

## Computerpraktikum zur Vorlesung Teilchenphysik für Fortgeschrittene

### Reconstructing Top Events in CDF Data

We are interested in  $t\bar{t}$ -events decaying via:

$$\begin{array}{lll} t \rightarrow bW^+ \rightarrow \ell^+ \nu_\ell & \text{with } \ell = e, \mu & \text{semileptonic} \\ \bar{t} \rightarrow \bar{b}W^- \rightarrow q\bar{q} & \text{with } q = u, d, s, c & \text{hadronic} \end{array}$$

The charge conjugate states are always implied.

#### Setting up

- Copy the file `ttbar.tar` from the web page of this course into your working directory and unpack it with the command `tar xzf ttbar.tar`. The Directory `Praktikum` is produced, which contains three files: `DrawTTbarPraktikum.cc`, `runTTbarPraktikum.cc`, `rootlogon.C`. In addition the directory contains two sub directories: `include`, `Results`.
- We will analyze the data contained in the directory `/home/staff/jwagner/TopData/Rootfiles`. Have a look at the directory `/home/staff/jwagner/TopData` and its sub directory `Rootfiles`. The root files we will use, are:
  - `tight_elmu_dataMiniTree.Praktikum.root`  
containing real preselected lepton + jets events (CDF data)
  - `ttopelMiniTree.Praktikum.root`  
containing  $t\bar{t}$  Monte Carlo (MC) events ( $m_t = 178$  GeV)
  - `atop_We_3pMiniTree.Praktikum.root`,  
`atop_We_4pMiniTree.Praktikum.root`,  
`atop_Wmu_4pMiniTree.Praktikum.root`  
containing background Monte Carlo events

- `ttop0zMiniTree.Praktikum.root`  
containing  $t\bar{t}$  Monte Carlo events ( $m_t = 175$  GeV)

All Monte Carlo samples are subjected to a full CDF detector simulation.

- Have a look at the directory `include`: In the files `TTbarPraktikum.cc` and `TTbarPraktikum.hh` you can implement your analysis code to select  $t\bar{t}$  events and to fill histograms. The files `HTfit.cc` and `HTfit.hh` will be used for the extraction of the  $t\bar{t}$ -signal using a fit. Routines from `PlottingRoutines.cc` can be used in the main program `TTbarPraktikum.cc` to get nice histograms. A shared library containing the routines from `TTbarPraktikum.cc`, `HTfit.hh`, `PlottingRoutines.cc`, ... are produced by the commando sequence `make clean` and `make lib`. Have also a look at the `Makefile`.
- The file `runTTbarPraktikum.cc` in the upper directory executes the routines from the shared library `TTbarPraktikum.so` produced before. This routine can be called in ROOT by the command `.x runTTbarPraktikum.cc(proc)`, where `proc` is one of the different process ID's:
  - `proc=6000`: real CDF data
  - `proc=600`:  $t\bar{t}$  Monte Carlo ( $m_t = 178$  GeV)
  - `proc=601`: W+4jets background Monte Carlo
  - `proc=602`: W+3jets background Monte Carlo
  - `proc=603`:  $t\bar{t}$  Monte Carlo ( $m_t = 175$  GeV)

In the ROOT script `run TTbarPraktikum.cc` the constructor of `TTbarPraktikum()` is called first. Then the Job is started with `BeginJob()`, files are read, and in `EventLoop()` a loop over all events is performed. Here, the routine `TTbarPraktikum::Event()` is called for each event, where several routines for the selection and the filling of histograms are called. Have a look at the file `TTbarPraktikum.cc` and here in particular at the routines `TTbarPraktikum::EventLoop()`, `TTbarPraktikum::Event()` and the routines called in the latter routine. In the end `EndJob(rfile)` is called and the histograms are stored in ROOT files in the directory `Results`:

- `TTbarPraktikum.Plots.dtype=1.proc=6000.root`  
Histograms filled with real CDF data
- `TTbarPraktikum.Plots.dtype=2.proc=600.root`  
Histograms filled with  $t\bar{t}$  Monte Carlo ( $m_t = 178$  GeV)
- `TTbarPraktikum.Plots.dtype=2.proc=601.root`  
Histograms filled with W+4jets background Monte Carlo
- `TTbarPraktikum.Plots.dtype=2.proc=602.root`  
Histograms filled with W+3jets background Monte Carlo

- `TTbarPraktikum.Plots.dtype=2.proc=603.root`  
Histograms filled with ttbar Monte Carlo ( $m_t = 175$  GeV)

They can be read by the root script `DrawTTbarPraktikum.cc`. This script will draw and fit the histograms.

In table 1 some usefull C++ and ROOT commands are summarized. The ROOT Trees used for this small analysis contain information about the reconstructed lepton, the missing transverse energy, the high energetic jets. The full Tree description is given in table 2. Note that the ‘truth 4-vectors’ will always be zero for real data.

<b>C++ Comands</b>	
For loop	for(int i=0; i<20; i++){...}
If statement	if(a>=b){...}else{...}
$\geq$	>=
$\leq$	<=
$:=$	=
=	==
$\neq$	!=
negation of statement	!( <i>statement</i> )
logical OR	
logical AND	&&
a = a + x	a += x
$\sqrt{a}$	sqrt(a)
<b>ROOT: TH1, TH2</b>	
Fill 1dim. histogram	histo→Fill(x);
Fill 2dim. histogram	histo→Fill(x,y);
Fill 1dim. histogram with weight w	histo→Fill(x,w);
Scale a histogram with a factor a	histo→Scale(a);
Get entries in histogram	histo→Integral();
Write histogram into a file	histo→Write();
<b>ROOT: TLorentzVector (lvector: (<math>P_x, P_y, P_z, E</math>))</b>	
Assign values	lvector.SetPxPyPzE(px, py, pz, e)
$P_{x,y,z}$	lvector.Px(), .Py(), .Pz()
$E$	lvector.E()
Invariant mass	lvector.M()
Pseudorapidity	lvector.Eta()
Rapidity	lvector.Rapidity()
Transverse momentum	lvector.Pt()
<b>ROOT: TVector2 (vector: (x,y))</b>	
get x	vector.X()
get y	vector.Y()
<b>ROOT: TMath</b>	
$x^a$	TMath::Power(x,a)
$\pi$	TMath::Pi()
$ a $	TMath::Abs(a)
$\ln(a)$	TMath::log(a)

Tabelle 1: Some usefull C++ and ROOT commands.

Variable Name	Type	Definition
procId	Int_t	ID characterizing the process (Data > 1000, MC < 1000)
<b>Charged lepton variables (<math>e</math> or <math>\mu</math>) - reconstructed</b>		
lep4	TLorentzVector	Four-momentum of the charged lepton
lepType	Int_t	Electron $\rightarrow$ 1; muon $\rightarrow$ 2
lepQ	Float_t	Charge of lepton
<b>Missing energy - reconstructed</b>		
misEt	TVector2	Missing transverse energy vector (Corresponds to px, py of neutrino)
<b>High energetic jets - reconstructed</b>		
nJets	Int_t	Number of high energetic jets in event
btag	Double_t	b-tag information for each jet (array of nJets entries)
jet4	TLorentzVector	Four-momentum of each jet (array of nJets entries)
<b>Event shape variable - reconstructed</b>		
HT	Double_t	Total transverse energy in the event
<b>Monte-Carlo truth</b>		
p4Top_lep_gen	TLorentzVector	4-vector of the semileptonically decaying t-quark
p4Top_had_gen	TLorentzVector	4-vector of the hadronically decaying t-quark
p4W_lep_gen	TLorentzVector	4-vector of the leptonically decaying W-boson
p4W_had_gen	TLorentzVector	4-vector of the hadronically decaying W-boson
p4Lep_gen	TLorentzVector	4-vector of the charged lepton from t-decay
p4Neutrino_gen	TLorentzVector	4-vector of the neutrino
QLep_gen	Double_t	Charge of the generated lepton from top

Tabelle 2: Description of the used preselected trees.

