

ROOT Mathematical Libraries

A. Kreshuk, L. Moneta, E. Offermann

CERN/PH-SFT

LCG AA Internal Review, 18 September 2006



Root Math Work Package



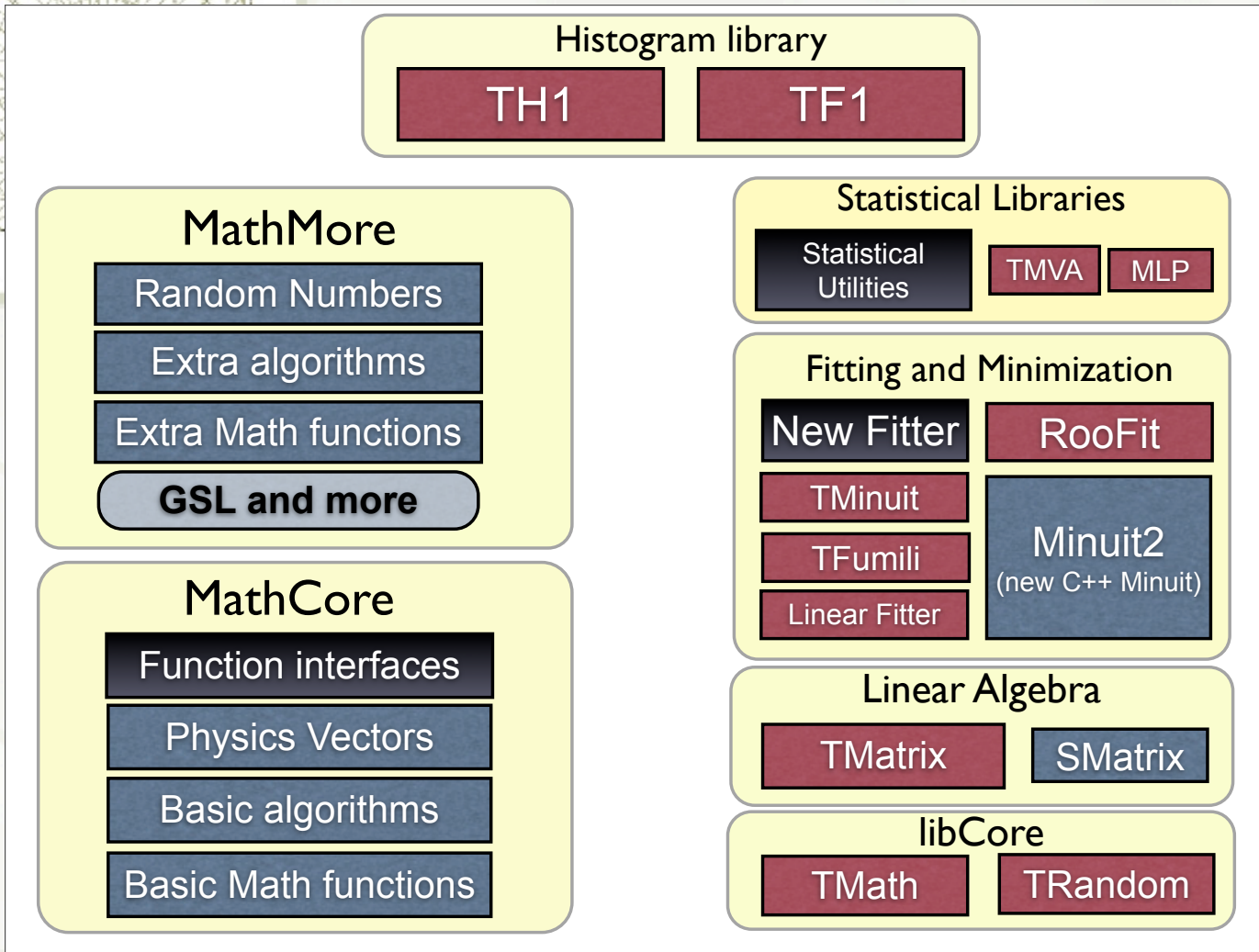
- ✦ Work package formed with ROOT-SEAL merge
- ✦ Main responsibilities for this work package:
 - ✦ Basic mathematical functions
 - ✦ Numerical algorithms
 - ✦ Random numbers
 - ✦ Linear algebra
 - ✦ Physics and geometry vectors (3D and 4D)
 - ✦ Fitting and minimization
 - ✦ Histograms (math and statistical part)
 - ✦ Statistics (confidence levels, multivariate analysis)

Outline

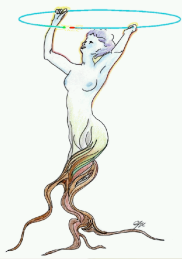


- ★ New ROOT Math Libraries : *MathCore* and *MathMore*
 - ★ Physics and vector package (*GenVector*)
 - ★ *SMatrix* package
 - ★ mathematical functions and numerical algorithms
- ★ Fitting and Minimization
 - ★ new C++ Minuit (*Minuit2*)
 - ★ Linear and robust fitter
 - ★ plans for new fitting classes
 - ★ fitting GUI (new fit panel)
- ★ Other recent developments:
 - ★ improvements in random numbers
 - ★ Histogram comparison
 - ★ *TMVA* (multivariate analysis)
 - ★ FFT, *SPlot*
- ★ Future plans

ROOT Math Libraries

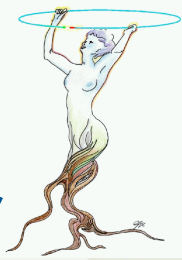


Packages and Dictionaries



- ✦ *MathCore*, *MathMore*, *SMatrix* and *Minuit2* can be released and built as independent components
 - ✦ *MathMore* requires GSL and released with GPL license
 - ✦ use *autoconf/automake/libtool* to configure and build
- ✦ Libraries built outside ROOT (with *auto*-tools) do not provide the CINT dictionary
- ✦ For template classes (like *GenVector* and *SMatrix*) the dictionary is provided for the most used types
 - ✦ `double`, `float` and `Double32_t`
 - ✦ dictionary is the dominant part of the library
 - ✦ 2 Mb on Linux of a 2.3 Mb library for *MathCore*

