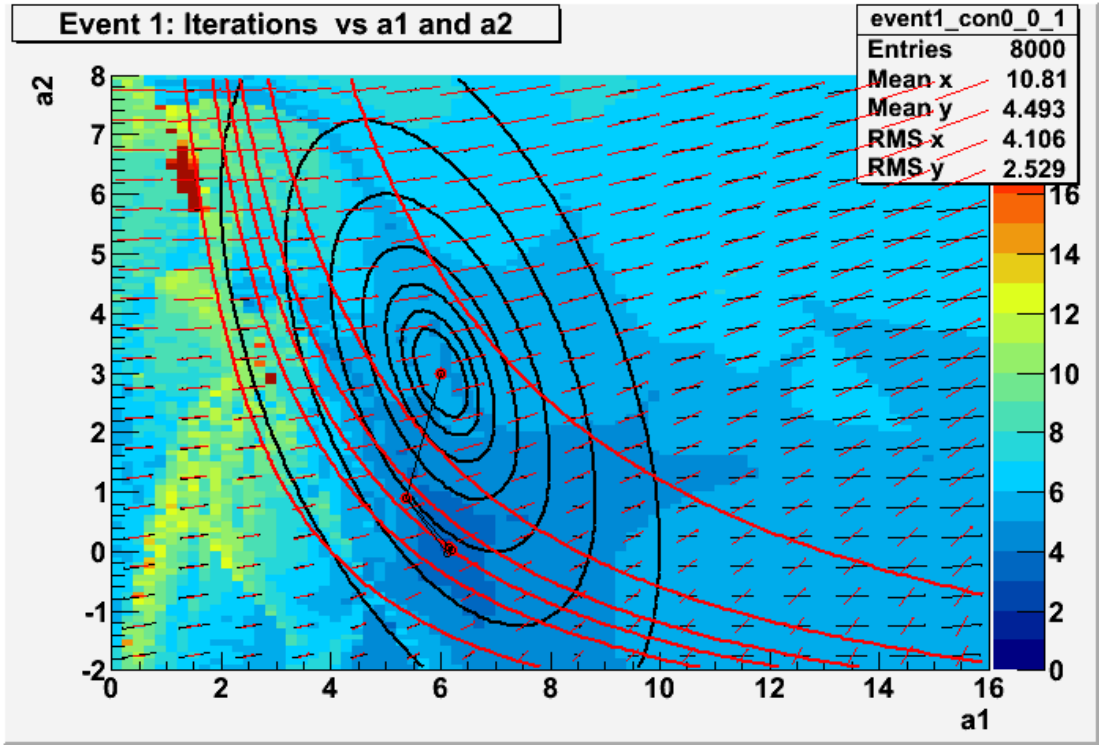
MarlinKinFit Status.

Status of the kinematic fit package in Marlin



Benno List, DESY -IPP
<Benno.List@desy.de>
Status of MarlinKinFit
ILD Software meeting
4.5.2010

HELMHOLTZ ASSOCIATION

DESY

Overview

- Introduction
- New debugging tools:
  - TextTracer and RootTracer
  - ParameterScanner
- The new fit engine: NewFitterGSL
- Performance of NewFitterGSL
- Soft Constraints with NewFitterGSL

Introduction

- MarlinKinFit provides classes to solve constrained kinematic fit problems
- FitObject: objects to be fitted, e.g. jets, neutrinos, leptons
  → FitObject incorporates information about parametrization,
  calculates its own 4-vector,
  stores error information
- Constraint: objects representing constraints
  → may constrain sum of px/py/pz/E of objects,
  or invariant mass (difference of masses) of objects
- Fitter: fit engine to set up and solve kinematic fit problem
  Method used: Lagrange multipliers

Example how to use the fitter

```
JetFitObject      taujet (43.2, 0.34,  1.33, 2.1, 0.1, 0.1);
JetFitObject      taumu  (27.3, 2.7,  -1.78, 0.2, 0.03, 0.03);
NeutrinoFitObject taunu  (18.3, 2.7,  -1.78, 10,  0.2, 0.2);

MomentumConstraint pxc (0, 1, 0, 0);       // sum(1*px_i) = 0
MomentumConstraint pyc (0, 0, 1, 0);       // sum(1*py_i) = 0
MomentumConstraint pzc (0, 0, 0, 1);       // sum(1*pz_i) = 0
MomentumConstraint ec  (1, 0, 0, 0, 91.2); // sum(1*E_i) = 91.2
MassConstraint mtauc (1.7);           // mass (taumu + taunu) = 1.7

pxc.addToFOList (taujet);   // px-constraint sums over 3 particles
pxc.addToFOList (taumu);
pxc.addToFOList (taunu);
// similar for pyc, pzc, ez, and mtauc...

NewFitterGSL fitter;

fitter.addFitObject (taujet);
fitter.addFitObject (taumu);
fitter.addFitObject (taunu);

fitter.addConstraint (pxc);
fitter.addConstraint (pyc);
fitter.addConstraint (pzc);
fitter.addConstraint (ec);
fitter.addConstraint (mtauc);

TextTracer tracer (std::cout);
fitter.setTracer (tracer);
fitter.setDebug (0);

double prob = fitter.fit();
```