## Exercises

### 1. $t\bar{t}$ -Signal Extraction

#### 1.1 Kinematics of $t\bar{t}$ -Events

Have a look at some Monte-Carlo truth quantities:

Plot the transverse momentum and the rapidity of the top quarks. Use for this, both top quarks, thus the semileptonically and hadronically decaying top quark.

Have a look at the two dimensional distributions: transverse momentum versus rapidity of the top quarks (semileptonically and hadronically decaying top quark combined), the transverse momentum of the anti-top versus the transverse momentum of the top quark and the rapidity of the anti-top versus the rapidity of the top. Consider for the latter two plots the relation of the quark type (top or anti-top) and

the charge of the lepton!  
 What do you conclude from all these plots?

*Instructions:*

Move to the directory `/Praktikum/include` and open the file `TTbarPraktikum.cc`. You have to add some code to `TTbarPraktikum.cc`. Initialize the histograms in the routine `TTbarPraktikum::PrepareHistos()` and fill these histograms in `TTbarPraktikum::FillTopMC()`. The recommended range, where  $p_t$  should be plotted, is 0-300 GeV with 30 bins. For the rapidity the range -2 to 2 and 30 bins are recommended. A variable out of the tree is read via `cEv->variable name`, for example `cEv->p4Top_lep_gen` to get the real four momentum of the leptonically decaying top quark. The ROOT class `TLorentzVector` provides some routines to calculate easily the transverse momentum (`.Pt()`, e.g. `cEv->p4Top_lep_gen.Pt()`), rapidity (`.Rapidity()`), mass (`.M()`). More information can be obtained on the web-page <http://root.cern.ch/root/html404/TLorentzVector.html>. In order to store the histograms into a root file call `histo->Write()` for each one or two dimensional histogram in the routine `TTbarPraktikum::EndJob(const char* rfile)`. Update the shared library with `make clean` and `make lib`. Afterwards execute `runTTbarPraktikum.cc` with the process ID 600. The command in ROOT is thus `.x runTTbarPraktikum.cc(600)`. Then, you can plot the histograms using the ROOT script `DrawTTbarPraktikum.cc`. Type in ROOT `.x DrawTTbarPraktikum.cc(1)`. This executable calls the routine `TTbarPraktikum::DrawTTbarMC(const char* rfile)`. The plots are stored as ps-file in the directory `Results` under the name `SigMCVariables_1.ps`.

## 1.2 Number of high energetic jets

Plot the number of high energetic jets of the preselected data sample ( $N_{jets} \geq 1$ ) and of the  $t\bar{t}$  Monte-Carlo. How do you interpret this plot? What cut would you apply to look for  $t\bar{t}$ -events in real data?

*Instructions:*

Add code to `TTbarPraktikum.cc`. The initialization of the histogram should be done again in `TTbarPraktikum::PrepareHistos()` (use `hTTbarEv[0]`), the histogram should be filled in `TTbarPraktikum::FillEventVariable()` and the histogram should be written to the root file in `TTbarPraktikum::EndJob()`. Do not forget to update the shared library with `make clean` and `make lib`. Execute again `runTTbarPraktikum.cc`. You have to run the program once for the real data (`proc=6000`) and once for the  $t\bar{t}$  Monte Carlo (`proc=600`). In order to plot the histograms the routine `TTbarPraktikum::DrawTTbarEv(const char* rfile1, const char* rfile2, const char* rfile3)` is used. In this routine the histogram `hdata[0]` is used for the data and `hmc[0]` is used for the Monte-Carlo.

