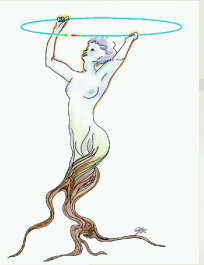
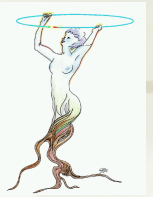


New Developments of ROOT Mathematical Libraries

W. Brown[1)], M. Fischler[1)], A. Kreshuk[2)], J. Marraffino[1)], L. Moneta[2)], A. Zsenei[2)]
[1)] Fermi National Accelerator Laboratory, Batavia, Illinois, USA
[2)] CERN – European Organization for Nuclear Research, Geneva, Switzerland

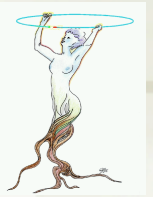
Computing in High Energy and Nuclear Physics

13-17 February 2006, T.I.F.R. Mumbai, India



Root Math Work Package

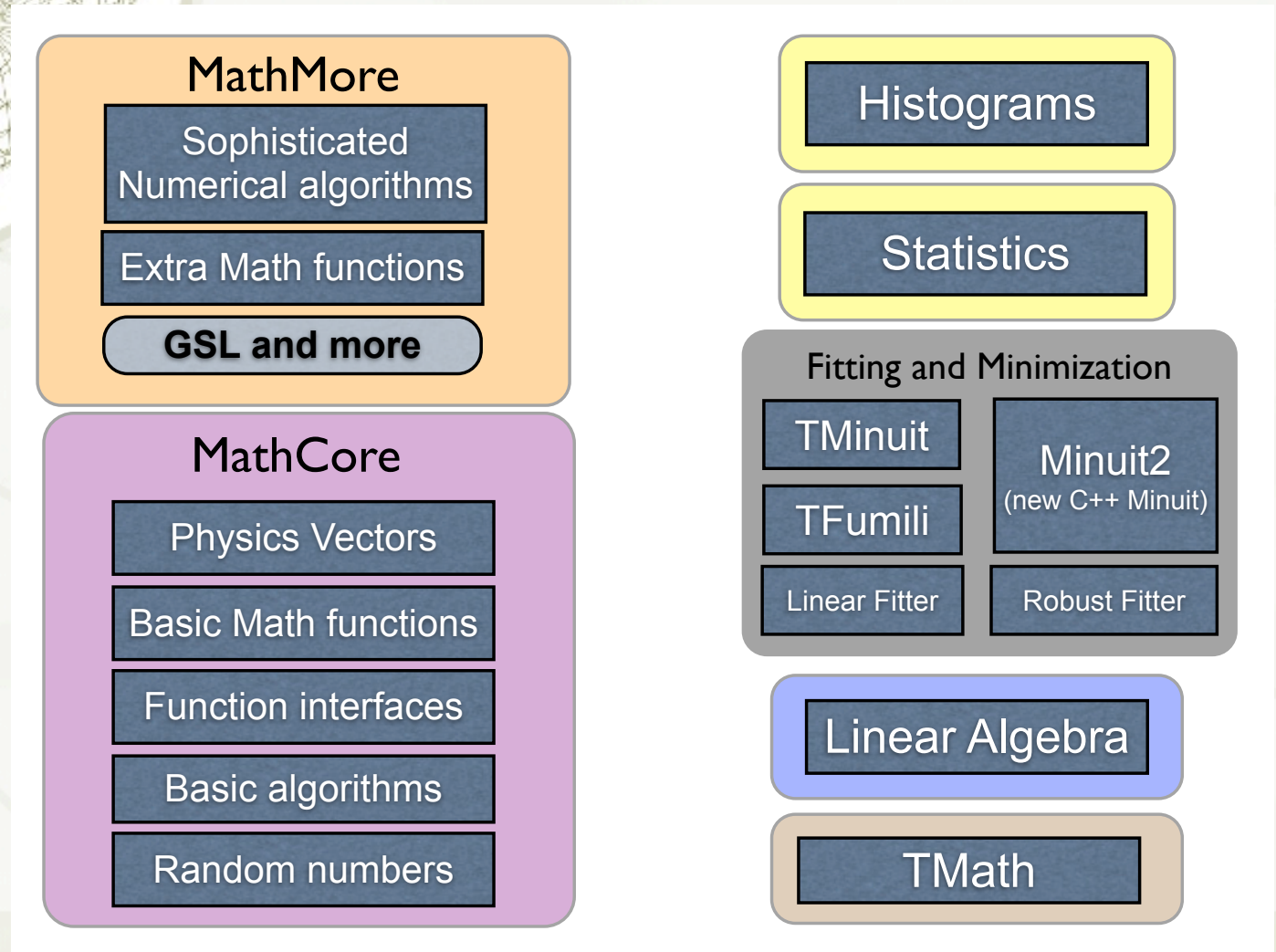
- Main responsibilities for this work package:
  - Basic Mathematical functions
  - Fitting and Minimization
  - Random Numbers
  - Linear Algebra
  - Physics and geometry Vectors (3D and 4D)
  - Histograms
  - Statistics (confidence levels, multivariate analysis, etc..)
- Will describe only the recent developments not all Math functionality present in ROOT

