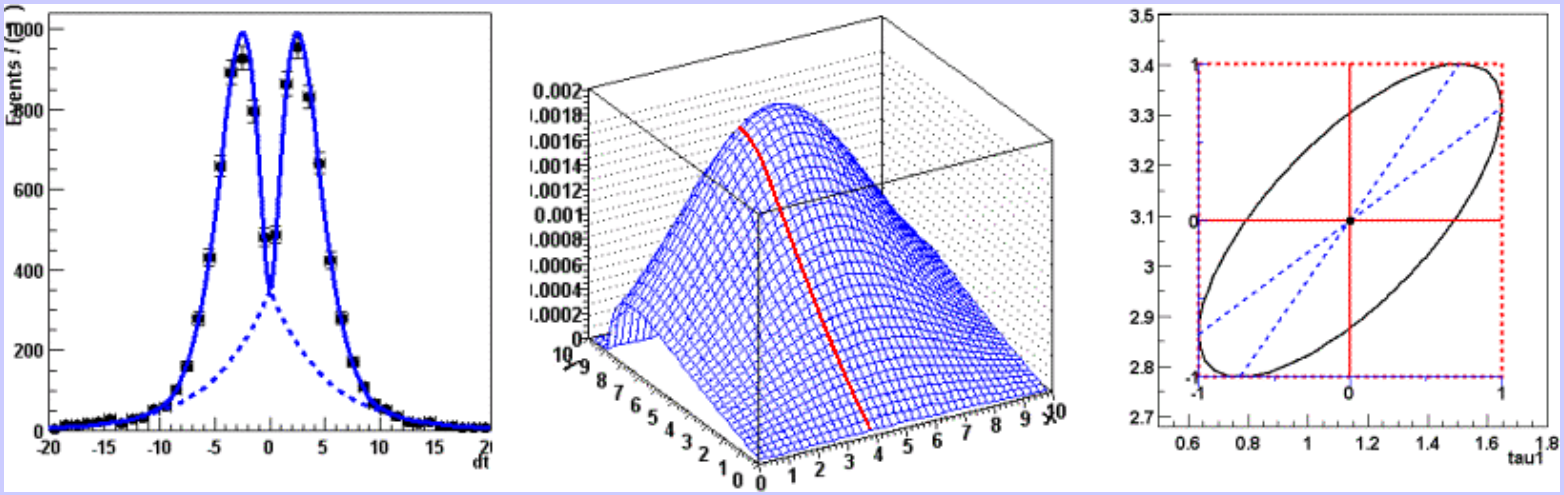
RooFit in BaBar



Chris Roat, Stanford University
BaBar Collaboration
ROOT2004
25 February 2004

Outline

- Overview of RooFit features
- Example #0: Simple Example
- Example #1: Categories, Sharing Variables
- Example #2: Toy MC Tools
- Example #3: Blinding

About RooFit

- Minutes to get started, lifetime to use all features
- Developed by David Kirkby and Wouter Verkerke
- Tool for Unbinned Maximum Likelihood fits
  - Can easy sum, multiply, convolve PDFs
  - Normalizations handled internally
- Optimizations, parallelization
- Nice plot creation facilities
  - Slicing, Projecting, plotting PDF components

Outline

- Overview of RooFit features
- Example #0: Getting Started
- Example #1: Categories, Sharing Variables
- Example #2: Toy MC Tools
- Example #3: Blinding

A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2})^{-1}"),
 gausFrac("Gaussian Fraction","Gaussian Fraction",0,1);
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",RooArgList(g,p),gausFrac);
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```

Fitted variables and parameters are “RooRealVar”

A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2})^{-1}"),
 gausFrac("Gaussian Fraction","Gaussian Fraction",0,1);
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",RooArgList(g,p),gausFrac);
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```

Many pre-defined PDFs!

A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2})^{-1}"),
 gausFrac("Gaussian Fraction","Gaussian Fraction",0,1);
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",RooArgList(g,p),gausFrac);
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```

Set parameters and generate MC “RooDataSet” in just a few lines!