`rfile1` is the ROOT-file with the real data and `rfile2` is the  $t\bar{t}$  Monte-Carlo. In order to get a direct comparison between both distributions normalize the Monte-Carlo distribution to the number of entries in the data. Tip: The number of entries in a histogram is obtained by `histo->Integral()` and a histogram can be scaled by a factor  $f$  using the command `histo->Scale(f)`. The code to plot both histograms in one canvas is then given again. Execute `DrawTTbarPraktikum.cc`, now with `draw=2`. Note that the command in `DrawTTbarPraktikum.cc` has to be `ana->DrawTTbarEv(rfile1, rfile2, rfile2)`, since the third file is not filled up to now. The plots are again stored as ps-file in the directory `Results`.

### 1.3 Study of Event Shape Variables

Study, which event shape variable could be used to discriminate background (W+3 jets, `proc=602`) and  $t\bar{t}$ -signal Monte-Carlo (`proc=600`). For this, select events with at least three jets ( $N_{jets} \geq 3$ ). Look at  $H_T$  (use the histogram `hTTbarEv[1]`), the total transverse energy in the event, the missing transverse energy (absolute value, `hTTbarEv[2]`), the transverse momentum (`hTTbarEv[3]`) and the pseudorapidity (`hTTbarEv[4]`) of the reconstructed charged lepton. Plot the distributions for background and signal in one histogram. Which variable has the largest discrimination power?

*Instructions:*

Add code in `TTbarPraktikum::PrepareHistos()`, `TTbarPraktikum::FillEventVariable()`, `TTbarPraktikum::DrawTTbarEv(const char* rfile1, const char* rfile2, const char* rfile3)`, where `rfile3` will contain histograms of the background, and `TTbarPraktikum::EndJob(..)`.

The recommended range of the variable  $H_T$  is 0-600 GeV, that of the missing transverse momentum and that of the transverse momentum of the charged lepton is for both 0-150 GeV and that of the pseudorapidity of the charged lepton is -1.5 to 1.5. For all cases 30 bins can be used. You have to execute `runTTbarPraktikum.cc` for the  $t\bar{t}$  Monte Carlo (`proc=600`) and for the W+3 jets background (`proc=602`). Draw the histograms using `DrawTTbarPraktikum.cc`, where now the command `ana->DrawTTbarEv(rfile1, rfile2, rfile2)` is replaced by `ana->DrawTTbarEv(rfile1, rfile2, rfile3)`. Note that you have to modify the routine `TTbarPraktikum::DrawTTbarEv(..)` for that.

### 1.4 Extraction of the $t\bar{t}$ Signal ( $N_{jet} \geq 3$ )

Use the variable  $H_T$  as discrimination variable and fit the  $t\bar{t}$ -template and the background template to the data. What  $t\bar{t}$ -signal fraction do you get? How can you improve that?

*Instructions:*

Fill a histogram `hTTbarEv[11]` with the variable  $H_T$ . Use the same range as before, but reduce the number of bins to 15. Execute `runTTbarPraktikum.cc` for the data (`proc=6000`), signal (`proc=600`) and background Monte Carlo (`proc=602`).

The code for the fit can be found in `HTfit.cc` and `HTfit.hh`. The theory prediction `ValFit[i]` for each bin  $i$  consist of two parts, namely the  $t\bar{t}$  signal contribution - `Ndata*par[1]*ValSignal[i]` and the background contribution - `Ndata*(1-par[1])*ValBG[i]`. Here `Ndata` is the total number of entries of the data, `par[1]` is a free parameter and gives the  $t\bar{t}$ -signal fraction. `ValSignal[i]` and `ValBG[i]` respectively are the to unity normalized signal and background distributions. These arrays and also the corresponding arrays for the data are filled from the histograms you produced before. Implement the function for the theory prediction `ValFit[i]` into `HTfit::fcn(..)`. In order to get that theory prediction, which describes the data in the best way, the function  $f_1$  has to be minimized in case of a least square fit and the function  $f_2$  has to be minimized in case of a likelihood fit (holds also for small statistics samples):

$$f_1 = \sum_i^{\text{nbins}} (\text{ValData}[i] - \text{ValFit}[i])^2 / (\text{ErrorData}[i])^2$$

$$f_2 = \sum_i^{\text{nbins}} 2 \cdot (\text{ValFit}[i] - \text{ValData}[i] \cdot \ln(\text{ValFit}[i]) + \text{ValData}[i] \cdot \ln(\text{ValData}[i]) - \text{ValData}[i])$$