Create Fit objects with parameters and their errors

Create Constraints

Define wich objects are summed for the constraints

Create the Fitter

Tell the fitter which fit objects and which constraints belong to the fit problem

Use a tracer to follow how the fit proceeds

Do the fit

DESY

New Debugging Tools: Tracers

- Tracer mechanism:

```
NewFitterGSL fitter;
TextTracer tracer (std::cout);
fitter.setTracer (tracer);
```

  Tracer object is called during the iteration process and can output/store information how the iterations proceed

- Two Tracer classes available:
  - TextTracer: Creates Text output
  - Root Tracer: Creates root tree
- Used mainly for development

Example output from TextTracer:

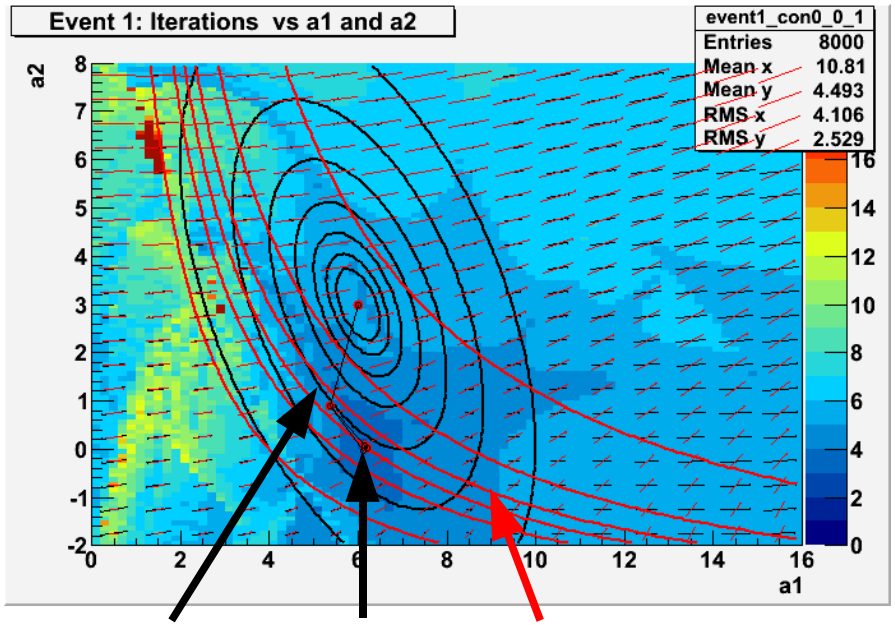
```
============== Starting fit ====================
Fit objects:
TestFitObject: (6 ± 1, 3 ± 1.5), chi2=0
Hard Constraints:
0 HyperbolaTestConstraint: 17+-8.18535
Value of alpha: 0
Value of detW: 0
Value of mu: 0
Value of phi: 0
Total chi2: 0 = 0 = 0(fo) + 0(sc)
Hard constraints: 17, scaled: 2.07688
Contribution to merit function: 0, scaled: 0
Merit function: 0, scaled: 0

        ....

============== Final result ====================
Fit objects:
TestFitObject: (6.14828 ± 1, 0.0661795 ± 1.5),
  chi2=4.74326
Hard Constraints:
0 HyperbolaTestConstraint: -1.24313e-06+-8.00542
Value of alpha: 1
Value of detW: -1.71152
Value of mu: 0.604563
Value of phi: 4.74326
Total chi2: 4.74326 = 4.74326 = 4.74326(fo) + 0(sc)
Hard constraints: 1.24313e-06, scaled: 6.89001
Contribution to merit function: 7.51549e-07,
  scaled: 4.16544
Merit function: 4.74326, scaled: 8.9087
============== Finished fit ====================
```