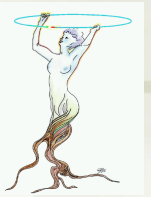


Outline

- 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*)
- Other recent developments (statistics)
  - Linear and Robust Fitter
  - *SPlot* : tool for signal/background discrimination

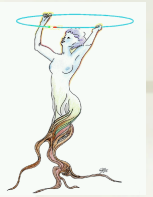
ROOT Math Libraries





Physics and Geometry Vectors

- Classes for 3D Vectors and LorentzVectors with their operations and transformations
  - Merge old ROOT Physics classes and CLHEP Vector and Geometry classes
- Requested by LHC experiments to be used in reconstruction, analysis and in the event data model
- Work done in collaboration with C++ experts from Fermilab computing group
  - M. Fischler, W. Brown and J. Maraffino
- New classes designed with cleaner and minimal interfaces
  - Try to avoid duplications and aim for stability

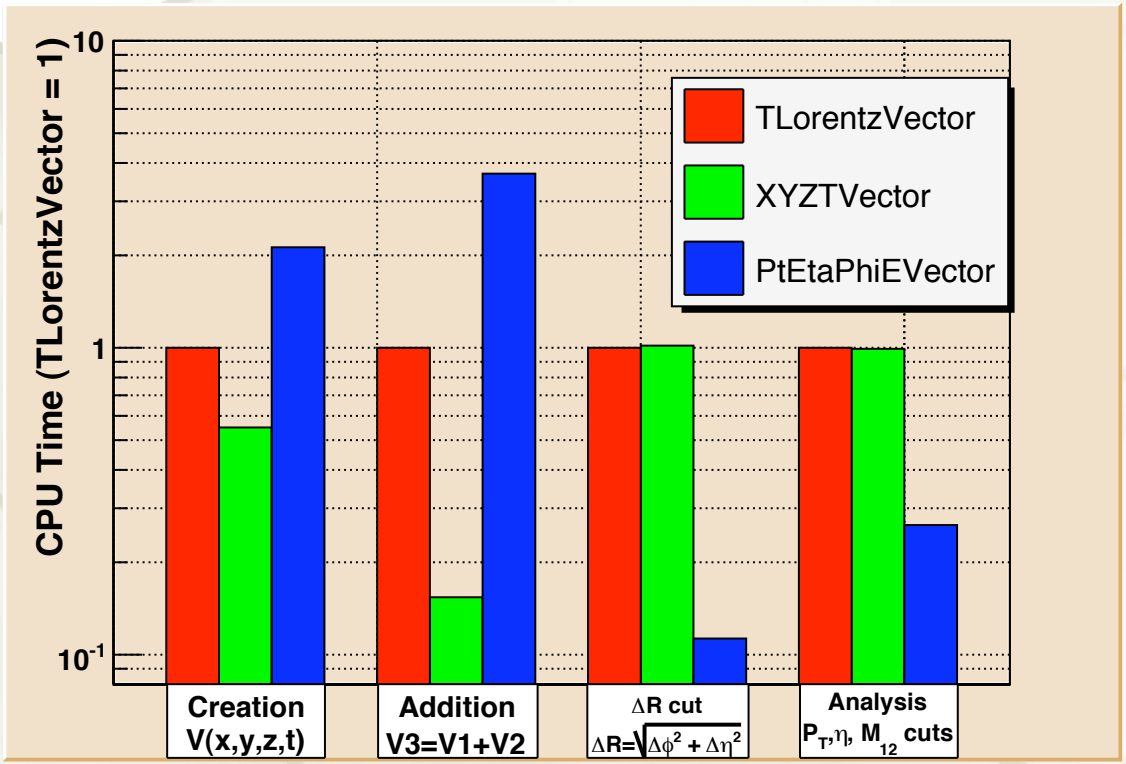


Generic Coordinate Systems

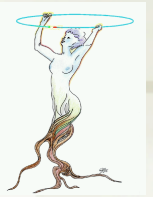
- Vector based on a generic coordinate system type:
  - `LorentzVector<PxPyPzE4D<double> >`
  - `LorentzVector<PtEtaPhiE4D<double> >`
- Coordinate System classes are template on the contained scalar type (e.g. `double` or `float`)
- Available coordinate system classes:
  - `Cartesian3D,Polar3D,Cylindrical3D,CylindricalEta3D`
  - `PxPyPzE4D,PxPyPzM4D,PtEtaPhiE4D,PtEtaPhiM4D`
- Use of typedefs to hide template complexity
  - `XYZTVector` for `LorentzVector<PxPyPzE4D<double> >`