In the code,  $f_1$  and  $f_2$  are labeled with `fMin`. `ValData[i]` is the number of data events in bin  $i$ , `ErrorData[i]` is the corresponding uncertainty and `nbins` is the number of bins used for the fit. Note that the command for  $\ln$  is in C++ and ROOT `log`. In order to get a very stable fit, first a  $\chi^2$ -fit is performed and than a likelihood fit. This is controlled by the parameter `par[0]`. For `par[0]==1` a  $\chi^2$ -fit is performed and for `par[0]==2` a likelihood fit is performed. Implement in `HTfit::fcn(..)`  $f_1$  and  $f_2$ .

What happens, when `ErrorData[i]` is 0? And what happens, when `ValData[i]` is 0?

The fit itself is then performed by the ROOT class `TMinuit`. Execute `DrawTTbarPraktikum.cc` with `draw=3 (sel=1)`. The result of the fit will be automatically visualized.

## 1.5 Improvement of the Signal to Background Ratio ( $S/B$ )

Extract the  $t\bar{t}$ -signal for the three additional cases:

1.  $N_{jets} \geq 3$  and at least one jet which has a  $b$ -tag

2.  $N_{jets} \geq 4$  and no  $b$ -tag
3.  $N_{jets} \geq 4$  and at least one jet which has a  $b$ -tag

How does the  $S/\sqrt{B}$  ratio change? For which scenario is the significance  $S/\sqrt{B}$  largest (useful for the discovery and cross section measurement)? Which scenario is the best for the determination of the top mass (clean sample needed)?

*Instructions:*

Extend the code of the routine `TTbarPraktikum::FillEventVariable()`. Fill histograms with the variable  $H_T$  for these three cut scenarios (`hTTbarEv[12]`: 1. scenario, `hTTbarEv[13]`: 2. scenario, `hTTbarEv[14]`: 3. scenario). Use the range 0-600 GeV with 15 bins for  $H_T$ . In order to apply the  $b$ -tag requirement you have to implement a loop over all high energetic jets  $i$  and to check if at least one of them has `btag[i]=1`. Execute `runTTbarPraktikum.cc` for the data (`proc=6000`), signal (`proc=600`) and the two background Monte Carlo samples (`proc=602 W+3 jets` and `proc=601 W+4jets`). Use again `DrawTTbarPraktikum.cc` with `draw=3` to perform the fits and to visualize the results. Note that the loop over parameter `sel` has to be extended (up to 4 now) for the three scenarios in `DrawTTbarPraktikum.cc`.

## 2. Determination of the Top Mass

For the determination of the mass the cut scenario 3 is used ( $N_{jets} \geq 4$  and at least one jet which has a  $b$ -tag).

### 2.1 Reconstruction of $t\bar{t}$ -Events

Reconstruct the four-momenta of the two top quarks:  $t \rightarrow bW^+ \rightarrow b\ell^+\nu_\ell$  (semileptonically decaying top quark) and  $\bar{t} \rightarrow \bar{b}W^- \rightarrow \bar{b}q\bar{q}$  with  $q = u, d, s, c$  (hadronically decaying top quark). The charged conjugated state is also implied. Use for the top pair reconstruction the four-momentum of the charged lepton, the four-momenta of the four jets and the missing transverse energy. Store the result for all possible hypotheses in the object `ttHyp` which is from type `TTbarHypotheses()` (see `TTbarHypotheses.hh` and `TTbarHypotheses.cc`).