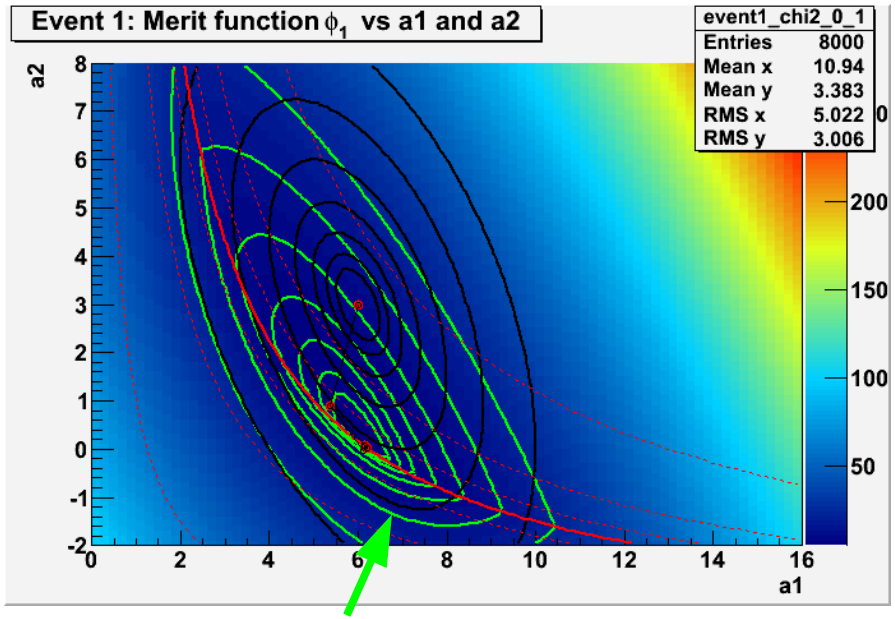
New Debugging Tools: Scanners

- Scan Parameters to create 2D-maps:

```
NewFitterGSL fitter;
ParameterScanner scanner (fitter);
scanner.doScan (0, 80, 0., 16., 1,    100,-2., 8., "event1_", "Event 1: ", mu);
IterationScanner itscanner (fitter);
itscanner.doScan (0, 80, 0., 16., 1,     100, -2., 8., "event1_", "Event 1: ");
```

Useful for visualisation





The New Fit Engine: NewFitterGSL

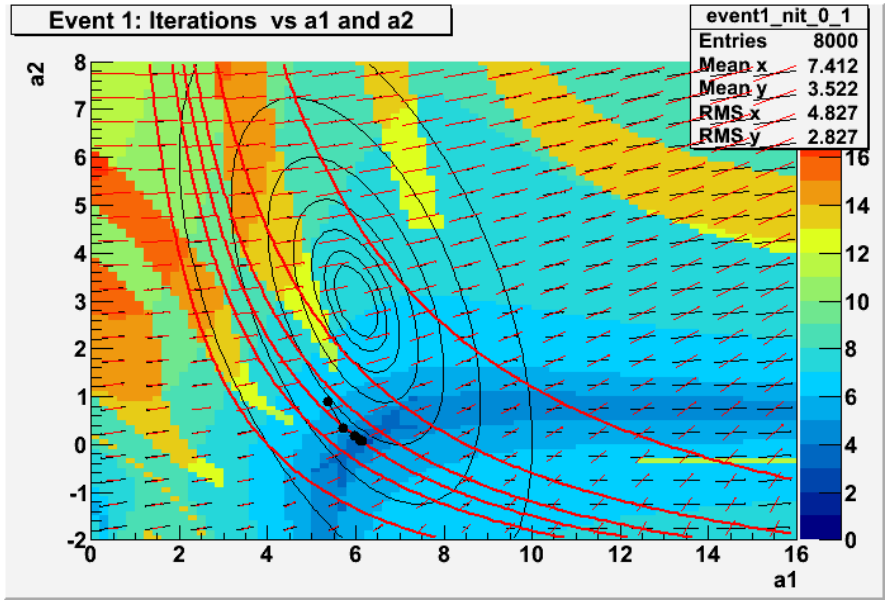
- Alternative fit engine to OPALFitterGSL, which remains the reference implementation
- Features:
  - Uses 2nd derivatives of constraints → better direction prediction
  - Better line search algorithm
  - Uses 2nd order corrections to avoid Maratos effect
  - **Can incorporate soft constraints**
- Soft Constraints: Instead of demanding exact fulfillment of a constraint, add a χ^2 term to achieve approximate fulfillment, with some tolerance
  - SoftGaussMomentumConstraint: implements χ^2 term
    $\chi^2 = (c_x \Sigma p_{x,i} + c_y \Sigma p_{y,i} + c_z \Sigma p_{z,i} + c_e \Sigma E_i)^2 / \sigma^2$
  - SoftGaussMassConstraint: implements χ^2 term
    $\chi^2 = ([\Sigma E_i]^2 - [\Sigma p_{x,i}]^2 - [\Sigma p_{y,i}]^2 - [\Sigma p_{z,i}]^2) / \sigma^2$

