4C00 Project Log

Daniel Beecher

Supervisor:
Dr. Mark Lancaster

March 25, 2004

1 Week 1

Research into the background of high-energy particle physics. Read through some of the 3C24 notes.

2 Week 2

Created a mock up of the project website. In order for it to be easy to read, the design was to be simple and with correct use of white-space. More research was carried out into the likes of parton distribution functions and basic kinematics.

3 Week 3

3.1 Wednesday, October 8, 2003

First meeting with Dr. Lancaster and the project title was decided upon. Read the DØ paper [1], in order to learn more about the techniques used in actual HEP experimentation; also read up on some techniques used for measuring p_T from experimental data. [2] Also began work on the actual website, replacing the previous mock page.

3.2 Thursday, October 9, 2003

Spoke to Dr. Lancaster in more detail about the project, to aid in writing the project outline. A $P_T(Z)$ form will have a detector simulation run upon it. This simulation will have to take into account various factors which hinder current experiments. This will then be fitted against the CDF data, then the $P_T(W)$ will be fitted against this. With a estimation for the transverse momentum of the W-boson, a more accurate value of M_W can be calculated. Finally, how $P_T(W)$ affects the uncertainty of the boson mass shall be considered, via a comparison with the actual value. Began to write the project outline.

3.3 Friday, October 10, 2003

Continued more work on the project outline. Finally worked out how to access the remote PC in the HEP cluster from the UCL WTS network. Played about with Root as a result.

4 Week 4

4.1 Monday, October 13, 2003

Project outline submitted, and HTML version put on the project website. The project title was changed from the old project to the new one.

4.2 Wednesday, October 15, 2003

The apparatus of the CDF detector was explained to the group in the weekly meeting with Dr. Lancaster. How to use SunONE/Forte[1] on the HEP machines was also shown. The work on the project is delayed until the Mark finished the sample Root file. Just what information the Root files create was explained, and how the data will be presented.

5 Week 5

5.1 Monday, October 20, 2003

Spoke to Dr. Lancaster about the delayed Root file. He is having trouble getting the JAS[2] classes working properly. While they came with a full documentation, he was having trouble understanding what the methods actually did.

He has written some C++ code that generates a histogram from a root file. He says the parts of the code that should be written are very similar to Java. The only part that was C++ specific was a section used to define memory locations, this would not change, so it can be defined elsewhere. All the program did was select energy data for all the particles listed as electrons, a few simple if loops.

Rather than work on the 'ugly' code , he will tidy it up for the meeting on Wednesday.

5.2 Wednesday, October 22, 2003

Dr Lancaster informed us all in the meeting that he has written some C++ code that would open a root file to allow analysis of it. This will allow histograms to be generated for various properties. Dr. Lancaster also went to show what he uses to generate the root files—a Monte Carlo simulation, HERWIG[3] (Hadron Emission Reactions With Interfering Gluons). The aim is to generate some useful histograms before the weekend using the code.

A greater understanding of what the project entails was reached, and what work is needed in the future.

- An analysis program needs to be developed to measure the transverse momenta of the $e^{\pm}$, and other properties of the decay of the Z^0 from $p\bar{p}$ collisions. This will be simulated using HERWIG.
- The theoretical models/distributions do not compare with experimental data because of limitations due to the detectors—these are to be modelled and a program will be written to 'smear' the HERWIG estimations.
- After the simulations side of the project is complete, work will need to begin on the experimental side. Data from the CDF [3] gives the measurements the detectors make. The particles involved in the reaction will have to be reconstructed from the experimental signatures. This will allow plots of actual data for the $Z^0 \to e^+e^-$ processes and comparison with the smeared HERWIG data.
- At this point a $P_T(Z)$ distribution should exist for both the simulated and actual data; the simulation then should be modified until it matches the actual data, giving a final $P_T(Z)$ distribution. This will allow a $P_T(W)$ to be created. The $P_T(W)$ will be used to estimate the mass of the $W^{\pm}$, and compared with the accepted mass, 80.450 GeV.

---

[1] SunONE, formerly Forte, is a freeware Java integrated development environment provided by the creators of Java, Sun Microsystems

[2] JAS is an collection of Java classes to use Root files.

[3] http://hepwww.rl.ac.uk/theory/seymour/herwig/

5.3 Thursday, October 23, 2003

12:00pm–2:00pm

Produced some sample histograms using the analysis framework written by Dr. Lancaster. While most of the code is illegible to those not versed in C++, it does not need to be changed. However, the number of histograms to be produced, and the techniques used to extract the data—from the HERWIG root file—have to be defined.

$$P_T = \sqrt{P_x^2 + P_y^2} \tag{1}$$

The code written by Dr. Lancaster checked for the $e^{\pm}$ with the highest P_T using equation (1)—See figure 1 for the resulting histogram.

The program also calculates the angle of scatter for the $e^{\pm}$ with the highest P_T, using equation (2), but this was later found to be incorrect, as results show (ref:figure 2).

$$\theta = \arccos\left(\frac{P_z}{\sqrt{P_x^2 + P_z^2 + P_z^2}}\right) \tag{2}$$

With the analysis program working correctly, the transverse momenta of both the electron and the positron is to be produced for the current HERWIG simulation before the next meeting. The pseudorapidity, η, is also be calculated, equation (3). This quantity is more useful than the scattering angle because it is an invariant when converting between the centre-of-mass frame and the lab frame.

$$\eta = -\ln\tan\frac{\theta}{2} \tag{3}$$

5.4 Friday, October 24, 2003

11:30am–4:00pm

After consideration, the formula used to calculate the scattering angle, equation (2), was actually correct, the problem may lie in how Dr. Lancaster wrote the code[4]. Spent time correcting the histograms produced yesterday so that they are more suitable for printing—Root saves them with a grey background, rather than a white background. The mouse commands used to achieve this can be saved as C macro, allowing automation of the graph formatting, but these are still to be learnt.

$$\eta = -\ln\tan\left[\frac{1}{2}\arccos\left(\frac{P_z}{\sqrt{P_x^2 + P_y^2 + P_z^2}}\right)\right] \tag{4}$$

The next set of histograms to produce will show the P_T of the highest P_T electron and positron, and well as the corresponding η, using equation (4).

The root file produced as a result was saved as `031024_z_pt_eta.root`. The labelling system shows the date the analysis was carried out[5] (for cross-referencing to this log) and the properties investigated. The histograms are as expected (see figures 3–6).

6 Week 6

6.1 Monday, October 27, 2003

14:00pm–15:00pm

Decided to look at how the P_T of the actual Z-boson. This shall be done by plotting the absolute difference between the $P_T(e^-)$ and the $P_T(e^+)$. The data was saved as `031027_z_pt_eta.root`.

[4]See `ExampleJob.cc`, dated 23/10/03

[5]'03' is for 2003, '10' for October, '24' is the date.

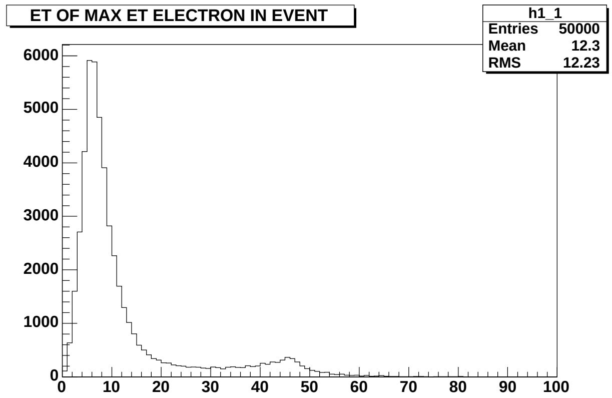

Figure 1: This is a sample histogram produced by Root.

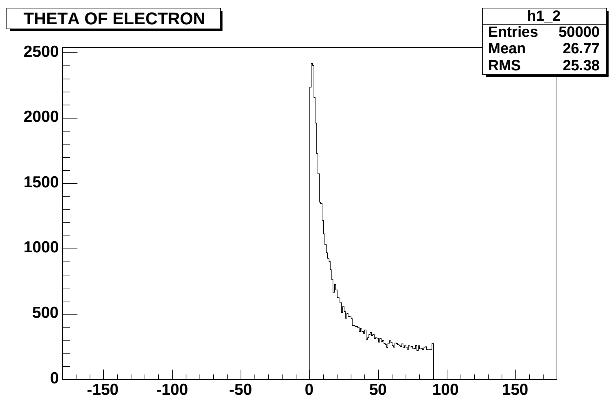

Figure 2: Another sample histogram, however the results are incorrect due to an error in the `ExampleJob.cc` file.

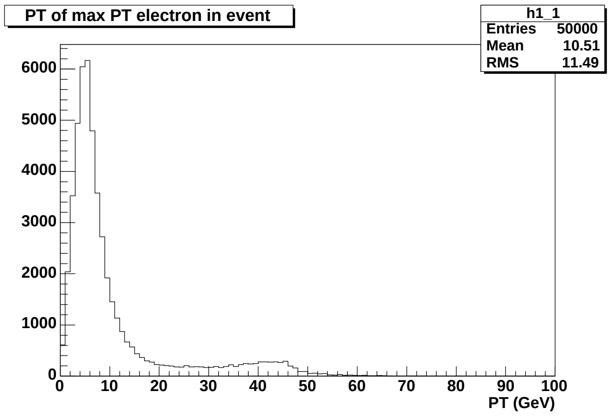

Figure 3: The P_T distribution for the highest P_T electron.

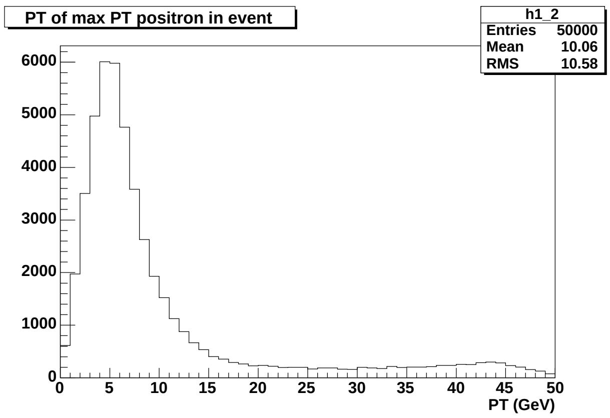

Figure 4: The P_T distribution for the highest P_T positron.

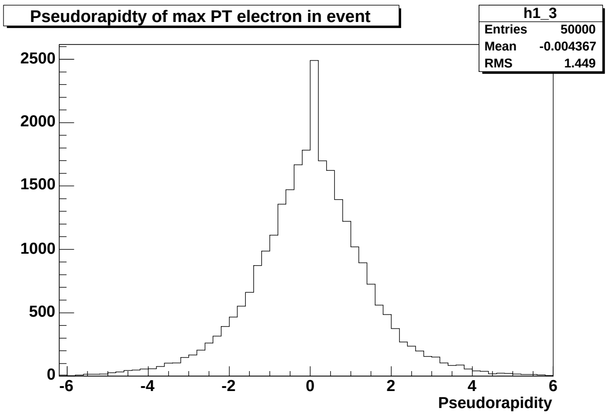

Figure 5: The η distribution for the highest P_T electron.

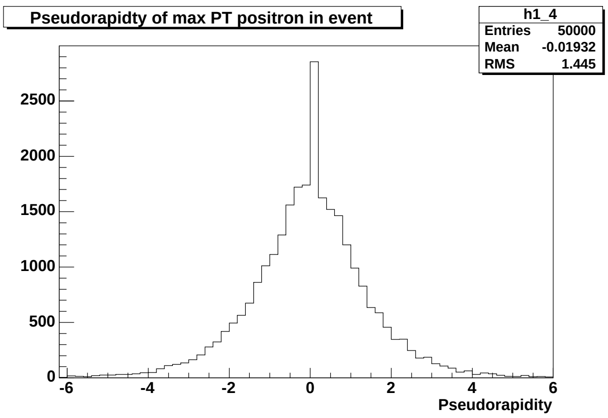

Figure 6: The η distribution for the highest P_T positron.

6.2 Wednesday, October 29, 2003

10:00am–2:45pm

Was told that the data/histograms that had produced were incorrect in the weekly meeting . Dr. Lancaster explained that the P_T distributions for the $e^{\pm}$ should show a higher transverse momentum, with an edge at half the mass of the of the Z-boson. All of the graphs will need to be recreated for the Z-boson, and for the W-boson as well.

$$\left(\frac{\Delta E}{E}\right)^2 = \frac{(0.135)^2}{E_T} + \kappa^2 \tag{5}$$

After the problems with the current plots are corrected, the direction the project would move in was discussed. The next step is to introduce a detector effect, the energy resolution of the CEM calorimeter, equation (5). This factor is supposed to be correct for the whole calorimeter, but each module has its own characteristic errors—the extra term is a correction to take this into account. However, the actual value changes frequently and will need to be found out. This effect occurs for the CEM, and as a result, the pseudorapidity range will decrease to $|\eta| \leqslant 1$.

The energy smearing will be coded into the analysis program, to smear the HERWIG data. Once this has be done, the next task will be to find a functional form for this new energy distribution (with as few variables as possible).

First, the current problems need to be solved. Rather than re-write all the code in the `ExampleJob.cc`, I will comment out the Z-boson specific histograms, and run the program using the $W^{\pm}$ HERWIG root file (just change the input file in 'steering.cards'). The data was saved as `031029_w_pt_eta.root` and the resulting plots are figures 7–12. Next, the same code was run using the Z^0 root file, and saved as `031029_z_pt_eta.root`, but the plots were the same as before (figures 3–6). Obviously, if the code works correctly for the $W^{\pm}$, but not the Z^0, then there must be a problem with the root file Dr. Lancaster produced from HERWIG. I informed him of the problem, and will have to wait for it to be corrected before I can continue with any Z-boson analysis.

The W-boson plots were correct, however. The P_T distributions for the e^- (figure 7) and e^+ (figure 8) both show a peak at approximately half the momentum) of the W-boson. They also both exhibit the characteristic 'edge' just after the peak. The graphs have this shape because, on average, the resultant electron (for W^- decay) or positron for W^+ decay) take half the energy, with the neutrino having taken the rest.

The pseudorapidity plots (figures 9–10) are harder to interpret. They are both centred about $\eta = 0$, which is reasonable, the emitted electron or positron will be emitted perpendicular to the colliding beams. That they take the form of a Dirac-delta function is worrying. A minor search on the internet for data on the pseudorapidity of the emitted $e^{\pm}$ proved to be unsuccessful, so I will need to ask Dr. Lancaster what the correct results should be.

The energies of the $e^{\pm}$ from the W boson events should also be looked at. To do this a small section of code was inserted into the `ExampleJob.cc` file to store the energy associated with that particular particle in the event (the code is already identifying the particles it is interested in). The code was run and the results stored in the `031029_w_pt_eta.root` file. As can be seen from the data, figures 11 and 12, both the electron and the positron show the correct energies (half the energy of the W-boson most of the time)

3:00pm–4:00pm

After consultation with Dr. Lancaster, it is obvious that there is a problem with the pseudorapidity data—a bug in the code. The first place to look would be in the formulae, but they are correct (after careful checking). In the end, the problem was with my actual code. When defining the pseudorapidity variable, '0.0' was used as a default value. This proved to be a problem, because the value was written into the histogram if the particle checked was neither a electron or positron. The solution was to change the default value so that it was outside the bin range of the data. The correct data plots were produced, figures 13–14.

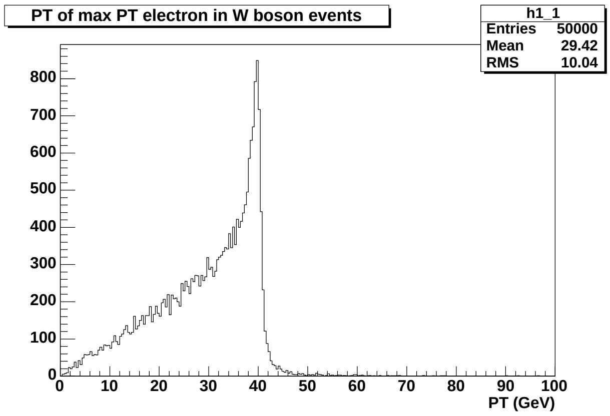

Figure 7: The P_T of the maximum P_T electron in the decay of the W-boson.

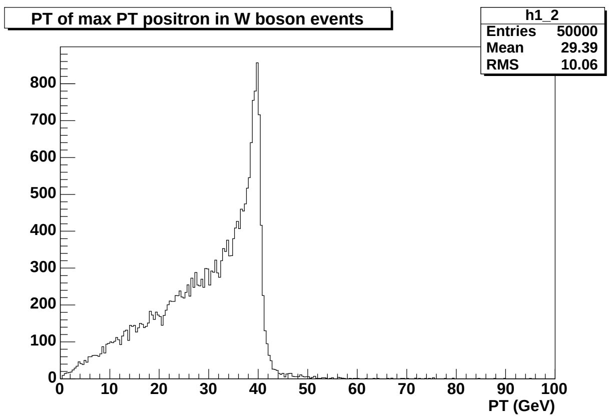

Figure 8: The P_T of the maximum P_T positron in the decay of the W-boson.

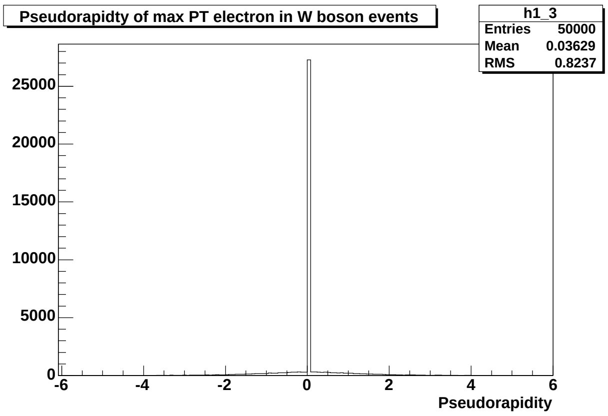

Figure 9: The η distribution of the maximum P_T electron in the decay of the W-boson.

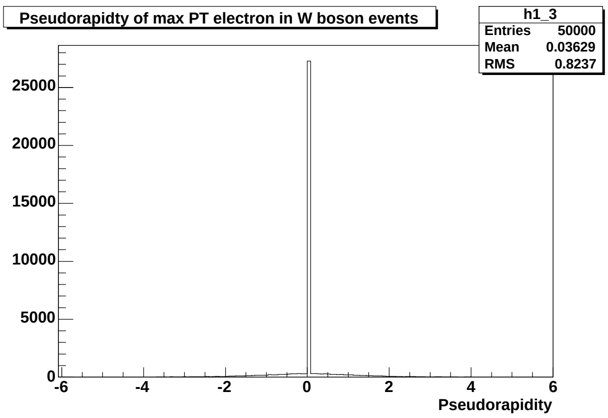

Figure 10: The η distribution of the maximum P_T electron in the decay of the W-boson.