- User can choose representation optimal for his use case

Generic Vector Performances

- Optimal run-time performance

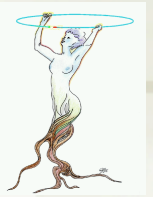


- user can choose best coordinate system
- no virtual calls and use of inline methods



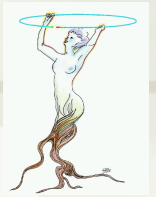
3D Points and Vectors

- Points and Vector distinction in 3D
  - Different transformations and operations
    - *PositionVector3D* class
      - Rotation and translation
      - Cannot be added and difference result is a DisplacementVector
    - *DisplacementVector3D* class
      - Only rotation (no translation)
      - Have addition and subtraction
  - Describe them as different types
    - dis-allowed transformations will not compile



Transformation

- 3D Rotations
  - 3x3 orthogonal matrix representation: `Rotation3D`
  - 3 Euler angles : `EulerAngles`
  - Direction Axis (Vector) + angle : `AxisAngle`
  - Quaternion (4 numbers) : `Quaternion`
  - Specialized rotations around X,Y,Z axis : `RotationX,Y,Z`
- 3D Transformation
  - Rotation + Translation: `Transform3D`
- LorentzRotation
  - Generic 4x4 matrix: `LorentzRotation`
  - Boost classes: `Boost, BoostX,Y,Z`



SMatrix Package

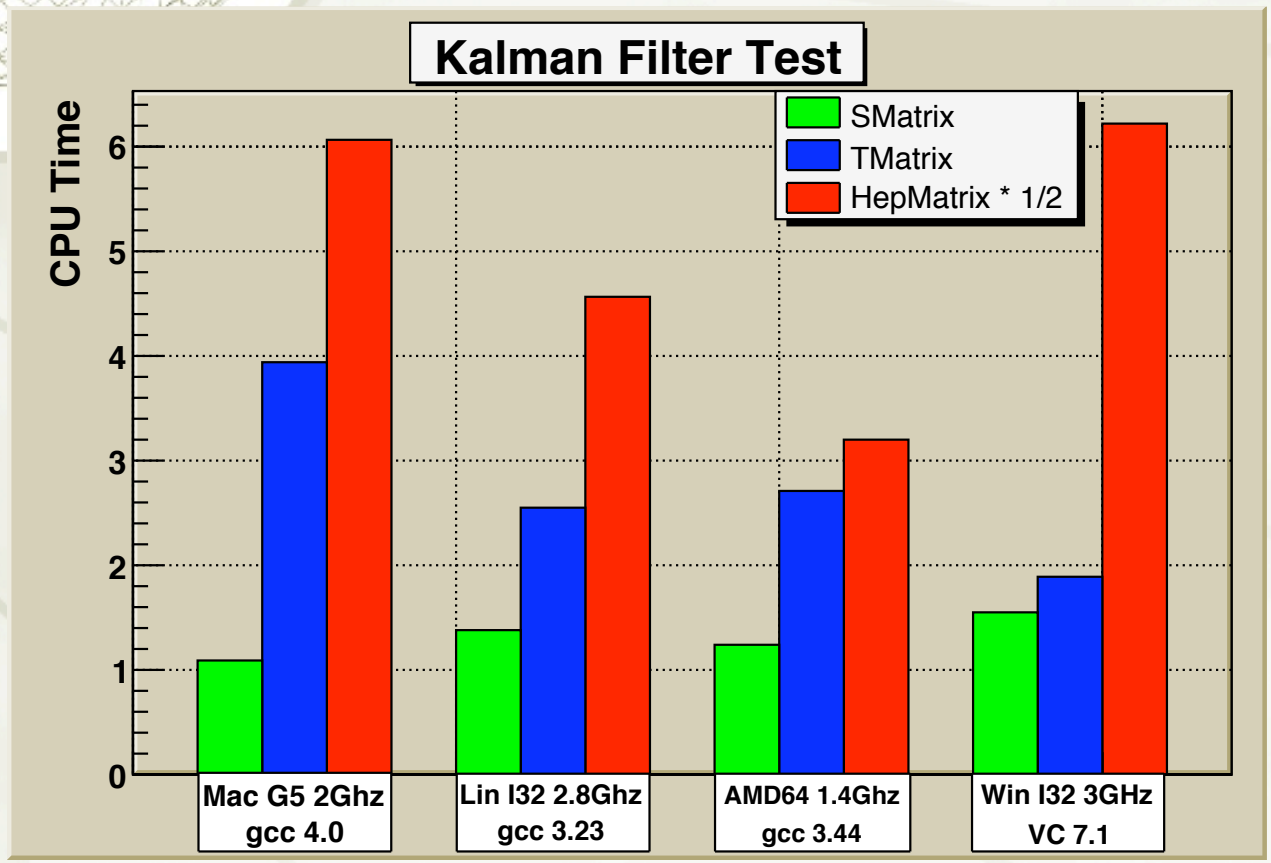
- package initially developed by T. Glebe for HeraB
- matrix and vector classes of arbitrary type
- for **fixed** (not dynamic) matrix and vector sizes :

```
SMatrix< double, 2 , 5>
SVector< double, 5 >
```