A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2})^{-1}"),
 gausFrac("Gaussian Fraction","Gaussian Fraction",0,1);
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",RooArgList(g,p),gausFrac);
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```

Fit with options

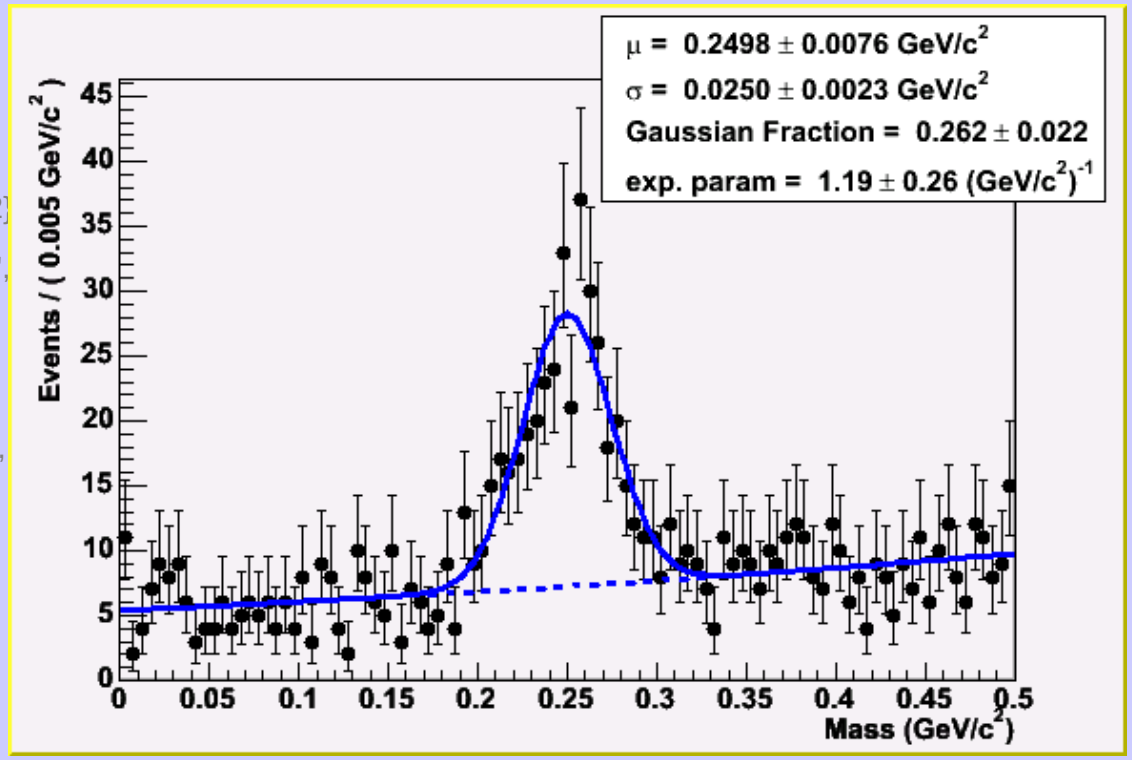
A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2})^{-1}"),
 gausFrac("Gaussian Fraction","Gaussian Fraction",0,1);
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",RooArgList(g,p),gausFrac);
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```

RooPlot (TH1) container helps handle PDF normalization, plotting subcomponets, and doing projections

A Simple Example

```
RooRealVar
 mass("mass","mass",0,0.5,"GeV/c^{2}"),
 mean("#mu","#mu",0,0.25,"GeV/c^{2}"),
 sigma("#sigma","#sigma",0,1,"GeV/c^{2}"),
 bgExp("exp. param", "exp. param",0,10,"(GeV/c^{2}
 gausFrac("Gaussian Fraction","Gaussian Fraction",
RooGaussian g("g","Gaussian",mass,mean,sigma);
RooExponential p("p","Polynomial",mass,bgExp);
RooAddPdf model("model","Gaussian+Polynomial",
mean.setVal(0.25);
sigma.setVal(0.03);
bgExp.setVal(1);
gausFrac.setVal(0.3);
RooDataSet *data= model.generate(mass,1000);
model.fitTo(*data,"m");
RooPlot *frame= mass.frame();
data->plotOn(frame);
model.paramOn(frame,data);
model.plotOn(frame);
model.plotOn(frame,Components(p),LineStyle(kDashed));
frame->Draw();
```