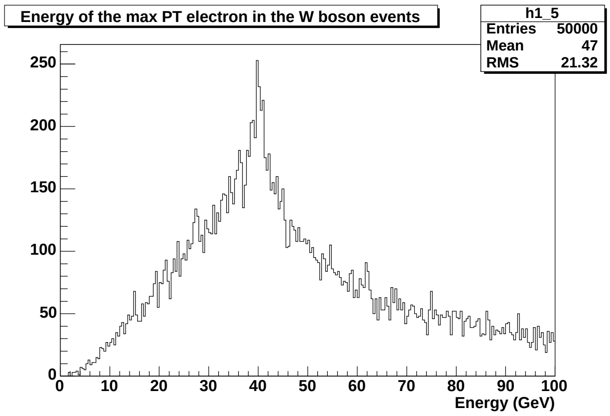

Figure 11: The energy of the maximum P_T electron in the decay of the W-boson.

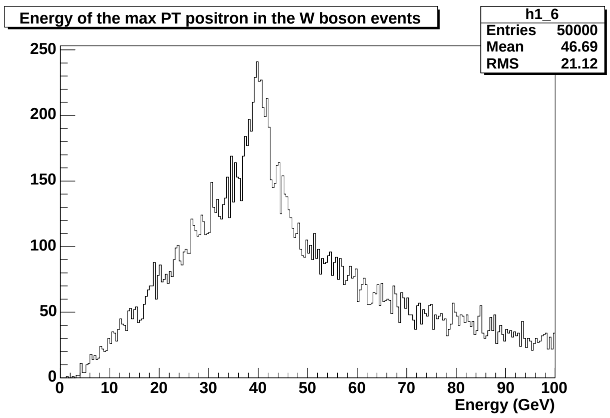

Figure 12: The energy of the maximum P_T positron in the decay of the W-boson.

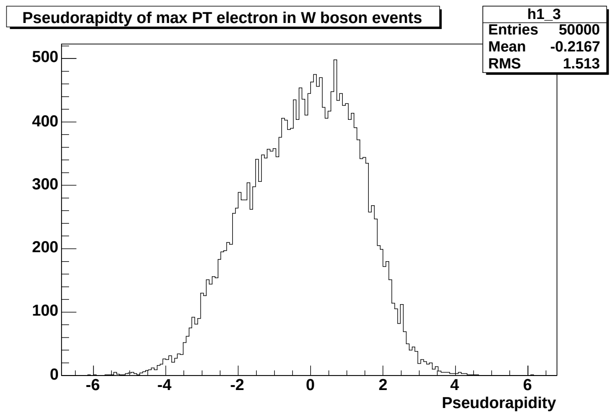

Figure 13: The η distribution of the max P_T electron in the decay of the W-boson.

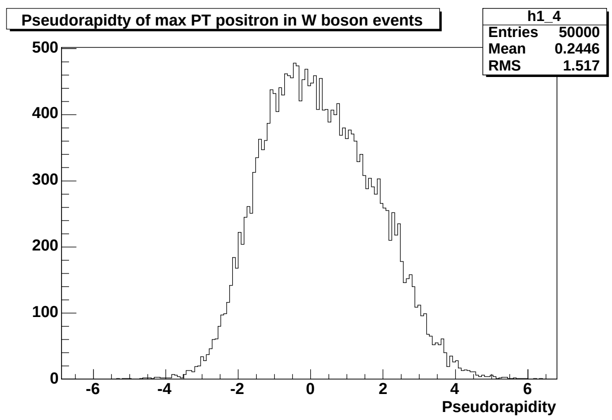

Figure 14: The η distribution of the max P_T positron in the decay of the W-boson.

6.3 Thursday, October 30, 2003

12:45pm–1:30pm

The correct pseudorapidity distributions (figures 13–14) both show, that in general, the emitted lepton is perpendicular to the path of the colliding $p\bar{p}$ beams. As expected (unlike the previous data) there is a width to the plots. The also seems to be an asymmetry to the individual plots, with the electron data having a larger 'tail' on the negative side, and vice versa for the positron. Having information about the neutrino would be helpful in trying to interpret what the differences in the plots mean—the e^- and e^+ being identical, excepting charge, so one would expect the data to be the same. Although, the W-boson has an asymmetry, the W^+ and W^- produce different results. This might explain why there seems to be a pseudorapidity bias in my data, but η data for the $W^\pm$ would need to be examined. I looked at Will's data[6] and there is an asymmetry to the W boson pseudorapidity, which could explain the differences between fig 13 and fig 14.

6.4 Friday, October 31, 2003

6:15pm–7:00pm

Updated the website with the working PDF documents.

7 Week 7

7.1 Wednesday, November 5, 2003

10:15am–3:00pm

Dr. Lancaster informed that the HERWIG file for the Z^0 had been corrected (he had actually sent an email on the 31th October, but I had not checked my email recently) and that there was little extra for me to do as I had not yet generated any proper histograms for the Z boson. This is to be rectified today. He did explain about how to apply the energy smearing in more detail.

$$E' = E + R\Delta E \tag{6}$$

(6) shows what effect (5) will have on an input energy. R is a random number with a Gaussian distribution (mean=0, width=1). Was also encouraged to investigate what effect κ on the energy smearing. If $0.01 \leq \kappa \leq 0.03$ is insignificant compared with the natural uncertainty in the energy, then it does not need to be calculated exactly. Should start with $\kappa = 0.025$.

The histograms for the Z^0 need to be generated. The same code that was used to generate the data for the W-boson will be used on the Z-boson, using the `Zee_feynman_only_50k.root`. The data will be saved as `031105_z_elec.root`. Initially, the range of energies in the plots were too small (0–100 GeV), so the histograms will be re-plotted with a larger energy range (0–500 GeV).

The P_T of the electron, figure 15 has a similar form to the same event in the W boson case (figure 7), as would be expected. The 'edge' of the data indicate a cut-off point at approximately half the mass of the Z_0.[7] This is the same for the positron distribution, figure 16.

The pseudorapidity plots, figures 17 and 18, are pretty much identical—both centred about zero and the distribution is zero at $|\eta| > 5$. This indicates that there are very few emitted $e^\pm$ at low angles, and the majority are emitted perpendicular to the colliding $p\bar{p}$ beams.

Both the energy distributions, figures 19–20, show a modal energy of half the Z^0 energy; on average the electron and positron carry the same amount of energy.

Dr. Lancaster mentioned a Root class that might prove helpful in my analysis, `TLorentzVector`. It can store (P_x,P_y,P_z,E) as a 4-vector, and, hopefully, it has details like the pseudorapidity as in-built methods.

[6] Will is investigating the effect that parton distribution functions have on the mass of the W boson, but he had produced η distributions using the same HERWIG file as I had - http://www.hep.ucl.ac.uk/∼reece/tex/notebook/week5/week5.pdf

[7] It should be apparent that the atomic unit for momentum is eV/c, but with c=1, this is just eV; the same applies to the mass, eV/c^2 becomes eV.

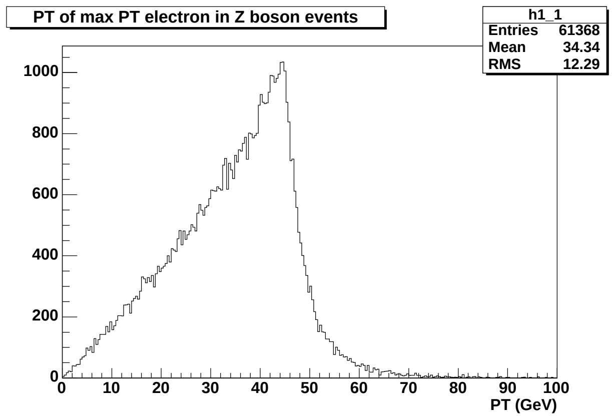

Figure 15: The P_T distribution for the highest P_T electron in the decay of the Z^0.

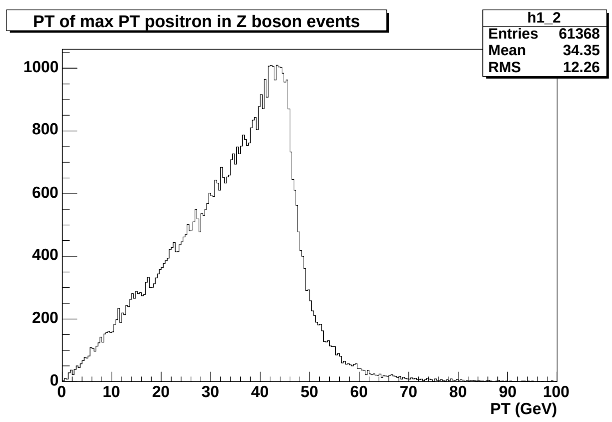

Figure 16: The P_T distribution for the highest P_T positron in the decay of the Z^0.

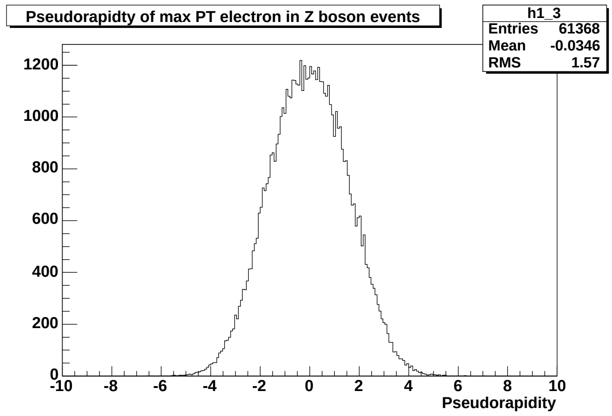

Figure 17: The η distribution for the highest P_T electron in the decay of the Z^0.

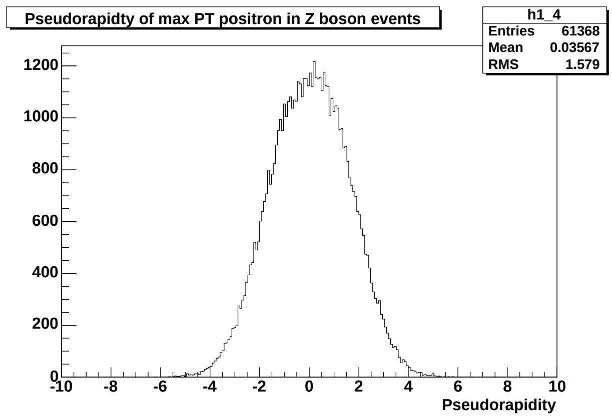

Figure 18: The η distribution for the highest P_T positron in the decay of the Z^0.

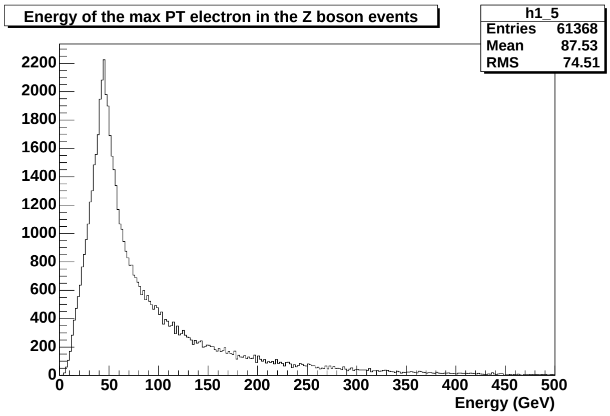

Figure 19: The energy distribution for the highest P_T electron in the decay of the Z^0.

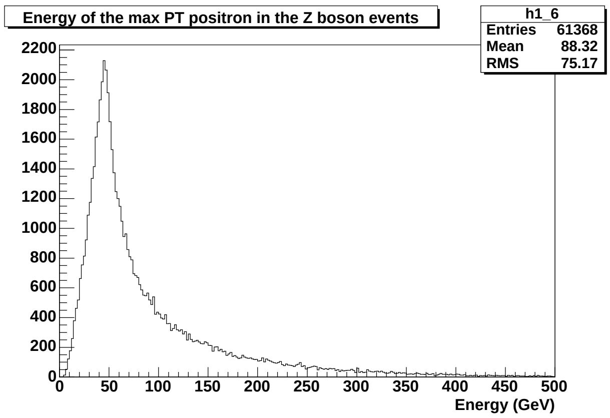

Figure 20: The energy distribution for the highest P_T positron in the decay of the Z^0.

The Root help files[8] indicate that such methods exist, but whether they will work is not clear. The only way to find out would be to re-write the current `ExampleJob.cc` code but using the `TLorentzVector` methods and comparing the results with my the plots generated today.

At the moment the C++ code is doing its best not compiling for some unknown reason, and Dr. Lancaster to ask him about how to implement the class he suggested so research into the generation of a Gaussian random number was carried out instead. A method [9] of converting a group uniform random numbers into a Gaussian distribution was found.

According to Dr. Lancaster, my problem lay in how the `TLorentzVector` object was initialised, so the code changed to use the default constructor and set the components with a different method. This yielded the same error when it tried to compile. Rather than waste too much time on it, one will leave the code as is, with the possibility of writing a C++ class to do the same job (if one has gathered a large enough understanding of the language).

7.2 Thursday, October 6, 2003

11:30am–1:45pm

$M_{0,Z}$ distribution needs to be investigated before the energy resolution effect into the analysis. The invariant mass (rest mass) can be calculated using (7).

$$m_0 = \sqrt{\frac{E^2}{c^4} - \frac{p^2}{c^2}} \tag{7}$$

One is unsure whether to calculate Z boson data using the boson information in the HERWIG file, or to calculate it from the properties of the decay products. For the moment, the decay products should be used, given that it is impossible to analyse the Z^0 directly in an experiment.

However, no data is being recorded in the histogram. It may be a problem with using atomic variables (i.e. setting constants such as c to 1), but if is changed c to be 1 in my code then the $M_{0,Z}$ value is around zero, which is clearly incorrect.

5:20pm - 6:00pm

Managed to get the 4-vector class to compile (after speaking to Dr. Lancaster, and realising that the class should be included at the top of the page:

```
#include 'TLorentzVector.h'
```

I was not able to check the data in Root to make sure the in-built methods were functioning properly, but that will be done tomorrow morning. (I was unable to check in Root because I was using a computer where I had only a text display to work with, and Root needs a graphical display).

7.3 Friday, October 7, 2003

1:00pm–4:25pm

The new code using the 4-vector class needs to be checked to make sure that it works. The histograms are the same as the ones generated using formulae worked out. The 4-vector class in the future due to the more elegant code that results. There is one difference, the $P_T(Z)$ distribution is now the correct shape, so the problem must have lain in how the code/formulae was written; this is yet more evidence for sticking with the 4-vectors. However, there is a small peak at a P_T of zero. This should not be the case because it is very unlikely that it would happen, so the peak is due to a bug in the code, binning data at zero when it should not be.

Having looked at various properties of $Z^0 \rightarrow e^+e^-$, it is time to finally build the energy uncertainty into the code. Dr. Lancaster told me that there was a function which generates a random number from a

[8] The help files can be found at http://root.cern.ch/root/htmldoc/ClassIndex.html

[9] http://www.taygeta.com/random/gaussian.html

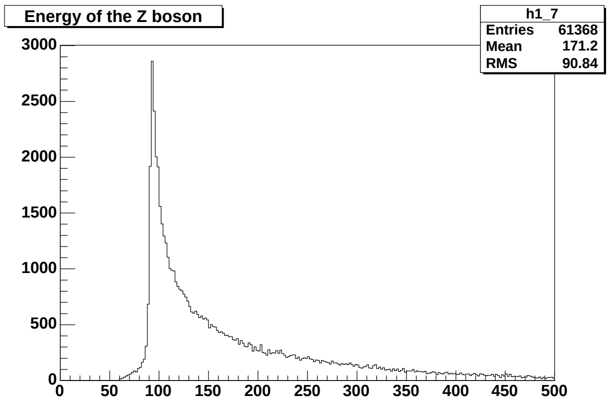

Figure 21: The energy distribution of the Z^0.

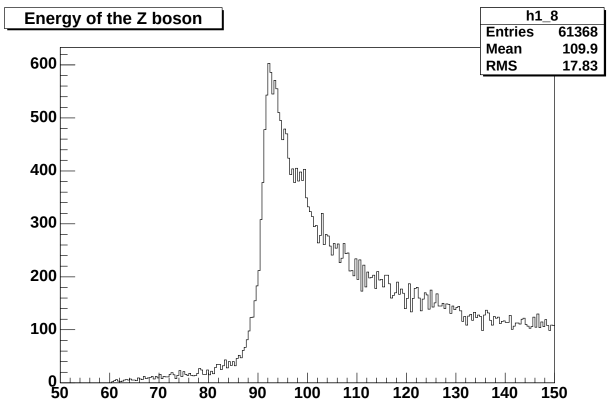

Figure 22: The energy distribution of the Z^0 over a smaller range to illustrate the peak energy more clearly.

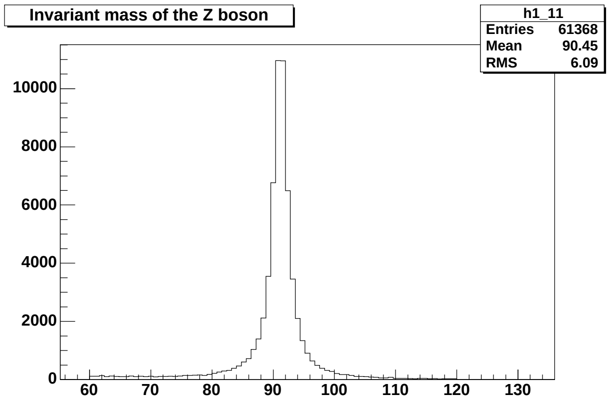

Figure 23: The invariant mass of Z^0 according to the HERWIG data.

Gaussian distribution (it defaults to having a mean of 0 and a width of 1). It may be something like `grand`, but one is not sure. But the appearance of the bug in the binning of the Z boson, yet again, halts progress on the energy resolution.

(4:00pm)

One is no closer to identifying the problem with the binning of the data. Changing then number of the bins serves to show that the same number of events are being placed in the zero bin, approximately 1200 events. Why they are there cannot be determined. Applying a cut to the data would remove the peak, but there would be a small amount of events with a $P_T(Z)$ of zero, so such crude measures cannot be used; the problem must be discovered before continuing, before the code becomes more complicated.

(4:20pm)

Still cannot find the solution to the problem. The code has been emailed to Dr. Lancaster for him to have a look at; the problem cannot be found and continuing is just wasting time at the moment.

8 Week 8

8.1 Tuesday, November 11, 2003

(4:50pm)

Received an email from Dr. Lancaster about the code sent to him. He does not think that it is a bug/problem on my part, more to do with the actual HERWIG data. He suggests reducing the number of bins to 50 and running in the range 0–50 GeV; this will produce correct looking histograms.

For the past few days one has been trying to get Root running on my computer at home (using Cygwin), and have now decided to install RedHat so that one can work on the project outside of the university.