*Instructions:*

The selection of events fulfilling the requirements  $N_{jets} \geq 4$  and at least one high energetic jet with a  $b$ -tag is performed by the routine `IsSelected()`, which is called automatically in `TTbarPraktikum::Event()` for the top mass exercise. Extend the code in `TTbarPraktikum::RecTop()`. The determination of the four-momentum of the charged lepton and the calculation of the two solutions for the  $z$ -component of the momentum of the neutrino (see introductory text) is already

performed. Furthermore a loop over these two  $P_{z,\nu}$  solutions is done, where the first one is the solution with the smaller  $|P_{z,\nu}|$  and the second one (in case of two real solutions) is the solution with the larger  $|P_{z,\nu}|$ . In order to have later a selection criterion of the hypotheses, a flag `Pnu=1` is set for the smaller  $|P_{z,\nu}|$  solution and `Pnu=0` for the other solution.

Calculate the energy of the neutrino and the four-momentum for the two possible cases and then the four-momentum of the leptonically decaying  $W$ -boson. Note that the four-vectors can be easily added via the operation `+`. In order to get the four-momentum of the semileptonically decaying top quark you have to loop over the high energetic jets. For the reconstruction of the hadronically decaying top quark you have to loop three times over the jets (two light quarks jets and one  $b$ -jet). Ensure, that the jets assigned as light quark jet do not have a  $b$ -tag (`btag=0`). Do not require an explicit  $b$ -tag for both jets assigned as  $b$ -jet. Furthermore, look after that you use the jets only once for a hypothesis and that you use only different combinations of light jets (jet 2 and jet 3 is for the combination of light jets the same as jet 3 and jet 2). Calculate the four-momentum of the hadronically decaying  $W$ -boson and of the hadronically decaying top quark. Store the four-momentum of the two top quarks in `ttHyp->P4Top_lep[sumHyp]` and `ttHyp->P4Top_had[sumHyp]`. Increase afterwards the number of hypotheses `sumHyp` by one. Store the number of hypotheses in `ttHyp->nhyp` after closing the loop.

Compile the code and produce the shared library `TTbarPraktikum.so`.

## 2.2 Selection of the Best Hypothesis

Find that hypothesis out of all hypotheses, which has `Pnu=1` and which has a minimal  $\chi^2$  but  $\chi^2 > 0$ . Store the index of this “best” hypothesis `sumHyp` in `ttHyp->idxBestChi2`.

*Instructions:*

Extend the code of `TTbarPraktikum::RecTop()` further. Calculate  $\chi^2$  according to equation (2) in the introductory text:

$$\chi^2 = \frac{(m_{W \rightarrow jj} - M_{W \rightarrow jj, exp.})^2}{\sigma_{M_{W \rightarrow jj, exp.}}^2} + \frac{(m_{top \rightarrow bl\nu} - m_{top \rightarrow bjj})^2}{\sigma_{\Delta M_{t, exp.}}^2} \quad (1)$$

Here,  $m_{W \rightarrow jj}$  is the reconstructed mass of the hadronically decaying  $W$ -boson and  $m_{top \rightarrow bl\nu}$  and  $m_{top \rightarrow bjj}$  are the reconstructed mass of the semileptonically decaying top and the hadronically decaying top quark, respectively. The central value of the measured hadronically decaying  $W$ -mass  $M_{W \rightarrow jj, exp.}$  and the width  $\sigma_{M_{W \rightarrow jj, exp.}}$  of that distribution as well as of the distribution of the mass difference of the two top

quarks  $\Delta M_{t,exp.}$  are determined to be:

$$\begin{aligned} M_{W \rightarrow jj,exp.} &= 79.3 \text{ GeV} \\ \sigma_{M_{W \rightarrow jj,exp.}} &= 18.0 \text{ GeV} \\ \sigma_{\Delta M_{t,exp.}} &= 32.1 \text{ GeV} \quad . \end{aligned}$$

Store in addition for each hypothesis  $P_{nu}$  and  $\chi^2$  in `ttHyp->PNu[sumHyp]` and `ttHyp->Chi2[sumHyp]` respectively. Compile the code and produce the shared library `TTbarPraktikum.so`.

