

Z^0 - Resonanz

Fortgeschrittenen Praktikum 2

Antonia Strübig und Nena Milenkovic

12. - 23. Oktober 2009

Inhaltsverzeichnis

1	Versuchsziel	3
1.1	Aufgabenstellung	3
2	Theorie	4
2.1	Elementarteilchen	4
2.1.1	Das Photon	4
2.1.2	Die Leptonen	4
2.1.3	Die Quarks	5
2.2	Wechselwirkungen bei hohen Energien	5
2.3	Elektroschwache Wechselwirkung	6
2.4	Wirkungsquerschnitt und Zerfallsbreiten	6
2.5	Vorwärts-Rückwärts-Asymmetrie A_{FB}	7
2.6	Elektron-Positron-Wechselwirkung	7
2.6.1	Bhabha-Streuung	7
2.6.2	Annihilation in Fermionenpaare	8
2.6.3	Strahlungskorrekturen	8
2.7	Der OPAL-Detektor	9
2.7.1	Aufbau des Detektors	9
2.7.2	Signaturen der Zerfallsereignisse	10
3	Auswertung	13
3.1	Theoretisch zu berechnende Werte	13
3.2	Analyse der Ereignisse mit GROPE	15
3.3	Analyse der Monte-Carlo-Daten	21
3.4	Analyse der echten OPAL-Daten	25
3.4.1	s-t-Kanal-Trennung	27
3.4.2	Berechnung der Z ⁰ -Parameter	29
3.4.3	Leptonenuniversalität	31
3.4.4	Anzahl Neutrino-Familien	31
3.4.5	Vorwärts-Rückwärts-Asymmetrie	32
4	Zusammenfassung	33
A	C++ Quellcode für ROOT	34

1 Versuchsziel

Im Z⁰-Versuch sollen die Masse und die Zerfallsbreite des Vektorbosons Z⁰ bestimmt werden. Anhand von echten Cern-Daten sollen grundlegende Gesetze des sogenannten Standardmodells überprüft werden. Außerdem soll man die Leptonen-Universalität nachweisen, die Anzahl der leichten Neutrinofamilien bestimmen und die Vorwärts- Rückwärts-Asymmetrie messen.

1.1 Aufgabenstellung

Es gibt zwei Aufgabenteile, im ersten Teil sollen verschiedenen theoretische Werte berechnet werden:

- Berechnen Sie die Zerfallsbreiten für die verschiedenen Fermionpaare und vergleichen Sie diese mit den in der Anleitung angegebenen Werten.
- Berechnen Sie die folgenden Größen: Gesamtbreite, hadronische Breite, “geladene“ leptonische Breite, “neutrale“ leptonische Breite (unsichtbare Breite), partielle Wirkungsquerschnitte am Resonanzmaximum.
- Um wieviel Prozent würde sich die Breite der Z⁰-Resonanz ändern, wenn der Zerfall in ein weiteres leichtes Fermionenpaar möglich wäre?
- Zeichnen Sie die erwarteten Formen der Winkelverteilungen für die Prozesse $e^+e^- \rightarrow e^+e^-$ und $e^+e^- \rightarrow \mu^+\mu^-$ auf. Trennen Sie im Falle von $e^+e^- \rightarrow e^+e^-$ die einzelnen Beiträge voneinander.
- Berechnen Sie die Vorwärts-Rückwärts-Asymmetrie für den Prozeß $e^+e^- \rightarrow \mu^+\mu^-$ bei $\sqrt{s} = 91.2, 89.2$ und 93.2 GeV.

Im zweiten Teil geht es um die Verarbeitung der Daten:

- Analysieren der Ereignisse mit dem Programm GROPE, um charakteristische Eigenschaften der verschiedenen Zerfallsereignisse des Z⁰ zu bestimmen.
- Bestimmen von Schnittkriterien zur Trennung der verschiedenen Ereignisse und Überprüfen der Effizienz an den Monte-Carlo-Daten.
- Trennung von s- und t-Kanal der Elektronenereignisse, um nur die Elektronen aus den Z⁰-Zerfällen zu erhalten.
- Berechnung der partiellen Wirkungsquerschnitte der verschiedenen Ereignisse für Energien um die Z⁰-Resonanz herum.
- Berechnung der Zerfallsbreiten der verschiedenen Fermion-Paare und berechnen der Gesamtzerfallsbreite des Z⁰.

- Überprüfung der Theorie in Hinsicht auf Leptonuniversalität und Anzahl der Neutrino-Familien.
- Berechnung der Vorwärts-Rückwärts-Assymtrie der Myonen, um den Weinbergwinkel am Resonanzmaximum zu berechnen.

2 Theorie

2.1 Elementarteilchen

Teilchen lassen sich zunächst mithilfe der Quantenmechanik in Fermionen (mit halbzahligen Spin) und Bosonen (mit ganzzahligen Spin) einteilen. Die Fermionen gliedern sich weiter auf in Leptonen, Baryonen und Quarks. Bei den Bosonen sind es das Photon, die Mesonen und die Vektorbosonen W^+ , W^- und Z^0 . Auf diese Teilchen soll in den nächsten Kapiteln gesondert eingegangen werden.

2.1.1 Das Photon

Man nimmt an, dass sich ein Photon immer mit Lichtgeschwindigkeit bewegt. Formal werden dem Photon die Masse 0 und der Spin 1 zugeordnet. Jedoch ist es nicht sinnvoll von einem Spin beim Photon zu sprechen, weshalb man die Helizität einführt, eine Projektion des Spins auf die Impulsrichtung. Dieser kann die Werte +1 und -1 annehmen. Ausnahmen sind die "virtuellen" Photonen, sie tauchen nur als ausgetauschte Quanten bei einem Wechselwirkungsprozess auf und an ihre Stelle kann auch das Z^0 -Boson treten. (siehe Kapitel: 2.3)

2.1.2 Die Leptonen

Zu dieser Gruppe gehören die Elektronen, die Myonen und die Tau-Leptonen und ihre jeweiligen Antiteilchen, die Neutrinos und Antineutrinos. Die Neutrinos nehmen nur an der schwachen Wechselwirkung teil, wohingegen die geladenen Leptonen auch noch an der elektromagnetischen Wechselwirkung teilnehmen.

	Ladung Q/e	Masse m/MeV	Neutrinos	Ladung Q/e	Masse m/MeV
Elektron e^-	-1	0,551	ν_e	0	0
Myon μ^-	-1	105,66	ν_μ	0	0
Tau τ^-	-1	1777	ν_τ	0	0

Tabelle 1: Leptonen

2.1.3 Die Quarks

Es existieren heute 6 uns bekannte Quarks:

		Ladung Q/e	Masse m/MeV
Up	$ u\rangle$	+2/3	1,5-3,0
Down	$ d\rangle$	-1/3	3-7
Charm	$ c\rangle$	+2/3	1250
Strange	$ s\rangle$	-1/3	95
Top	$ t\rangle$	+2/3	170900
Bottom	$ b\rangle$	-1/3	4200

Tabelle 2: Quarks

Aus den Quarks setzen sich die Hadronen zusammen, die in Mesonen und Baryonen unterteilt werden. Mesonen bestehen aus Quark und Antiquark, Baryonen aus jeweils drei Quarks. Jedem Quark wird ein weiterer Freiheitsgrad zugeordnet, nämlich die Farbe (“colour”).

2.2 Wechselwirkungen bei hohen Energien

Es gibt 4 verschiedene Wechselwirkungen, wobei jedoch nur drei davon von der Theorie des Standardmodells beschrieben werden. Für die Gravitation existiert noch keine Theorie, sie spielt aber für unseren Versuch auf Grund der sehr geringen Massen der Elementarteilchen keine Rolle.

elektromagnetische Wechselwirkung

Die elektromagnetische Wechselwirkung betrifft alle geladenen Teilchen. Sie wird durch Austausch eines Photons übertragen und durch die Feldtheorie der Quantenelektrodynamik beschrieben.

schwache Wechselwirkung

Die schwache Wechselwirkung wirkt auf alle Elementarteilchen, sie wird durch Austausch von den Vektorbosonen W , W^+ oder Z^0 vermittelt. Da die Massen der Austauschteilchen mit 80 und 91 GeV sehr gross sind, hat die schwache Wechselwirkung nur ein sehr geringe Reichweite.

starke Wechselwirkung

Die starke Wechselwirkung koppelt an die Farbladung von Quarks und Gluonen. Die 8 verschiedenen Gluonen, sind Austauschteilchen der starken Wechselwirkung, aber da sie

selber ebenfalls Farbladung tragen, können Gluonen auch untereinander wechselwirken. Das Potential der starken Wechselwirkung wird als

$$V_q = -\frac{4}{3} \cdot \frac{\alpha_s}{r} + kr$$

geschrieben.

2.3 Elektroschwache Wechselwirkung

Auf der Suche nach einer Eichtheorie für die schwache Wechselwirkung, wurden die drei Eichbosonen W^+ , W^- und W^0 eingeführt (entspricht SU(2)-Symmetrie). Die Kopplungsstärke dieses Feldes ist g , wobei folgender Zusammenhang zur Fermi-Konstante G_F besteht:

$$G_F = \frac{\sqrt{2} \cdot g^2}{8 \cdot M_W^2} = 1,1663 \cdot 10^{-5} \text{GeV}^{-2}$$

M_W stellt hier die Masse des W-Bosons dar.

Da sich die schwache Wechselwirkung bei hohen Energien sehr ähnlich zur elektromagnetischen Wechselwirkung verhält, kamen Glashow, Salam und Weinberg auf die Idee beide Wechselwirkungen zu vereinen. Die U(1)-Symmetrie der elektromagnetischen Wechselwirkung führt auf ein weiteres Eichboson Y^0 . W^- und W^+ sind mit ihren physikalischen Teilchen identisch. Y^0 mischt mit W^0 zum Photonfeld und zum dazu orthogonalen Z^0 -Feld.

$$\begin{aligned} A &= W^0 \sin(\theta_w) + Y^0 \cos(\theta_w) \\ Z^0 &= W^0 \cos(\theta_w) + Y^0 \sin(\theta_w) \end{aligned}$$

θ_w ist der Mischungswinkel der Felder, der sogenannte Weinbergwinkel.