8.2 Wednesday, November 12, 2003

(8:00pm)

RedHat Linux is running on my PC at home, and a version of Root (v10.01) has been installed to allow the analysis of data at home. Hopefully one will able to transfer the analysis framework onto the machine, allowing me to work from home, rather than use the slower machines at UCL (while the HEP machines are fast, the UCL WTS is hideously slow and cumbersome).

8.3 Thursday, November 13, 2003

(4:30am)

Managed to get all the files needed to run the C++ analysis program[10], but could only execute this file; attempting to recompile the code brought up a host of errors. It would seem that some of the custom classes used are needed:

```
#include "ExampleJob.hh"
#include "HepEvtBlock.hh"
#include "RootFile.hh"
#include "AnalysisSteer.hh"
#include "Logger.hh"
#include "Histos.hh"
```

These files are missing from my computer, will need to explore Dr. Lancaster's workspace to find them, before finding out what else is wrong.

1:00pm–1:50pm

Looking inside Dr. Lancaster's data analysis directory[11], the missing were found and copied all the relevant files—not all the files could be copied as one did not have the relevant permissions. Putting these in a directory on my PC and changing the link in the `makefile` to point to there should work.

Tested Dr. Lancaster's theory about how the data will look if the change the bin size is changed; two new histograms were created, one with 50 bins in the range 0–50 GeV, and one with 250 bins in the range 0.2–50 GeV (ignoring the data at zero), figures 24 and 25 respectively. Figure 24 does, indeed, mimic the expected distribution, but it is very 'blocky' and coarse. Ignoring the erroneous data at zero, figure 25, results in data similar to the expected distribution. One would personally prefer to use the latter technique to solve the problem, but will ask Dr. Lancaster about which he thinks would be preferable to use, whether my idea of ignoring zero is viable, or not.

8.4 Friday, November 14, 2003

6:20pm–6:50pm

Having copied the files forgotten yesterday, one is going to attempt to edit my local directories and files so that the `ExampleJobAnalysis` can be compiled locally. Copying the contents of Dr. Lancaster's directory next to my data analysis directory, and edited the `Makefile`—not being well versed in C++, one believes that the contents of this file control how the C++ files in the directory are compiled. There is a link to Dr. Lancaster's directory in the file:

```
LIB_OBJS = ControlCards.o Logger.o  Histos.o Destroyer.o
  RootFile.o AnalysisSteer.o HeaderBlock.o HepEvtBlock.o
  THBook.o ElectronAnalysis.o
TEST_OBJS = ExampleJob.o ML_4C00 =  /home/Dr.
Lancasterl/4C00_analysis
```

[10] `ExampleJobAnalysis.exe`

[11] `/home/Dr.  Lancasterl/4C00_analysis/` on the HEP machines

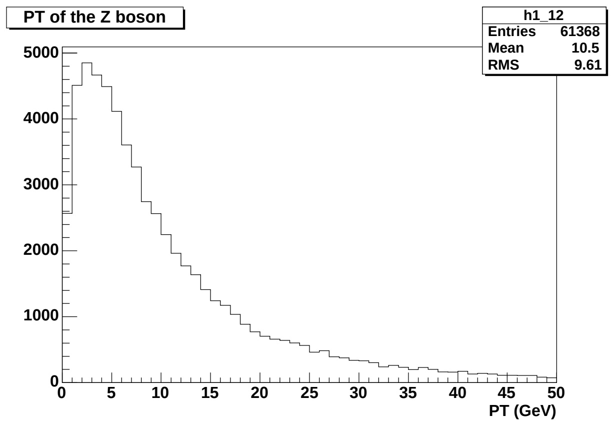

Figure 24: The P_T distribution of the Z boson using only 50 bins in the range 0–50 GeV

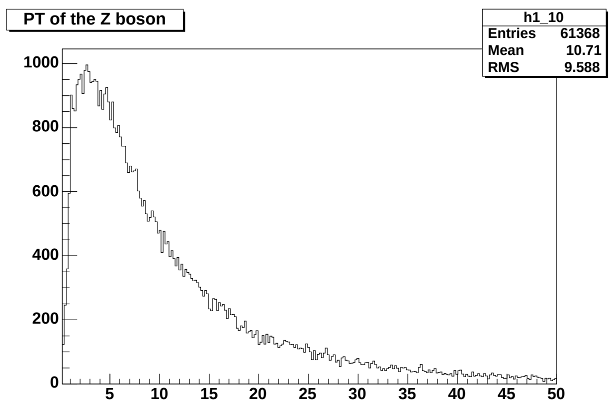

Figure 25: The P_T distribution of the Z boson using 250 bins in the range 0.2–50 GeV

Changing this to point to the relevant directory on my machine does not help matter, the compiler still cannot find the files to include. But after looking through the directory the mistake is found. Linux is case sensitive, whereas Windows is not. Copying the files from the HEP machine to the WTS machine being used, then copying those files to the zip disk[12] resulted in Windows converting all the filenames to lowercase. After changing the filenames back to what there are supposed to be, the code now compiles and one can work from home, rather than the sluggish WTS.

8:45pm–10:30pm

According to my earlier meetings with Dr. Lancaster, the energy resolution of the CEM in the CDF detector is defined by equations (5) and (6). Combining the two equations leads to an actual equation that can used to create z bosons with smeared energies:

$$E(\kappa) = E\left(1 + R\sqrt{\frac{0.135^2}{E_T} + \kappa^2}\right) \tag{8}$$

One is not sure about the transverse energy, E_T (assuming that it is the energy in the transverse plane), for the moment one shall assume that the proportion of energy in the transverse plane is the same as the proportion of momentum in the said plane, thus:

$$E(\kappa) = E\left(1 + R\sqrt{\frac{0.135^2 P}{EP_T} + \kappa^2}\right) \tag{9}$$

.

In both equations (8) and (9), R refers to a random number, with a Gaussian distributions centred about zero with a width of 1.

(10:20pm)

The idea to generate histograms automatically from 0.01–0.03 is not going to plan. After fighting through the inevitable bugs and problems with the casting of variables, the code compiled, only for a run-time error to spoil things. For the record, the program said:

```
>>INFO:Nov 14 22:16:10:main: Running on 61368 Events

 *** Break *** segmentation violation
 Generating stack trace...
 0x08059457 in main + 0x13a3 from ./ExampleJobAnalysis.exe
 0x42015574 in __libc_start_main + 0xe4 from /lib/tls/libc.so.6
 0x08058025 in TFile::TFile[in-charge](char const*, char const*,
char const*, int) + 0x31 from ./ExampleJobAnalysis.exe
Aborted
```

This does not mean much to me, apart from knowing that whatever `TFile` is, or does, it does not like what I have given it.

9 Week 9

9.1 Monday, November 17, 2003

(12:45pm)

Spoke to Dr. Lancaster about how to call a random Gaussian number (had been working with a uniformly distributed random number to get the code working), he showed the function that will generate it:

[12] I am not aware of how to mount a zip drive onto a computer I am using SSH to access.

```
gRandom->Gaus()
```

This would generate a random number from a Gaussian distribution with a mean of zero and a width of one. Also found out how to superimpose histograms:

```
TF1->Draw("SAME")
```

When drawing the actual histogram, rather than using the `->Draw()`, using `->Draw("SAME")` will plot the histogram over the last one plotted. This should come in handy when comparing the κ plots.

(1:45pm)

Yet another problem has arisen, the code that Dr. Lancaster supplied me to generate a random number does not work on my PC. First guess is that this must be a function missing. There are a few possible solutions: delve deeper into the HEP accounts to look for the version of compiler they use, or any files needed (like what one did with the original missing files); download and install a random number generator for use at home; work solely at UCL, which is not really an option. Either way, no more work can be done until this is overcome.

4:45pm–7:15pm

The reason that the code would not compile with `gRandom->Gaus()` is because the method is specific to Root, and I had not included the relevant file header. So after adding `#include "TRandom.h"` the file would compile.

The $P_T(Z^0)$ plots for $0.01 \leq \kappa \leq 0.03$ shall now be generated. My code will produce histograms for $P_T(Z)$, but not E(Z); the bug is being looked into.

(7:10pm)

It was not a bug in the code, but data being binned outside the range of the histogram. There might be a problem; all the histograms are either identical (the P_T plots) or extremely similar (the energy plots). One will need to speak to Dr. Lancaster tomorrow to see what is supposed to be smeared by the energy resolution and by what amount.

(10:30pm)

The formula for the application of the energy smearing, (9), may be incorrect; a mistake with the calculation of E_T. This is how to find the transverse energy:

$$\begin{aligned} E^2 &= m_0^2c^4 + p^2c^2 \qquad (10) \\ &= m_0^2c^4 + \left(p_x^2 + p_y^2 + p_z^2\right)c^2 \qquad (11) \\ \therefore E_T &= c\sqrt{p_x^2 + p_y^2} \qquad (12) \end{aligned}$$

9.2 Tuesday, November 18, 2003

7:00pm–8:00pm

In order to see whether the implementation of the energy resolution is correct, the normal and smeared distributions for the $P_T(Z)$, $E(Z)$ and $m_0(Z)$ will be plotted against one another.

(7:50pm)

Individual plots for the factors listed above can be produced, I am working on putting them on the same histogram. It appears to be too complicated to plot over existing data at moment, until research further into the Root classes shows the way to do it.

Figures 26 and 27 are identical, obviously there is something wrong with my application of the CEM effect ct to the P_T, it will be investigated later. Figures 28 and 27 show the energy distributions of the Z^0 for a no smearing case and the $\kappa = 0.025$ case respectively. There smeared data does have a broader distribution, as expected, but it is not that easy to see, Laying both plots on the same histogram to illustrate the differences more clearly. The smeared data is more broad due to the greater uncertainty. Figures 30 and 31 illustrate how the energy smearing affects the data more clearly, but the reasoning behind the broadening is the same as for the energy broadening.

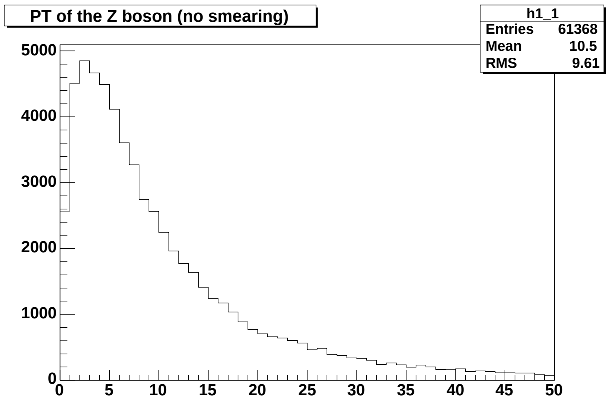

Figure 26: The P_T distribution of the Z boson with no smearing of the energy.

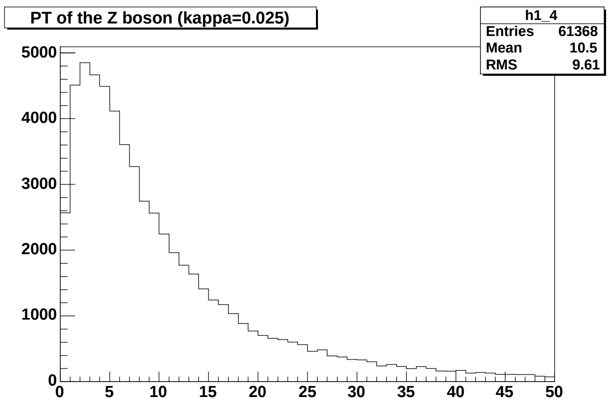

Figure 27: The P_T distribution of the Z boson with the CEM resolution effect applied ($\kappa = 0.025$)

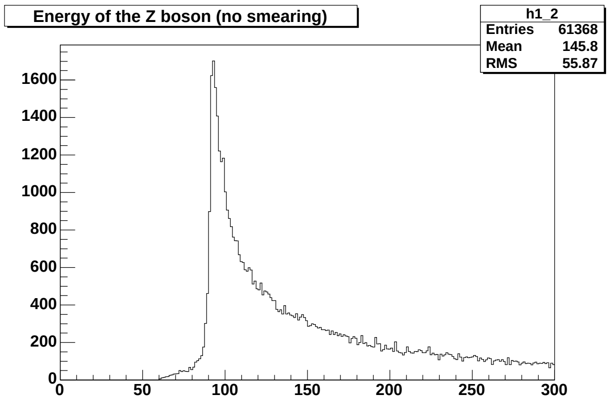

Figure 28: The energy distribution of the Z boson with no smearing of the energy.

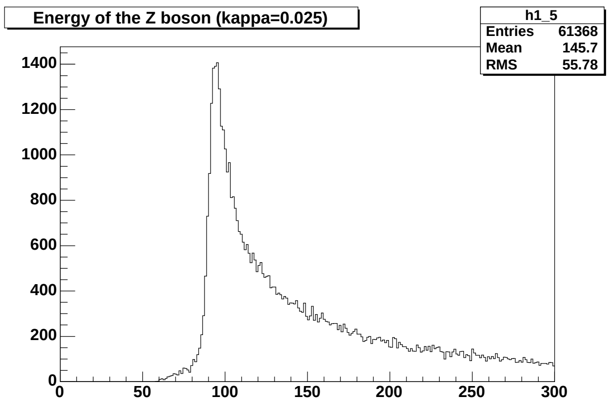

Figure 29: The energy distribution of the Z boson with the CEM resolution effect applied ($\kappa = 0.025$)

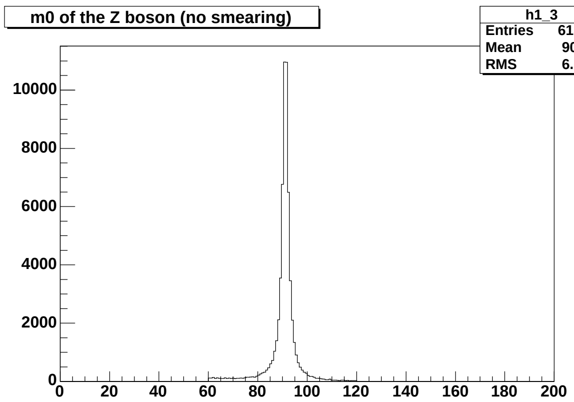

Figure 30: The m_0 distribution of the Z boson with no smearing of the energy.

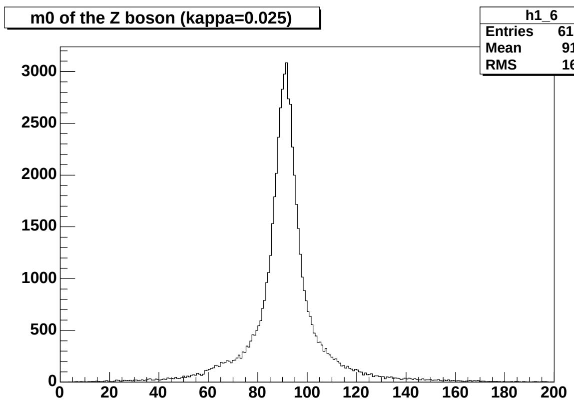

Figure 31: The m_0 distribution of the Z boson with the CEM resolution effect applied ($\kappa = 0.025$)

9.3 Wednesday, November 19, 2003

10:00am–11:30pm

A few things to check up on were discussed in the meeting:

- The reason that my P_T data was not smearing was because the CEM effect was not applied correctly. Therefore the uncertainties in P_x, P_y and P_z will need to be calculated.
- The energy smearing can checked by plotting a graph of $(E - E_{smear})/E$, which should be a Gaussian.

To find the uncertainty in the particle momenta:

$$\begin{aligned} p_x &= p \sin\theta \cos\phi & (13) \\ p_y &= p \sin\theta \sin\phi & (14) \\ p_z &= p \cos\theta & (15) \\ p &= \sqrt{E^2 - m_o^2} & (16) \end{aligned}$$

Assuming that the uncertainty of an function $f = f(a, b, \ldots)$ is :

$$(\Delta f)^2 = \left(\frac{\partial a}{\partial f}\right)^2 (\Delta a)^2 + \left(\frac{\partial b}{\partial f}\right)^2 (\Delta b)^2 + \cdots \quad (17)$$

Thus, the particle momenta uncertainties:

$$\begin{aligned} (\Delta p)^2 &= \left(\frac{dp}{dE}\right)^2 (\Delta E)^2 & (18) \\ &= \left(\frac{E}{\sqrt{E^2 - m_0^2}}\right)^2 (\Delta E)^2 & (19) \\ \therefore \Delta p &= \frac{E\Delta E}{\sqrt{E^2 - m_0^2}} & (20) \\ & & (21) \end{aligned}$$

At the moment, there is no angular smearing:

$$\frac{dp_x}{dp} = \frac{dp_y}{dp} = \frac{dp_z}{dp} = 1 \quad (22)$$

$$\therefore \Delta p = \Delta p_x = \Delta p_y = \Delta p_z = \frac{E\Delta E}{\sqrt{E^2 - m_0^2}} \quad (23)$$

2:00pm–8:00pm

The momenta smearing (23) will be applied to the CEM effect. In addition, those particles which do not satisfy $|\eta| \le 1$ will be ignored, this is because the CEM detects in this range, and the effect should only be applied in this region.

(7:30pm)

Histograms have been generated for the CEM effect, comparing the original data with $\kappa = 0.025$ data (figures 32–34), and comparing $\kappa = 0.01$ with $\kappa = 0.03$ data (figures 35–37). No cuts were used in the end because number of particles left dropped to a sixth, and there was little noticeable smearing from the CEM resolution[13]

[13] There are other energy resolution effects for other pseudorapidities, when these are added to the analysis I can begin to employ cuts on the data.

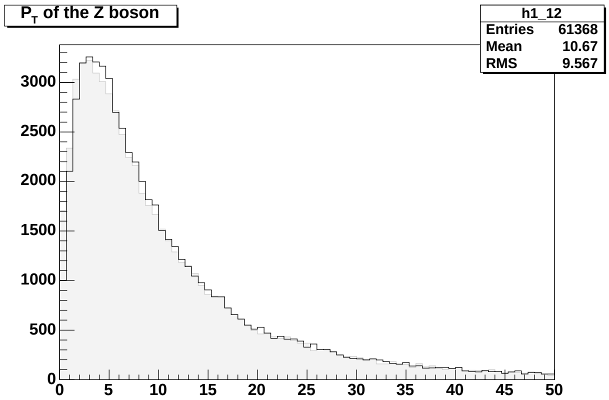

Figure 32: The P_T distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa = 0.025$. No cuts have been made.

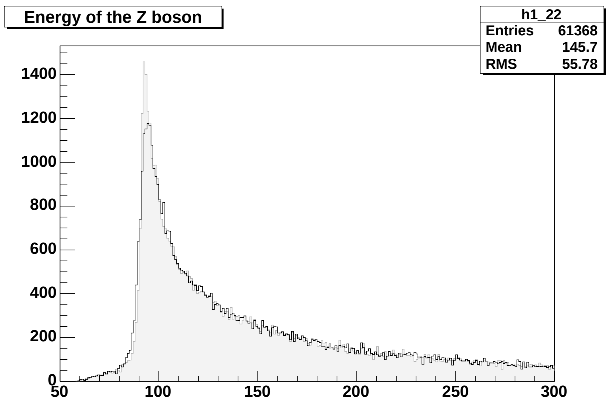

Figure 33: The energy distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa = 0.025$. No cuts have been made.