Physics and Geometry Vectors

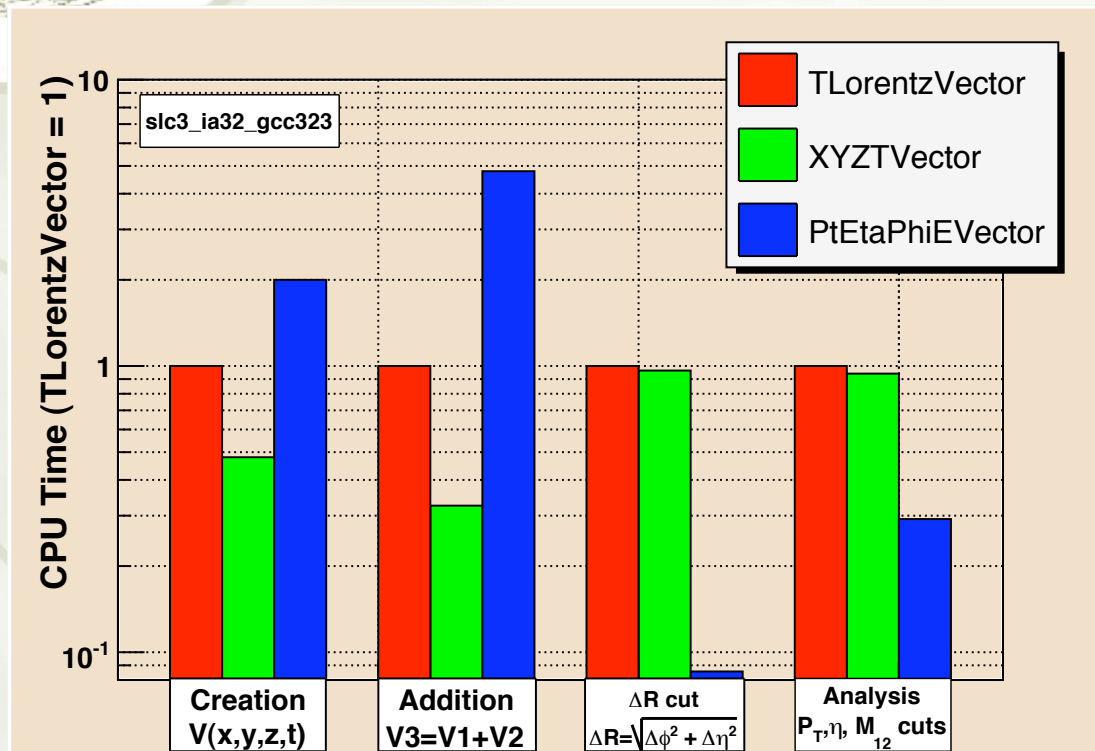


- ★ Classes for 3D and 4D vectors with their operations and transformations (rotations)
 - ★ functionality as in CLHEP *Vector* and *Geometry* packages
- ★ Work done in collaboration with Fermilab computing group (*M. Fischler, W. Brown and J. Marraffino*)
- ★ Main features of the new classes:
 - ★ generic scalar contained type
 - ★ i.e. single or double precision
 - ★ generic coordinate system concept
 - ★ i.e. cartesian, polar and cylindrical
- ★ Used now by CMS and LHCb

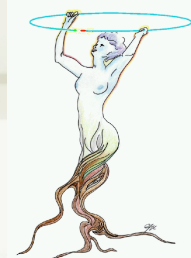
GenVector Performances



★ Optimal run-time performance

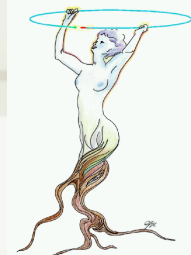


- ◆ no virtual calls and use of inline methods
- ◆ user can choose best coordinate system



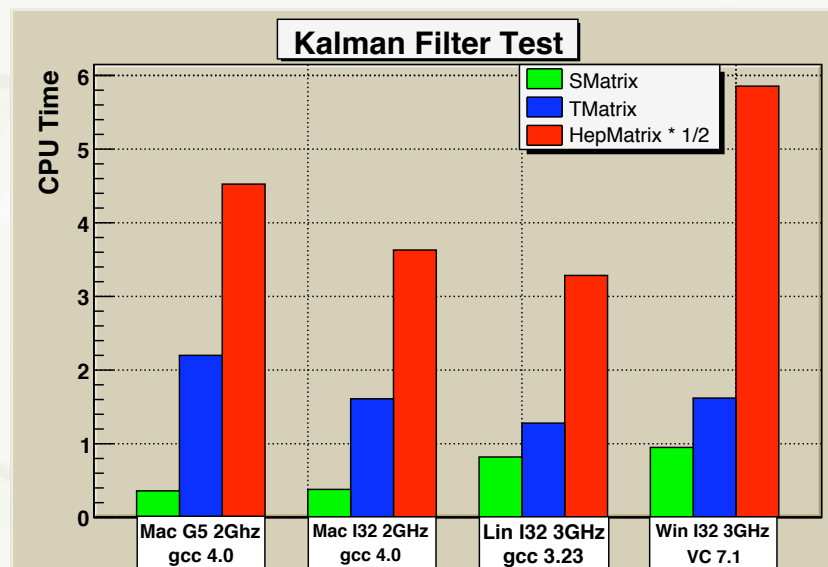
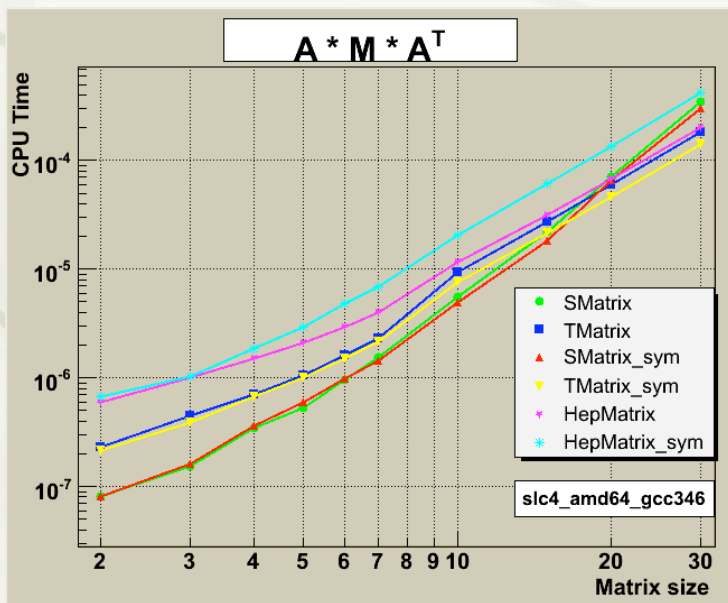
SMatrix Package