Improved Convergence

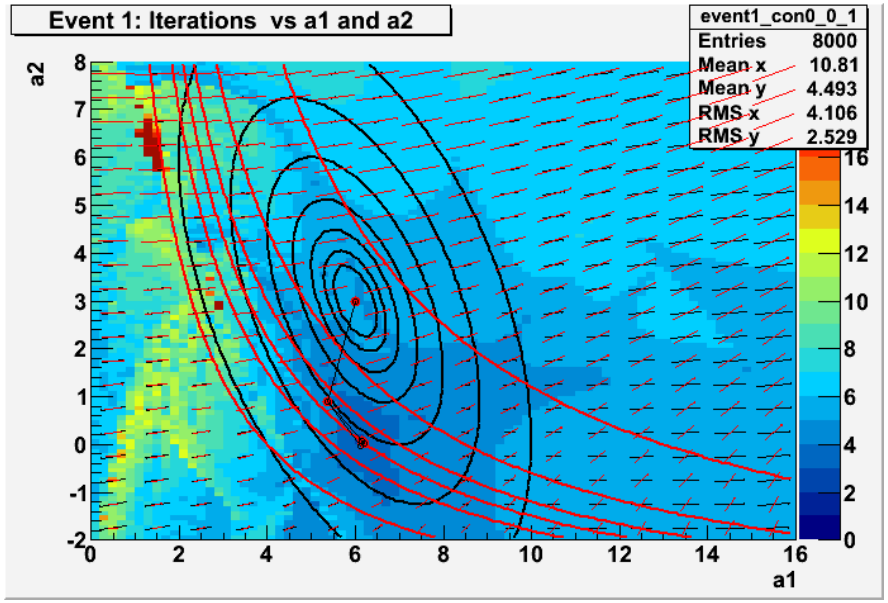
- NewFitterGSL generally needs fewer iterations to converge than OPALFitterGSL, but each iteration takes longer
- Overall: NewFitterGSL converges somewhat better, but your mileage may vary, so try them both