The affect of the smearing on the P_T of the Z^0 is still not that noticeable, figure 32. which one will want to talk with Dr. Lancaster about; if this is not what should happen, then there is either something wrong with my data analysis, or my implementation. Although, the distribution has slightly shifted to the right, as it should. The energy distribution, figure 33, is slightly broadened by the CEM resolution effect, this is just due to the added uncertainty in the energy, the distribution has become well peaked at the mass/energy of the Z^0. The invariant mass distribution, figure 34, is more sensitive to the effects of the CEM resolution, for it exhibits the greatest degree of broadening for $\kappa = 0.025$. Again, the value of the invariant mass has become less well defined due to the uncertainty in the data.

The CEM smearing effect for $\kappa = 0.01$ and $\kappa = 0.03$ were plotted against on another to see how sensitive the Z^0 properties are to a change in κ.[14] The P_T distribution, figure 35, does not show much variation between the $\kappa = 0.01$ or $\kappa = 0.03$ cases. There is little variation in the energy distribution (figure 36) either. The invariant mass distribution, figure 37, does broaden with the larger κ. At this moment, there does not seem to be that much variation coming from κ in the range $0.01 \leq \kappa \leq 0.03$. Figure 38 is a check made to ensure that the CEM effect is being applied correctly. It shows the distribution of the additional energy, and should take the form of a Gaussian distribution, given that the smearing is applied with a Gaussian. As can be seen, the effect is being applied correctly, and the width of the Gaussian increases with κ; this is expected because the amount of additional energy is proportional to κ.

9.4 Friday, November 21, 2003

(3:00pm)

The new graphs (figures 32–37) were shown to Dr. Lancaster, and he commented that the difference between the $\kappa = 0.01$ and $\kappa = 0.03$ data was greater than the sensitivity of the CDF data—κ will need to be known in greater detail. The energy resolution formula of the plug's calorimeter, the PEM was given, which detects for $1 < |\eta| \leq 2.4$.

$$\left(\frac{\Delta E}{E}\right)^2 = \left(\frac{0.16}{\sqrt{E_T}}\right)^2 + \kappa_2^2 \tag{24}$$

This is similar to the CEM energy resolution, (6), with κ_2 in the range $0.01 \leq \kappa \leq 0.08$ and the 0.16 may actually vary up to 0.19 (16–19%). Both of these will be tested to see their effects upon the P_T, energy and invariant mass distributions. The appropriate cuts will also be applied to the data: all electrons or positrons with $\eta > 2.4$ will be ignored, and those remaining will have the relevant energy smearing, be that CEM or PEM. The η distributions of the electrons and positrons, figures 17 and 18 show that a majority of the emitted particles are in the η range specified. Setting the pseudorapidity any higher may lead to misleading results; the angle is getting closer to the background scatter coming from the colliding $p\bar{p}$ beams.

9.5 Saturday, November 22, 2003

2:45pm–5:20pm

The PEM energy resolution effect will be applied to the data using:

$$E(\kappa_2) = E\left(1 + R\sqrt{\frac{0.16^2}{E} + \kappa_2^2}\right) \tag{25}$$

Where R is a random number with a Gaussian distribution (mean=0; width=1). Initially, the PEM effect will be applied alongside the CEM effect (with $\kappa_1 = 0.025$) and κ_2 will be changed from 0.01 to 0.08; κ_2 will affect the data more than κ_1, so it's affect must be investigated.

(5:20pm)

Too much time has been spent working on code that would be able to generate histograms with varying κ_1 and κ_2, it does not seem compatible with the `TLorentzVector` classes. Rather than waste more time on this, less advanced code will be used, one that will hold one κ constant, and vary the other.

[14] κ will be in the range 0.01–0.03, see my meeting with Dr. Lancaster on 5th November 2003.

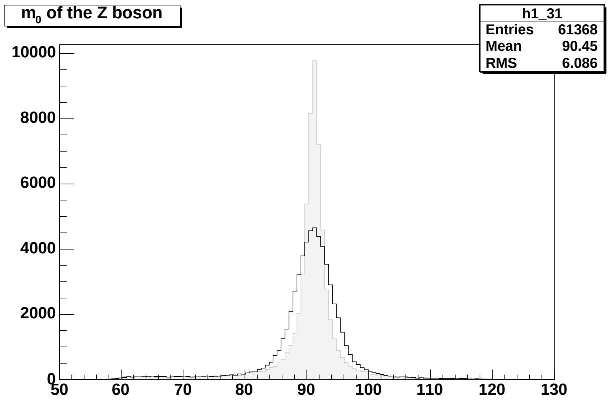

Figure 34: The m_0 distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa = 0.025$. No cuts have been made.

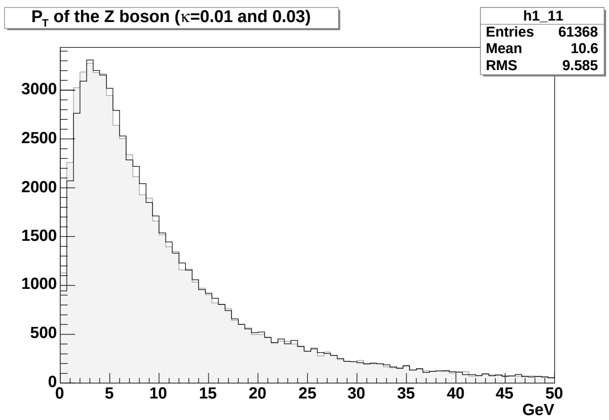

Figure 35: The P_T distribution of the Z boson. The filled area shows the CEM smearing with $\kappa = 0.01$, the line on top is $\kappa = 0.03$. No cuts have been made.

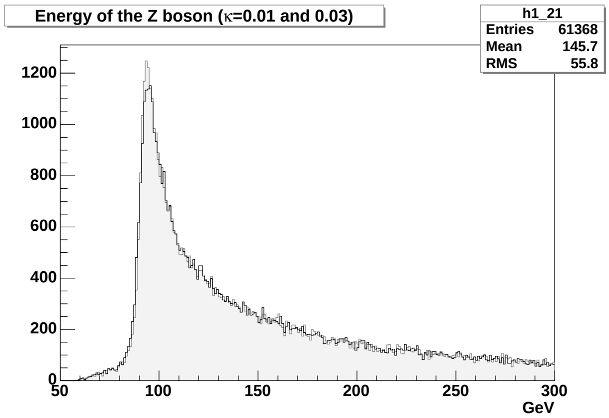

Figure 36: The energy distribution of the Z boson. The filled area shows the CEM smearing with $\kappa = 0.01$, the line on top is $\kappa = 0.03$. No cuts have been made.

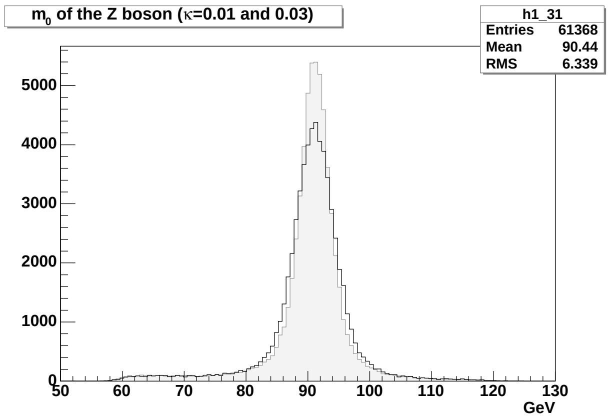

Figure 37: The m_0 distribution of the Z boson. The filled area shows the CEM smearing with $\kappa = 0.01$, the line on top is $\kappa = 0.03$. No cuts have been made.

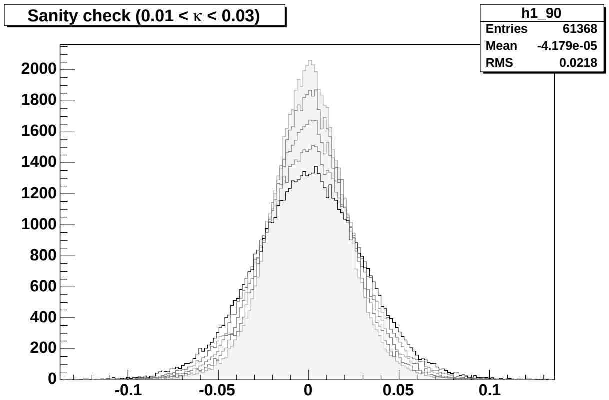

Figure 38: This shows the check of the CEM smearing. The filled line shows the $\kappa = 0.01$, and the uppermost black line is $\kappa = 0.03$. The lines in between are steps of $\kappa = 0.005$.

9.6 Sunday, November 23, 2003

7:30pm–9:00pm

The code was written to vary the PEM κ_2 with a constant CEM κ_1, and a root file produced.

10 Week 10

10.1 Monday, November 24, 2003

8:00pm–9:00pm

The data generated yesterday has been converted into graphs, figures 39–45. The P_T distribution, 39, shows how the combined CEM ($\kappa_1 = 0.025$) and PEM ($\kappa_2 = 0.01$) smearing affect the data. As expected, there is a shift to a higher P_T due to the energy smearing. Figure 40 shows a small broadening of the energy distribution, due to the uncertainty in the energy due to the CEM/PEM effect. The invariant mass distribution, figure 41, is the most sensitive to the effect of adding the PEM smearing.

The PEM smearing for $\kappa_2 = 0.01$ was compared with $\kappa_2 = 0.08$, the extremes of κ_2, so that the sensitivity of the data to κ_2 can be looked at. Its affect on P_T, figure 42, is to shift the peak to the right—increase the P_T, which is expected. The energy distribution, figure 43, is broadened slightly with the increased κ_2, again, as expected. The same is true for the invariant mass of Z^0, figure 44, but to a greater extent.

The PEM application check, figure 45, should show the distribution of the effect the PEM resolution has on the energy, which is the expected Gaussian.

How the CEM smearing affects the combined CEM/PEM smearing should be investigated next.

10.2 Tuesday, November 25, 2003

(5:10pm)

While researching about the CDF apparatus, one found a webpage which listed a few technical papers about various detectors in the apparatus,[15] and the resolutions of the CEM and PEM for the planned CDF2 [4] experiment, which may be of use in the future.

[15] http://www-cdf.fnal.gov/physics/techpub.html

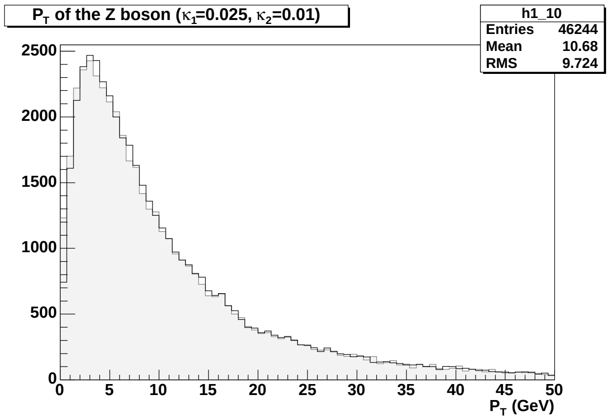

Figure 39: The P_T distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.025$ and the PEM effect with $\kappa_2 = 0.01$. All particles with $|\eta| > 2.4$ were rejected.

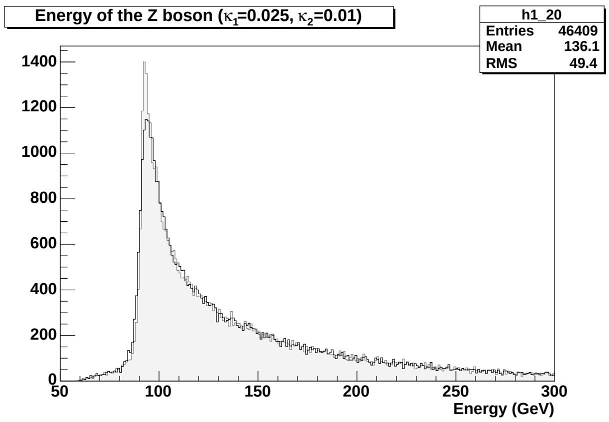

Figure 40: The energy distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.025$ and the PEM effect with $\kappa_2 = 0.01$. All particles with $|\eta| > 2.4$ were rejected.

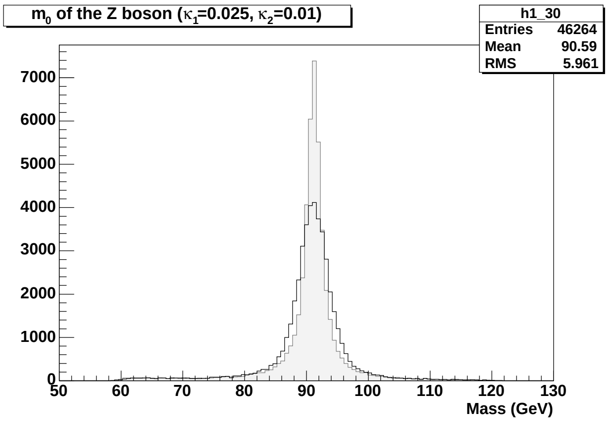

Figure 41: The m_0 distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.025$ and the PEM effect with $\kappa_2 = 0.01$. All particles with $|\eta| > 2.4$ were rejected.

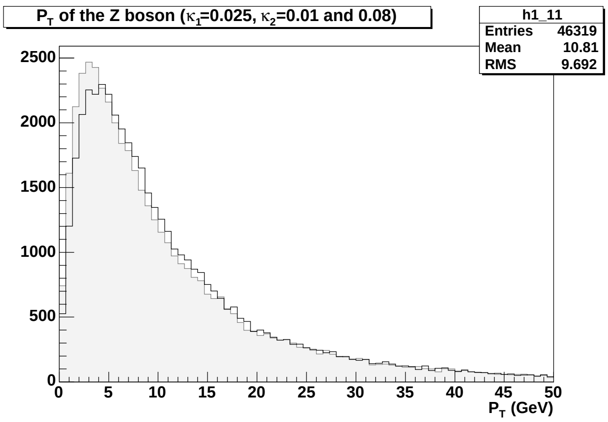

Figure 42: The P_T distribution of the Z boson. The filled area shows the PEM smearing with $\kappa_2 = 0.01$, the line on top is $\kappa_2 = 0.08$; both lines have the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

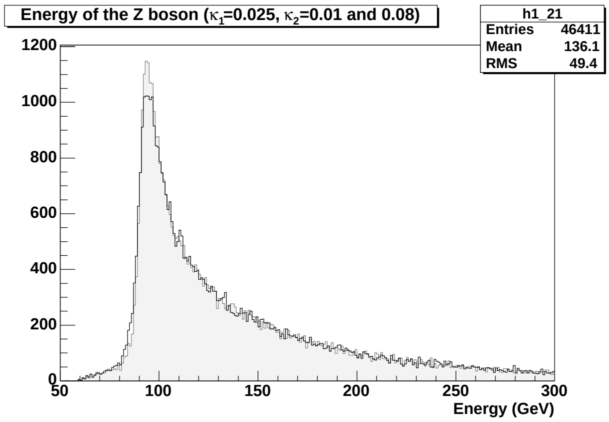

Figure 43: The energy distribution of the Z boson. The filled area shows the PEM smearing with $\kappa_2 = 0.01$, the line on top is $\kappa_2 = 0.08$; both lines have the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

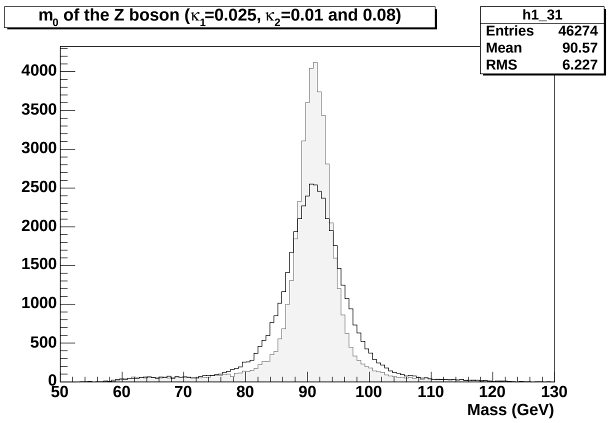

Figure 44: The m_0 distribution of the Z boson. The filled area shows the PEM smearing with $\kappa_2 = 0.01$, the line on top is $\kappa_2 = 0.08$; both lines have the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

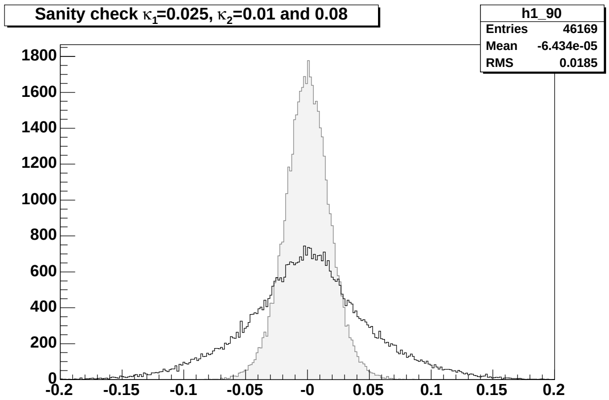

Figure 45: This shows the check of the PEM smearing. The filled line shows the $\kappa_2 = 0.01$, and the black line is $\kappa_2 = 0.08$; both lines have the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

10:10pm–11:50pm

The PEM is expected to have $\kappa_2 = 0.032$, so the PEM resolution effect will be kept at this value, and the CEM effect varied with $0.005 \leq \kappa \leq 0.03$.

Figures 46–51 show the effect that the CEM smearing has on the measurements when the PEM effect is being held constant. As can be seen, the data is much less sensitive to its effect. This is expected because the range of variation in the κ_1 is much smaller than in the PEM case (where $0.01 \leq \kappa_2 \leq 0.08$, which is quite a big change when taken in context).

10.3 Wednesday, November 26, 2003

10:20am–2:50pm

The effect that the 16% term in the PEM resolution, equation (24) will be looked at next. Holding κ_1 at 0.025 (2.5%) and κ_2 at 0.032 (3.2%), the 'term' will be varied from 0.16 to 0.19 (16–19%).