2.4 Wirkungsquerschnitt und Zerfallsbreiten

In diesem Versuch sollen die Wirkungsquerschnitte und Zerfallsbreiten bestimmt werden. Diese kann man aus den Streuamplituden berechnen.

$$\frac{d\sigma}{d\Omega_3} = \frac{1}{64\pi^2 s} \cdot \frac{|\vec{p}_3|}{|\vec{p}_1|} \cdot \sum |T_{fi}|^2$$

Die \vec{p}_n sind die Vierervektoren der ein- und auslaufenden Teilchen und die T_{fi} die Einträge der Streumatrix.

Für Zwei-Körper-Zerfälle erhält man als weiteres Ergebnis die Zerfallsbreite Γ aus:

$$\Gamma = \frac{|\vec{p}_1|}{8\pi M^2} \sum |T_{fi}|^2$$

Für die Herleitung sei auf [3] verwiesen.

2.5 Vorwärts-Rückwärts-Asymmetrie A_{FB}

Die Vorwärts-Rückwärts-Asymmetrie ist definiert als:

$$A_{FB} = \frac{\int_0^1 \frac{d\sigma}{d\cos\theta} d\cos\theta - \int_{-1}^0 \frac{d\sigma}{d\cos\theta} d\cos\theta}{\int_0^1 \frac{d\sigma}{d\cos\theta} d\cos\theta + \int_{-1}^0 \frac{d\sigma}{d\cos\theta} d\cos\theta}$$

Aus ihr kann man den Weinbergwinkel bestimmen. Asymmetrien sind sehr gute Messgrößen zur Betätigung der elektroschwachen Theorie.

Sie kommt dadurch zustande, dass unterhalb und oberhalb des Z⁰-Maximums die elektromagnetische Vektor- und die schwache Axial-Vektor-Wechselwirkung interferieren und am Maximum selbst die schwache Vektor-Wechselwirkung mit der Axial-Vektor-Wechselwirkung interferieren.

Bei Z⁰-Resonanz ergibt sich für den oberen Term:

$$A_{FB,max} = 3 \left(\frac{g_v}{g_a} \right)^2 = 3 \left(\frac{I_3 - 2\sin^2(\theta_w)}{I_3} \right)^2$$

2.6 Elektron-Positron-Wechselwirkung

2.6.1 Bhabha-Streuung

Die Bhabha-Streuung bezeichnet die elastische Elektron-Positron-Streuung ($e^+e^- \rightarrow e^+e^-$). Es sind hier mehrere Streuprozesse möglich. Zum einen die Annihilation in Photonen und die Streuung durch Austausch eines virtuellen Photons. Berücksichtigt man die schwache Wechselwirkung, vermittelt statt eines Photons das Z⁰-Boson. Also ergeben sich insgesamt 3 Streuprozesse:

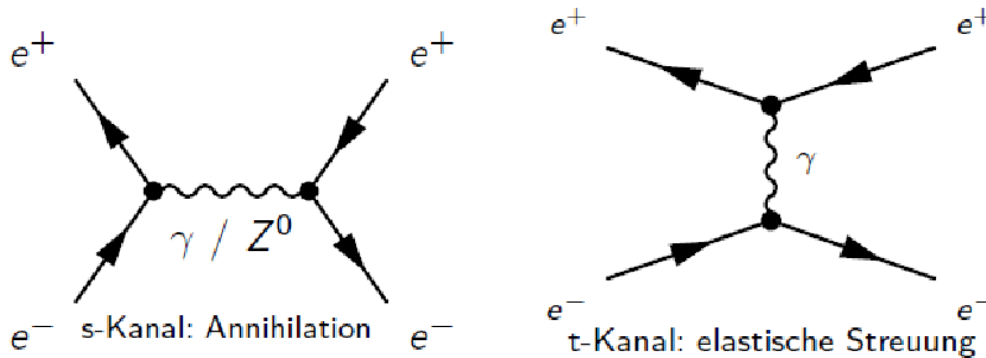


Abbildung 1: Titel

Die Annihilationsprozesse werden als s-Kanal- und die Streuprozesse als t-Kanal-Reaktionen bezeichnet. Für diesen Versuch ist es wichtig die s-Kanal-Ereignisse zu selektieren, da nur hier reelle Z⁰-Bosonen erzeugt werden. Man nutzt hierfür die Winkelabhängigkeit der differentiellen Wirkungsquerschnitte aus. Denn für kleine Streuwinkel θ dominieren die t-Kanal-Reaktionen ($\frac{d\sigma}{d\Omega} \propto (1 + \cos^2\theta)$) und für große die s-Kanal-Reaktionen ($\frac{d\sigma}{d\Omega} \propto (1 - \cos\theta)^{-2}$).

2.6.2 Annihilation in Fermionenpaare

Es sind durchaus auch Prozesse möglich mit beliebigen Fermion-Antifermion-Paaren im Endzustand. Dies kann durch den Austausch eines Photons oder eines Z⁰-Bosons erfolgen, also s-Kanal-Ereignissen.

Uns interessieren nur Ereignisse mit Energie nahe der Z⁰-Masse. Der Wirkungsquerschnitt wird hier durch den Z⁰-Austausch Term dominiert, der die typische Breit-Wigner-Form für den Austausch eines Spin=1 Teilchens hat. Für die Partialbreite erhält man dann:

$$\begin{aligned}\Gamma_f &= \frac{\sqrt{2}}{12\pi} \cdot N_c^f \cdot G_F \cdot M_Z^2 \cdot [(g_V^f)^2 + (g_A^f)^2] \\ &= \frac{\sqrt{2}}{12\pi} \cdot N_c^f \cdot G_F \cdot M_Z^2 \cdot [(I_3^f - 2 \cdot Q_f \cdot \sin^2 \theta_w)^2 + (I_3^f)^2]\end{aligned}$$

mit

$$\begin{aligned}N_c^f &= \text{Anzahl Farbladung}, \quad M_Z = \text{Masse } Z^0, \quad I_3^f = \text{Isospin}, \\ Q_f &= \text{Fermionladung} \quad \text{und} \quad \theta_w = \text{Weinbergwinkel}\end{aligned}$$

Die Gesamtbreite des Z⁰-Bosons ergibt sich dann aus der Summe der Partialbreiten:

$$\Gamma_Z = \Gamma_e + \Gamma_\mu + \Gamma_\tau + \Gamma_h + \Gamma_\nu$$

Die Wirkungsquerschnitte errechnen sich aus:

$$\sigma_f(s) = \frac{12 \cdot \pi}{M_Z^2} \cdot \frac{s \Gamma_e \Gamma_f}{(s - M_Z^2)^2 + (s \frac{\Gamma_Z}{M_Z})^2}$$

Und am Resonanzmaximum ($s = M_Z^2$):

$$\sigma_f^{peak} = \frac{12\pi}{M_Z^2} \cdot \frac{\Gamma_e}{\Gamma_Z} \cdot \frac{\Gamma_f}{\Gamma_Z}$$

Die genaue Herleitung entnimmt man erneut aus [3].

2.6.3 Strahlungskorrekturen

Die gemessenen Daten müssen noch auf Verluste aufgrund von Detektorineffizienzen oder Selektionskriterien korrigiert werden, bevor man sie mit der Theorie vergleichen kann. Zudem müssen noch Strahlungskorrekturen berücksichtigt werden, man unterscheidet reelle, virtuelle und QCD-Korrekturen. Die reellen Strahlungsprozesse setzen sich aus der Anfangsbremsstrahlung, der Endbremsstrahlung und deren Interferenz zusammen. Virtuelle Strahlungsprozesse sind im Gegensatz zu Bremsstrahlungskorrekturen durch den gleichen Endzustand wie im Falle der Born-Näherung charakterisiert. Für hadronische Endzustände müssen auch Gluonenabstrahlungsprozesse berücksichtigt werden.

Da die Berechnung der Strahlungskorrekturen sehr aufwendig ist, werden uns diese in einer Tabelle angegeben.

2.7 Der OPAL-Detektor

Die Daten, die wir zur Verarbeitung erhielten, stammen vom OPAL-Detektor am LEP-Speicherring des CERNs. Wir werden deswegen kurz „unseren“ Versuchsaufbau beschreiben (Abb.2).