- ✦ Package initially developed by T. Glebe for HeraB
- ✦ Matrix and vector classes of arbitrary type
- ✦ For **fixed** (not dynamic) matrix and vector sizes :
 - `SMatrix< double, 2 , 5>`
 - `SVector< double, 5 >`
- ✦ Complementary and **NOT** a replacement of *TMatrix*
- ✦ Optimized for small matrix sizes:
 - ✦ use expression templates to avoid temporaries
- ✦ Support for symmetric matrices (thanks to J.Palacios, LHCb)
 - ✦ storage of only $n * (n+1) / 2$ elements
- ✦ Support for basic operations and matrix inversion
 - ✦ not full linear algebra functionality
- ✦ Used by LHCb, CMS and now ATLAS

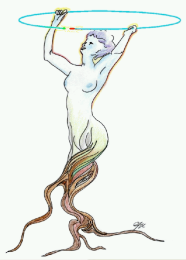


SMatrix Performances

- ◆ Comparison ROOT (*TMatrix*/*SMatrix*) and CLHEP (*HepMatrix*)
 - ◆ CPU performances in matrix operations varying the sizes
 - ◆ CPU performances in the Kalman filter update equations
 - ◆ sizes: 2x2, 2x5, 5x5 with addition, multiplication and inversion
 - ◆ Useful exercise also for *TMatrix* (achieved substantial improvements)



Mathematical Functions



★ Special Functions:

- ★ use interface proposed to C++ standard:

```
double cyl_bessel_i (double nu, double x);
```

- ★ large variety of functions complementing what exists in *TMath*

★ Statistical Functions:

- ★ Probability density functions (pdf)
- ★ Cumulative distributions (lower tail and upper tail)
- ★ Inverse of cumulative distributions
- ★ Coherent naming scheme. Example chi2:

```
chisquared_pdf  
chisquared_prob, chisquared_quant,  
chisquared_prob_inv, chisquare_quant_inv
```



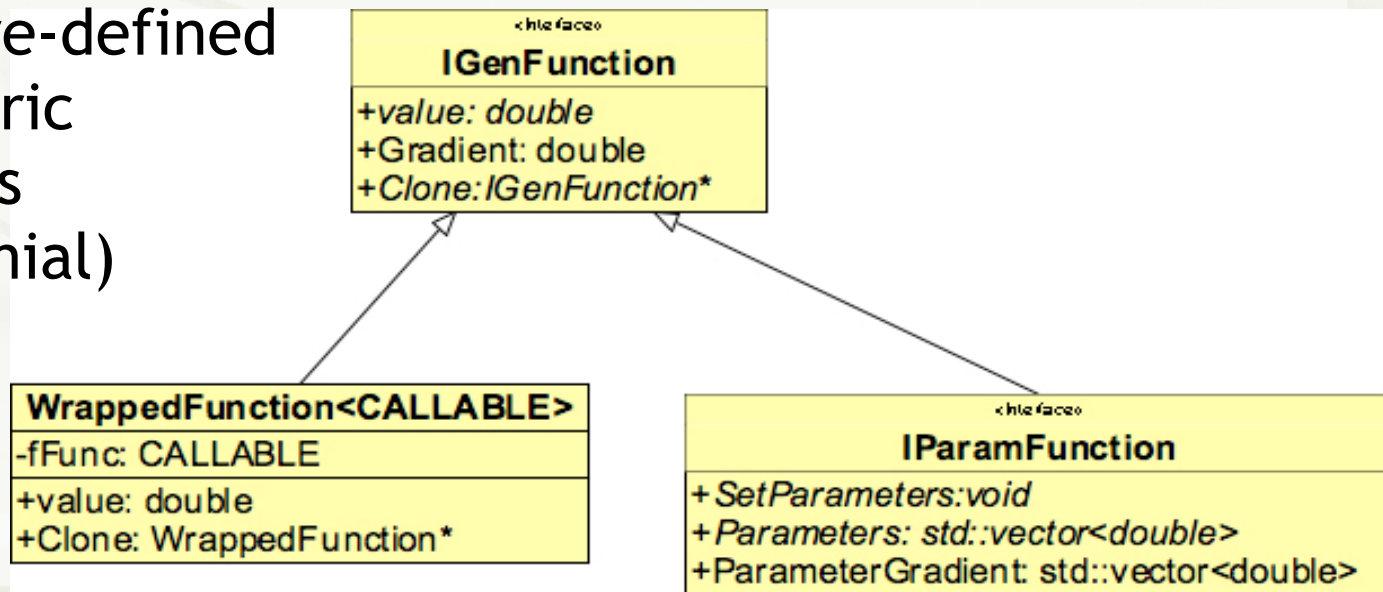
Numerical Algorithms

- ✦ C++ interface to GSL numerical algorithms
- ✦ Have now in the *MathMore* library algorithms for 1D functions:
 - ✦ **Numerical Derivation**
 - ✦ central evaluation (5 points rule) and forward/backward
 - ✦ **Numerical Integration**
 - ✦ adaptive integration for finite and infinite intervals
 - ✦ **Root Finders**
 - ✦ bracketing and polishing algorithms using derivatives
 - ✦ **Interpolation**
 - ✦ linear, polynomial and Akima spline
 - ✦ **Chebyshev polynomials** (for function approximation)

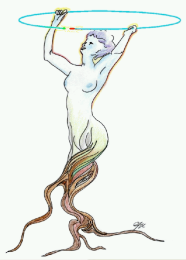


Function Interface

- ✦ Minimal interface used by all numerical algorithms:
- ✦ abstract classes (**IGenFunction** and **IParamFunction**)
- ✦ template **WrappedFunction** class to wrap any C++ callable object (functors, C free function, etc..)
- ✦ set of pre-defined parametric functions (Polynomial)



Fitting and Minimization



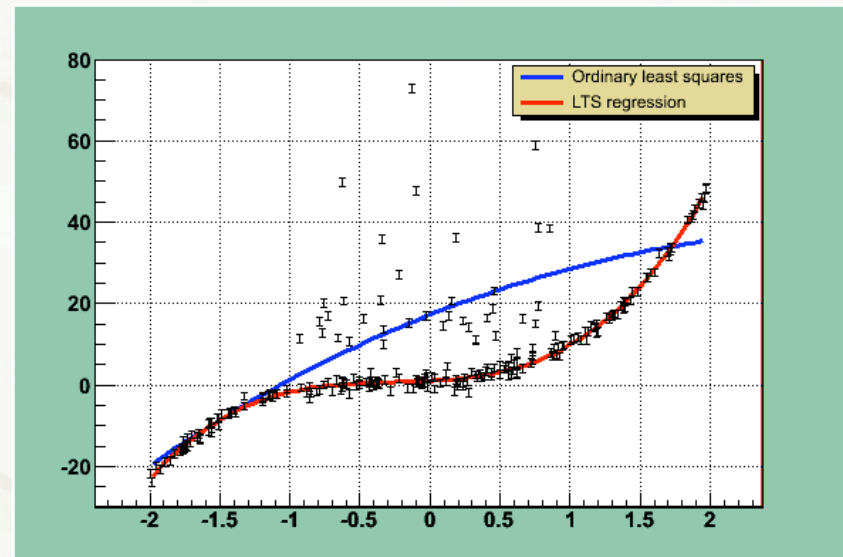
- ★ New C++ version of Minuit (*Minuit2*) in ROOT v5.08
 - ★ adapted SEAL packages to ROOT coding convention
 - ★ implemented a ROOT fitter interface (*TVirtualFitter*)
- ★ Same basic functionality as in old version
 - ★ *Migrad*, *Simplex*, *Minos* algorithms
- ★ Extended functionality:
 - ★ single side parameter limits
 - ★ added *Fumili* method for Chi2 and likelihood fits
- ★ OO package for generic function minimization
 - ★ easy to extend by inserting new minimization algorithms
 - ★ plan to add constrained minimization

