOT Canvas or via the context menu of any ROOT data object which is suitable for fitting. The context menu is available after a right mouse click on an object. The users can select via the fit panel a data set, from the available list of ROOT objects, and a fit model function. The fit function can be choosen from the list of pre-defined functions, like gaussian or polynomials, from user defined functions, which are available in the global ROOT list of functions, or from previously used functions, which have been stored with the data object. In addition, users can modify the fit function formula expression or enter a new one. A dialog can be used for setting the initial values of the parameters. Users can also control the parameters status with the possibility to fix some of them or setting their limits. The *General* tab provides the user interface for selecting the fit method whenever possible, like χ^2 or binned likelihood when fitting histograms. In addition, it is possible to set anyone of all the available fitting and drawing options. The fit range can also be controlled using a slider. The *Minimization* tab offers an easy choice of minimizer (like Minuit or Minuit2), minimization methods (like Migrad or Simplex) and the possibility to control the minimization parameters such as error definition, maximum tolerance, the maximum number of iterations, and print options. The functionality included in the latest release gives the possibility to draw contour plots or performing scan of the objective function. This is available via the advanced drawing dialog which can be opened from the *General* tab. The fit panel can be used for fitting all kind of ROOT data objects, including unbinned maximum likelihood fits of ROOT `TTree` objects. Extensions of this user interface are planned for building more complex fit function models by interfacing to RooFit toolkit [7] classes describing probability density functions.

5. Advanced Fitting and Statistical Tools

Advance fitting is possible by using the RooFit package [7], which is being distributed together with ROOT. RooFit can be used to build complex models, by combining probability density functions describing the observed events. The model can then be used to perform maximum likelihood fits, to produce advanced plots of the obtained fit results, and also to generate toy Monte Carlo samples for various studies.

A new package is currently being developed to provide advanced statistical functionality in ROOT in a consistent way. This packaged, named RooStats [8], is based on the RooFit classes for describing probability density functions or likelihood functions. It aims to satisfy the requirements of the LHC experiments for statistical data analysis by providing tools for calculating confidence intervals and for performing statistical hypothesis tests. It can then be used for estimating the discovery significance at LHC. The idea is to include the major statistical techniques for performing these calculations which have been approved by the experiment statistical committees. By using common interfaces different techniques, based on both frequentists and bayesian statistics, can be used easily, and results can be compared. In addition, by profiting from recent RooFit developments, like the `RooWorkSpace` class, which can be used to store all the modeling information including the likelihood functions and the data sets, the results can be shared between different experiments and their combination can also be performed using a consistent statistical treatment.

6. Conclusions

ROOT contains already a large variety of mathematical and statistical tools required for the analysis of LHC data. An effort is on-going to consolidate the existing libraries by improving the algorithms, making them easier to use and increasing their modularity to gain in long term maintainability. The recent developments of the fitting and minimization classes described in this paper provide an example of these improvements. The needs and the feedback received from users working on data analysis and reconstruction of the data are as well taken into account in this consolidation process. RooStats is a new package developed together with contributors from the CMS and ATLAS experiments. The final aim is to provide advanced mathematical and statistical tools which are considered standard by the community.

References

[1] L. Moneta et al., *New Developments of ROOT Mathematical Software Libraries*, proceeding to ACAT-2007.

[2] F. James, *MINUIT Reference Manual*, CERN Program Library Writeup D506.

[3] M. Hatlo *et al.*, *Developments of Mathematical Software Libraries for the LHC experiments*, *IEEE Transactions on Nuclear Science* **52-6**, 2818 (2005)

[4] S. Yashchenko, *New method for minimizing regular functions with constraints on parameter region*, Proceedings of CHEP'97 (1997).

[5] M. Galassi et al, *The GNU Scientific Library Reference Manual* - Second Edition, ISBN = 0954161734 (paperback). See also the url `http://www.gnu.org/software/gsl`.

[6] A. Lazzaro and L. Moneta, *MINUIT Package Parallelization and applications using the RooFit Package* , these proceedings.

[7] W. Verkerke and D. Kirkby, *The RooFit Toolkit for data modeling*, proceedings to PHYSTAT05. See also the url `http://roofit.sourceforge.net`.

[8] W. Verkerke, *Statistical Software for the LHC*, proceedings of the PHYSTAT LHC Workshop, CERN 27-29 June 2007, 169. See also the url `http://twiki.cern.ch/twiki/bin/view/RooStats/WebHome`.