Map: Number of Iterations versus starting point

OPALFitterGSL



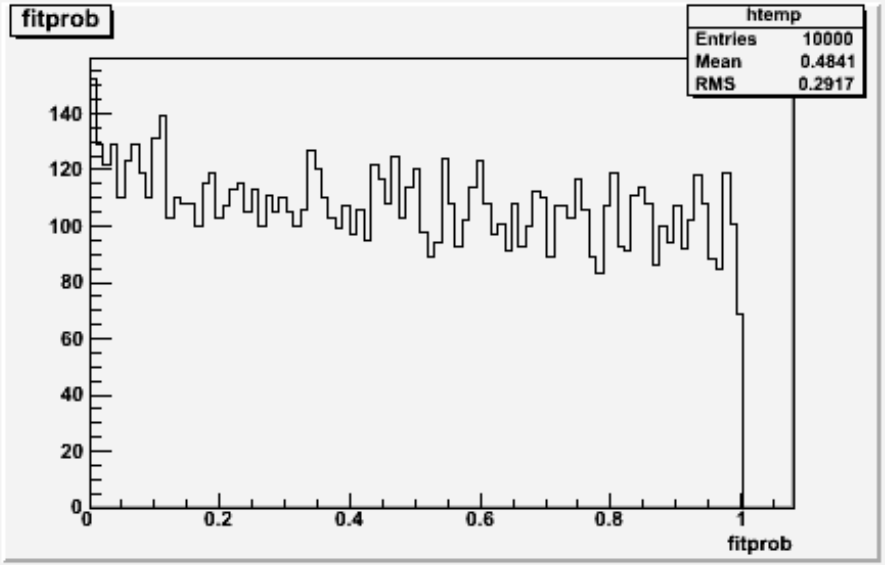
NewFitterGSL

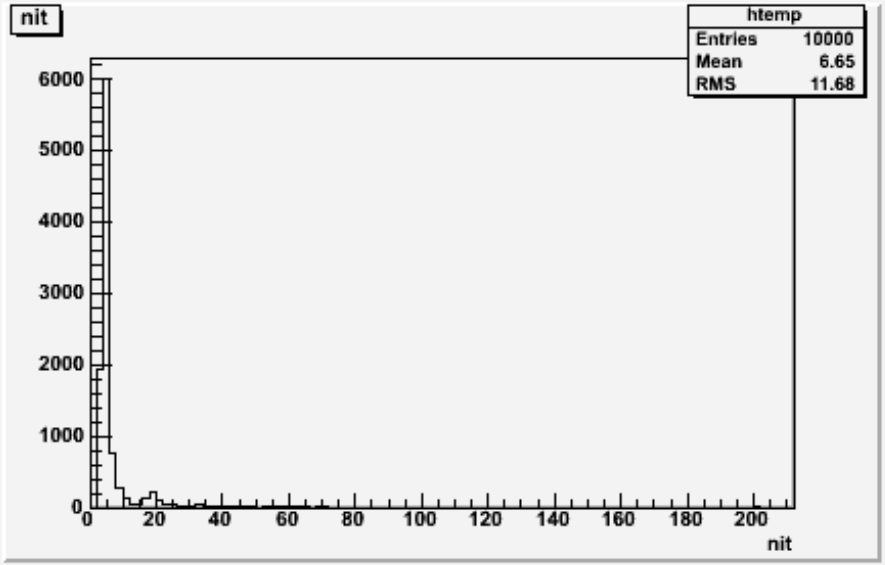


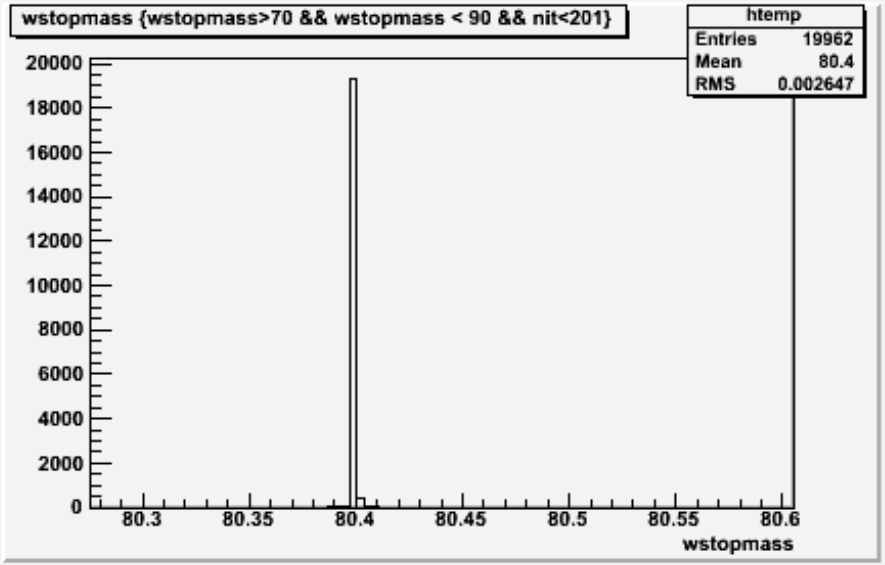
Performance NewFitterGSL

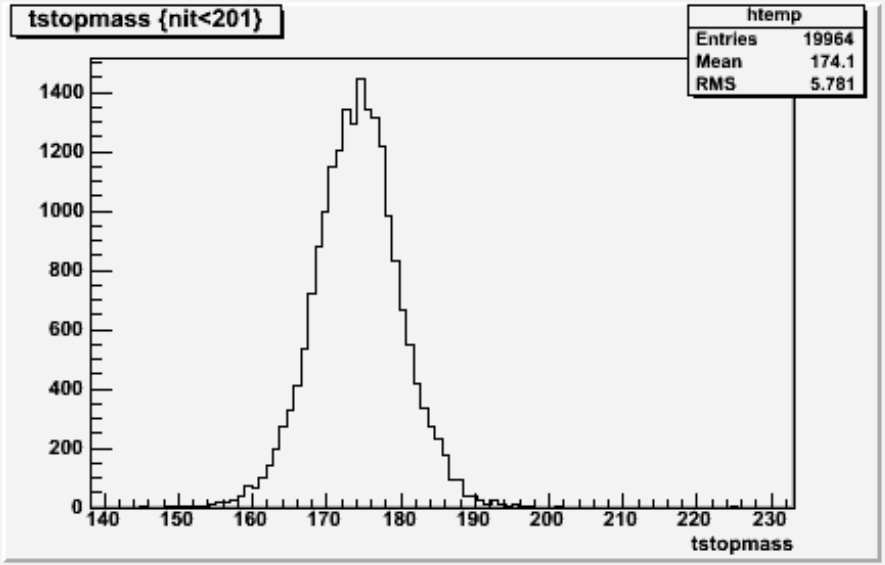
- Test sample:
  - **Toy** MC, e+e- → t t → bℓν bjj at 500GeV
  - Perfect jet pairing
  - 7 constraints: px, py, pz, E, m(W1) = mW, m(W2)=mW, m(t1) = m(t2)
- Times on my Laptop (Intel Core-2 Duo P8400, 2.26GHz) for 10k events
- NewFitterGSL: 17.8s, 0.10% failure rate
- OPALFitterGSL: 9.7s, 0.19% failure rate
- Replace 3 Hard Mass constraints by Soft constraints:
  NewFitterGSL: 86s, 3.2% failure rate
- Disclaimer: This is only a technical check! This toy MC generates just smeared 4-vectors, no realistic physics + detector effects included!