  - dictionary can be generated only for pre-defined types
- complementary and **NOT** a replacement of *TMatrix*
- optimized for small matrix sizes:
  - use expression templates to avoid temporaries
- support for vector-matrix arithmetic operations and matrix inversion
  - not full linear algebra functionality

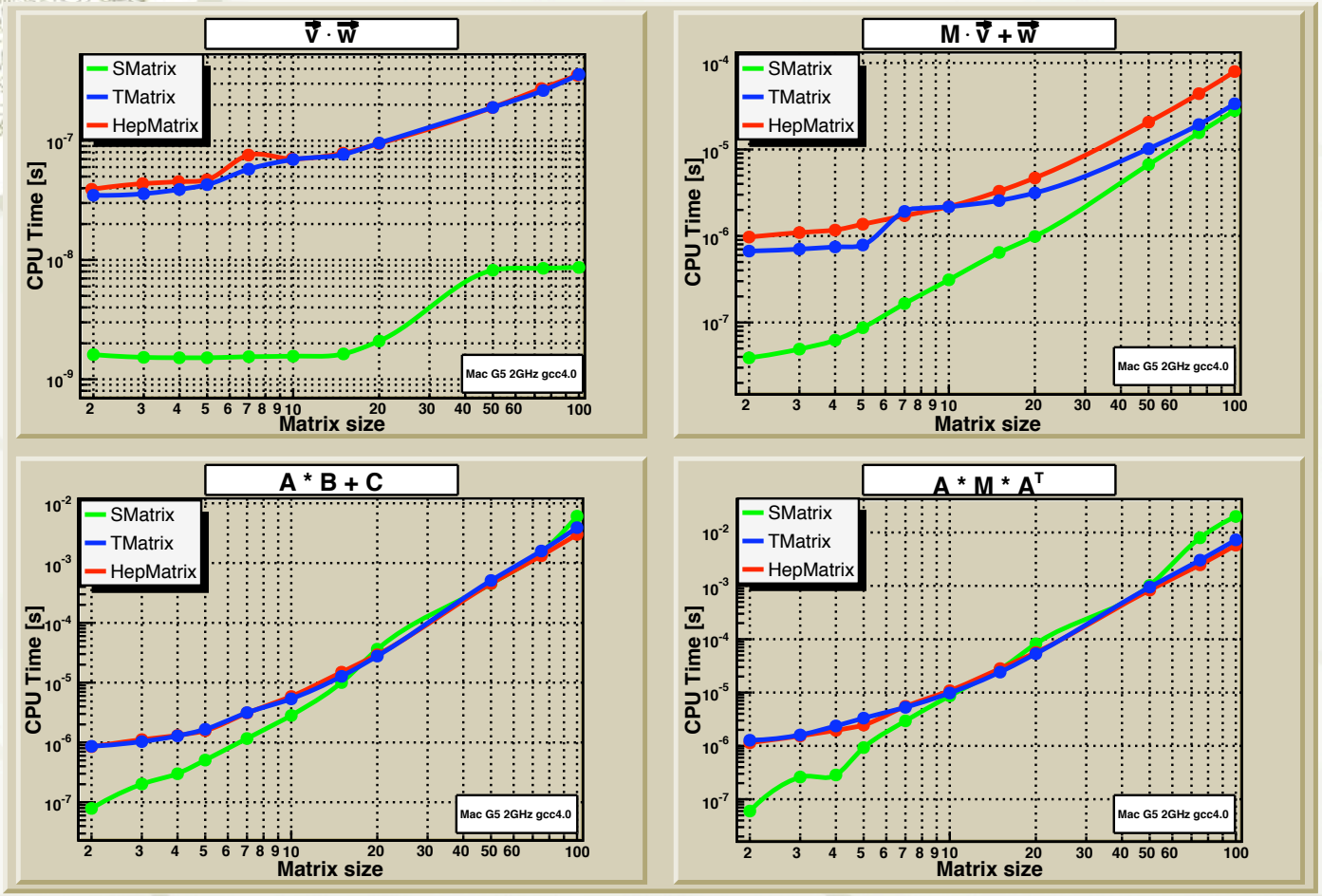
Matrix Performances

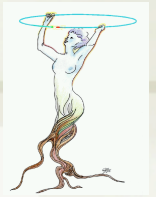
Comparison ROOT (*TMatrix*/*SMatrix*) and CLHEP (*HepMatrix*)



- Kalman Filter update equations
- Matrix sizes:
  - 2x2