RooPlot can make a very nice plot with axis labels, units, Poisson errors, and parameter box

Outline

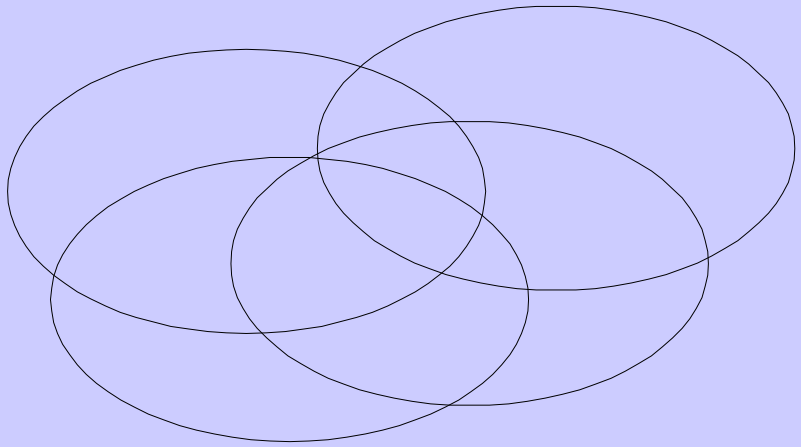
- Overview of RooFit features
- Example #0: Simple Example
- Example #1: Categories, Sharing Variables
- Example #2: Toy MC Tools
- Example #3: Blinding

What is a RooCategory?

- Represents a discrete variable
  - Naturally discrete: charge, flavor
  - Bins: momentum, mass (sideband/signal/unused)
  - Type of data: lepton/hadron, mixed/unmixed
- Makes subdivisions in a dataset
- Fit parameters may be arbitrarily split among categories

Where Categories Help

- An old friend:
  - Make control sample to measure a signal or background shape (with errors)
  - Fit data with measured shape to get a result
  - Vary shape within errors and examine variation in the result
- Measure properties in overlapping datasets:



Extra bookkeeping needed to avoid counting data items more than one time

Example: Counting λ_c-proton pairs

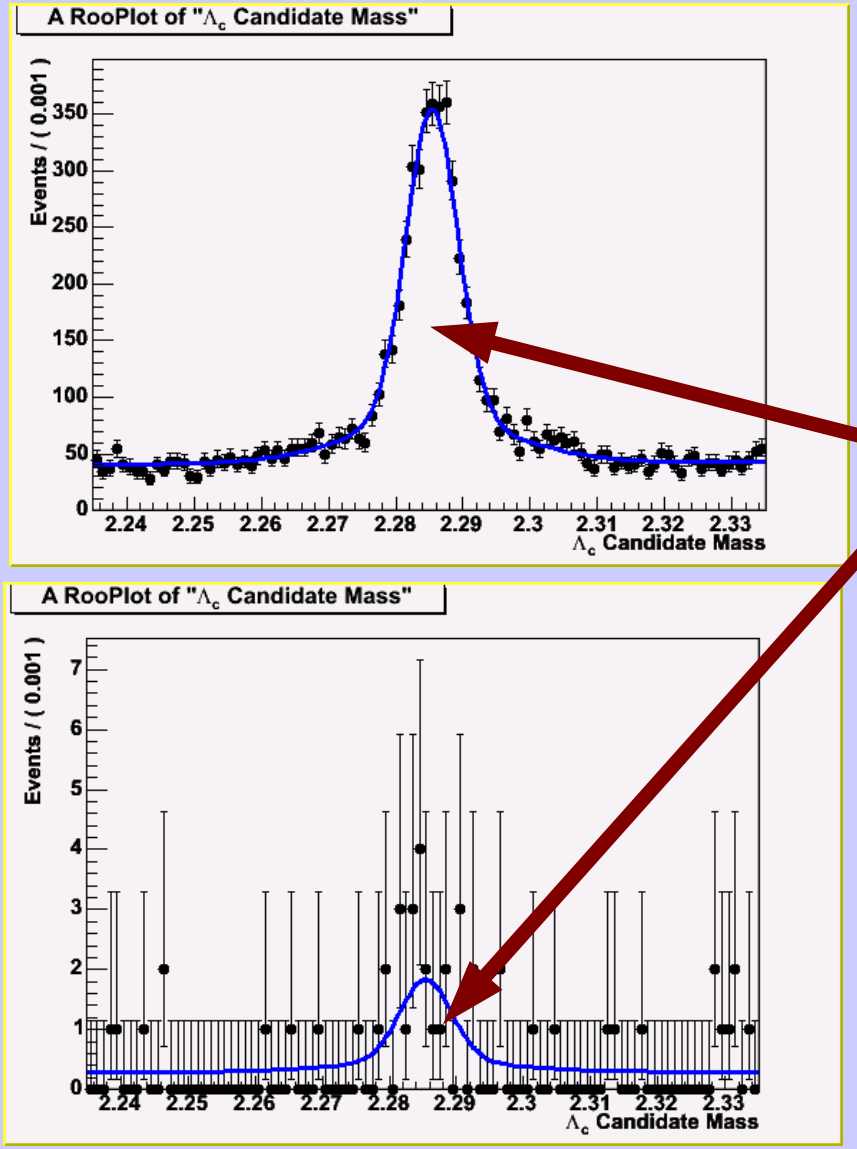
- λ_c invariant mass determined using all λ_c, not just those in pairs
- λ_c in multiple pairs need to be handled correctly
- Also use a fake proton control sample

Shape heavily determined by this subsample

We desired to divide data sample like the following:

	0 Fake	1 Fake	2 Fake	…
0 Real	$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
1 Real	$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
2 Reals	$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
…	…	…	…	…

Sharing Parameters



- Simultaneously fit all sub-samples
- Can make plots of any subsample and it's sub-PDF
- Double Gaussian widths and relative proportions “shared”
- Signal fraction and background shape separate for each sub-sample
- All correlation parameters correctly calculated

Outline

- Overview of RooFit features
- Example #0: Simple Example
- Example #1: Categories, Sharing Variables
- Example #2: Toy MC Tools
- Example #3: Blinding

Toy Monte Carlo Generation

- Class to easily generate and fit

```
RooMCStudy mgr(genpdf,fitpdf,vars,”gopts”,”fopts”);
mgr.generateAndFit(1000000,100); // 1M expts of 100
```