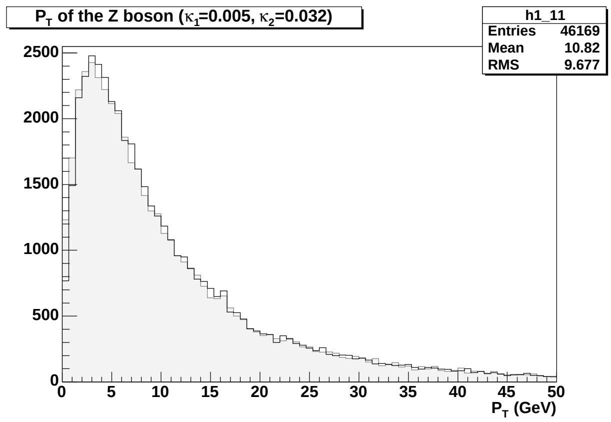

Figure 46: The P_T distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.005$ and the PEM effect with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

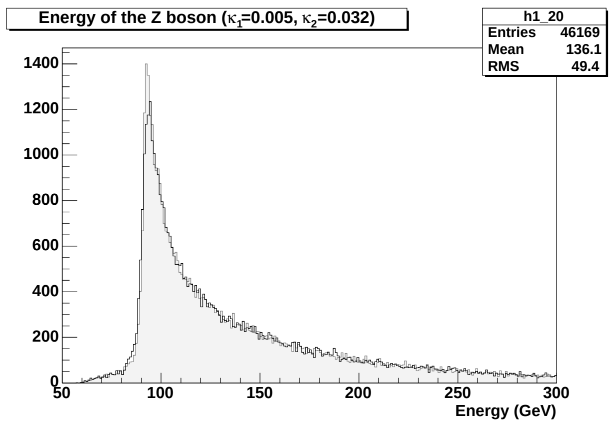

Figure 47: The energy distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.005$ and the PEM effect with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

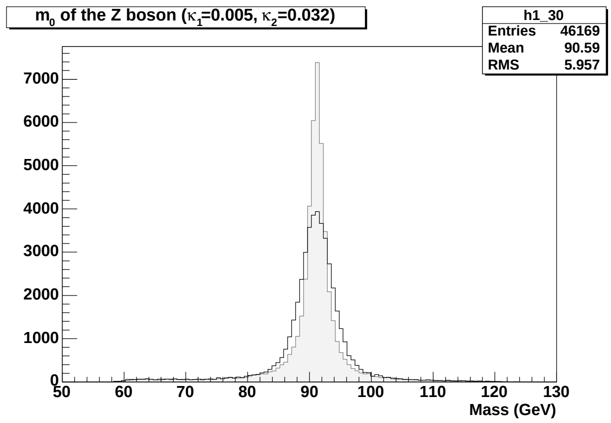

Figure 48: The m_0 distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.005$ and the PEM effect with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

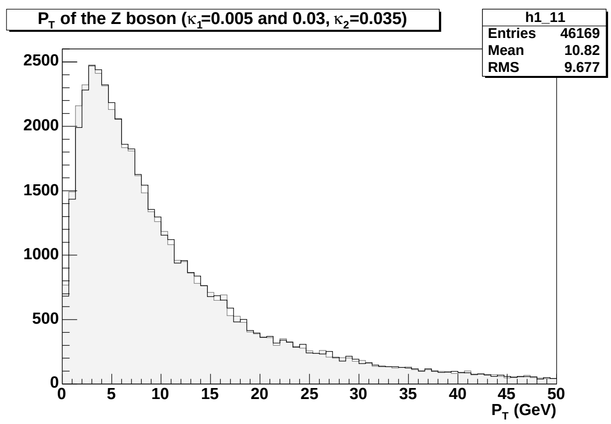

Figure 49: The P_T distribution of the Z boson. The filled area shows the CEM smearing with $\kappa_1 = 0.005$, the line on top is $\kappa_1 = 0.03$; both lines have the PEM effect applied with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

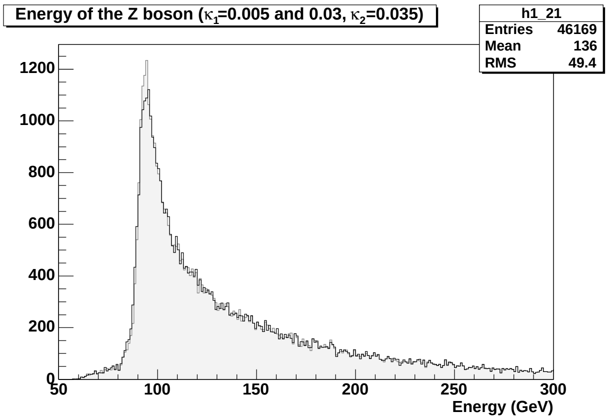

Figure 50: The energy distribution of the Z boson. The filled area shows the CEM smearing with $\kappa_1 = 0.005$, the line on top is $\kappa_1 = 0.03$; both lines have the PEM effect applied with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

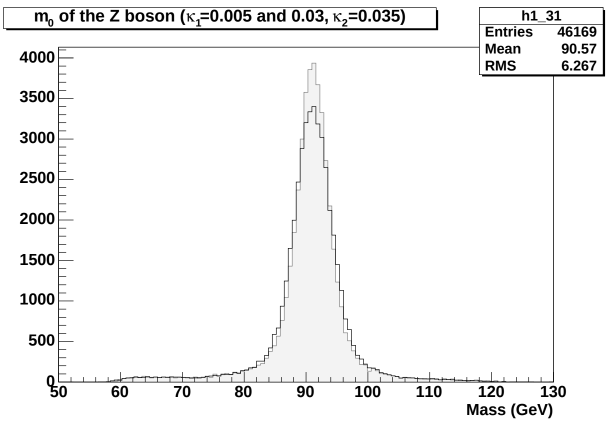

Figure 51: The m_0 distribution of the Z boson. The filled area shows the CEM smearing with $\kappa_1 = 0.005$, the line on top is $\kappa_1 = 0.03$; both lines have the PEM effect applied with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

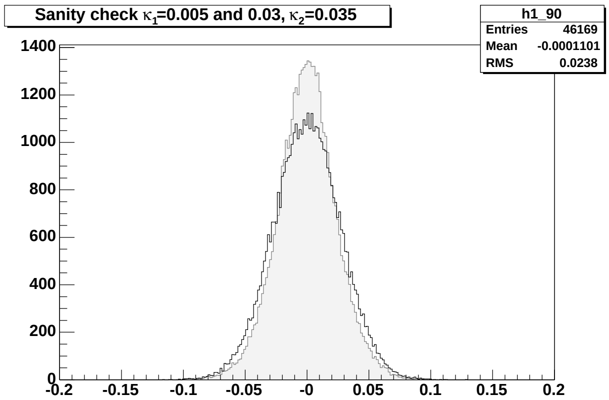

Figure 52: This shows the check of the CEM smearing. The filled line shows the $\kappa_1 = 0.005$, and the black line is $\kappa_1 = 0.03$; both lines have the PEM effect applied with $\kappa_2 = 0.032$. All particles with $|\eta| > 2.4$ were rejected.

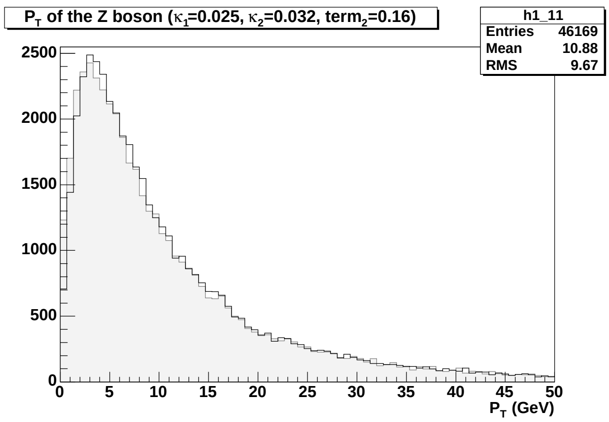

Figure 53: The P_T distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0 = .005$ and the PEM effect with $\kappa_2 = 0.032$ and $term_2 = 0.16$. All particles with $|\eta| > 2.4$ were rejected.

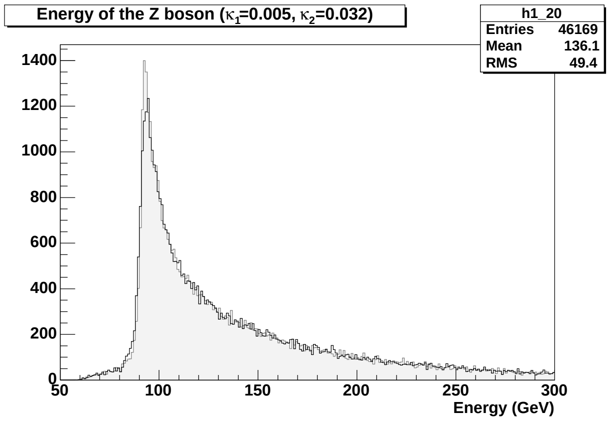

Figure 54: The energy distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.005$ and the PEM effect with $\kappa_2 = 0.032$ and $term_2 = 0.16$. All particles with $|\eta| > 2.4$ were rejected.

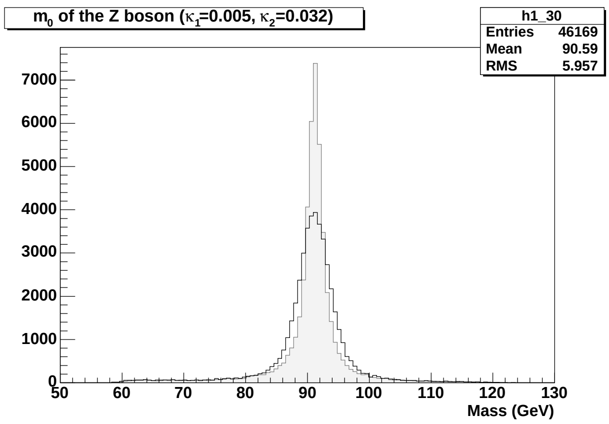

Figure 55: The m_0 distribution of the Z boson. The filled area shows the original data, the line on top shows CEM resolution effect with $\kappa_1 = 0.005$ and the PEM effect with $\kappa_2 = 0.032$ and $term_2 = 0.16$. All particles with $|\eta| > 2.4$ were rejected.

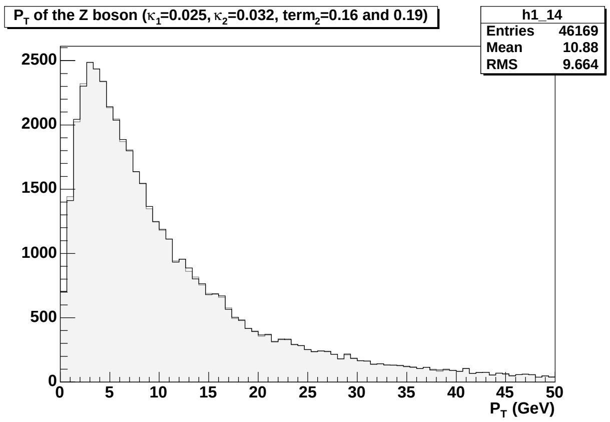

Figure 56: The P_T distribution of the Z boson. The filled area shows the PEM smearing with $term_2 = 0.16$, the line on top is $term_2 = 0.19$; both lines have the PEM effect applied with $\kappa_2 = 0.032$ and the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

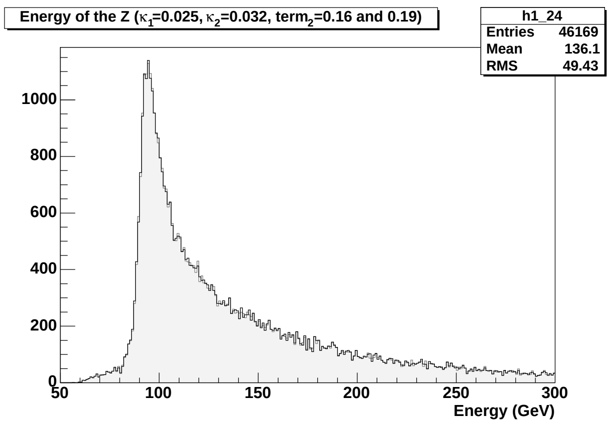

Figure 57: The energy distribution of the Z boson. The filled area shows the PEM smearing with $term_2 = 0.16$, the line on top is $term_2 = 0.19$; both lines have the PEM effect applied with $\kappa_2 = 0.032$ and the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

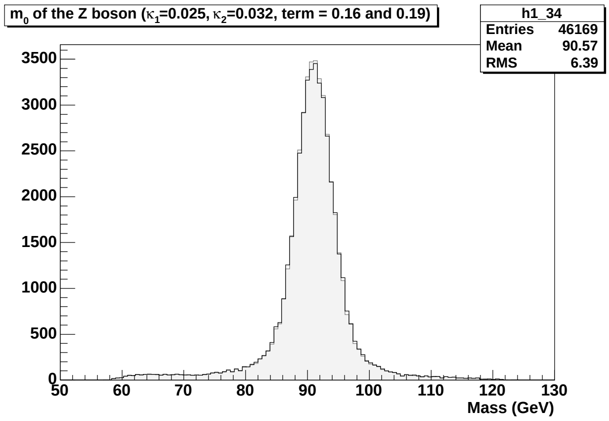

Figure 58: The m_0 distribution of the Z boson. The filled area shows the PEM smearing with $term_2 = 0.16$, the line on top is $term_2 = 0.19$; both lines have the PEM effect applied with $\kappa_2 = 0.032$ and the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

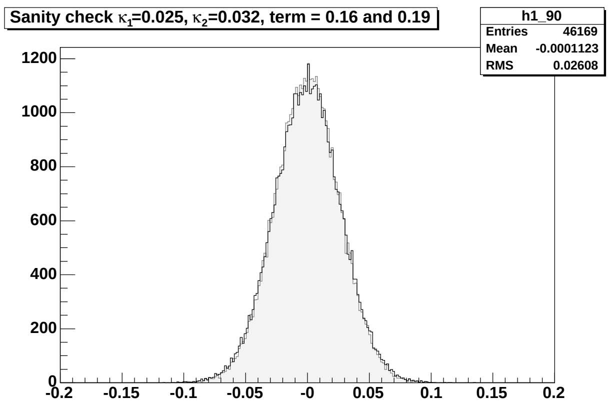

Figure 59: This shows the check of the CEM smearing. The filled area shows the PEM smearing with $term_2 = 0.16$, the line on top is $term_2 = 0.19$; both lines have the PEM effect applied with $\kappa_2 = 0.032$ and the CEM effect applied with $\kappa_1 = 0.025$. All particles with $|\eta| > 2.4$ were rejected.

Figure 53 shows that the PEM smearing applied with $\kappa_2 = 0.032$ and a major term of 16%. The affect on the distribution is as expected. This is the same for the energy and m_0 distributions, figures 54 and 55 respectively.

The P_T, energy and invariant mass distributions for the changing 'term' show that the error due to any uncertainty in its value will be negligible.

(2:40pm)

Found some useful diagrams in the Fermilab website of the CDF apparatus:

10.4 Friday, November 27, 2003

(2:10pm)

Spoke to Dr. Lancaster about where the project is heading. The data generated has shown that κ is going to be important—the change in the data is greater than the natural uncertainty. The work been carrying out on the CEM and PEM resolutions does not take such factors as electron acceptance into account. The next part of the project will involve real data from the CDF. Dr. Lancaster has created a ASCII file, `zee_central_only_cdf_real.data` which contains CDF data, and a Root file which contains m_0 distributions for various M_z and κ (CEM smearing only).

Any data point recorded can fit a number of distributions, and one is to write a program that will calculate this probability, $P(M_Z, \kappa)$ for any given data point $(M_{Z,i}, \kappa_i)$, and then determine the most likely distribution. This will require further research into likelihood fitting, interpolation algorithms and minimisation functions (there is a minimisation function built into Root).

For the moment, research into the algorithms and theories is needed.

11 Week 11

11.1 Monday, December 1, 2003

11:00am–3:00pm

Attempted to find some interpolation routines on the internet to be used in the next part of the project. Most information on the likes of bicubic spline interpolation is about image-resizing plug-ins—2D interpolation is crucial when increasing the size of an image—but little on the actual programming behind them. Also trying to find information on minimisation (Root's `TMinuet`) and integration

(2:45pm)

There is a company[16] that provides free source code for various analysis functions (as a library that needs to be compiled and installed). Rather than this, a copy of a book like '*Numerical Recipes*' may be of more use. All the copies of '*Numerical Recipes in C++*' were out of the library, but there were copies of '*Numerical Recipes in C*', which should suffice for the moment.

11.2 Tuesday, December 2, 2003

(12:00am)

Dr. Lancaster cancelled this week's meeting, but he was sent an email outlining the intended method to complete this section of the task.

Attempted to write a independent program in C++, rather than just methods in a larger program, as had been done previously. Tried to implement some file I/0 (the $M_{Z,i}$ will be read from a separate file), but it failed.

[16] Magic Software http://www.magic-software.com

11.3 Wednesday, December 3, 2003

10:00am–11:50am

Received an email from Dr. Lancaster explaining that the method was not quite what he had in mind. He outlined what was required in his email, but discussing it with him when he gets back will help in my understanding. For the moment, one will work on getting the CDF data reading section working.

Time spent researching into C++ programming on the internet. Still could get the program to work as wanted. The file I/O is causing a lot of trouble.

11.4 Thursday, December 4, 2003

The program was written, and saved as `example.cc`[17]. It reads the data from the file and stored the data as an array of doubles. However, the program loses the final decimal when writing the values into the array; no idea why it does this, but it may prove to be a problem.

12 Week 12

12.1 Tuesday, December 9, 2003

Spoke to Dr. Lancaster about the how the fitting program was to be implemented. He explained how it was to work.

12.2 Wednesday, December 10, 2003

Dr. Lancaster gave me a library of all the example C programs from '*Numerical Recipes in C*' and told me where all the source code was in his file space on the network. Now have set routines to interpolate the data. Also downloaded some old Fortran programs of his, which complete the same task as the fitting program. This should make it easier to understand what the program is supposed to do.

13 Week 13

13.1 Tuesday, January 9, 2004

13:20pm–5:00pm

The array that stores the data from the CDF should really in inside a custom class, so one shall attempt to define the class, `realData`. This class with contain a private array of doubles and public methods for setting and returning elements in the array.

(2:30pm)

Researching how to write classes because my knowledge is limited.

(4:45pm)

Succeeded in getting the `realData.cc` (and its `realData.h`) file to compile, but am having trouble linking them so that my test program `realDataTest.cc` can see if the class is working properly.

[17]in a subdirectory called `4C00_fitter`

14 Week 14

14.1 Monday, January 12, 2004

(12:30pm)

Updated the website pages so that they point to the correct links.

14.2 Sunday, January 18, 2004

3:45pm–5:00pm

Worked on converting the Fortran program into C++. The files in the program are `lhood.f` (main program), `dfint.f`, `lgamma.f`, `savgol.f`, and `smoothr.f`. To begin with, converting some of the routines (from the likes of Numerical Recipes) into C++. The `lgamma.f` just returns $\ln \Gamma(x)$ for $x > 0$. This function was taken from '*Numerical Recipes*' the code in C++ being `gammaln.c`[18].