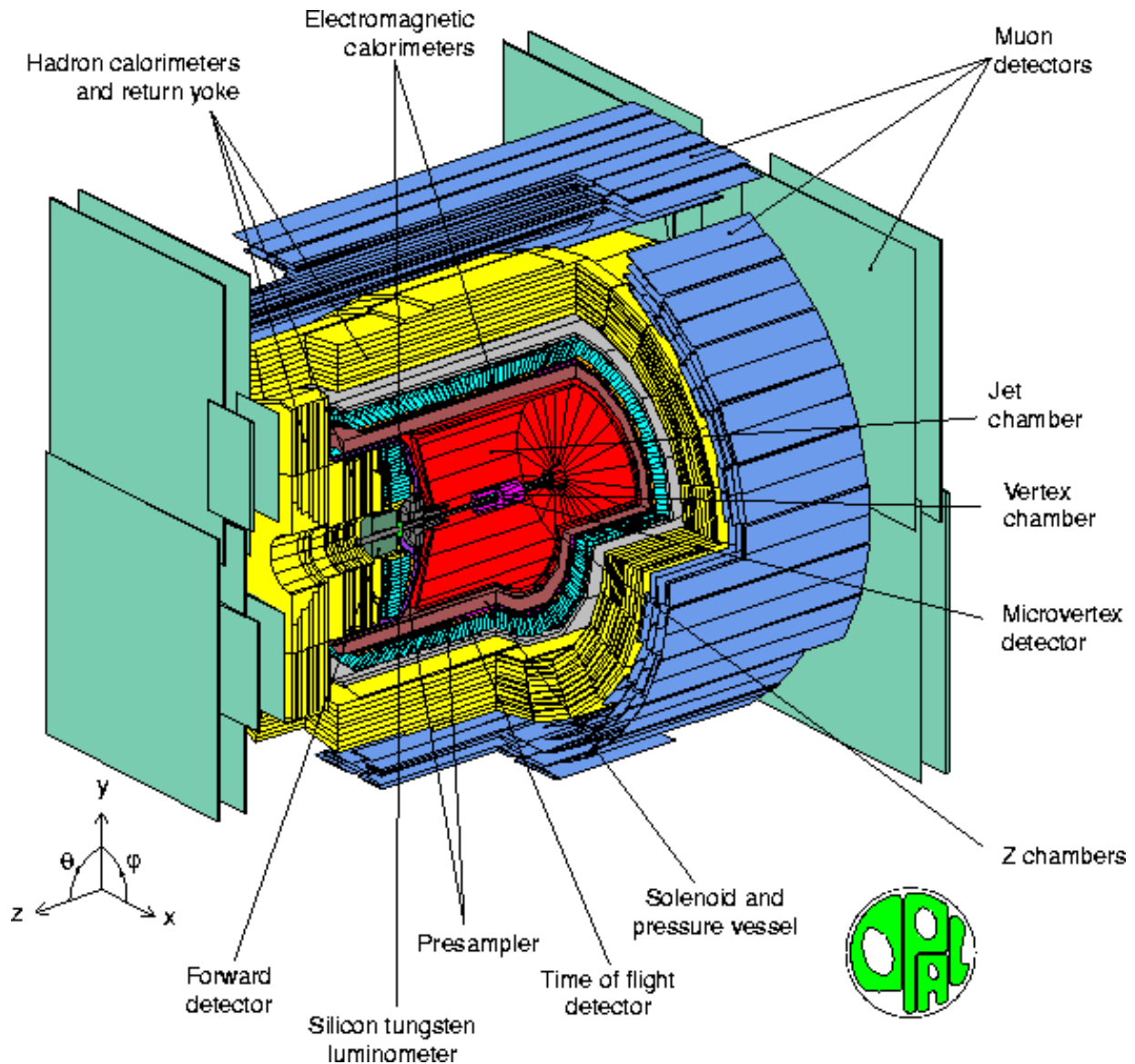


Abbildung 2: Schematische Darstellung des OPAL-Detektors

2.7.1 Aufbau des Detektors

Der Detektor besteht aus vielen verschiedenen Komponenten, die es ermöglichen alle Arten von Zerfällen zuzuordnen. Die Komponenten werden ihrer Reihenfolge nach von innen nach

aussen beschrieben.

- **μ -Vertex-Detektor und Vertex-Kammer**

Der μ -Vertex-Detektor liegt direkt an der Strahlachse und ist ein Sziliziumdetektor. Die anschliessende Vertex-Kammer ist ein Vieldrahtproportionalzähler. Beide Komponenten haben ein sehr hohes Auflösungsvermögen.

- **Jet-Kammer**

Die Jet-Kammer besteht aus vielen Zähldrähten, die parallel zur Strahlachse angeordnet sind. Sie nimmt den Energieverlust der Teilchen auf und wird ausserdem zur Spurrekonstruktion benutzt.

- **Z-Kammer**

Die 24 Z-Kammern sind ebenfalls Vieldrahtproportionalzählrohre, deren Drähte radial angeordnet sind und somit eine sehr gute Auflösung in z-Richtung liefert.

- **Solenoidspulen**

Um den Zentraldetektor wird ein starkes Magnetfeld angelegt, welches ermöglicht die Impulse der Teilchen zu bestimmen. Zudem kann man es für eine einfache Ladungszuordnung benutzen.

- **Time-of-Flight-System**

Dieses System wird zur Bestimmung der Flugzeit der Teilchen und zum Triggern des Detektors benutzt.

- **Elektromagnetisches Kalorimeter**

Das elektromagnetische Kalorimeter besteht aus Bleiglasblöcken und wird zur Bestimmung der Energie und Position von elektromagnetischen Schauern benutzt.

- **Hadronisches Kalorimeter**

Das hadronische Kalorimeter ermöglicht die Messung der Position und Energie von hadronischen Schauern.

- **Myon-Kammern**

Die Myon-Kammern dienen zur Detektion von Myonen, die nicht mit den anderen Komponenten wechselwirken und ungehindert den eigentlichen Detektor verlassen.

- **Vorwärtsdetektor**

Der Vorwärtsdetektor misst mittels Bhabha-Streuung die Luminosität des Strahls.

2.7.2 Signaturen der Zerfallsereignisse

Da wir die Detektordaten zu den verschiedenen Ereignissen zuordnen sollen, ist es für uns von besonderem Interesse zu wissen, welche Spuren die verschiedenen Ereignisse in den Detektorkomponenten hinterlassen. Abb. 3 zeigt schematisch die Signaturen, der von uns untersuchten Teilchen.

Geladene Teilchen zeichnen sich generell durch ihre gekrümmte Spur in der Jet-Kammer aus.

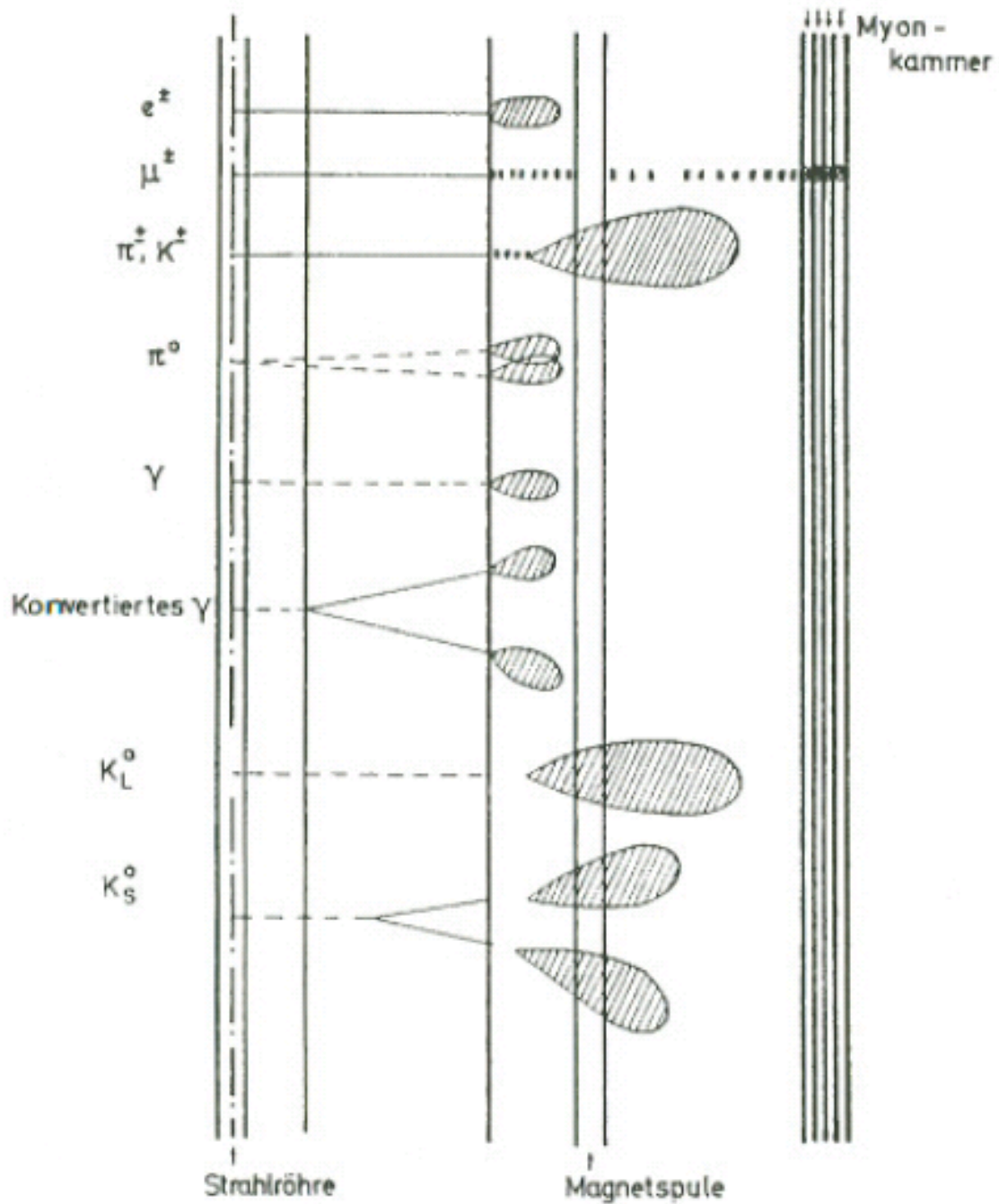


Abbildung 3: Signaturen verschiedener Teilchen im OPAL-Detektor

- **Elektronen und Positronen**

e^- und e^+ geben ihre gesamte Energie im elektromagnetischen Kalorimeter ab.

- **Myonen**

Myonen geben nur sehr wenig Energie in Kalorimetern ab und erzeugen als einzige Teilchen ein Signal in den Myon-Kammern.

- **Geladene Pionen und Kaonen**

Diese Teilchen beginnen als Schauer im elektromagnetischen Kalorimeter, der sich dann bis ins hadronische Kalorimeter fortsetzt.

- **Tau-Leptonen**

Tauonen zerfallen sehr schnell, meist in Pionen und Neutrinos. Somit erzeugen sie eine ähnliche Spur wie Pionen, aber es fehlt auf Grund der Neutrinos Transversalimpuls. Neutrale Teilchen hinterlassen keine Spur in der Jet-Kammer.

- **Photonen**

Photonen haben eine ähnliche Signatur wie e^- , e^+ , nur mit fehlender Spur in der Jet-Kammer. Zudem können sie bei sehr hohen Energien in ein $e^- - e^+$ - Paar konvertieren. Dann werden sie durch zwei Spuren, die V-förmig auseinandergehen und zwei Schauer im elektromagnetischen Kalorimeter identifiziert.

- **Neutrale Pionen**

π^0 zerfällt sofort in zwei Photonen, die V-förmig auseinanderfliegen und zwei Schauer im elektromagnetischen Kalorimeter erzeugen. Es dürfen dabei keine Spuren in der Jet-Kammer gefunden werden.