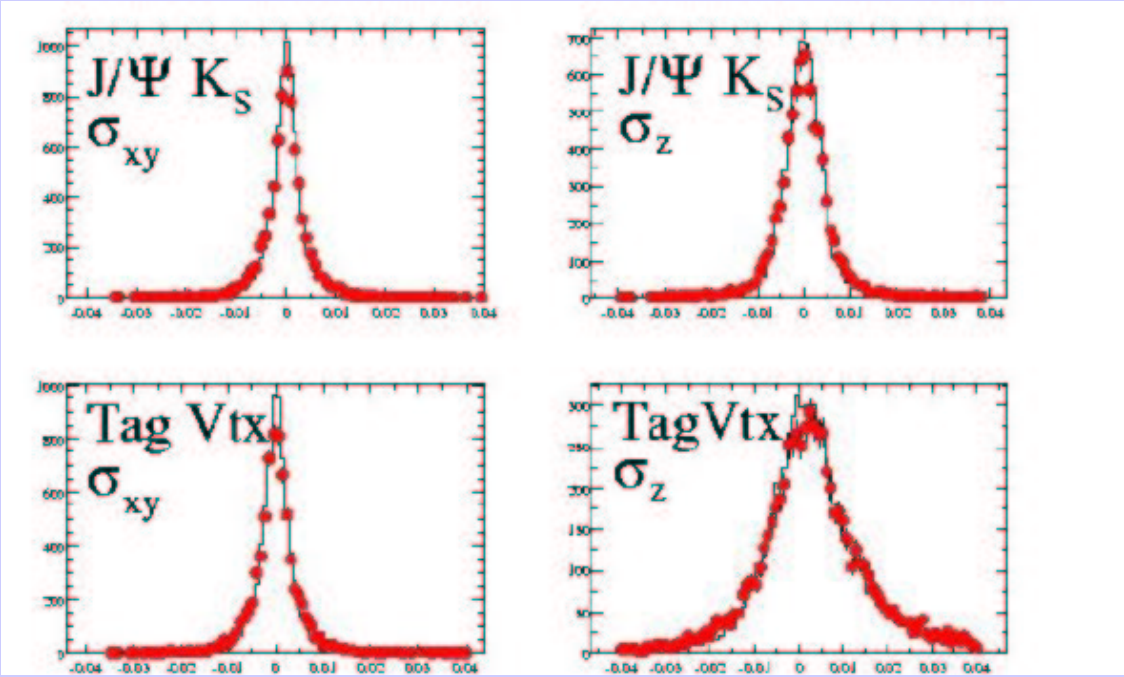
- Data and fit results are managed internally

- Very simple to plot results!

```
mgr.plotParamOn( aVar.frame() )->Draw();
mgr.plotError(aVar,emin,emax)->Draw();
mgr.plotPull(aVar,min,max,bins,boolShowGausFit)->Draw();
```

Effect of Silicon Tracker Damage

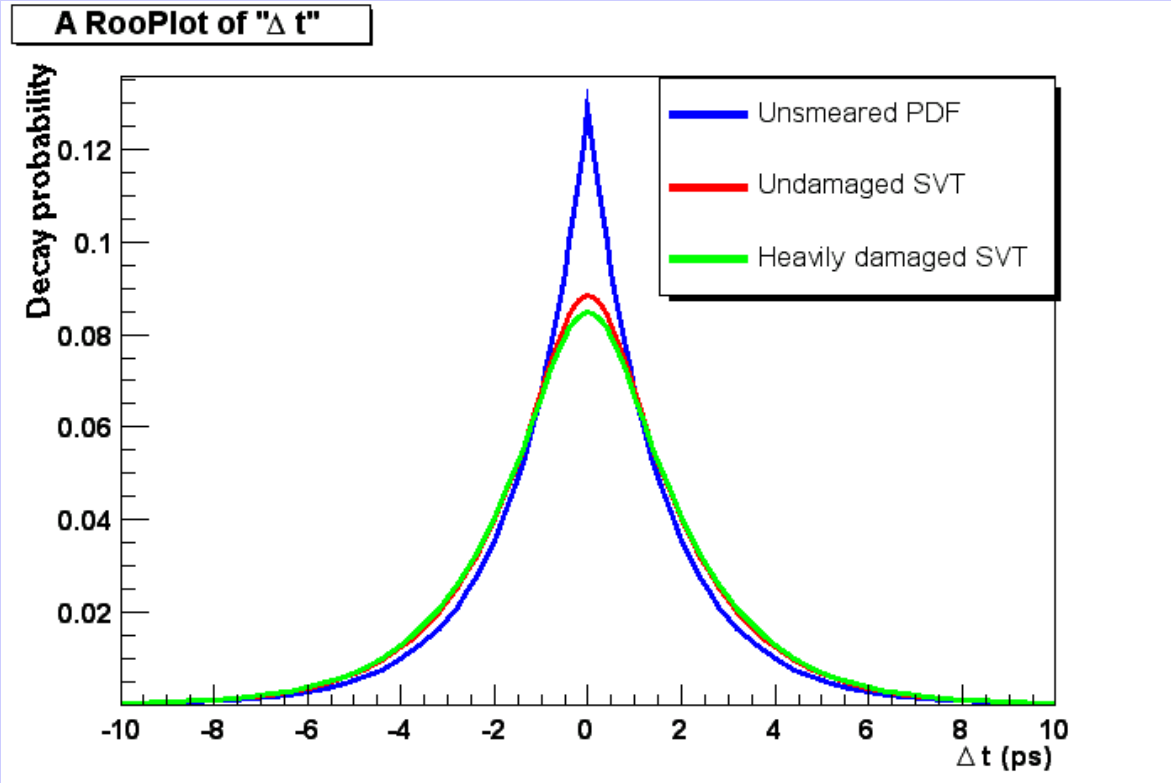
- From full detector simulation, determine resolution degradation with increasing damage (not using RooFit yet)