`savgol.f` is a Fortran implementation of Savitsky-Golay smoothing, which can be found, again, in '*Numerical Recipes in C*'[19]. In trying to compile it, more files are required: some matrix methods, `ludcmp.c` and `lubksb.c` [20]. These files also required another set of files: `nrutil.h` and `nrutil.c`, and minor modification of the header file (some methods were missing the overloaded definitions used in the actual source).

15 Week 15

15.1 Monday, January 19, 2004

(3:40pm)

Worked out why there may be as many compilation errors, the compiler was not used correctly. Having read about makefiles[21], one now know how to use the compiler and linker correctly.

15.2 Wednesday, January 21, 2004

1:00pm–4:45pm

Working in writing the fitting program.

(2:30pm)

Having too many problems with memory allocation and scope. Decided to rewrite the file reading program using the `<vector>` container class.

15.3 Saturday, January 24, 2004

2:00pm–7:00pm

The `realData` class is working, next is to develop the master index class, to retrieve and store the index all the histograms in the the root file. Could not find a way to extract the histograms for the file.

7:00pm

After some research at the Root webpage[22], one found a user guide which described file I/O. Inspection of the `Makefile` written by Dr. Lancaster to compile the `ExampleJobAnalysis.exe` file showed how to compile and link the C++ code to be able to use the Root C++ classes. Then to use them, one would only have to include the relevant headers.

[18] §6.1, p214
[19] §14.8, p650ff
[20] LU Decomposition of a matrix, §2.3, p46–48
[21] http://users.actcom.co.il/∼choo/lupg/tutorials/writing-makefiles/writing-makefiles.html
[22] http://root.cern.ch

15.4 Sunday, January 25, 2004

10:30am–6:30pm

With the new knowledge about Root file I/O, the classes for setting up the master ID array and interpolation array need to be written.

(1:00pm)

The class for creating the master ID array has been written and seems to be working correctly. The source files are `masterIndex.cc` and `masterIndex.h`. To be able to extract histograms using this class is very important, but one did not know how to cast an integer to a character array/string. To solve the problem, crude methods to format numbers from, for example, '12' to 'h12' have been developed.

(5:00pm)

While writing the class for the interpolation array (`iArray.cc` and `iArray.h`) a problem in the formatting methods has become apparent. When it formats '100', '100' results, not the expected 'h100'; trying to eradicate the bug from the code only succeeding in finding new number for the formating to crash at.

(5:40pm)

Bug found and corrected.

(6:15pm)

The initial interpolation array class has been written and appears to be working correctly.

16 Week 16

16.1 Tuesday, January 27, 2004

4:00pm–6:00pm

Spoke to Dr. Lancaster, earlier, about my progress with the fitting program, and how the `TMinuit` actually returns the values of M_Z and κ being looked for. At the moment, the program can generate a 3D array with M_Z, κ and the histogram parameters making up the dimensions. Thus with an interpolation routine, a histogram profile can be generated for any value of M_Z,κ inside the range of data. This is what will be done. An arbitrary (M_Z,κ) will be used to select a histogram profile. Then an analysis of the fit to the CDF data, from the text file, will be made (the profile will need to be normalised by setting the integral of all the data to unity before the fit). This value is given to Minuit, which will then generate a new pair of M_Z,κ values. A new histogram profile will be interpolated if need be. This is again normalised and checked against the CDF data. This process will repeat until Minuit finds a reasonable minimum. At this point the best fit values of M_Z and κ will be returned.

Thus, the next task is to build the interpolation routines in to the program. '*Numerical Recipes*' has a cubic spline algorithm that will be employed.

(6:00pm)

Tricubic spline interpolation of the array may not be what is required. This method would return $y(M_Z, \kappa, z_i)$ where x_i is a certain point along the (M_Z,κ) histogram. But the (M_Z,κ) histogram is to be normalised, so I think that a bi-cubic spline interpolation of each z_i would generate a crude profile that could then be normalised, and the CDF data checked against (this would involve more interpolation between z_i).

Thus, a method that takes (M_Z,κ) and the 3D array as an input and returns an array of z_i needs to be written next.

16.2 Wednesday, January 28, 2004

12:00pm–4:30pm

The bicubic interpolation routine has been finished, and compiles, but it does not return reasonable values. It looks like there has been a mistake with a pointer reference, from the values returned.

(4:30pm)

Bug has been isolated to being within the `numRep::splint` method. The other methods seem to general reasonable results, except itself and those that those that depend on it.

16.3 Thursday, January 29, 2004

11:00am–5:20pm

The bug in the interpolation routine, thought to be in the `splint` method needs to be found. However, should this prove a fruitless task, implementing a different version of 2D interpolation may be required.

(12:20pm)

Theres may be a problem with way `a*a*a-a` and `b*b*b-b` is calculated. After checking, these values are supposed to be this large. With this in mind, the bug may be in the way the second derivatives are calculated, `numRep::spline` and `numRep::splie2`.

(12:45pm)

`spline` and `splint` are working correctly, having been tested and returned the expected data at (90.25,1.0), '5408.16'. The next data point is (90.25,1.2), '5328.15'; the routines gave an estimation of '5339.92' for (90.25,1.1). This shows that the cubic spline routines are working correctly. Thus, the creation of the 2D array of second derivatives should contain the error. This cannot be the source of the error, upon checking the array of second derivatives used for testing with the current 2D array that is being generated. This leaves `splin2` with the bug.

(4:00pm)

The bug has been found and corrected for. The `splint` functions in the suspected method were being called with an incorrect value, an X-value from an array, rather than the Y input value that it should have been. The next task is to check the consistency of the interpolated profiles with a histogram in the master Root file.

(5:20pm)

The (91.001,1.001) histogram was compared with the (91,1) histogram 'h84'. They are pretty much identical, so the interpolation routines are working, saved a `fit_check.root`. Now the fit parameter routines have to be written so that Minuit can run.

17 Week 17

17.1 Monday, February 2, 2004

3:30pm–4:30pm

In order to check the interpolation routines are working, Dr. Lancaster had suggested looking at how they vary around a grid point at (91,1) to (91.125,1.2), saved in the `fit_check.root`.

The next routine to build into the program is the normalisation routine. Currently, it is a crude summation of all the bins to find the integral; this is not good enough, for the minimisation will interpolate across the bins. Thus an accurate, yet fast integration algorithm is needed.

(4:30pm)

'*Numerical Recipes in C++*' have many algorithms for integration[23], but many do not seem relevant to the task at hand. The mathematics apply, but the actual implementations require in input function, rather than an array of data. On second thoughts, this can easily be adapted for. The program is going to need to be able to interpolate between bins when the CDF data is checked during the likelihood fitting. Thus, the function asked for will be this routine.

17.2 Wednesday, February 4, 2004

1:20pm–6:00pm

The minimisation routine, using `TMinuit` needs to be developed, so first some minor research into the Root class is required.

(6:00pm)

There are problems with the arrays in the program; they seem exist at some point, then not the next. This may be due to how all the arrays are declared as pointers. All the code in the classes should be rewritten to use the vector class instead. This will make the program slower, but less prone to errors.

17.3 Thursday, February 5, 2004

12:00pm–6:10pm

The reason for the error in the log likelihood fitting would be down an incorrect formula used. This was corrected, but the histogram profile arrays were becoming 'strange' near the end of the execution of the program. It seems that closing a Root file caused another person problems, so it was not closed until the very end of the code—this solved the problem. Yet, there are still bugs. While the log likelihood calculation code should be correct, it is returning '0' all the time, and the values during the calculations alternate between '0' and '-0', very bizarre. This was a minor bug in the code that was corrected.

(2:40pm)

A serious error in the code has been uncovered. The interpolation is creating histograms that have negative bulges as one moves toward the 'edges' of the (M_Z,κ) range. This is not acceptable. First ideas were that it was an artifact of using cubic spline interpolation, and that it may be rectified using polynomial interpolation. This idea was quickly nullified by a quick check at the Numerical Recipes forum on the Internet[24]: the polynomial to interpolate the data would need to be of the order of 9 and 16—the theory is that the polynomial needed to interpolate the data must be of order one less than the number of data points)—which would be too large for a stable polynomial.

(6:10pm)

In an effort to use the integration code Dr. Lancaster gave me, a new class was created to store the profiles in. This is an improvement on the program as it more object-orientated than before. One is also now aware of why the log likelihood fitting is returning 'nan' (not a number) outside a certain range. Inspection of the interpolates profiles at this point reveal that there are 'bulges' caused by the splines, these cannot be integrated during the normalisation process and the the fitting function cannot cope further along the line. While the problems with the interpolation must be corrected, it would also be good form to code an error checking feature into the program that detects this run-time error. However, correcting the bulging profiles is a more pressing concern.

[23]§4, p133ff
[24]http://www.numerical-recipes.com/forum/index.php

17.4 Saturday, February 7, 2004

4:30pm–8:00pm

Attempts to control the splines by explicitly specifying the derivatives at the edges of the array has not provided much success, the arrays returned are full of 'nan'. While the cubic spline interpolation is fine for the calculation of the best fit—with 160 data points, polynomial interpolation with polynomials of an order of 159 would be wildly unstable—the profile extraction procedure needs more refinement.

Polynomial interpolation will be used for the profile extraction, there is a chance that it will be unstable, but the bicubic spline method produces histograms that are not acceptable for the fitting process.

The polynomial interpolation routine has been written, `polint`, but it breaks down after two calls when called from a for-loop whilst checking. It is also inaccurate further from the centre of the distribution, as expected with polynomials; this level of inaccuracy (at (91,0.5) '8770.98' when a value about '5500' is expected) is not good enough.

It was at this point that one noticed that while the range of the individual M_Z profiles is 70–110 GeV, the range of input histogram M_Z profiles is 90–92 GeV; the negative histograms were appearing at M_Z's of 89 and lower. This would have been extrapolating the data outside of the interpolation region, thus the histograms with negative regions are not that surprising, given the lack of accuracy with extrapolation. When the log likelihood is checked in the 90–92 interval, the program worked correctly.

17.5 Sunday, February 8, 2004

4:00pm–

Found some information about how to use TMinuit in Root[25] and it says that TMinuit expects the function that it will minimise to be a static function; my program uses member variables and member functions, which will not work with TMinuit.

18 Week 18

18.1 Tuesday, February 10, 2004

1:30pm–4:00pm

`FCN` can accept data input (even though it is an external static function) using a `GetObjectFit`, which allows one `TObject` to be passed in. Thus I need to rewrite all me code so that all the member functions that are needed are in one class, and have that class be derived from the base `TObject` class. Then I should be able to pass it, and it's methods into the minimisation function.

18.2 Thursday, February 12, 2004

12:00pm–4:40pm

Was shown that the reason why the code was not compiling was due to the TMinuit library not being set. Once the link to it was defined, the code compiled correctly.

Now the program fails during while Migrand runs; a memory error somewhere, it seems.

(4:30pm)

The program now seems to work and gives M_Z of 91.2487±0.0046 and κ of 1.919±0.018 as the best fit for the data.

[25]http:// the place

18.3 Sunday, February 15, 2004

3:00pm–4:40pm

The `ControlCards` class was integrated into the fitting program. Rather than having to recompile the code to change parameters, one only needs change the data in an external file, `fitter.cards` and 'tell' the program where to find the file: typing `export CONTROL_CARDS=fitter.cards` to set an environment variable for the `ControlCards` class to locate.

19 Week 19

19.1 Wednesday, February 12, 2004

1:00pm–5:10pm

Having spoke with Dr Lancaster, the fitting program is just about finished. The integration routine being used is still too crude, implementing the routine he gave would be the best option. Last time I tried to integrate this routine into the program, the program crashed because of a segmentation fault. Where the fault lies can be found with the use of a debugger, `gdb`.

(1:50pm)

The fault seems to be to do with the arrays. This problem was encountered earlier, the program seems to run out of memory. The exact part of the program that fails is not the integration itself, but recreating the `profile::splines` array for the new normalised data. There is also an unnerving difference results of the two methods. For a (92,1.9) profile, summing over the total bins gives an integral of 1.27434×10^7, whereas the Gaussian Legendre method estimates it to be 3.18325×10^6, a factor of four out. One must be incorrect.

(3:00pm)

Having isolated the problem, and managed to stop it appearing in the test code (`main_testing.cc`) the problem reappears when the program in compiled to work with Minuit (`main_minuiut.cc`).[26]

(4:15pm)

Integrating the profule by a summation of the bins can be used; having been reminded that the bin contents needed to be multiplied by the bin widths (the area of each bin), which had been forgotten. After correcting this basic mistake, the two integration methods were compared. The summation now gave a value of 8.18358×10^6 for the area under the graph, matching the Gaussian Legendre method up to four significant figures. Re-writing all the code to use vectors of doubles may have helped work out why the program seemed to run out of memory, but would have been time consuming. With the knowledge that the 'crude' method was not as crude as first though, the program should now be finished.

19.2 Thursday, February 19, 2004

1:00pm–4:30pm

The fitting program is not returning the correct results for the best fit. The error estimated being returned by TMinuit are too small. A histogram of the raw data was created [27] so that the mean and RMS of the data could be compared with the best fit profile from the fitter[28]. The means were 90.05 and 89.35, and the RMS, 5.425 and 4.506. The differences are too great, and indicate a problem with the fitting program.

[26] It should be noted, that rather than have a complicated main source code to debug the program, there is a separate testing code to the actual program code. While this would not highlight bugs in the main file, it would allow testing of the various objects and classes used. To pick which version of the program to compile, there is an option in the `Makefile` which identifies the filename of the main code.

[27] `040219_Z_actual.root`

[28] `040219_FitterOUT.root`

The first check is upon the interpolation. A liner interpolation of two histograms with differing M_Z will be determined using Root, with the result checked against what the interpolation estimates it to be.

The basic check will involve taking two adjacent histograms in the Root file[29] (90.625,1.8) and (90.750,1.8), which are 'h58' and 'h68', respectively. A third histogram will be formed from the initial two: create a blank histogram in Root, before adding each histogram to it with 50% weighting. This will produce a histogram at (90.6875,1.8) via a linear interpolation. An unnormalised histogram generated by the `Fitter` should be identical.

Figure 60 shows that the two estimates do not agree. The error in the program may lie in the interpolation routines.

19.3 Friday, February 20, 2004

5:00pm–7:00pm

Looking at figure 60 it seems that they interpolated profile differs from the Root profile by a bin. In fact when inspected, it would appear that the Fitter profile begins a bin too early. Thus, recreating the figure, but offsetting the array by one bin. The resultant figure shows that the `Fitter` interpolation agrees with the Root interpolation. Thus, the error in the best fit may be a systematic error. To test for this, the same test will be carried out at different points in the (M_Z,κ) plane.

The interpolation has been checked at four points in the (M_Z,κ) plane: (90.6875,1.8), (91.8125,1.8), (90.6875,1.8) and (91.8125,1.8), figures 61–64. The linear interplation is the same as the `Fitter` interpolation in each case, the error in the `Fitter` program does not appear to lie in the interpolation algorithm.

20 Week 20

20.1 Tuesday, February 24, 2004

1:30pm–4:300pm

Generated a histogram comparing the best fit from `Fitter` with a normalised profile of the raw data. As can be seen from figure 65, the best fit is obviously not correct. Using the top and bottom edges of the bins, figures 66–67, does not have an affect on the best fit either.

20.2 Wednesday, February 25, 2004

12:30pm–4:00pm

20.3 Thursday, February 26, 2004

12:00pm–

21 Week 21

21.1 Tuesday, March 2, 2004

4:00pm–6:00pm

With only a few weeks left until the project deadline, it does not make much sense to continue trying to find the problems in the fitting program. The options are to either implment a calibration factor into the program (thus allowing the best fit to be tweeked into the right place) or using another fitting method. With no viable alternative to the current program, calibrating it to fit the data seems to be the only choice, albeit, an unsatisfactory one.

[29]'Adjacent' with regards to the location of the histogram in the (M_Z,κ) plane.

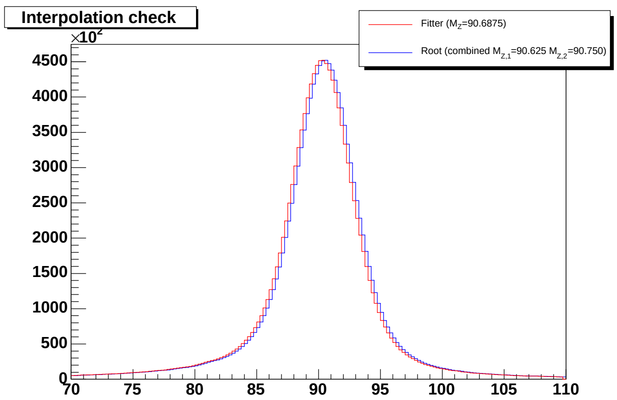

Figure 60: A check of the interpolation routines used: a linear interpolation and the `Fitter` estimate of the histogram profile with M_Z=90.6875 and κ=0.8.

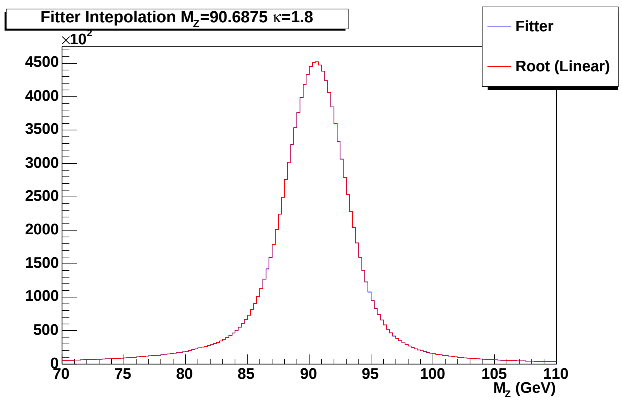

Figure 61: A check of the interpolation routines used: a linear interpolation (with $M_{Z,1}$=90.625 GeV, $M_{Z,2}$=90.750 GeV) and the `Fitter` estimate of the histogram profile with M_Z=90.6875 GeV and κ=1.8.

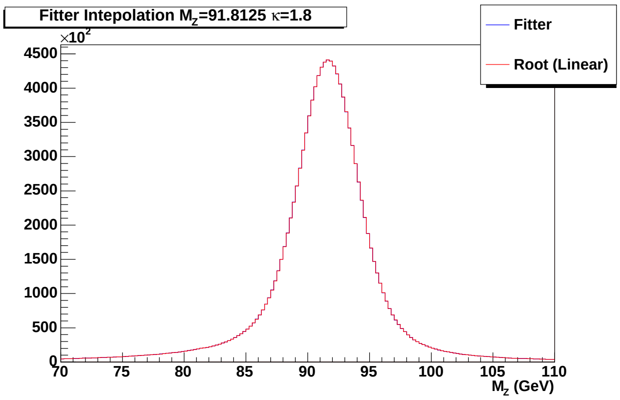

Figure 62: A check of the interpolation routines used: a linear interpolation (with $M_{Z,1}$=91.750 GeV, $M_{Z,2}$=91.875 GeV) and the `Fitter` estimate of the histogram profile with M_Z=91.8125 GeV and κ=1.8.