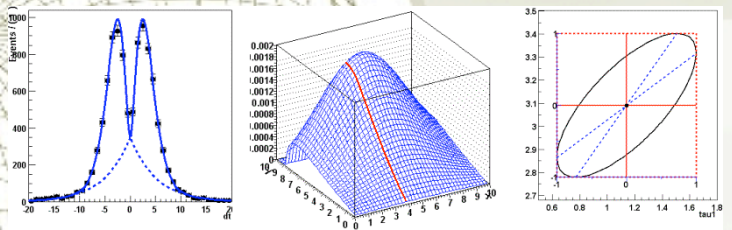
Linear and Robust Fitter



- ★ **TLinearFitter** class to fit function linear in the parameters (e.g Polynomial)
 - ★ direct solution by solving a linear system
 - ★ can be 10-15 times faster than Minuit
- ★ Robust Fitting
 - ★ outliers removal
 - ★ use of Least Trimmed Square (LTS) regression

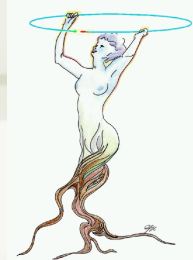
```
Graph.Fit("pol3", "rob=0.75", -2, 2);
```





RooFit

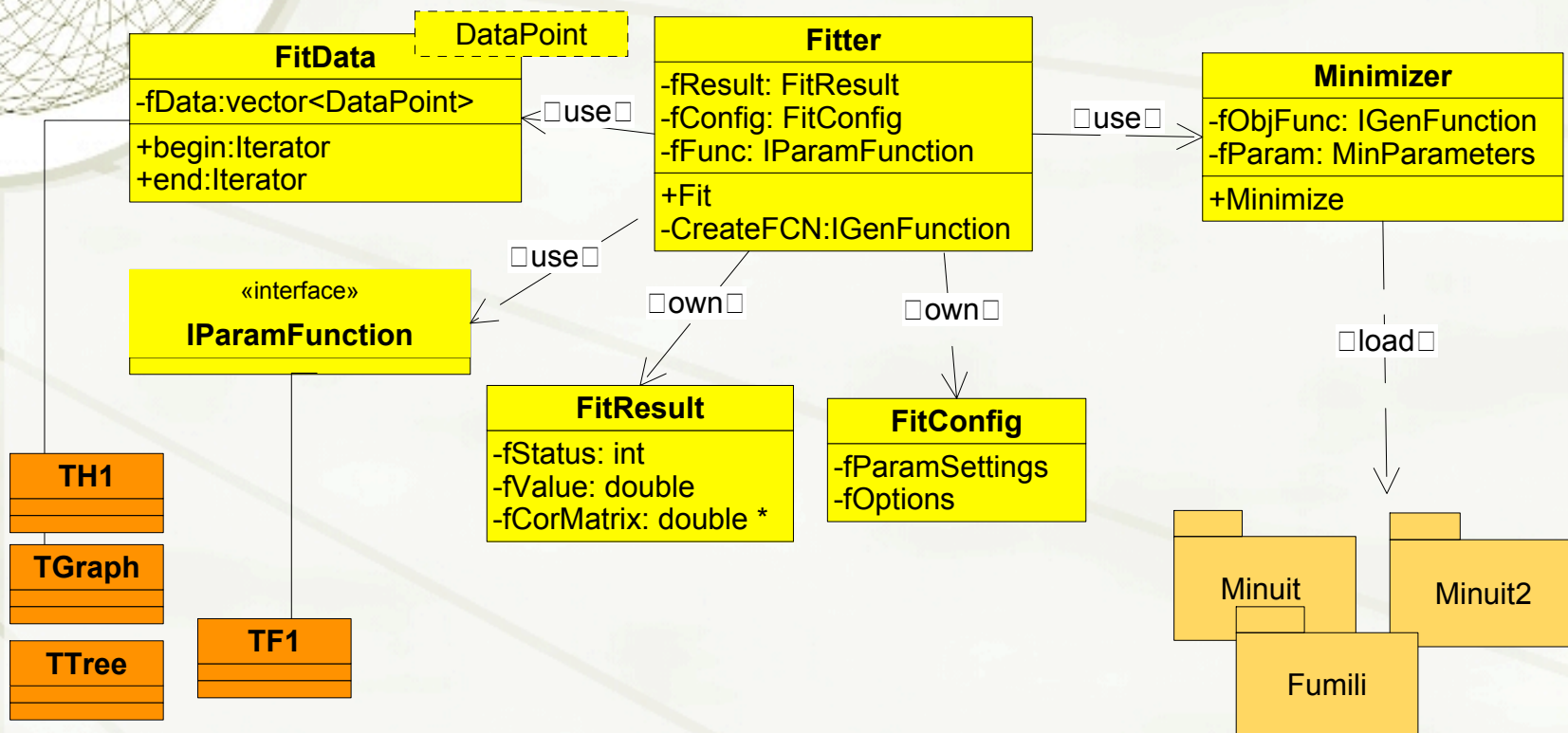
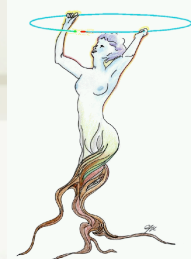
- ◆ **RooFit** package added in ROOT version 5
 - ◆ developed in *BaBar* (V. Werkerke and D. Kirkby)
 - ◆ fitting framework for sophisticated fitting
 - ◆ support for various fitting methods
 - ◆ based on *TMinuit*
 - ◆ OO description of p.d.f.
 - ◆ addition, convolution, automatic normalization
 - ◆ provides extra functionality (toy MC, advanced plotting)
- ◆ Not yet fully integrated in ROOT (built from a tar file)
- ◆ Large and complex package
 - ◆ question of long term maintenance