- **Neutrale Hadronen**

Neutrale Hadronen werden durch ihren hadronischen Schauer identifiziert, auf den keine Spur zeigt.

- **Neutrinos**

Neutrinos können durch OPAL nicht detektiert werden, da sie nur schwach wechselwirken. Allerdings kann man das Auftreten von Neutrinos durch fehlenden Transversalimpuls nachweisen.

3 Auswertung

3.1 Theoretisch zu berechnende Werte

Zerfallsbreiten

Die theoretischen fermionischen Zerfallsbreiten berechnen sich aus (s. Kap.2.6.2):

$$\Gamma_f = \frac{\sqrt{2}}{12\pi} \cdot N_c^f \cdot G_f \cdot M_Z^3 \cdot \left[\left(I_3^f - 2Q_f \sin^2(\theta_W) \right)^2 + \left(I_3^f \right)^2 \right]$$

In Tab.3 finden sich die Werte, die wir eingesetzt haben.

Fermion	Q _f	$g_A^f = I_3^f$	$g_V^f = I_3^f - 2Q_f \cdot \sin^2 \theta_w$
e ⁻ , μ ⁻ , τ ⁻	-1	-1/2	-1/2+2sin ² θ _w
ν _e , ν _μ , ν _τ	0	1/2	1/2
u, c	2/3	1/2	1/2-4/3 sin ² θ _w
d, s, b	-1/3	-1/2	-1/2+2/3 sin ² θ _w

Tabelle 3: Werte für die theoretische Berechnung der Zerfallsbreiten

Wir erhielten damit folgende Werte:

$$\Gamma_e = \Gamma_\mu = \Gamma_\tau = 83,39 MeV$$

$$\Gamma_u = \Gamma_c = 284,67 MeV$$

$$\Gamma_d = \Gamma_b = \Gamma_s = 366,92 MeV$$

$$\Gamma_{\nu_e} = \Gamma_{\nu_\mu} = \Gamma_{\nu_\tau} = 165,45 MeV$$

$$\Gamma_{geladeneLeptonen} = \Gamma_e + \Gamma_\mu + \Gamma_\tau = 250,19 MeV$$

$$\Gamma_{neutraleLeptonen} = 3 \cdot \Gamma_{\nu_e} = 496,35 MeV$$

$$\Gamma_{hadr.} = 2 \cdot \Gamma_u + 3 \cdot \Gamma_d = 1670,11 MeV$$

somit ergibt sich für die theoretische Gesamtbreite des Z⁰:

$$\Gamma_Z = \Gamma_{geladeneLeptonen} + \Gamma_{neutraleLeptonen} + \Gamma_{hadr.} = 2,417 GeV$$

Die theoretischen Wirkungsquerschnitte am Resonanzmaximum von Z⁰ berechnet man mittels

$$\sigma_f = \frac{12\pi \Gamma_e \Gamma_f}{M_Z^2 \Gamma_Z^2}$$

Wir berechnen die theoretischen Werte zu

$$\sigma_e = \sigma_\mu = \sigma_\tau = 2,1024 nb$$

$$\sigma_{\nu_e} = \sigma_{\nu_\mu} = \sigma_{\nu_\tau} = 4,1716nb$$

$$\sigma_u = \sigma_c = 7,1766nb$$

$$\sigma_d = \sigma_s = \sigma_b = 9,2514nb$$

Wäre es möglich, dass Z⁰ in ein weiteres leichtes Fermionenpaar zerfällt, würde sich die Zerfallsbreite um die jeweilige Fermionenzerfallsbreite erhöhen:

- für geladene Leptonen: $\frac{\Gamma_e}{\Gamma_Z} = 3,4\%$
- für neutrale Leptonen: $\frac{\Gamma_{\nu_e}}{\Gamma_Z} = 6,8\%$
- für u-ähnliche Quarks: $\frac{\Gamma_u}{\Gamma_Z} = 11,8\%$
- für d-ähnliche Quarks: $\frac{\Gamma_d}{\Gamma_Z} = 15,2\%$

Winkelverteilungen

Die theoretische Winkelverteilung des s- und t-Kanals sieht folgendermassen aus:

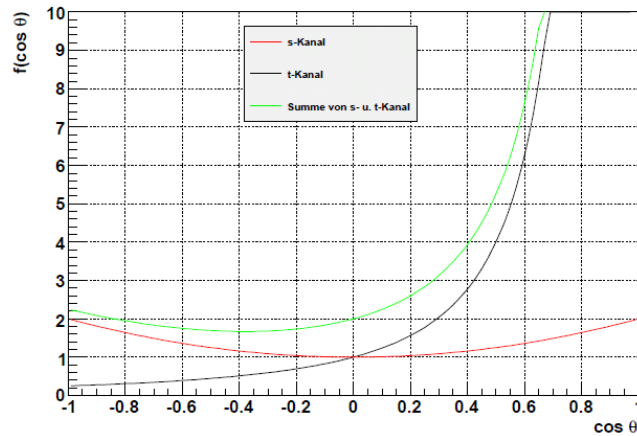


Abbildung 4: Winkelverteilung von s- und t-Kanal und deren Summe

Vorwärts-Rückwärts-Asymmetrie

Die Vorwärts-Rückwärts-Asymmetrie der Myonen kann man mit folgender Formel berechnen. Die sich daraus ergebenden Werte finden sich in Tabelle 4.

$$A_{FB} = -\frac{3}{4} \cdot \frac{2g_A^f g_V^f Q_f \Re(\chi) + 4g_A^e g_V^e g_A^f g_V^f |\chi|^2}{Q_f^2 - 2g_A^f g_V^f Q_f \Re(\chi) + ((g_A^e)^2 + (g_V^e)^2)((g_A^f)^2 + (g_V^f)^2) |\chi|^2}$$

	$\sqrt{s} = 89,225 \text{ GeV}$	$\sqrt{s} = 91,225 \text{ GeV}$	$\sqrt{s} = 93,225 \text{ GeV}$
$\sin\theta_w=0,21$	-0,938	0,0761	0,2317
$\sin\theta_w=0,23$	-0,1640	0,0234	0,1963
$\sin\theta_w=0,25$	-0,1949	0,0041	0,1905

Tabelle 4: Theoretische Werte Vorwärts-Rückwärts-Asymmetrie

3.2 Analyse der Ereignisse mit GROPE

Zuerst liessen wir uns von GROPE verschiedene Zerfallsereignisse aufzeichnen, um Übung in der Zuordnung der verschiedenen Ereignisse und ein Gefühl für die richtige Position der Schnitte zu bekommen.

Nachfolgend findet sich für jede Ereignisart, die jeweils typische grafische Darstellung. GROPE zeigt dabei einen Schnitt durch den Detektor und die Rekonstruktion der Teilchenspuren, ausserdem sind oben die gesammelten Datenwerte angegeben.

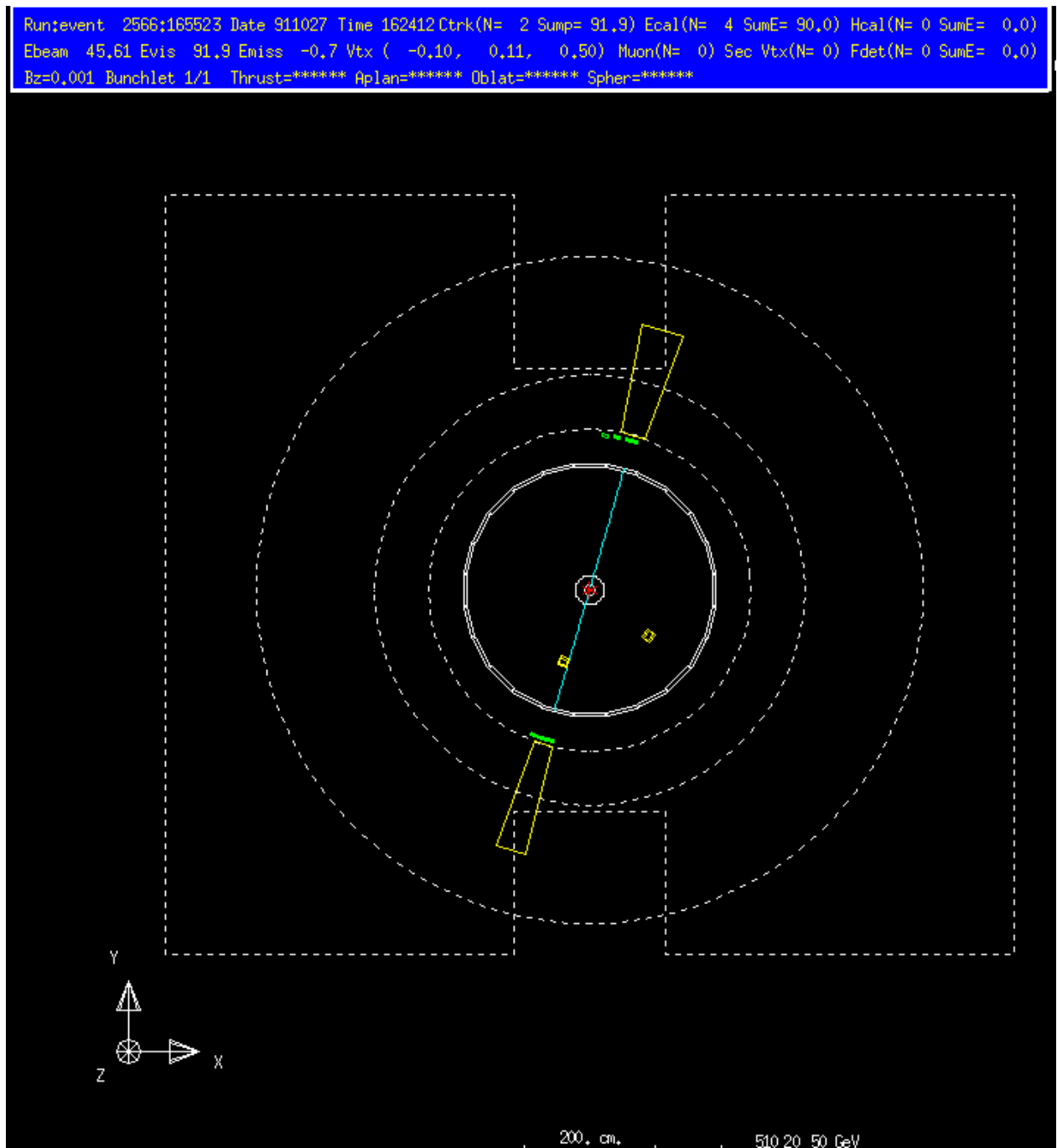
In Tab.5 findet sich eine Zusammenstellung der zur Verfügung stehenden Daten, in Tab.6 sind einige Beispiele der charakteristischen Werte für verschiedene Ereignisse aufgeführt.