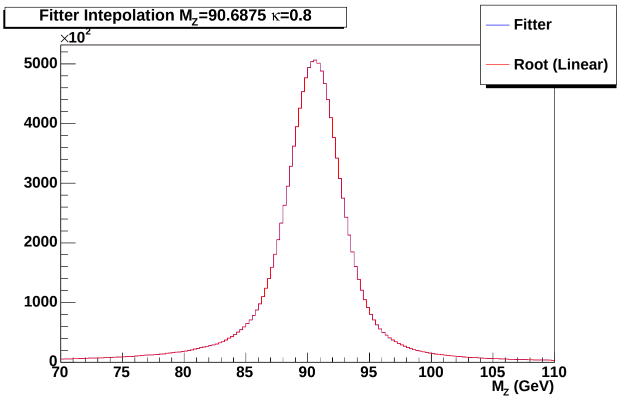

Figure 63: A check of the interpolation routines used: a linear interpolation (with $M_{Z,1}$=90.625 GeV, $M_{Z,2}$=90.750 GeV) and the `Fitter` estimate of the histogram profile with M_Z=90.6875 GeV and κ=0.8.

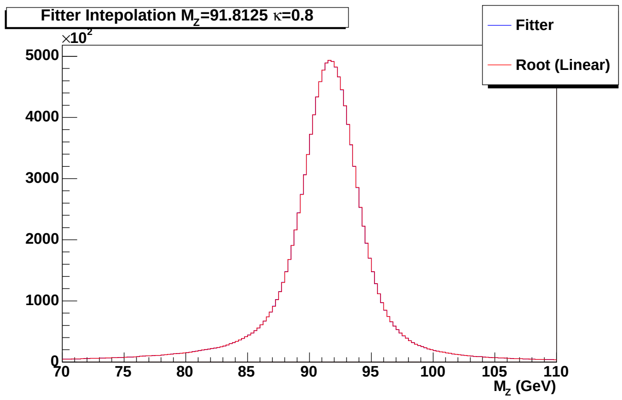

Figure 64: A check of the interpolation routines used: a linear interpolation (with $M_{Z,1}$=91.750 GeV, $M_{Z,2}$=91.875 GeV) and the `Fitter` estimate of the histogram profile with M_Z=91.8125 GeV and κ=0.8.

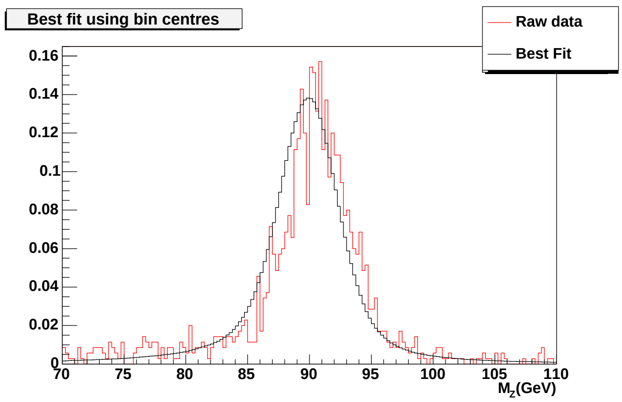

Figure 65: The raw M_Z data with the `Fitter` best fit profile. The fit was made in the region $80 \leq M_Z \leq 100$ GeV and taking the centre of the bin labels.

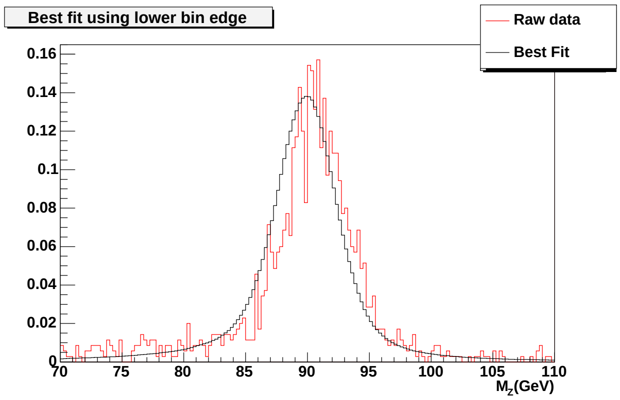

Figure 66: The raw M_Z data with the `Fitter` best fit profile. The fit was made in the region $80 \leq M_Z \leq 100$ GeV and taking the bottom edge of the bin labels.

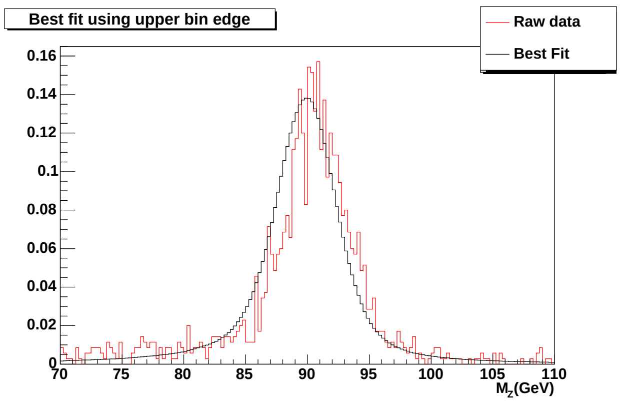

Figure 67: The raw M_Z data with the `Fitter` best fit profile. The fit was made in the region $80 \leq M_Z \leq 100$ GeV and taking the top edge of the bin labels.

Once the calibration factor has been implemented, the functional form for the $P_T(Z)$ will need to be determined. `Fitter` can fit a two parameter distribution, therefore, a Root file of function forms (configured in a similar way to the (M_Z,κ) file) can be fitted against actual CDF $P_T(M)$ data.

21.2 Wednesday, March 3, 2004

12:30–7:20pm

The functional form for the $P_T(Z)$ should be determined, leaving fitting program to work on later. The functional form to be used is the following (where X = $P_T/50$ and Γ is a Gamma function):

$$\frac{\mathrm{d}\sigma}{\mathrm{d}p_T} = \frac{X^{P_4}}{\Gamma(P_4+1)}\left[(1-P_1)\,P_2^{P_4+1}e^{-P_2X} + P_1P_3^{P_4+1}e^{-P_3X}\right] \tag{26}$$

From this, a smeared P_T form, P_T^{smear}, can be dervied:

$$P_T^{smear} = P_T^{true} \otimes f_{smear} \tag{27}$$

The smearing function in (27) is the effect of the detector resolution on the data.

Since $P_4 \in \mathbb{R}^+$, the `TMath::Gamma` method can be used, rather than implment a more general algorithm.

(4:00pm)

Need to write a small program to extract PT data from a Root file, so that there is some data for the program to fit with. The original Root file[30] contains information for the e^- (id=11), e^+ (id=-11) and (id=23)[31]. A small program can read the Root file, extract the $P_T(Z)$ data, and write it to an ASCII file to be used to test the `PTFit` program.

(7:00pm)

Test data has finally been created. It was proving too difficult to write a program that read the Root file from scratch, so it was decided to use the `ExampleJopAnalysis.exe` program. This did not work either, the program would hang during the first loop, only writing one value to the text file. In the end, a histogram with 61368 bins (the same number of events in the file) was created, and each bin was weighted with the respective $P_T(Z)$. This histogram was saved to a Root file. This Root file was then read, and the data from the histogram extracted and written to a file. It can only be assumed that something in Dr. Lancaster's framework was causing problems with the code in the `ExampleJobAnalysis.exe` attempt.

21.3 Thursday, March 4, 2004

12:00pm–5:00pm

A program to find the four paramaters with `TMinuit` has been written[32]. The code worked out the χ^2 for the functional form when fitted against the data from HERWIG. However, the parameters returned from the fit do not produce a form that fits the data.

The reason for this is simple: the χ^2 is incorrect. The program compares each $P_T(Z)$ with the likelihood of occcurance, as determined by (26). Rather, the data should be put into a histogram, normalised, then checked against (26).

(2:00pm)

The new code has been written[33] and run, but the fits are still incorrect. `TMinuit` is still sensitive to the input values, the problem may lie in the actual data being fitted against. This data has a spike around 3 GeV and is misleading `TMinuit`, figure 69.

[30] `Zee_feynman_only_50k.root`, used in earlier stages of the project when investigating the effect κ had upon the data.
[31] http://pdg.lbl.gov/2002/montecarlorpp.pdf
[32] 040303_findparam.cc
[33] 040304_findparam.cc

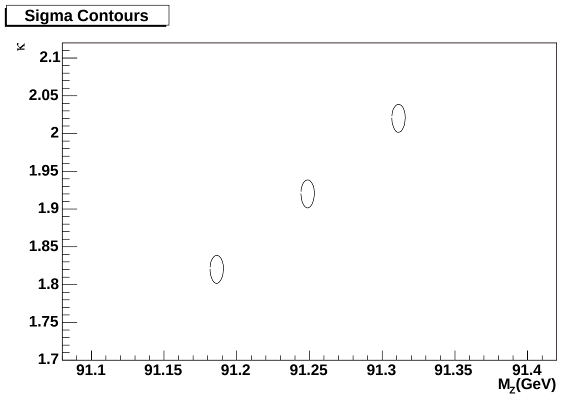

Figure 68: The σ contours for the respective best fits in figures 65–67.

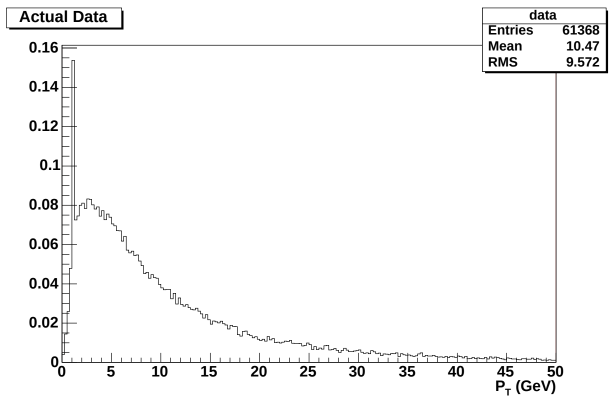

Figure 69: The $P_T(Z)$ data being used. The spike in the data is causing `TMinuit` to return incorrect fits.

Dr. Lancaster says he will generate a file of correct P_T data to use, but it will be a few hours before he has the time to do so. In the meantime, work on how to apply the smearing to the P_T^{true} will be looked at.

There are two ways to do it. One way would be to apply to convolute the two funtions analytically, and to use the resultant form in the fitting. A different way would be to systemmatically check the differences between P_T^{true} and P_T^{smear} for HERWIG generated data. Using the recontructed $P_T(Z)$ (the differences between the electron/positron transverse momenta), the functional form can be re-weighted so that is describes the smeared data, before χ^2 testing.

As far as one knows, a way to take the convolution of two functions involves multiplying the two functions together in Fourier-space. This is not a trivial process for (26) given than Fourier transforms on three-parameter functions is demmanding on computational time, let alone a five-parameter one, (28). Nethertheless, this method would be more accurate than the cruder re-weighting scheme.

$$F(Q_1 \dots Q_T) = \int_{\infty}^{\infty} \dots \int_{\infty}^{\infty} f(P_1 \dots P_T) e^{-2\pi i (P_1 Q_1 + \dots + P_T Q_T)} \mathrm{d}Q_1 \dots \mathrm{d}Q_T \tag{28}$$

After more thought, there are two smearing functions, one for each electron and positron:

$$P_T^{smear} = P_T^{true}\left(p_T^{e^-}, p_T^{e^+}\right) \otimes f_{smear}^{e^-}\left(p_T^{e^-}, E_{e^-}\right) \otimes f_{smear}^{e^+}\left(p_T^{e^+}, E_{e^+}\right) \tag{29}$$

(9:00pm)

The analytical idea will not work as planned. The fourier transform of (26), (28), takes 3–4 minutes to compute (using '*Mathmatica*', and it does not have a nice form, comprised of recursive fourier transforms. Thus, the energy smearing will be achieved by re-weighting the P_T^{true} distribution.

21.4 Thursday, March 4, 2004

12:00pm–2:00pm

Dr. Lancaster has generated a file of HERWIG data. The file contains the (P_X,P_Y,P_Z) for each electron and the weight of the event. This data was used to generate a correct histogram to test against. However, `TMinuit` still will not return sensible values for the four parameters. A graph of (26) was generated, and highlights a problem in the code. As can be seen in figure 70, the shape of the curve is nothing close to what is to be expected.

5:00pm–8:30pm

There may be a problem with the Gamma function algorithm being used. To check, an algorithm used to generate $\ln\Gamma(z)$ will be compared with `TMath::Gamma`. They are both the same, so the problem does not lie in the gamma function.

(5:45pm)

Found the bug in the code, the formula was using X^4 rather than X^{P_4}. The parameters should be extracted from the data with a working functional form now.

Table 21.4 shows the results from `PTFit`, the program written to find the functional form that best fits the data. The results are unsatisfactory, the starting values are sitting on the expected minimum; the starting values are the parameters used to generate the data in HERWIG. These starting values were used because `TMinuit` would return functional forms that were amazingly incorrect. As can be seen in figure 72, the functional form follows the data, but it not a satisfactory fit, not when `PTFit` was effectively told where the minimum in the χ^2 was.

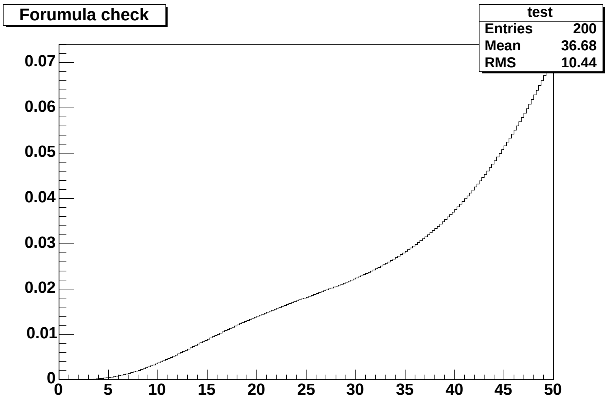

Figure 70: The functional form, (26) as generated with the parameters used to create the `ptz_4vec.dat` data file.

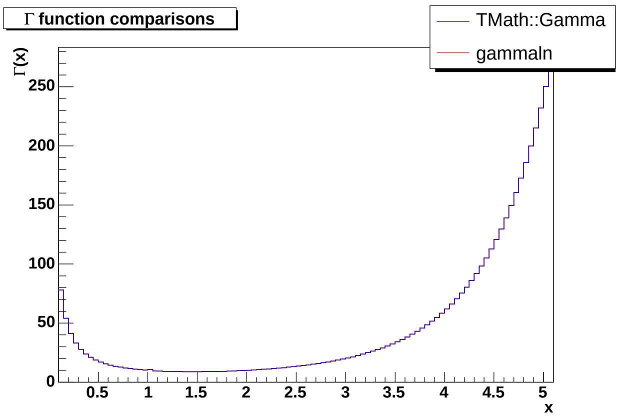

Figure 71: A comparison of Γ functions: `TMath::Gamma` compared with `TMath::Exp(gammaln)`.

	Value	Error	Starting value
P1	0.5135	63.9611	0.5135
P2	6.45936	5.86499	0.2767
P3	7.8193	5.05797	7.819
P4	0.417	2.14856	0.4170

Table 1: The four parameters in (26) that best fit the data in `ptz_4vec.dat` generated using `040305_findparam.cc`

22 Week 22

22.1 Monday, March 8, 2004

12:00pm–5:00pm

Having no idea what the problems with the fitting are, the χ^2 function was changed. The functional form was binned into a histogram the same size as the one that contained the data. Both were then normalised to unity via a summation of the bin contents, and the χ^2 calculated for the two histograms. This method produced a much more satisfactory fit that before, with `PTFit` being able to find a minimum when given (0.5,0.5,10,0.5) as input parameters (with `Migrand` using a strength of '1'). The values returned were then used as starting parameter for `Migrand` running in a more fussy mode (a strength of '2' and making sure that the minimum and errors were as accurate as possible). The parameters returned, table 22.1 were much closer to the actual parameters used to generate the data, than the results last Friday, table 21.4.

	Value	Error	Starting value
P1	0.57532	35.6052	0.5
P2	0.236938	29.1206	0.5
P3	7.92869	37.057	10
P4	0.417341	4.2766	0.5

Table 2: The four parameters in (26) that best fit the data in `ptz_4vec.dat` generated using `040308_findparam.cc`

The error do seem large, and looking at the code, the data has not been normalised properly (forgotten to sum over the area of the bins). The corrected code produces:

	Value	Error	Starting value
P1	0.572916	19.3792	0.5
P2	0.234915	15.5454	0.5
P3	7.92695	9.85936	10
P4	0.417133	1.08426	0.5

Table 3: The four parameters in (26) that best fit the data in `ptz_4vec.dat` generated using `040308_findparam.cc` with the correct normalisation.

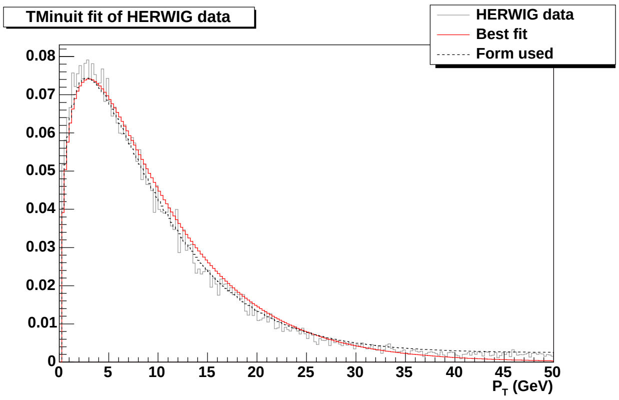

Figure 72: The best fit functional form (using the parameters generated with `040305_findparam.cc`) compared with the input $P_T(Z)$ HERWIG data (as reconstructed from the electronic decay products of the Z^0)

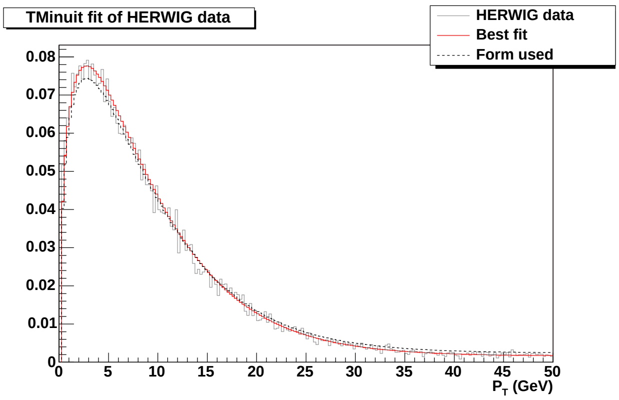

Figure 73: The best fit functional form (using the parameters generated with `040308_findparam.cc`) compared with the input $P_T(Z)$ HERWIG data (as reconstructed from the electronic decay products of the Z^0)

Figure 73 shows the new fit, which looks much more appropriate than figure 72. Yet the error still seem large, their effect on the functional form will be looked into next.