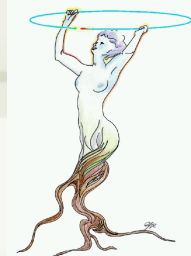


New Fitting classes

- ★ Have a simplified version of *RooFit* to replace current *TVirtualFitter*
 - ★ *RooFit* will still be needed for complex fits
- ★ Core fitting classes independent of other ROOT library
 - ★ dependence on *libHist* (TH1 and TF1) will be only at an outer level
- ★ Use function interfaces (i.e. *IParamFunction*) defined in *MathCore*
- ★ Define interface for minimization
 - ★ choose minimizer at run-time (using plug-in manager)

New Fitter Design





New Fit Panel (*Fit GUI*)

- ✦ Develop a new Fitting GUI for ROOT data analysis objects (*TH1*, *TGraph* and *TTree*)
 - ✦ to improve quality and functionality of old one
 - ✦ easier for user to drive and control the fits
- ✦ Provides functionality for:
 - ✦ function combinations
 - ✦ parameter definition (setting values, fixing, etc...)
 - ✦ select fitting methods (Chi2, likelihood, linear fit)
 - ✦ choose fitting options
 - ✦ pick out minimizer library (Minuit, Minuit2, Fumili)
 - ✦ advanced drawing options (residuals, CL, contours)

New Fitting GUI



★ prototype being developed

★ panel to control function parameters already released

New Fit Panel
Object: LikelihoodMinos:TH1D

General | Minimization

Function
Predefined: gaus
Operation: NOP ADD CONV

Selected Function: gaus
Set Parameters...

Options
Method: Binned Likelihood
 Linear fit
Robust: 0.0 No Chi-square

Fit Options
 Integral Improve fit results
 Improve errors All weights = 1
 Use range Add to list

Draw Options
 SAME
 Do not draw

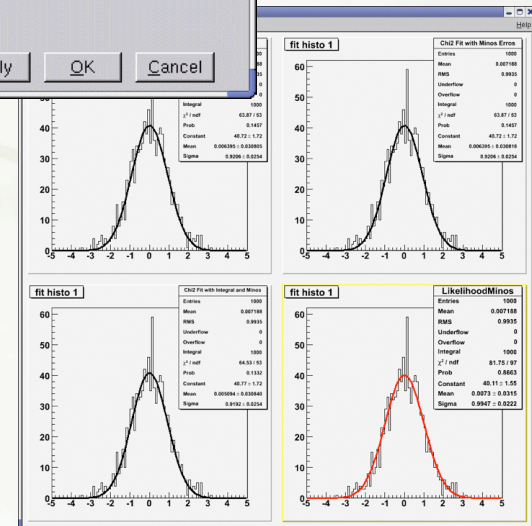
Print Options
 Default Verbose Quiet

Fit Reset Close

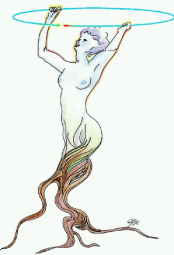
Name	Fix	Value	Min	Set Range	Max
Constant	<input type="checkbox"/>	40.1058	35.4464		44.7652
Mean	<input type="checkbox"/>	0.00730011	-0.0870921		0.101692
Sigma	<input type="checkbox"/>	0.994726	0.927995		9.94719

Immediate preview

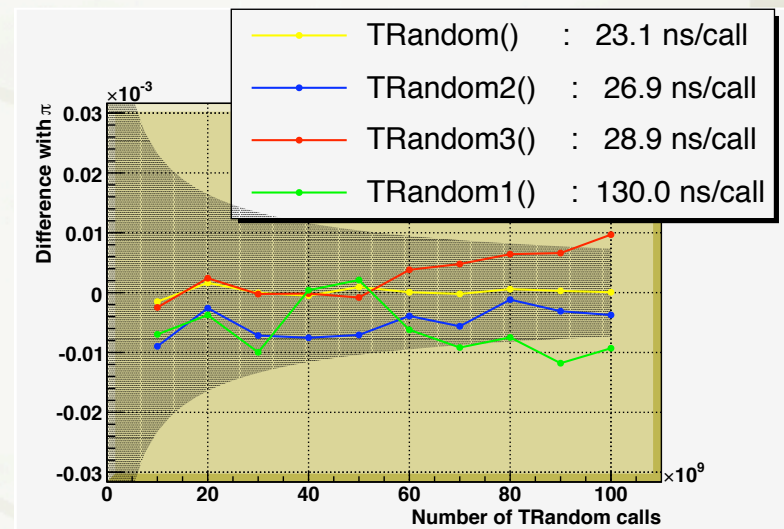
Reset Apply OK Cancel



Random Numbers Improvements



- ★ default is now Mersenne-Twister generator (*TRandom3*)
 - ★ fast and excellent pseudo-random quality
- ★ replace obsolete *TRandom2* with TausWorth generator
- ★ add RanLux generator (*TRandom1*)
- ★ use a better linear congruential for *TRandom*
 - ★ old one had seeding problems and a not uniform coverage
 - ★ need to maintain for backward compatibility a generator based on a state of only 32 bits
 - ★ very short period ($2^{31} \sim 10^9$)
 - ★ strongly discourage its use in any statistics application