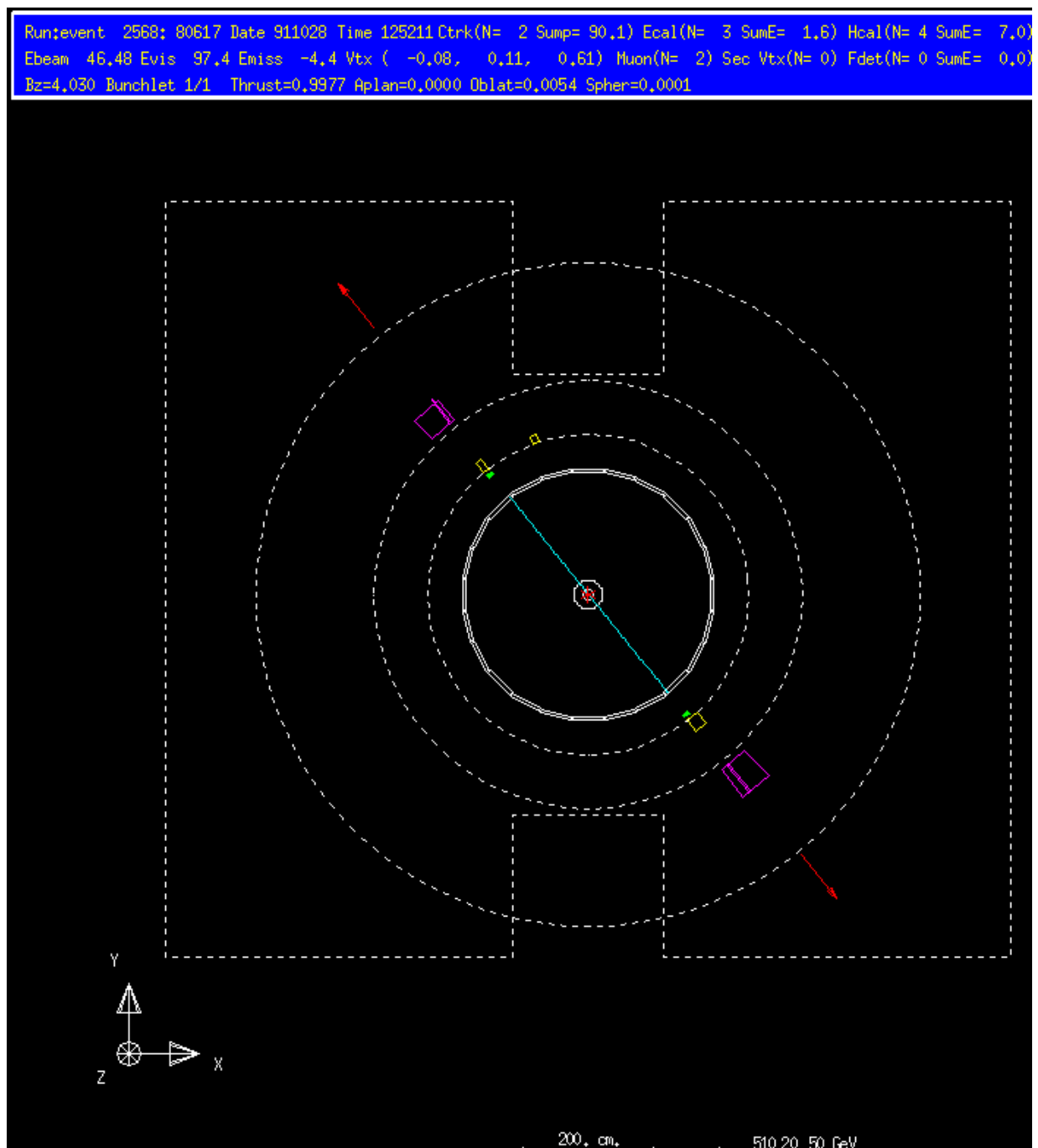
Bedeutung	Name in GROPE	Name in Dateisatz
Anzahl der geladenen Spuren	N	NCharged
Gesamtimpuls	sump	Pcharged
Energie im elektr. Kalorimeter	Ecal (SumE)	E_ecal
Energie im hadronischen Kalorimeter	Ecal (SumE)	E_hcal
Strahlenergie	Ebeam	E_lep
Streuwinkel		cos_thet

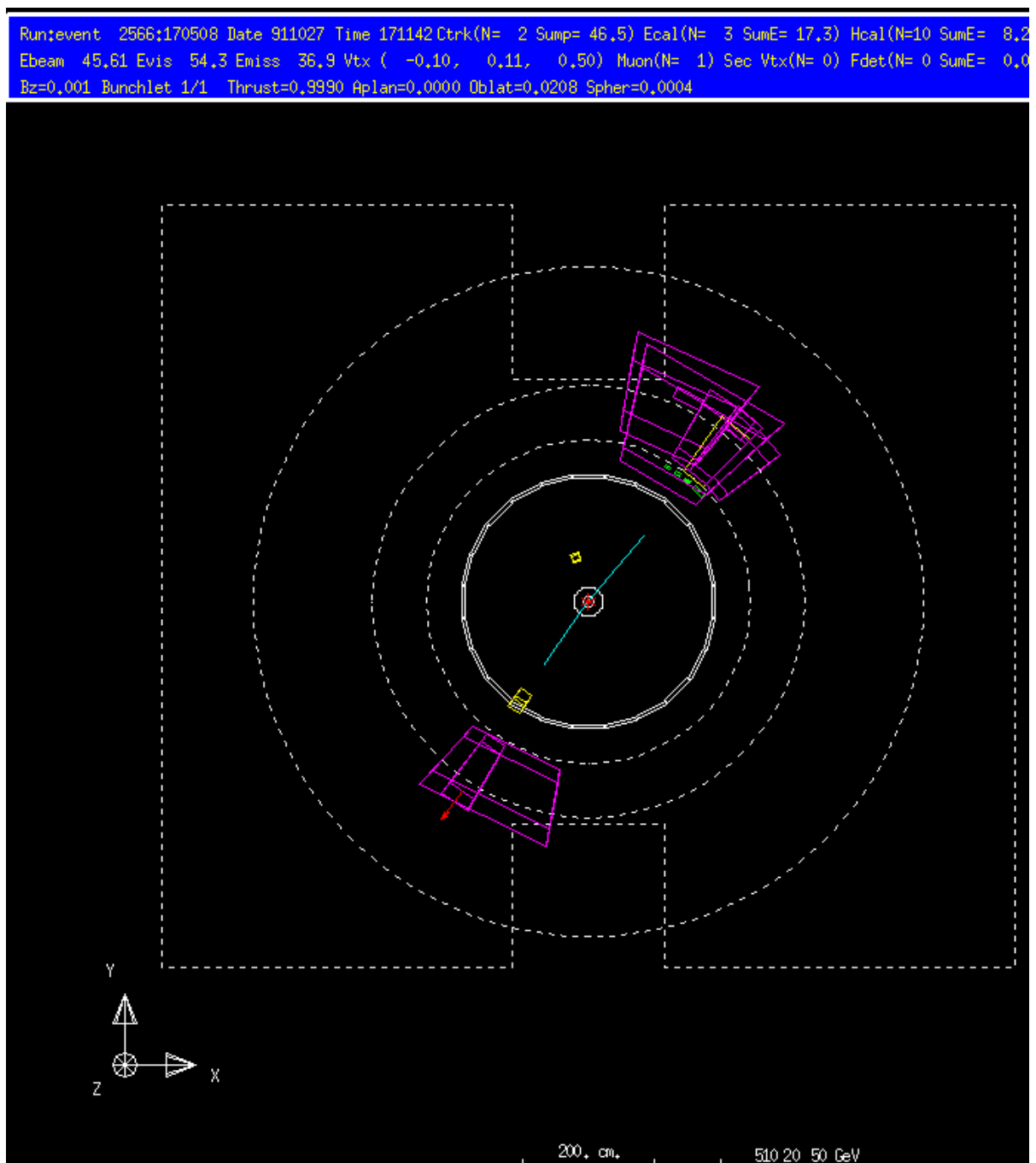
Tabelle 5: Zur Verfügung stehende Messparameter