  - 2x5, 5x2
  - 5x5
- Operations:
  - addition
  - multiplication
  - inversion

Matrix Operation Performances





Mathematical Functions

- Special Functions:
  - use interface proposed to C++ standard:

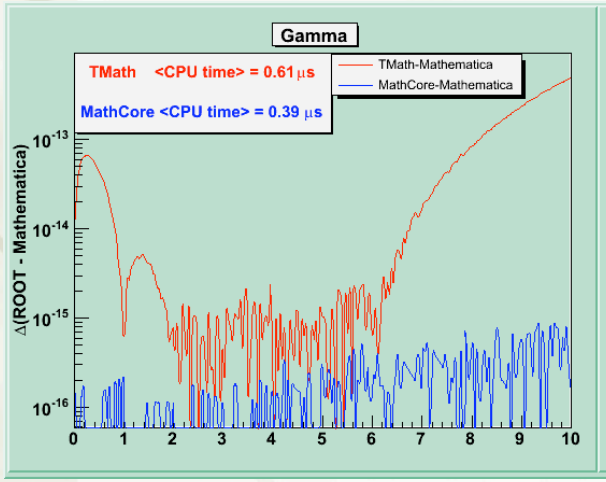
```
double cyl_bessel_i (double nu, double x);
```

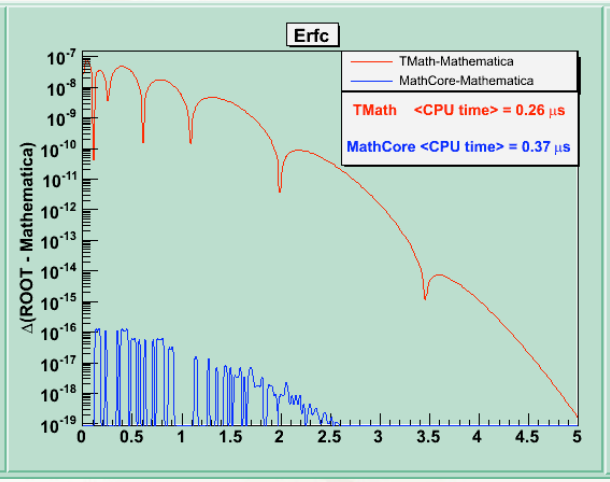
- Statistical Functions:
  - Probability density functions (pdf)
  - Cumulative dist. (lower tail and upper tail)
  - Inverse of cumulative distributions
  - Coherent naming scheme. Example chi2:

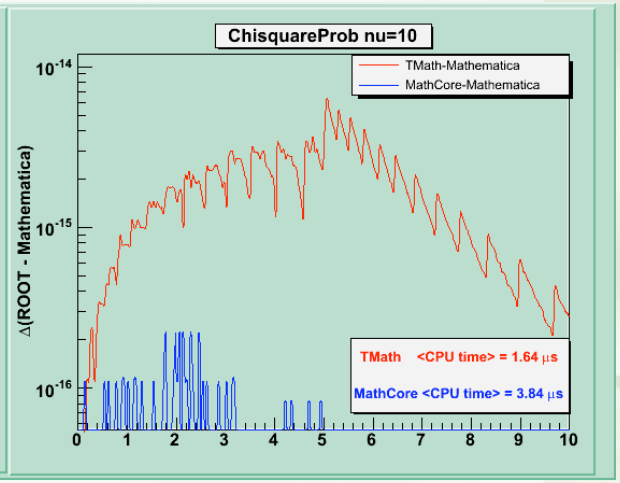
```
chisquared_pdf
chisquared_prob, chisquared_quant,
chisquared_prob_inv, chisquare_quant_inv
```

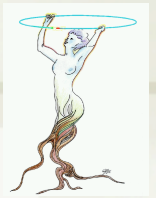
Mathematical Functions Tests

- Compare difference with Mathematica and NagC
  - improved precision with *MathCore*/*MathMore* functions
  - some penalty in time observed
    - especially for some GSL based functions (worse x 2)







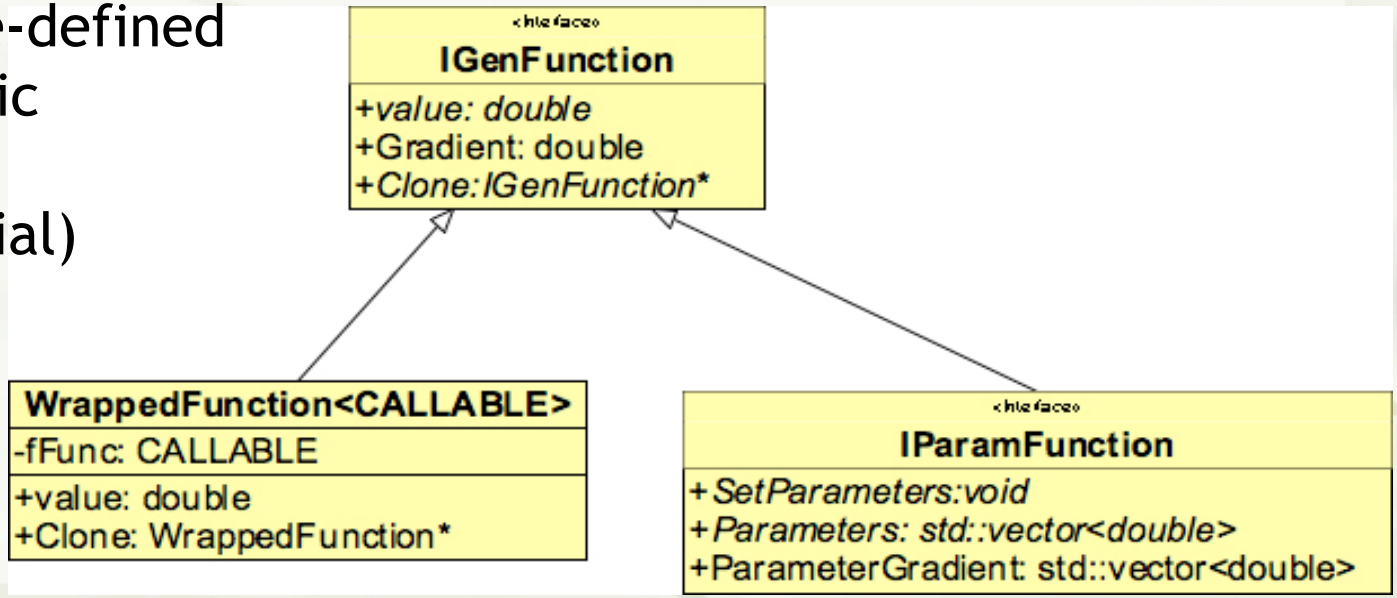


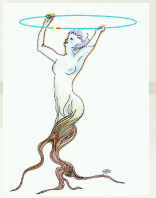
Numerical Algorithms

- C++ interface to GSL numerical algorithms
- Currently present are 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 algorithm:
- 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)