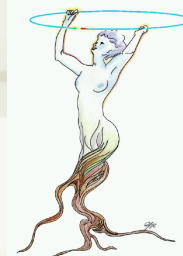


Random Numbers

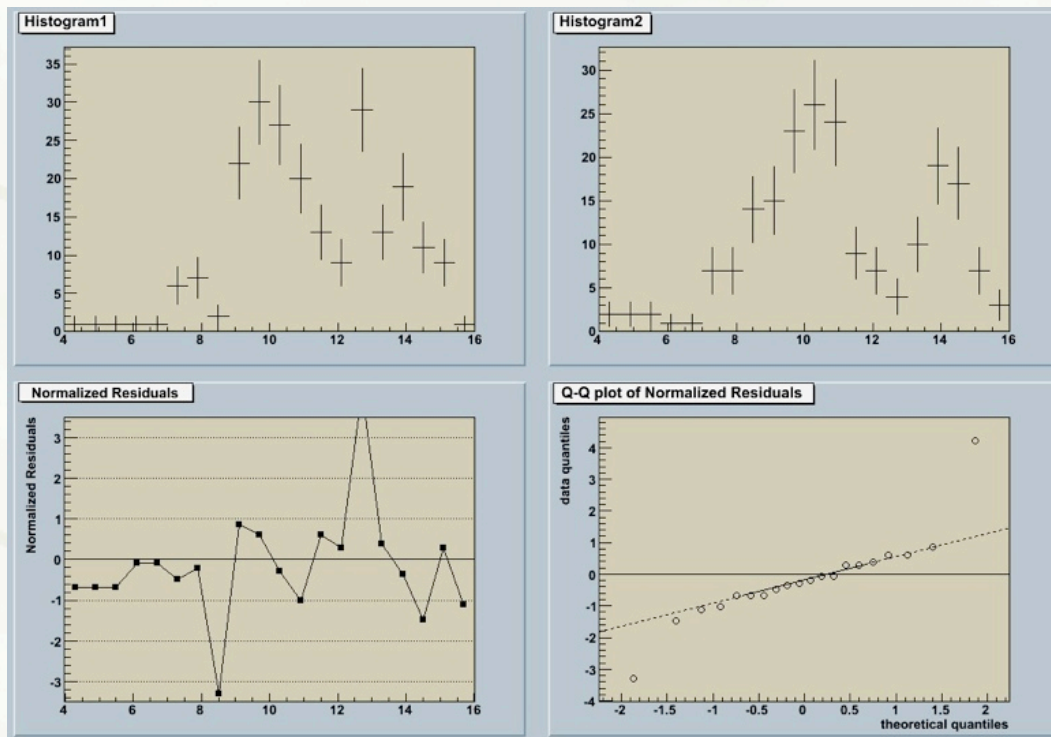


- ◆ Added in *MathMore* interface for GSL random
- ◆ Improved generation of random distributions:
 - ◆ Poisson for large N
 - ◆ performance improvements for others (Landau)
- ◆ Fermilab computing group has developed a new random package following C++ standard proposal
 - ◆ foreseen as a CLHEP replacement
 - ◆ no plan for the moment to include it in ROOT
 - ◆ will be proposed to be included in Boost (uses some Boost classes)
- ◆ Need to review with experiments and Geant4 their future needs for random numbers (after CLHEP)

Histogram Comparison



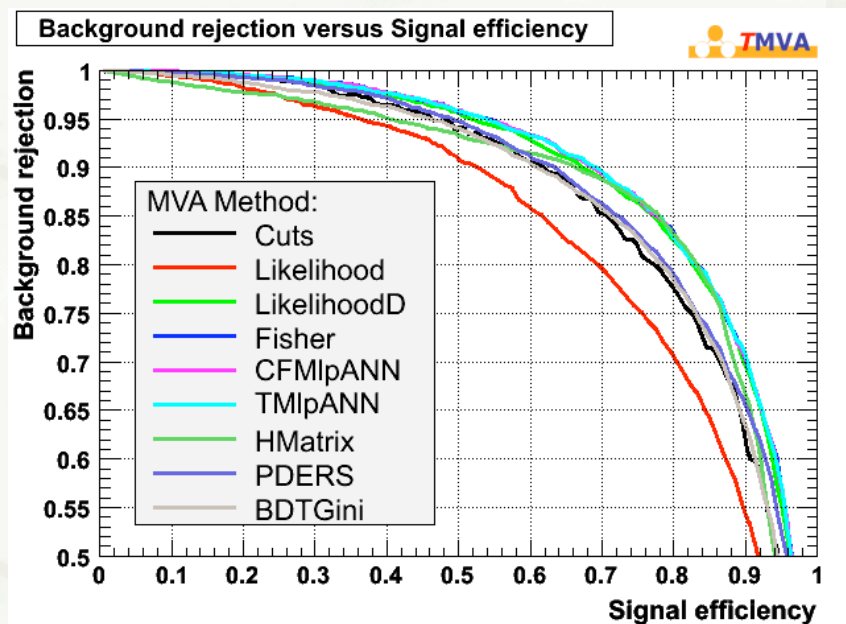
- ◆ Improvements in Chi2 test for comparing histograms
 - ◆ algorithm from *N. Gagunashvili* and implemented in C++ by *D. Haertl*
 - ◆ add possibility to use weighted histograms
 - ◆ comparison of histograms with different scales
 - ◆ produce normalized residuals

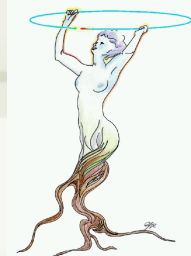


TMVA



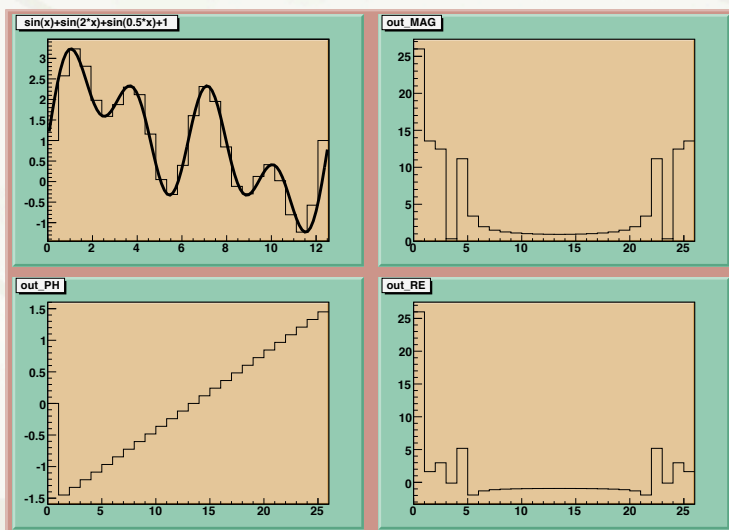
- ★ New package for multivariate analysis distributed in ROOT
 - ★ from *A. Hocker, J. Stelzer, H. Voss, K. Voss, X. Prudent*
- ★ Provides various methods for signal/background discrimination:
 - ★ Rectangular cut optimization
 - ★ Correlated likelihood estimator
 - ★ Multi-dimensional likelihood estimator
 - ★ Fischer discriminant
 - ★ H-matrix estimator
 - ★ Artificial Neural network (2 implementations)
 - ★ Boosted decision trees





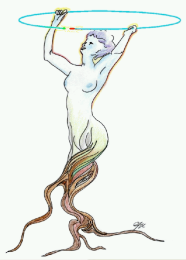
FFT

- ◆ Included in ROOT a common base class (*TVirtualFFT*)
 - ◆ add a functions to use it from TH1 (*TH1::FFT*)
- ◆ Implemented an interface to the popular *FFTW* package (see www.fftw.org)
 - ◆ support for one and multi-dimensional transforms
 - ◆ support for complex and real transformations