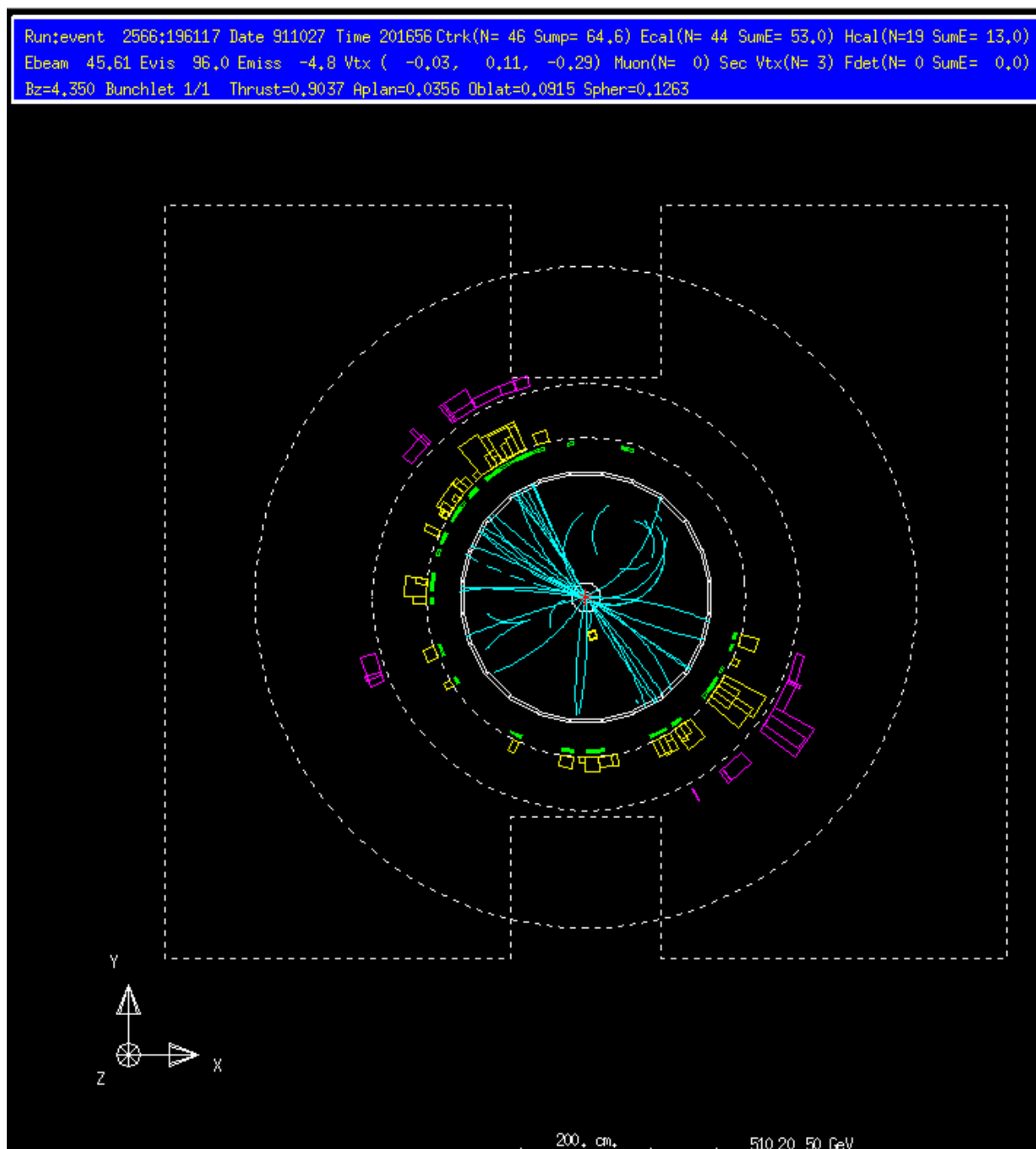
Ereignis	Ncharged	Pcharged	E_ecal	E_hcal
Elektron	2	91,9	90,0	0
	2	80,9	86,8	0
	2	87,4	93,2	0
Myon	2	90,1	1,6	7
	2	93	1,6	8,7
	2	89,1	2,3	8,5
Tau	5	74,0	51,1	10,2
	2	46,5	17,3	8,2
	2	30,8	1,6	6,3
Quarks	15	37,7	37	14,1
	36	45,3	53,2	7,7
	46	64,6	53	13

Tabelle 6: Beispiele für typische Messwerte

Abbildung 5: GROPE-Darstellung eines $e^+ e^-$ -Ereignisses

Abbildung 6: GROPE-Darstellung eines $\mu^+ \mu^-$ -Ereignisses

Abbildung 7: GROPE-Darstellung eines $\tau^+ \tau^-$ -Ereignisses

Abbildung 8: GROPE-Darstellung eines $q\bar{q}$ -Ereignisses

3.3 Analyse der Monte-Carlo-Daten

Um unsere Schnitte zu bestimmen, liessen wir uns mit ROOT für alle 4 Messparameter die Anzahl der verschiedenen Ereignisse als Histogramm ausgeben. Durch die Histogramme (Abb.9,10,11,12) kann man sehr gut erkennen für welche Parameter und Werte besonders saubere Schnitte möglich sind.

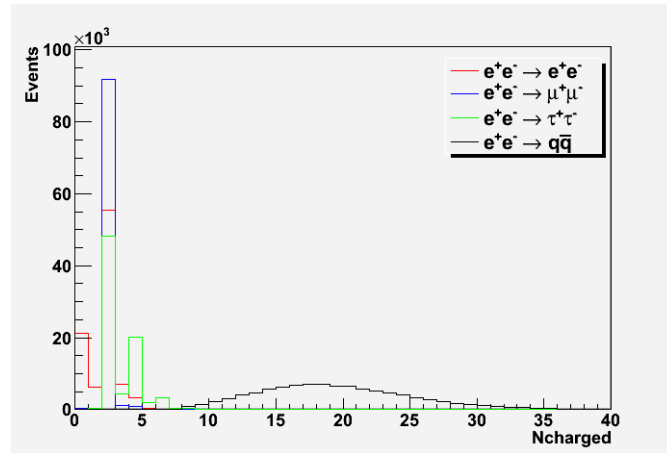


Abbildung 9: Verteilung der Ereignisse für Ncharged

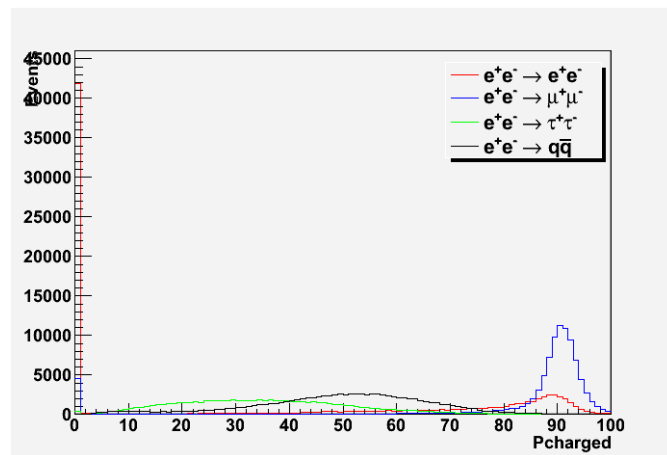
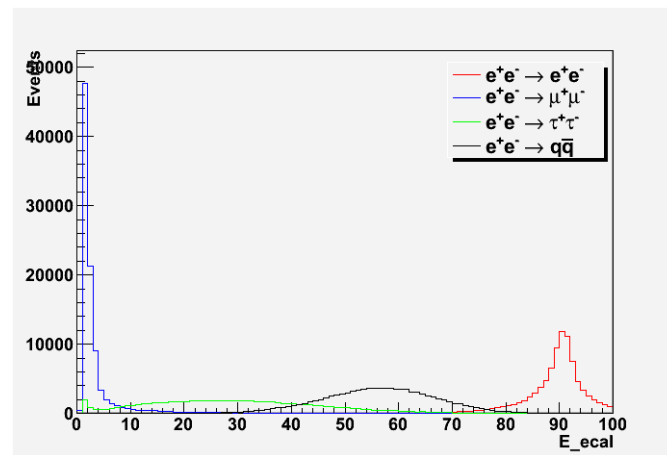
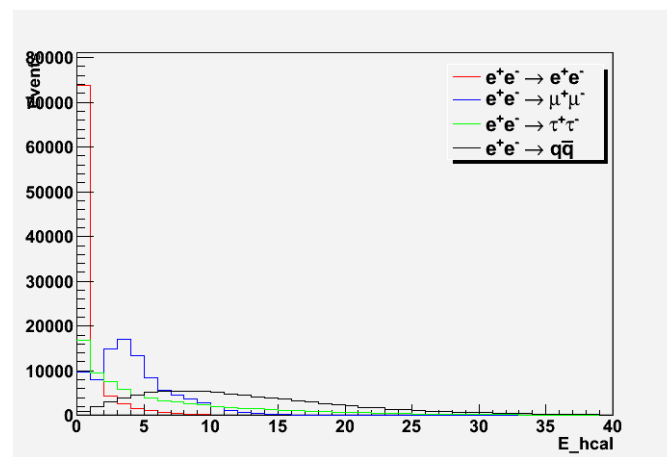


Abbildung 10: Verteilung der Ereignisse für Pcharged

Abbildung 11: Verteilung der Ereignisse für E_{eal}Abbildung 12: Verteilung der Ereignisse für E_{hcal}

Wir entschieden uns für folgende Schnitte (Tab.7):

	Ncharged	Pcharged	E_eal	E_hcal
Elektronen	<5		>80	
Myonen		>80	<10	
Tau	<7	>0 und <70	<70	
Quarks	>7			

Tabelle 7: Schnittparameter

Abb.13,14,15 und 16 zeigen die Schnitte angewandt auf die Monte-Carlo-Dateien.