Solid: undamaged
Points: heavy damage

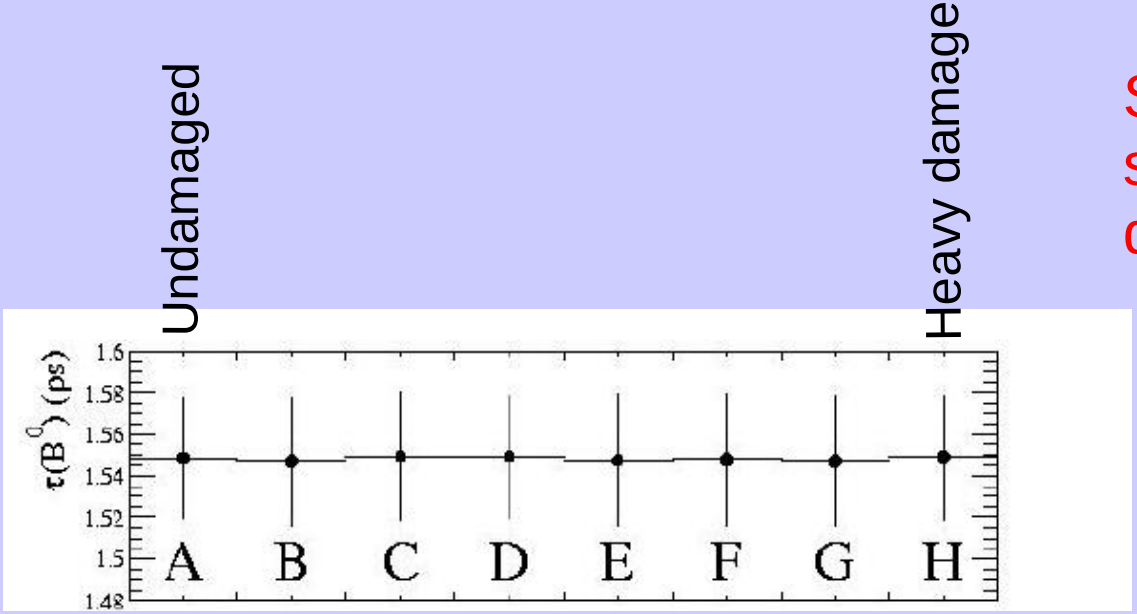
Expected δt

- Use convolution to make a RooPlot of B decay time difference



Effect on B Lifetime

- From Toy MC experiments, determine
  - Average lifetime measurement
  - Spread in average lifetime measurements
- Each point represents results from hundreds of toy experiments:



Shows no significant degradation!!