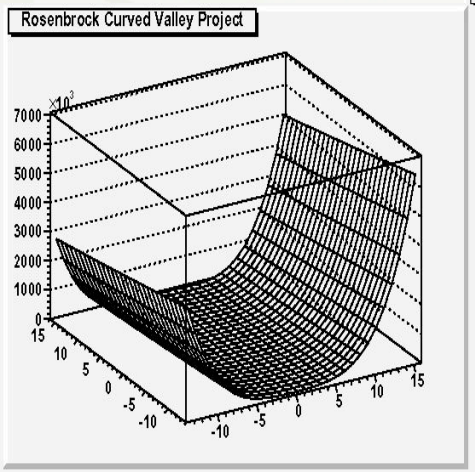


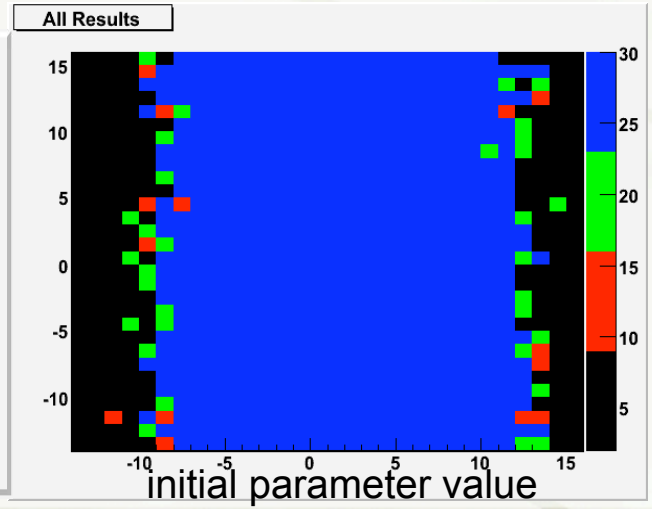
Fitting and Minimization

- *RooFit* package: framework for sophisticate fitting
- New C++ version of Minuit (*Minuit2*) in ROOT v5.08
  - implemented a ROOT fitter interface
- Same basic functionality as in old version
  - Migrad, Simplex, Minos algorithms
- Extended functionality:
  - Single side parameter limits
  - Added Fumili minimization method for Chi2 and likelihood minimization
- OO package for generic function minimization
  - Easy to extend by inserting new minimization methods

Minuit2 Tests

- Going under extensive performance and convergence tests
  - Same function calls needed to find minimum as in old version
    - Basically the same performance
- Example: test convergence with RosenBrock function