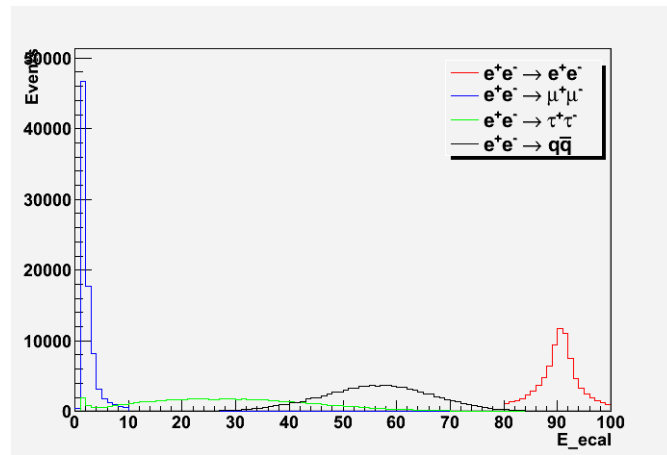


Abbildung 13: Verteilung der Ereignisse für E_cal nach Anwendung der Schnitte

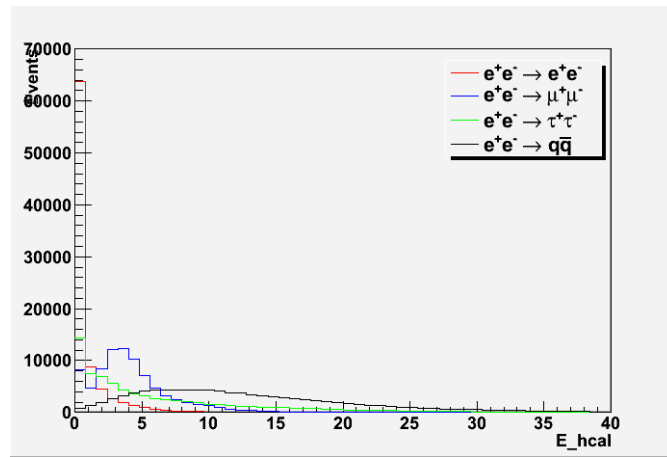


Abbildung 14: Verteilung der Ereignisse für E_hcal nach Anwendung der Schnitte

Die Effizienzmatrix M_{eff} für unsere Schnitte berechnet sich aus der Anzahl der Ereignisse nach dem Schnitt zur Gesamtanzahl Ereignisse von 100000.

$$M_{eff,ij} = \frac{N_{accepted,ij}}{N_{gesamt}}$$

Wobei i den Datensatz bezeichnet und j den Schnitt für das gesuchte Ereignis (1 = Elektron, 2 = Myon, 3 = Tau, 4 = Quark). Das Programm zur Berechnung der Effizienzmatrix befindet sich, wie die anderen Programme zur Auswertung auch, im Anhang (4).

Wir erhielten folgende Effizienzmatrix:

$$M_{eff} = \begin{pmatrix} 0,86 & 0 & 0,003 & 0,00006 \\ 0,00001 & 0,82 & 0,029 & 0 \\ 0,001 & 0,0004 & 0,75 & 0,005 \\ 0,00002 & 0 & 0,005 & 0,98 \end{pmatrix}$$

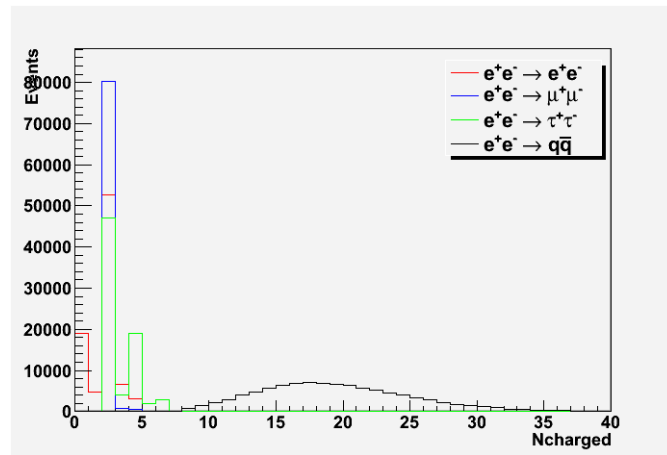


Abbildung 15: Verteilung der Ereignisse für Ncharged nach Anwendung der Schnitte

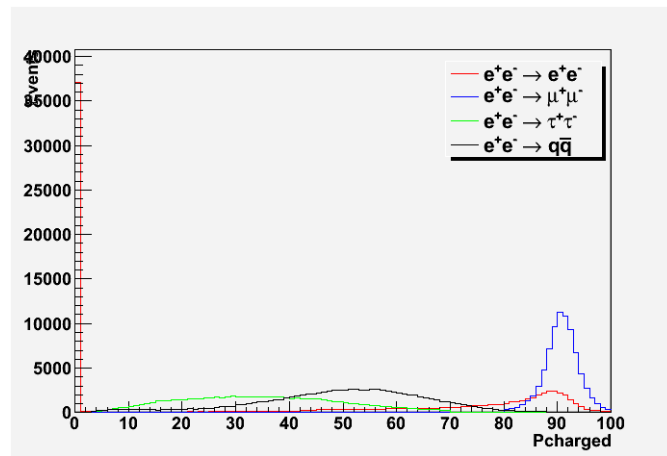


Abbildung 16: Verteilung der Ereignisse für Pcharged nach Anwendung der Schnitte

Die ideale Effizienzmatrix ist die Einheitsmatrix, das würde bedeuten, dass unsere Schnitte nicht nur absolut rein sind, sondern auch hohe Effizienz aufweisen. Im Versuch ist dies jedoch nicht zu realisieren, vorallem da die Tau-Ereignisse sich mit den anderen Ereignissen stark überlappen und es schwer ist, sie klar abzutrennen. Unsere Matrix ist ein guter Mittelwert zwischen Reinheit und Effizienz, ein Grossteil der gesuchten Ereignisart wird durchgelassen, ohne zu viele Ereignisse der anderen Arten dazuzufügen.

Der Fehler der Effizienzmatrix berechnet sich aus:

$$\sigma_{M_{eff,ij}} = \sqrt{\frac{M_{eff,ij}(1 - M_{eff,ij})}{N_{gesamt}}}$$

$$\sigma_{M_{eff}} = \begin{pmatrix} 0,001 & 0 & 0,0002 & 0,00002 \\ 10^{-5} & 0,001 & 0,0005 & 0 \\ 0,0001 & 0,00006 & 0,001 & 0,0002 \\ 10^{-4} & 0 & 0,0002 & 0,0005 \end{pmatrix}$$

Um im folgenden Schritt die wahre Anzahl Ereignisse aus den echten Daten zu erhalten, brauchen wir das Inverse der Effizienzmatrix, da gilt:

$$\begin{aligned} N_{beobachtet} &= N_{wahr} \cdot M_{eff} \\ \Rightarrow N_{wahr} &= N_{beobachtet} \cdot M_{eff}^{-1} \end{aligned}$$

Die Inverse ergibt sich zu

$$M_{eff}^{-1} = \begin{pmatrix} 1,16 & 2,17 \cdot 10^{-6} & -0,0049 & 0,00004 \\ 0,00005 & 1,23 & -0,047 & 0,0003 \\ -0,002 & -0,0006 & 1,334 & -0,007 \\ -0,00001 & 2,9 \cdot 10^{-6} & -0,007 & 1,025 \end{pmatrix}$$

Der Fehler der Inversen betragt

$$\sigma_{M_{eff}^{-1}} = \begin{pmatrix} 0,001 & 2 \cdot 10^{-9} & 0,0001 & 0,00002 \\ 7 \cdot 10^{-6} & 0,001 & 0,0003 & 2 \cdot 10^{-7} \\ 7 \cdot 10^{-5} & 0,00004 & 0,001 & 0,0002 \\ 10^{-5} & 3 \cdot 10^{-9} & 0,0002 & 0,0005 \end{pmatrix}$$

und berechnet sich aus Gausscher Fehlerfortpflanzung.

$$\sigma_{M_{eff}^{-1},ij} = \frac{1}{M_{ii}M_{jj}} \cdot \sqrt{(\sigma_{M_{ij}})^2 + \left(\frac{M_{ij}}{M_{jj}} \cdot \sigma_{M_{jj}}\right)^2 + \left(\frac{M_{ij}}{M_{ii}} \cdot \sigma_{M_{ii}}\right)^2}$$

3.4 Analyse der echten OPAL-Daten

Nach der sorgfaltigen Auswahl der Schnitte anhand der simulierten Monte-Carlo-Daten, kann man jetzt die Schnitte auf die echten Daten anwenden, die aus dem OPAL-Experiment am LEP-Speicherring am CERN stammen.

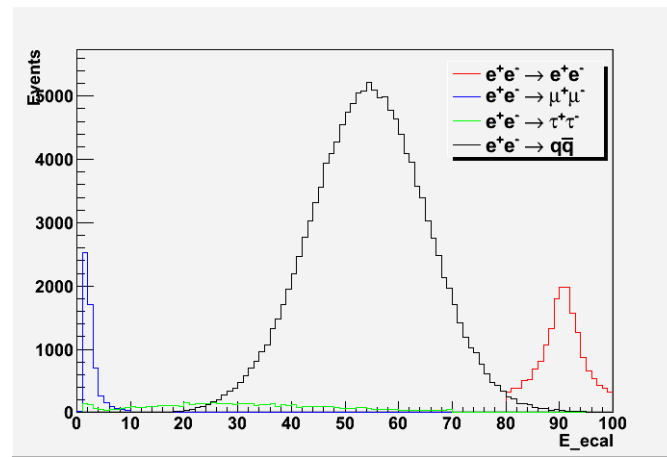
Wir verwendeten fur unsere Auswertung „daten_1.root“.

Tab.8 enthalt die verschiedenen Strahlenergien und Luminositaten fur „daten_1.root“.