### 2.3 Reconstructed Top Mass Distribution

Produce the reconstructed mass distribution of the semileptonically decaying top quark and of the hadronically decaying top quark and the distribution of the averaged reconstructed top mass  $((m_{t \rightarrow b\ell\nu} + m_{t \rightarrow bjj})/2)$  for signal  $t\bar{t}$  Monte Carlo events ( $m_t = 178 \text{ GeV}/c^2$ ). Use for these distributions the best hypothesis.

*Instructions:*

Have a look at `TTbarPraktikum::PrepareHistos()`. The histograms are already initialized. Fill the mass histograms in the routine `TTbarPraktikum::FillRecTop(...)` and write the histograms in the routine `TTbarPraktikum::EndJob(...)` into the root file. The four-momentum of the semileptonically decaying top quark for the  $i$ -th hypothesis is obtained by `ttHyp->P4Top_1lep[i]`. From `ttHyp->idxBestChi2` you know  $i$  of the best hypothesis. After compiling the code and producing the shared library `TTbarPraktikum.so` you have to execute `runTTbarPraktikum.cc` with `proc=600`. Then, you can plot the histograms using the ROOT script `DrawTTbarPraktikum.cc` with the parameter `draw=4`.

### 2.4 Determination of the True Top Mass

Determine the true top mass in the data from the reconstructed mass of the semileptonically decaying top quark. Perform for this a double Gaussian fit (sum of two Gaussians) to the data distribution, where the means  $\mu_1, \mu_2$  and the widths  $\sigma_1, \sigma_2$  of the two Gaussians are parametrized as a function (polynom of first order) of the true top mass  $m_t$ , e.g.  $\sigma_1 = \sigma_1(m_t) = \alpha \cdot m_t + \beta$ . The assumption of a linear dependence of the widths and the means of the double Gaussian on  $m_t$  seems to be a good approximation. In this fit the number of free parameters is reduced to 3 due to this parametrization compared to 6 for the usual double Gaussian fit.

Use the two Monte Carlo samples ( $m_t = 178 \text{ GeV}$ : `proc=600` and  $m_t = 175 \text{ GeV}$ : `proc=603`) to determine the linear dependence of the widths and the means of the double Gaussian on the true top mass  $m_t$ . For the two Monte Carlo samples, perform a double Gaussian fit (six free parameter) to the distribution of the reconstructed

mass of the semileptonically decaying top quark. Compute the gradient  $\alpha$  and the intersection point  $\beta$  to describe the running of the means and widths of the double Gaussian with  $m_t$  by using the fit results obtained for the two Monte Carlo samples. What is the result for  $m_t$  in the data?

*Instructions:*

Execute `runTTbarPraktikum.cc` with `proc=6000,600` and `603`. The code for the double Gaussian fit for the two Monte Carlo samples can be found in the routines `MrecFit(..)` (fit function) and in `TTbarPraktikum::DoubleGaussFit(..)`. The fitting routines are called in the main routine for the top mass determination, namely `TTbarPraktikum::MassFit(..)`. Fit the Monte-Carlo mass distributions using `DrawTTbarPraktikum` with `draw=5`.

The code for the double Gaussian fit for the data sample can be found in the routines `MrecFitData(..)` (fit function) and in `TTbarPraktikum::DoubleGaussFitData(..)`. Extend the code of `TTbarPraktikum::MassFit(..)` in order to fit also the mass distribution of the data. Compute the gradient `alpha[i]` and the intersection point `beta[i]` to describe the running of the fit parameters  $i$  with  $m_t$ . Use for that the predefined variables `alpha[i]` and `beta[i]` and the fit results of the two Monte Carlo samples. In the fit function for the data `MrecFitData(..)` the correct `alpha[i]` and `beta[i]` are automatically used. The fit for the data is called by `DoubleGaussFitData(..)`. Create a canvas for the data and draw the histogram together with the fit result. After producing the shared library `TTbarPraktikum.so` you can rerun `DrawTTbarPraktikum` with `draw=5`.