OPALFitterGSL, Hard Constraints

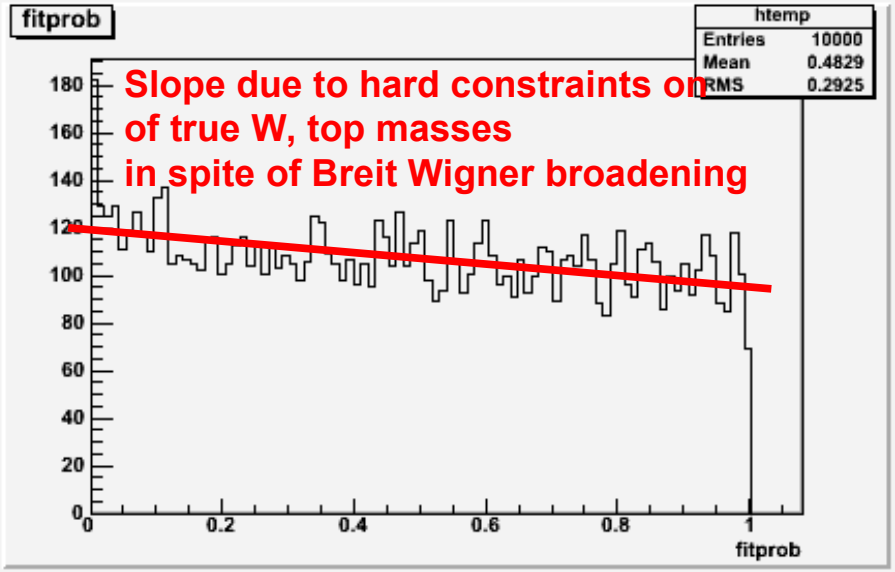


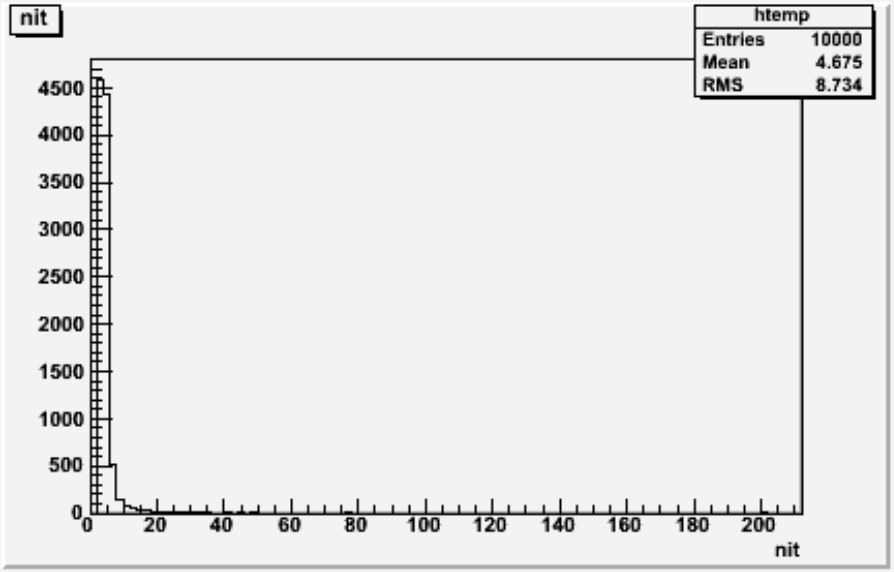


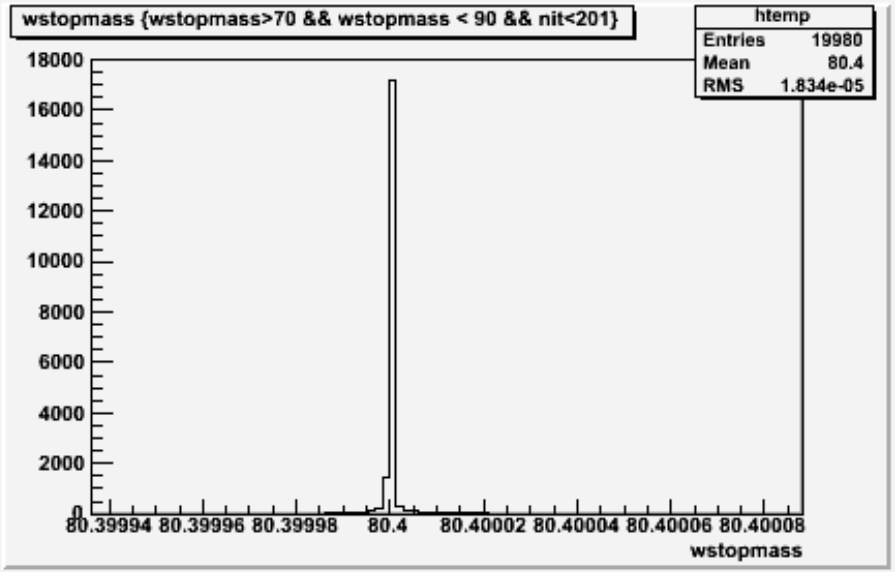


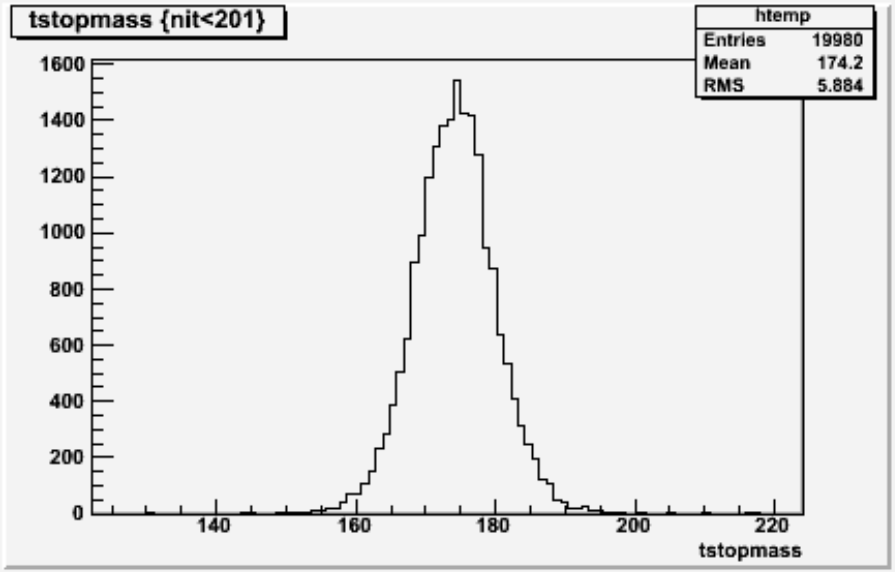