22.2 Tuesday, March 9, 2004

1:30pm–5:00pm

The CEM resolution effect needs to be built into the fitting program. The functional form needs to be smeared (to account for detector effects) before it is fitted against data coming from the CDF. The preferable way would be to analytically convolute (26) with the smearing function, but this it too demmanding and time costly. Instead, HERWIG data will be used to estimate the effect of CEM smearing.

HERWIG data for the electron and position will be used to reconstruct the Z^0, but the particles will have their energies and momenta smeared before recontruction to produced a smeared Z^0. Thus, histograms of smeared and unsmeared $P_T(Z)$ can be produced. These will be normalised to unity, and then compared. By taking the ratio of each histogram to one another, bin by bin, one can create an array that will contain an estimate as to how unsmeared P_T data can be smeared; the functional form histogram can be smeared by multiplying each bin by the respective smearing ratio, then the smeared form can be fitted to the CDF data.

The smearing will be applied as earlier in the project.

$$E(\kappa) = E^{obs} + R\Delta E \tag{30}$$

$$P_i(\kappa) = P_i^{obs} + R\Delta P_i \tag{31}$$

Where $i = x, y$ for the components in the transverse plane. While the input files do not contain the energy of the electons, this can be easily worked out taking the contribution of the momentum and mass. In both the above equations, R is a random number with a Gaussian distribution (mean of 0 width of 1). The magnitude of the uncertainty in the data can be determined using:

$$\Delta E = \sqrt{\frac{0.135^2 P}{E P_T} + \kappa^2} \tag{32}$$

$$\Delta P_i = \frac{E\Delta E}{\sqrt{E^2 - m_0^2}} \tag{33}$$

It would be preferable to fit to some actual experimental data, but one does not have any to use at this time. To test the code, the smeared functional form will be fitted against smeared HERWIG data, figure 74.

(4:30pm)

Spoke to Dr. Lancaster; found out that there was a file of actual experimental data to fit, which will be done next. He was uncertain about how good the fit was, but using more events will reduce the statistical uncertainty in the data, allowing for a more accurate assessment.

10:00pm–11:00pm

The file of actual data, `zee_pxpypx.dat`, is causing problems with the fit. After searching, the problem was identified: there were so few events in the test file (appox. 1200) that some bins at the higher end of the P_T scale were empty. This would have given a partial χ^2 value of ∞, meaning that `TMinuit` was trying to find the minimum of a funciton that was returning ∞ as a minimum. To solve the problem, all the bins will have one event added to them, but only for this particular file. The problem should not crop up again for data with more events.

22.3 Wednesday, March 10, 2004

11:00pm–5:45pm

The `ControlCards` feature is to be built into the code, so that it does need to be recompiled every time a variable is changed. In addition, the value of χ^2 is put on the graphs generated to show the goodness of fit.

So, far the progam was being run with a κ of 1.9 (as derived from the `Fitter` best fit on the M_Z distribution). This value was found to be too inaccurate at the time, thus 1.742 is being used instead (a value that Dr. Lancaster had determined in previous work).

The `TMinuit` output with the corrected κ:

```
FCN=50.5253 FROM MIGRAD    STATUS=CONVERGED    146 CALLS    147 TOTAL
                    EDM=1.9371e-05    STRATEGY= 1   ERROR MATRIX ACCURATE
 EXT PARAMETER                                  STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  P1           1.18738e-01   1.73392e+00   4.75165e-04   2.54345e-03
  2  P2           1.15151e-02   1.38219e-01   3.81889e-05   1.19211e-02
  3  P3           6.04338e+00   4.73643e-01   5.47698e-04   4.22782e-03
  4  P4           4.08365e-01   6.32559e-02   1.04494e-04  -1.95132e-02
```

Figure 75, shows the smeared functional form fitted against the CDF data. The deterctor simulation does smear the data as expected, the but the resultant form is crude. This can be rectified by using more HERWIG events to smooth out the statistical uncertainties.

Dr. Lancaster has provided a file with 4.6 million HERWIG events for the detector simulation. Hopefully, this will produce a smoother smerared form. In addition the χ^2 function now ignores any bins that have no entries, so that an empty bin does not caush the fitter to crash. He commented that the fit looks as if it is off by a bin. After inspection of the code, a bug was identified and corrected for: when plotting the final best fit curve, the lower bin edge was being used as the value of the bin, thus, the first bin would use '0' and be empty as a result. The code was changed to use the bin centres when generating the final curve, which gives a more pleasing result (it should be noted that this was mearly a problem with prodicing a plot at the end of the fitting, it does not affect the fitting, and the bug does not apply to the fitting).

The output from `TMinuit` with the correct χ^2 and increased HERWIG events:

```
FCN=45.4475 FROM MIGRAD    STATUS=CONVERGED     142 CALLS    143 TOTAL
                    EDM=1.05303e-05    STRATEGY= 1  ERROR MATRIX ACCURATE
 EXT PARAMETER                                  STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  P1           1.20890e-01   2.26231e+00   5.69955e-04   1.05104e-03
  2  P2           9.67534e-03   1.48348e-01   3.95979e-05   8.25396e-03
  3  P3           6.03412e+00   4.70683e-01   5.15226e-04   1.09825e-02
  4  P4           4.17118e-01   6.42726e-02   9.87493e-05  -8.12944e-02
```

As can be seen in figure 76 the smeared form is much smoother than figure 75, increasing the number of events being used in the simulation has removed the fluctuations from the curve. The total number of bins has be written into the legend so that the χ^2 value has meaning.

22.4 Thursday, March 11, 2004

12:00pm–

There is a tiny error in the χ^2 function, there are less data points than they are bins in the histogram if some of the bins are empty. The code was changed to keep track of the empty bins and to take this into account when displaying the χ^2 goodness on the graph.

A more serious error has arisen: the smearing has been applied incorrectly, the error in the energy should be:

$$\Delta E = E\sqrt{\frac{0.135^2 P}{E P_T} + \kappa^2} \tag{34}$$

When this is applied to the code, the energy smearing is far too large. The reason for this is κ: at the size of κ one is using (1.742) then its affect on the energy would be an uncertainty of 170% of the input energy, this cannot

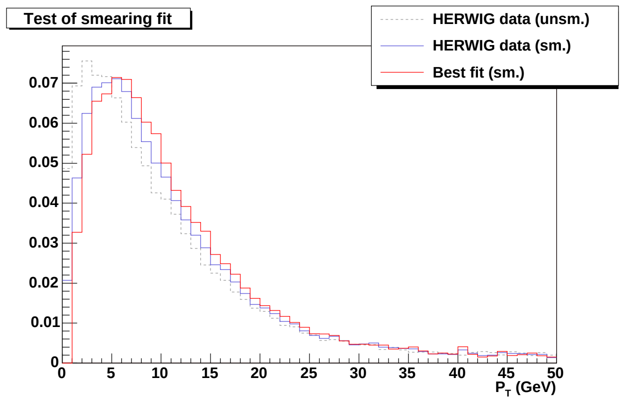

Figure 74: The smeared funtional form fitted to smeared HERWIG data.

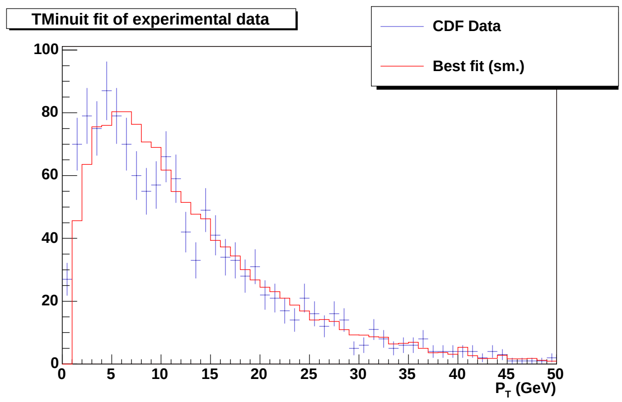

Figure 75: The smeared functional form fitted to $P_T(Z)$ data from the CDF, the smearing is carried out by a detector simulation using approximately 60,000 HERWIG-generated events.

be the case! After consultation with one of the original CDF papers [5], it is obvious that 1.742 represents 1.742%, and that the value to be used in the code is 0.01742. The correct fit was generated:

```
FCN=38.9175 FROM MIGRAD    STATUS=CONVERGED     158 CALLS    159 TOTAL
                  EDM=2.45823e-06    STRATEGY= 1  ERROR MATRIX ACCURATE
 EXT PARAMETER                                  STEP         FIRST
 NO.   NAME      VALUE           ERROR          SIZE      DERIVATIVE
  1  P1           1.32150e-01   1.66751e+00   2.81498e-04  -3.25119e-03
  2  P2           2.99996e-02   2.78857e-01   4.79196e-05   2.68518e-02
  3  P3           6.81461e+00   4.73778e-01   5.16074e-04   1.63663e-03
  4  P4           6.02520e-01   6.64390e-02   9.77457e-05  -1.81156e-02
```

The correct best fit, figure 77, has a 'better' peak than the previous plots, more than likely, this is because the dectector simulation has been applied correctly and the correct κ value was used.

23 Week 23

23.1 Monday, March 15, 2004

11:00pm–6:00pm

Wanted to check the goodness of fit for the unsmeared form. The code was edited to that the χ^2 fit calculated a smeared and unsmeared fit. The resultant figure, figure 78 shows that the simulation is actually worse than the unsmeared fit. However, the CDF data is not well defined in the peaked region, inspection by eye would suggest that the smeared form would fit the data better if there was more data to smooth out the fluctuations in the CDF data. Only with more data could one really say whether the simulation is useful or not.

Began work on the fit to find the M_W error[34]. The fit will involve using the covariant matrix from `TMinuit` and creating a new set of parameters by diagonalising the matrix. This will be used, with an external file, to reweight the $P_T(Z)$ form into a $P_T(W)$ form, and then to derive a transverse mass. Repeating this 10000 times and plotting the masses will enable a mass uncertainty to be estimated.

(3:40pm)

Having a bad time trying to get the FORTRAN matrix diagonalisation method to work. Researching for project talk instead.

(5:10pm)

Decided that the current implementation of the detector simulation was flawed. Changed it show that every time a particle has a property smeared, a new random number is used. The output from `TMinuit`:

```
FCN=39.572 FROM MIGRAD    STATUS=CONVERGED     142 CALLS      143 TOTAL
                  EDM=3.64816e-06    STRATEGY= 1
   ERROR MATRIX UNCERTAINTY   6.3 per cent
 EXT PARAMETER                                  STEP         FIRST
 NO.   NAME      VALUE           ERROR          SIZE      DERIVATIVE
  1  P1           1.31104e-01   1.02751e-02   5.29646e-05  -2.47900e-02
  2  P2           3.01858e-02   2.62166e-02  -1.58396e-04   1.52558e-01
  3  P3           6.82980e+00   4.89461e-01  -2.30229e-03  -1.18655e-02
  4  P4           6.07037e-01   6.92533e-02  -2.66912e-05   2.19748e-03

FCN=39.572 FROM MIGRAD    STATUS=CONVERGED      72 CALLS       73 TOTAL
                  EDM=9.22771e-07    STRATEGY= 2 ERROR MATRIX ACCURATE
 EXT PARAMETER                                  STEP         FIRST
 NO.   NAME      VALUE           ERROR          SIZE      DERIVATIVE
  1  P1           1.31178e-01   1.60112e+00   2.88840e-04  -5.76760e-04
  2  P2           3.02028e-02   2.70505e-01   4.87924e-05   8.46226e-03
  3  P3           6.83030e+00   4.72813e-01   5.19942e-04  -1.71109e-04
  4  P4           6.07108e-01   6.63039e-02   9.82685e-05  -2.29328e-04
```

[34] all code in `4C00_mwfit` directory

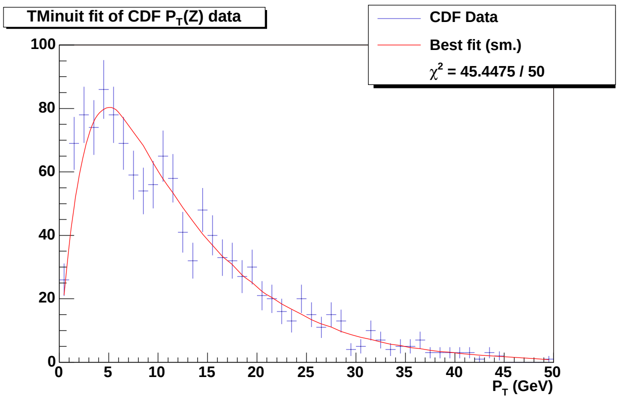

Figure 76: The smeared functional form fitted to $P_T(Z)$ data from the CDF, the smearing is carried out by a detector simulation using approximately 4.6 million HERWIG-generated events.

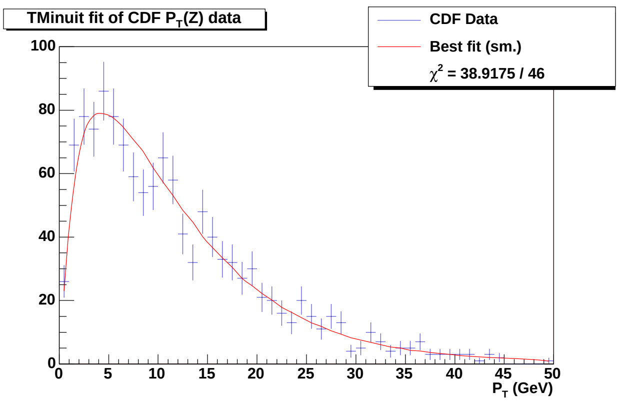

Figure 77: The smeared functional form fitted to $P_T(Z)$ data from the CDF, the smearing is carried out by the corrected detector simulation using approximately 4.6 million HERWIG-generated events.

23.2 Sunday, March 21, 2004

(8:20pm)

Changed the `Fitter` program so that it compares histograms bin by bin. This solved the problem with χ^2 fitting in the other `TMinuit` work and should work here. The program now converges upon a nicer result than before:

```
FCN=123.398 FROM MIGRAD    STATUS=CONVERGED      66 CALLS       67 TOTAL
                     EDM=1.87797e-07  STRATEGY=1    ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  Mz           9.11241e+01   1.30046e-02   6.95451e-05   1.39657e-02
   2  Kappa        1.50650e+00   2.36782e-01   1.31271e-03  -2.22785e-03
```

χ^2 123.383 (139) 170 bins (70–110)
χ^2 85.4373 (77) 85 bins (70–110)

Fitting over only the 80–100 GeV region:

```
FCN=76.8078 FROM MIGRAD    STATUS=CONVERGED      67 CALLS      68 TOTAL
                     EDM=6.34657e-08  STRATEGY=1   ERROR MATRIX ACCURATE
 EXT PARAMETER                                   STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  Mz           9.11229e+01   1.36073e-02   5.70814e-05   7.74096e-03
  2  Kappa        1.51052e+00   2.29922e-01   1.01381e-03  -1.32762e-03
```

χ^2 76.1488 (79) 130 bins (80–100)
χ^2 47.1093 (40) 45 bins (80–100)

23.3 Thurdsay, March 25, 2004

In conclusion:

The project has shown that the transverse momentum of the Z^0 can be used to represent the $W^{\pm}$. This was done by analysing Monte Carlo events generated by HERWIG.

The effect of the resolution of the CDF's Central Electromagnetic calorimeter upon various properties of the Z^0 was also investigated upon the HERWIG data.

How the detector simulation depended upon κ_{CEM} was investigated, and found to be significant. In order to find this κ_{CEM}, a bicubic interpolation fit was made to find the rest mass distribution of the Z^0 that contained the κ_{CEM} that best described the data. This talk was not as successful as hoped. A value of κ_{CEM} was found to be $1.5105\% \pm 0.2299\%$ but it did not compare too favourably with the expected value of 1.742%. This was more than likely the result of a lack of experience in C++ programming; a person more experienced in the language would have built exception handling into the program from start, which may have reduced the time spent chasing bugs and errors.

The value of κ_{CEM} was used to complete the modelling the CEM, and this was used with an ad hoc functional form to build a theoretical prediction for the observed transverse momentum of the Z^0. This was fitted against CDF data to find the four parameters that generated the best form; these parameters were reasonable with believable errors. The χ^2 for the fit was respectable, and it passed a 76% significance test.

A estimate for the corresponding uncertainty on the measured transverse mass of the $W^{\pm}$ was not completed; there was not enough time to complete the program, and integrating the Fortran subroutines into C++ was tricky. With more experience of Fortran, the analysis should have been completed, it was the implementation that cause the delay in working.

Is believed that whether the CEM simulation is valid, or not, could only be determined with more transverse momenta data from the CDF. The χ^2 fits of the smeared and original functional form both fit the data, but the error bars on the data are large enough to allow this.

Throughout the project work was hampered by bug and other programming related issues. Given that C++ was an unfamiliar language at the start of the project, progress was slower than liked when faced with difficult programming tasks. However, the language was picked up, with analysis programs were being written from scratch and could generate histograms that could be viewed in Root immediately.

It is hoped that the final section of the project could have been completed within a fortnight if it were to be continued.

Modelling some of the other detector, such as the Plug EM calorimeter, would improve the simulation; the effect of κ_{PEM} was significant when the PEM was modelled along with the CEM. Taking other detector effects into account would also be beneficial.

References

[1] DØ Collaboration, B Abbot *et al.*, *Phys Rev. D*, **61**, 032004 (2000)

[2] M Lancaster and D Waters, *J. Phys. G: Nucl. Part. Phys.*, **26**, (2000)

[3] CDF Collaboration, T Affolder *et al.*, *Phys Rev. Lett.* (1999)

[4] E Gerchtein and M Paulini, "CDF Detector Simulation Framework and Performance" *Computing in High Energy and Nuclear Physics* (2003

[5] L Balka *et al.*, "The CDF central electromagnetic calorimeter " *Nucl.Inst.Phys.A*, **267**, 272–279 (1988)

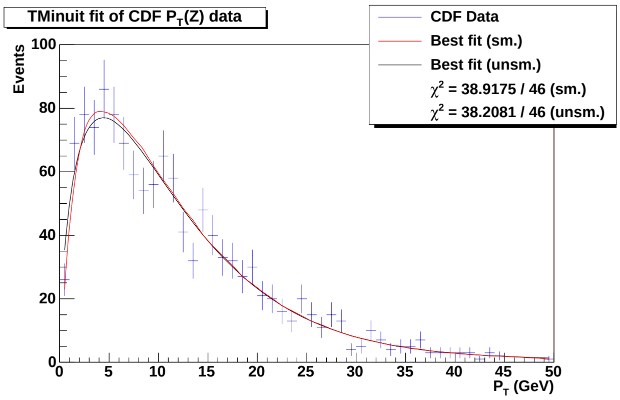

Figure 78: The smeared and unsmeared functional forms fitted to $P_T(Z)$ data from the CDF, the smearing is carried out by the simulation using approximately 4.6 million HERWIG-generated events.