- ◆ *TFFTComplex* for complex input/complex output transforms
- ◆ *TFFTRealComplex* for real input/complex output
- ◆ *TFFTComplexReal* for complex input/real output
- ◆ *TFFTReal* for real input/output

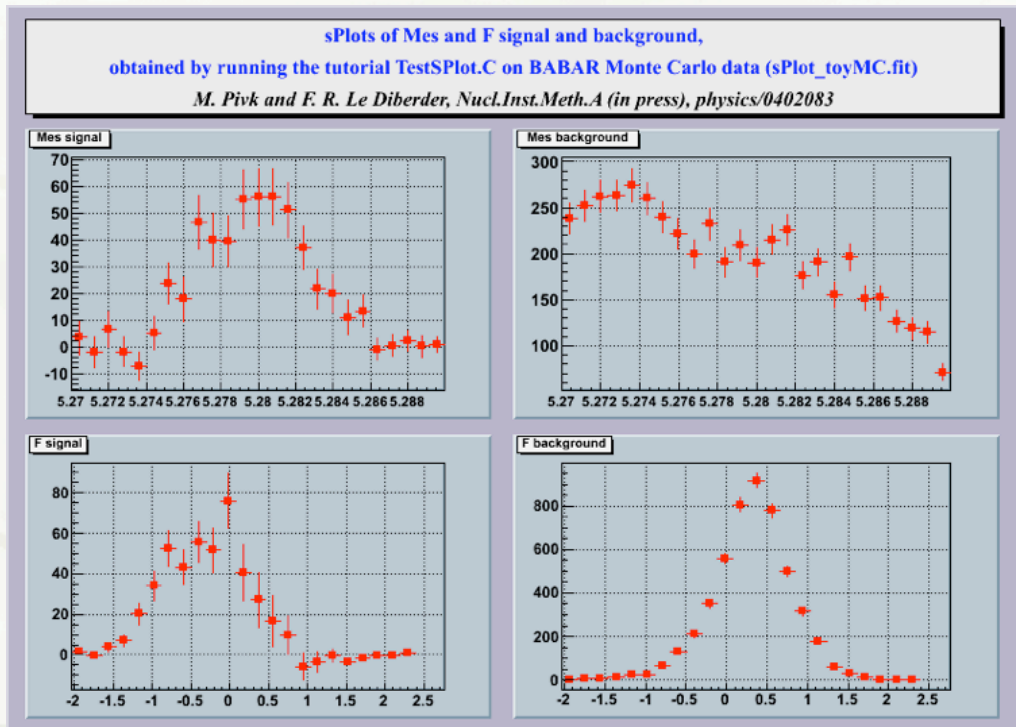
SPlot

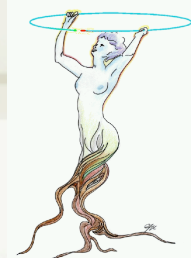


- ◆ new tool used to access the validity of maximum likelihood fits for discriminating signal from background

- ◆ from *M. Pivk and F.R. Le Diberder*

- ◆ *SPlot* gives unbiased distributions of the control variables
 - ◆ independently for all the various sources of events
 - ◆ no use of the control variables knowledge





Future Plans

★ *MathCore* and *MathMore*

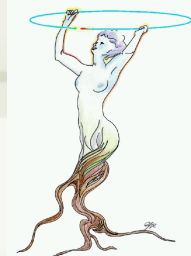
- ★ integration with ROOT analysis objects, like Histogram and Function classes
- ★ complete in *MathMore* the GSL wrapper
 - ★ quasi-random numbers, multi-dimensional integration
- ★ work on requests and feedback from the experiments
 - ★ CMS and LHCb started using *MathCore* and *SMatrix*
 - ★ ATLAS started using *SMatrix* for the track fit

★ Complete new ROOT fitting classes and Fit GUI

- ★ easier to use various fitting and minimization methods

★ Integrate *UNURAN*

- ★ package for generating non-uniform random numbers
- ★ from Wien statistics group (*J. Leydold*)



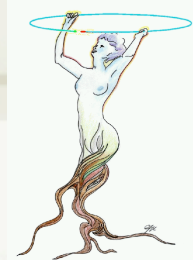
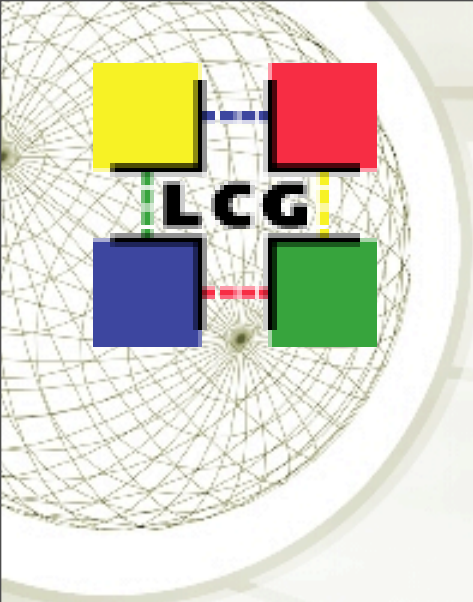
Documentation

- ★ Online doc based on Doxygen (and *THtml*) for the new classes of *MathCore*, *MathMore*, *SMatrix* and *Minuit2*
 - ★ provided for every new ROOT release
 - ★ example for latest 5.13.02 :
 - ★ http://seal.web.cern.ch/seal/MathLibs/5_13_02/SMatrix/html/index.html
- ★ Written a new Math chapter in the ROOT 5.12 User Guide (*chapter 13, 225-247*)
 - ★ describe random numbers, *MathCore* (*GenVector*), mathematical functions, *SMatrix*
 - ★ see <ftp://root.cern.ch/root/doc/chapter13.pdf>
- ★ Separate docs exist for other packages (*Minuit2*, *RooFit*)



References

- ★ *MathCore* online doc: <http://seal.web.cern.ch/seal/MathLibs/MathCore/html/index.html>
- ★ *MathMore* online doc: <http://seal.web.cern.ch/seal/MathLibs/MathMore/html/index.html>
- ★ *SMatrix* online doc: <http://seal.web.cern.ch/seal/MathLibs/SMatrix/html/index.html>
- ★ *Minuit2* online doc: <http://seal.web.cern.ch/seal/MathLibs/Minuit2/html/index.html>
- ★ *RooFit* homepage: <http://roofit.sourceforge.net/>
- ★ *TMVA* homepage: <http://tmva.sourceforge.net/>
- ★ *FFTW* homepage: <http://www.fftw.org/>
- ★ *Histogram comparison* paper: <http://arxiv.org/abs/physics/0605123>
- ★ *SPlot* paper: <http://arxiv.org/abs/physics/0402083>
- ★ *UNURAN* homepage: <http://statmath.wu-wien.ac.at/unuran/>
- ★ C++ Random number proposal:
 - ★ <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n2079.pdf>