NewFitter, Hard Constraints

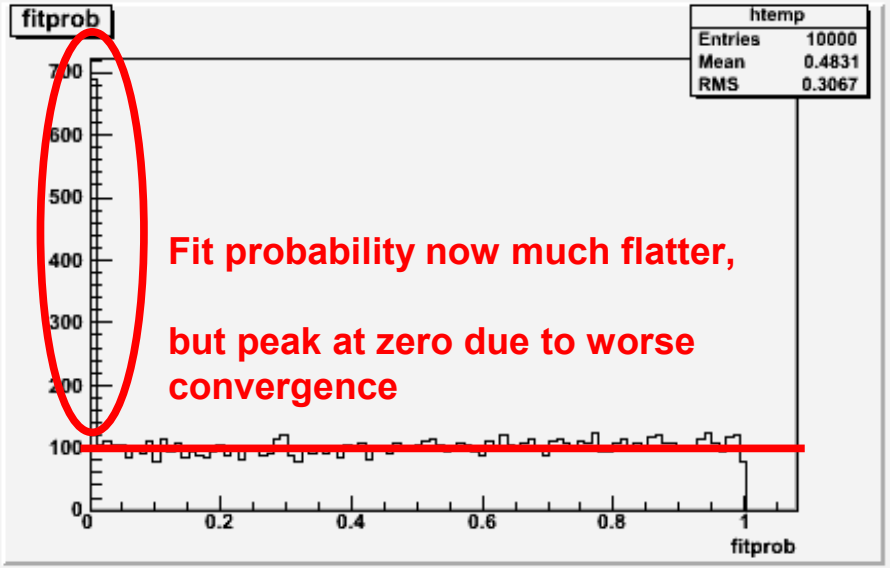


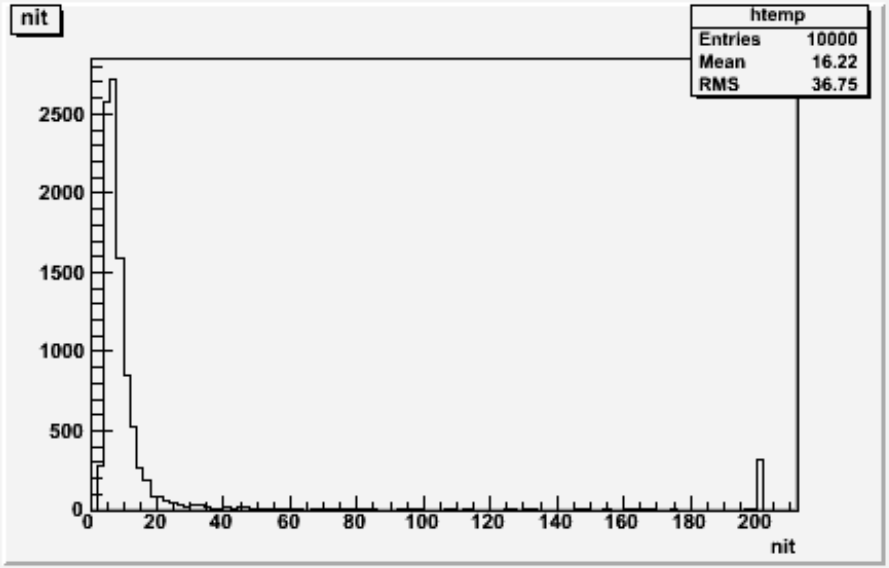


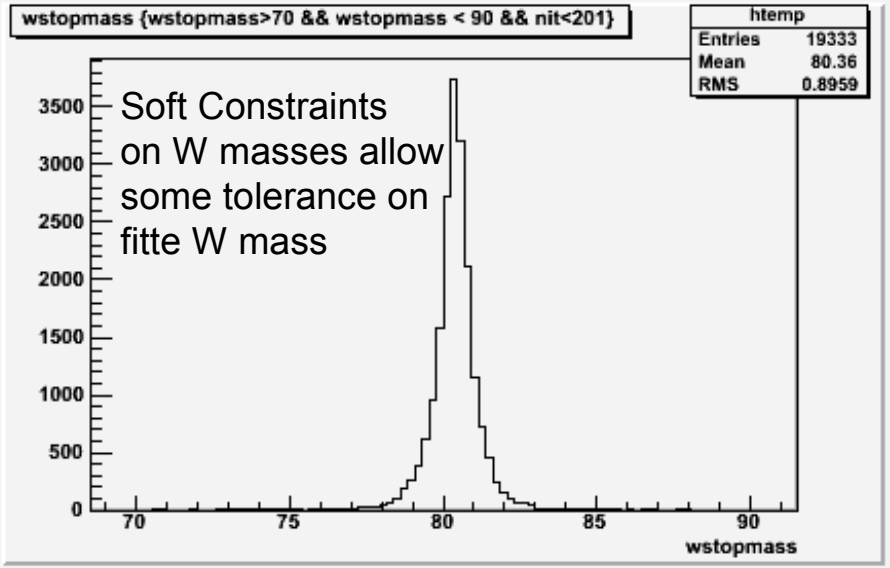


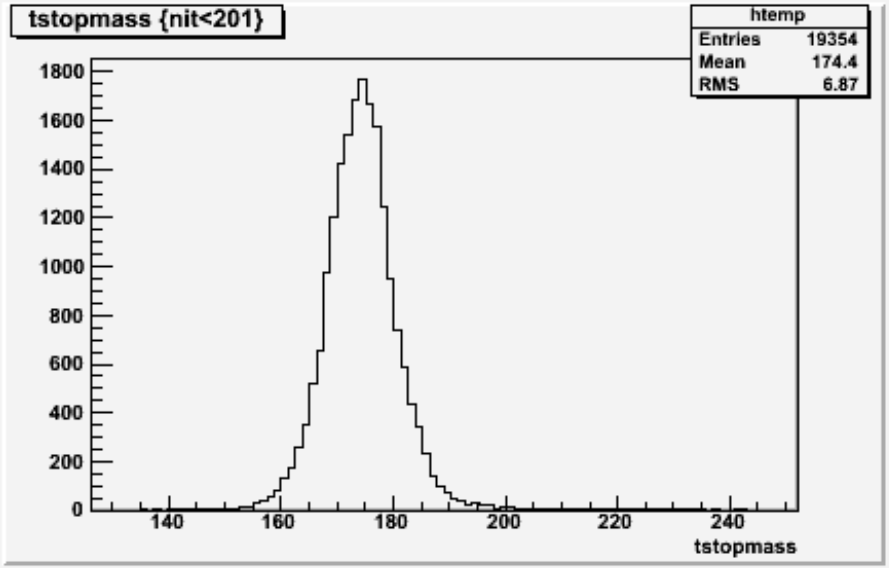


NewFitterGSL, Soft Constraints









Conclusions

- The NewFitter is (expected to be a bit) better than the Old fitter, albeit slower
- Soft constraints can be used with the NewFitter
- Try it and have fun