Minuit2 and TMinuit converge

only Minuit2 converges

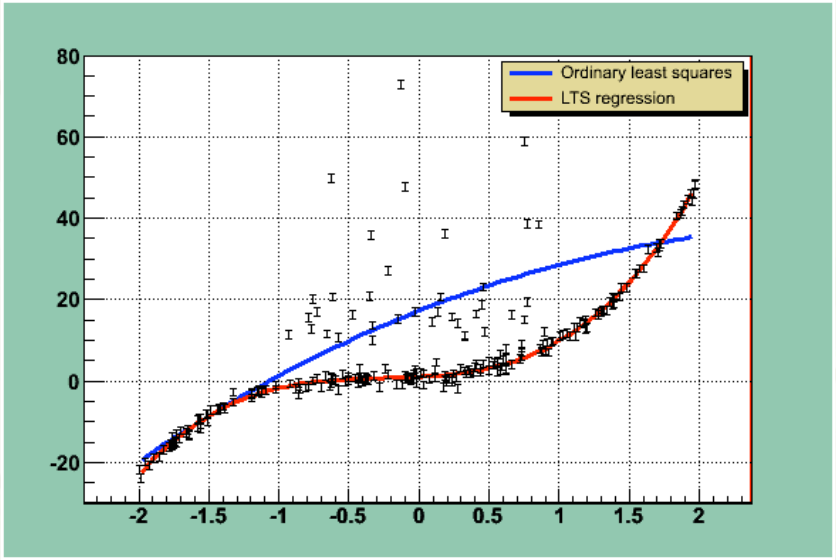
only TMinuit converges

Minuit2 and TMinuit fail

Linear and Robust Fitter

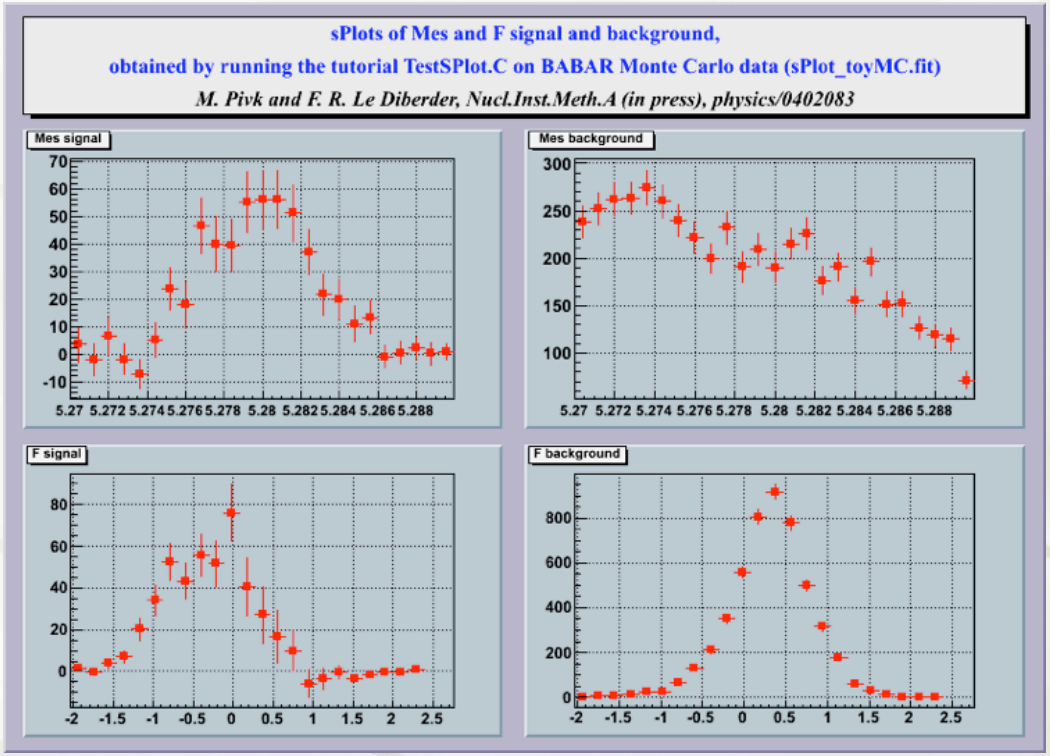
- `TLinearFitter` class to fit function linear in the parameters (e.g Polynomial)
  - direct solution by solving a linear system
  - can be 10-15 times faster than Minuit
- Robust Fitting
  - outliers removal
  - use of Least Trimmed Square (LTS) regression

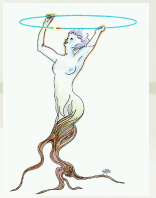
```
Graph.Fit("pol3", "rob=0.75", -2, 2);
```



SPlot

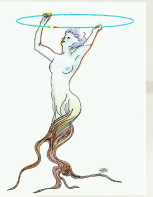
- new tool used to access the validity of maximum likelihood fits for discriminating signal from background
- ***SPlot*** gives unbias distributions of the control variables
  - independently for all the various sources of events
  - no use of the control variables knowledge





Future Plans

- *MathCore* and *MathMore*
  - integration with ROOT analysis objects, like Histogram and Function classes
  - investigate new random number package based on C++ standard (developed by Fermilab team)
- Improve and redesign ROOT fitter interface
  - give the user an easy possibility to use the various fitting and minimization methods
  - easy way in constructing complex fitting functions
- Linear Algebra
  - use template in the *TMatrix* classes (already in CVS)
  - have a symmetric n x n *SMatrix* class



Future Plans and Conclusions

- Further additions:
  - classes for FFT using FFTW and GSL
  - new statistical algorithms :
    - cluster algorithms from R package
    - multidimensional smoothing algorithms
    - boosted trees techniques for signal discrimination
  - develop a framework for estimating confidence intervals in presence of nuisance parameters
    - typical cases encountered at LHC
- Work on requests and feedback from the experiments
  - LHCb started adopting *MathCore* and *SMatrix*
  - CMS is planning to use them in their new software