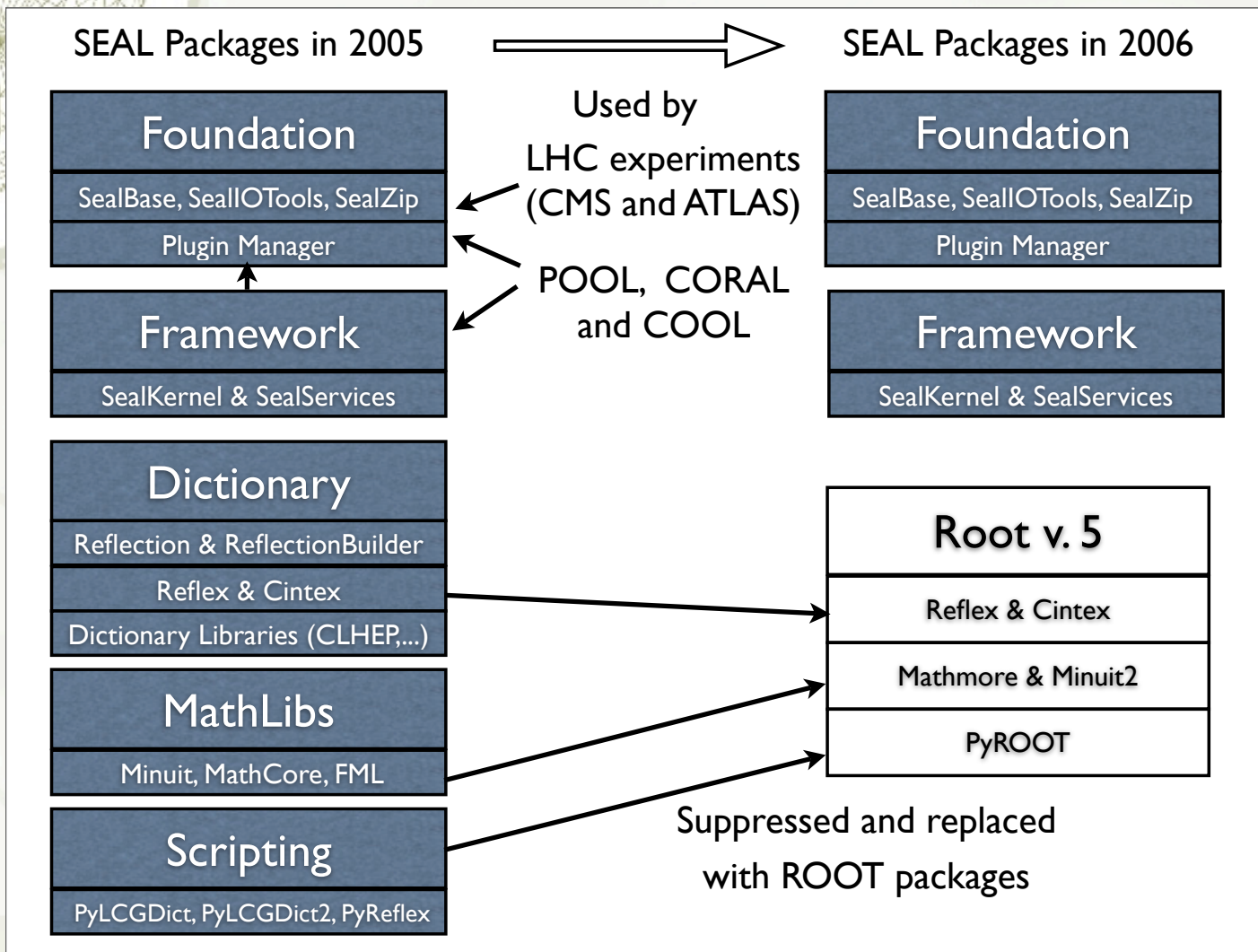
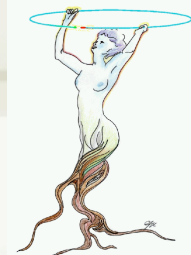


SEAL

Status and Plans

- ✦ Work package of the ROOT project with responsibility
 - ✦ facilitate migration of packages in ROOT
 - ✦ maintenance
 - ✦ some software not migrated so far in ROOT
 - ✦ requested by the LHC experiments and other LCG projects

Seal Migration Status



Current Status of SEAL



- ✦ SEAL release contains only these subsystems:

- ✦ *Foundation* :

- ✦ *SealBase*, *SealIO*, *SealZip*, *SealUtil*

- ✦ set of utility classes
- ✦ used by POOL, CORAL, COOL
- ✦ used by CMS and ATLAS

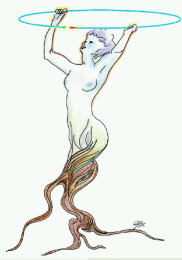
- ✦ *PluginManager* (based on *SealBase*)

- ✦ used directly by POOL in the Storage Service
- ✦ used by CMS in the new framework

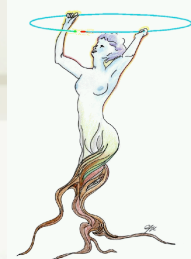
- ✦ *Framework* (component model)

- ✦ based on *PluginManager*
- ✦ dependent on Boost (uses Boost ref counted pointers)
- ✦ used by CORAL (and COOL) and CMS

Current SEAL Status (2)



- ★ SEAL contains also *MathLib* (with *FML* and *PyFML*)
 - ★ it will be suppressed when new fitting classes will be also available in ROOT
- ★ No direct dependency anymore from ROOT
 - ★ only at the level of testing
- ★ SEAL is in maintenance mode
 - ★ no new developments
 - ★ latest SEAL release built in April
 - ★ new release foreseen with few bug fixes and MAC OSX support
- ★ Problem discovered by COOL in the component model
 - ★ no support for multi-thread operations



Outlook for SEAL

- ✦ Investigate if worth keeping *Foundation* and *Framework* in a separate SEAL project
 - ✦ main client is POOL/CORAL/COOL
 - ✦ to satisfy the requirements (multi-thread support) some extra effort is needed
- ✦ plug-in manager could be moved/merged in ROOT
 - ✦ on-going studies on a plug-in manager based on Reflex
 - ✦ much simpler than current one in SEAL
 - ✦ could merge with current ROOT plug-in manager
 - ✦ experiments and POOL could then use this one