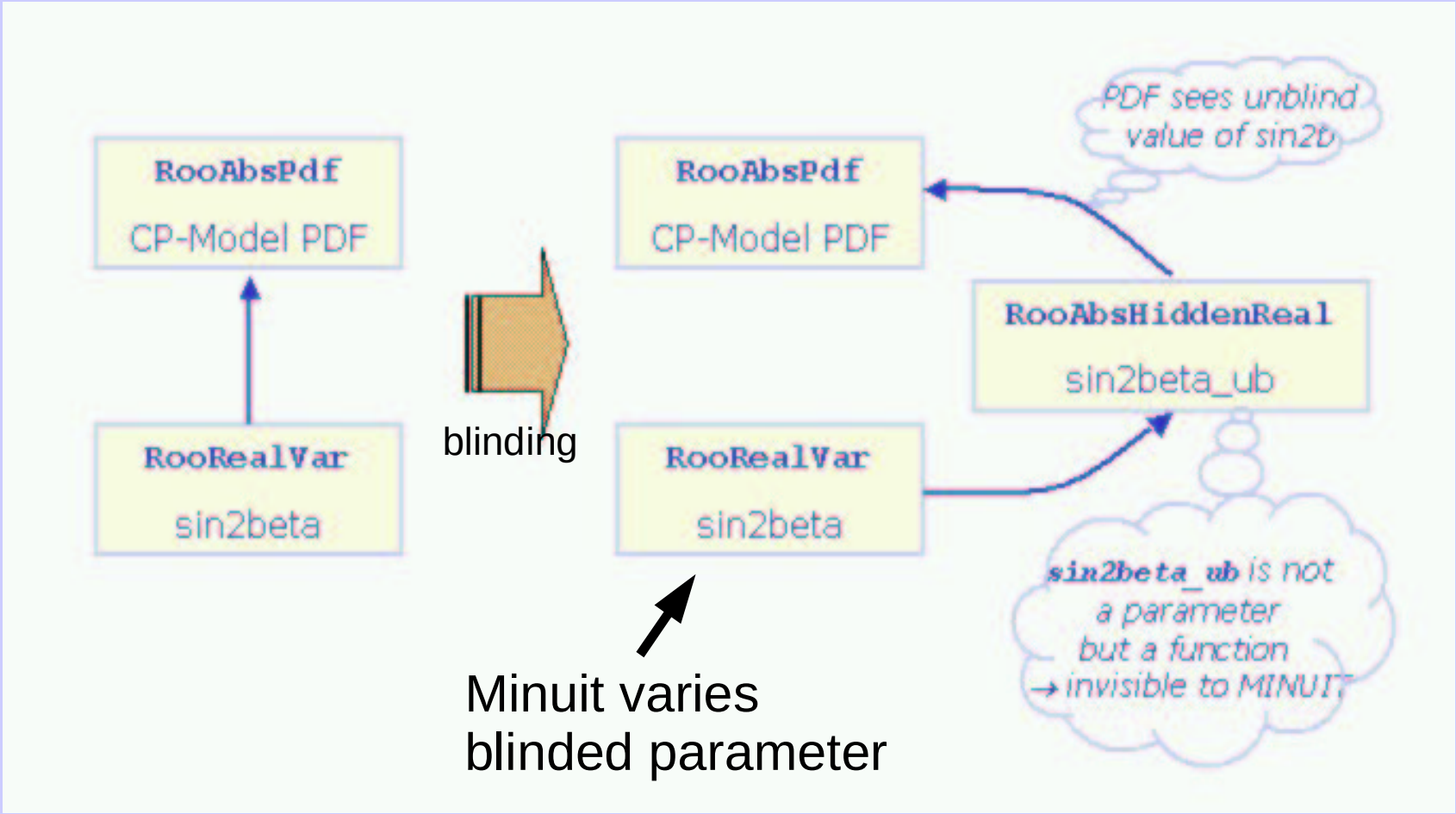
Outline

- Overview of RooFit features
- Example #0: Simple Example
- Example #1: Categories, Sharing Variables
- Example #2: Toy MC Tools
- Example #3: Blinding

What is Blinding in RooFit?

- Hiding the result to avoid biasing the ana-lyst :-)
- Transformation of result can be arbitrary:
  Offset, sign change, rescaling, etc.
- Parameters of transformation are randomly generated from a string:
  Ex: “I would really like to finish this thesis!”
- Unblinded value is given only to your PDF calculation, so even MINUIT is blind!

Blinding for sin2β



The End

- RooFit is easy to use
- Well thought out: very flexible, versatile
- Used in Babar analysis & detector studies

- Download RooFitCore and RooFitModels:
  http://roofit.sourceforge.net

- Tutorials and lots of documentation on website