Strahlenergie E/GeV _{lep}	Luminosität L	σ_L
88,48	675,859	5,712
89,47	543,627	4,831
90,23	419,776	3,975
91,23	3122,204	22,318
91,97	639,838	5,577
92,97	479,24	4,482
93,718	766,838	6,498

Tabelle 8: Energien und Luminositäten daten_1

Durch Anwenden unserer Schnitte erhielten wir folgende Ereignis-Histogramme (Abb.17,18,19,20):

Abbildung 17: Ereignisse für E_{eal} nach Anwendung der Schnitte auf daten_1

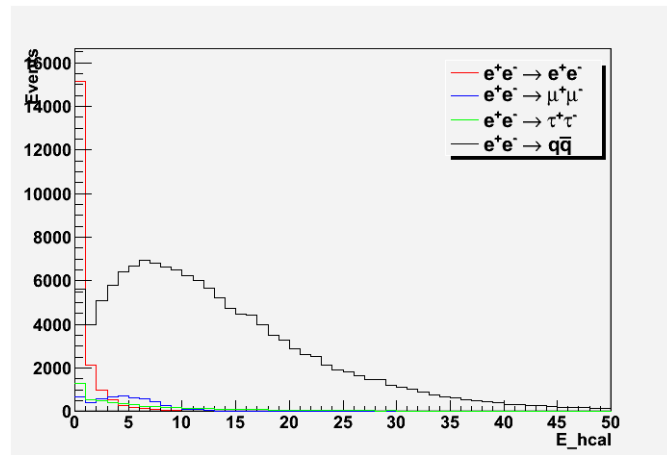


Abbildung 18: Ereignisse für E_hcal nach Anwendung der Schnitte auf daten_1

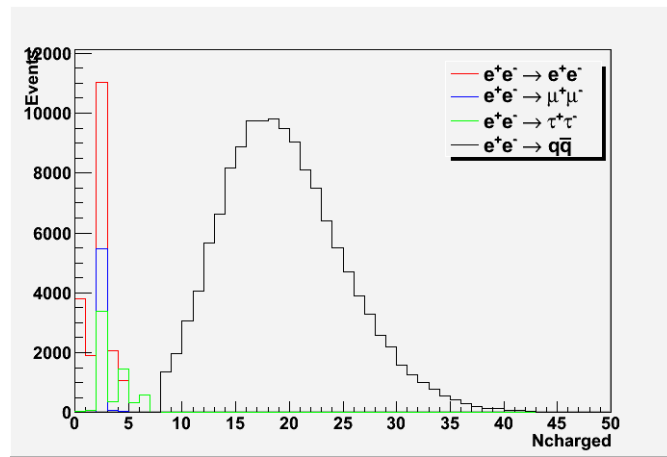


Abbildung 19: Ereignisse für Ncharged nach Anwendung der Schnitte auf daten_1

3.4.1 s-t-Kanal-Trennung

Die so erhaltenen Elektronen-Ereignisse müssen noch weiter verarbeitet werden, da in ihnen nicht nur die gesuchten Ereignisse aus den Elektron-Positron-Kollisionen (**s-Kanal**) enthalten sind, sondern auch reine Streueignisse (**t-Kanal**) (s. Bhabha-Streuung 2.6.2). Der t-Kanal ist proportional zu $\frac{1}{(1-\cos\theta)^2}$, der s-Kanal hingegen zu $(1 + \cos^2\theta)$. Um die Ereignisse zu trennen, trugen wir sie gegen $\cos\theta$ auf und fitteten eine Funktion der Form

$$F = s \cdot (1 + x^2) + t \cdot \frac{1}{(1 - x)^2}$$

an die Verteilung an (Abb.21). Jetzt kann man einen Vorfaktor $\xi(E)$ für die Elektronen-Ereignisse aus den Fit-Parametern berechnen, indem man den Quotienten aus der Anzahl s-Kanal-Ereignisse N_s zur Gesamtanzahl e⁻-Ereignisse N_{ges} nimmt. Da die Verteilung abhängig von der Strahlenergie ist, muss man die Anpassung für alle 7 Energien einzeln

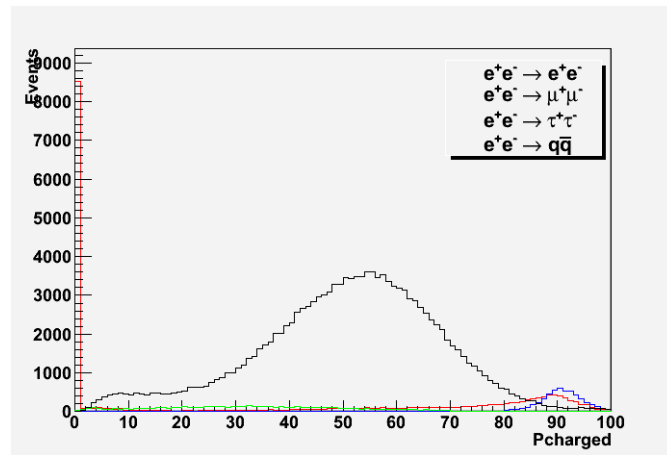
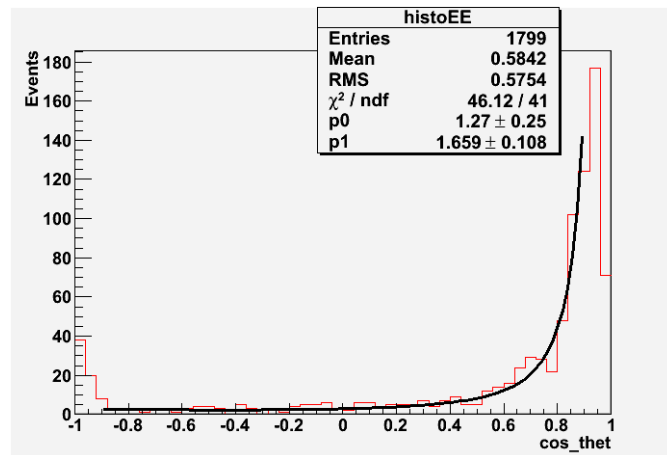


Abbildung 20: Ereignisse für Pcharged nach Anwendung der Schnitte auf daten_1

durchführen.

$$\xi(E) = \frac{\int_b^a s \cdot (1 + \cos^2\theta) d\cos\theta}{\int_b^a t \cdot \frac{1}{(1 - \cos\theta)^2} d\cos\theta} \quad a = 0,98 \text{ und } b = -0,98$$

Den Fehler σ_{st} für $\xi(E)$ schätzten wir ab, indem wir die jeweiligen Integralgrenzen leicht änderten. Wir haben nun für jede Energie einen 4er-Vektor mit den wahren Ereigniszahlen

Abbildung 21: Ereignisverteilung für $\cos\theta$ mit s+t-Fit

$N_{wahr,i}$, wobei i erneut die jeweilige Ereignisart bezeichnet.

Der Fehler für die wahren Ereigniszahlen $N_{wahr,i}$ berechnet sich jetzt aus:

$$\sigma_{wahr,i} = \sqrt{\sum_{j=1}^4 (N_{wahr,i} \cdot \sigma_{M_{eff,ji}^{-1}})^2 + (M_{eff,ji}^{-1} \cdot \sqrt{N_{wahr,i}})^2}$$

Für die Anzahl Elektronen geht noch der Fehler aus der s-t-Kanal-Trennung ein:

$$\sigma_{wahr,e^-} = \sqrt{(\xi(E) \cdot \sigma_{wahr,1})^2 + (\sigma_{st} \cdot N_{wahr,1})^2}$$

Die jeweiligen Wirkungsquerschnitte ergeben sich zu:

$$\sigma(E)_i = \frac{N_{wahr,i}}{L(E)} + \kappa(E)$$

mit $L(E)$ = Luminosität, $\kappa(E)$ = Strahlungskorrektur

$L(E)$ und $\kappa(E)$ stammen aus Tab.8 und waren in der Datei „daten_1.lumi“ angegeben.

3.4.2 Berechnung der Z⁰-Parameter

Die berechneten Wirkungsquerschnitte trägt man jetzt für jede Ereignisart gegen die Strahlenergie auf und fittet ein Breit-Wigner-Kurve an:

$$f_{B-W}(s) = \frac{12\pi}{M_Z^2} \frac{s \cdot \Gamma_e \cdot \Gamma_f}{(s - M_Z^2) + \left(\frac{s^2 \Gamma_Z^2}{M_Z^2}\right)} \cdot 0,3894 \cdot 10^{-6}$$

Aus den Fit-Parametern erhält man die gesuchte Masse M_Z und die Zerfallsbreite Γ_Z von Z⁰. Γ_e und Γ_f sind die Zerfallsbreiten des Elektrons und des jeweiligen Fermions. s ist das Quadrat der Strahlenergie und $0,3894 \cdot 10^6$ nb der Umrechnungsfaktor von GeV².

Wir fitteten die Elektronen-Ereignisse als Erstes, da die Zerfallsbreite des Elektrons in die anderen 3 Fits einfließt.

Abb.22,23,24,25 zeigen unsere Breit-Wigner-Fits für Elektronen-, Myonen-, Tau-Leptonen- und Quarkereignisse. Wobei $p0=\Gamma_e$, $p1=M_Z$ und $p2=\Gamma_Z$ bezeichnet.

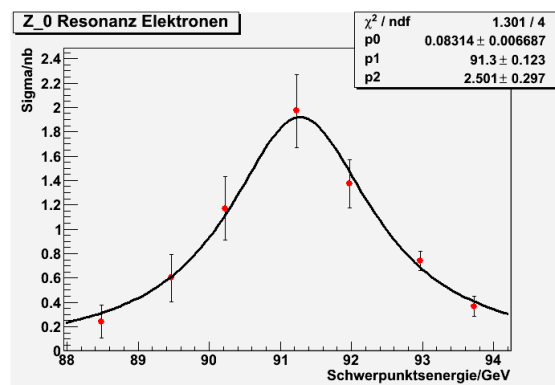


Abbildung 22: Breit-Wigner für Elektronen-Ereignisse

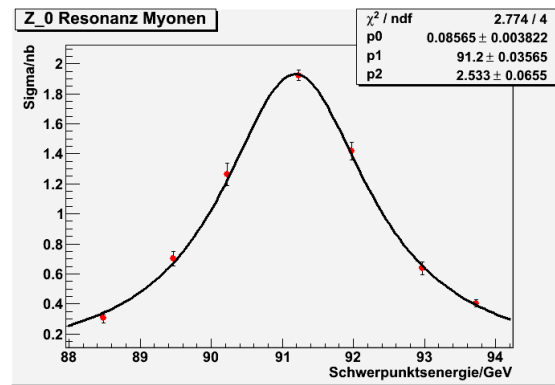


Abbildung 23: Breit-Wigner für Myonen-Ereignisse

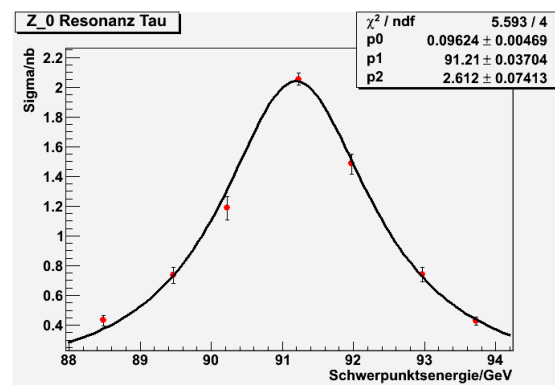


Abbildung 24: Breit-Wigner für Tau-Ereignisse

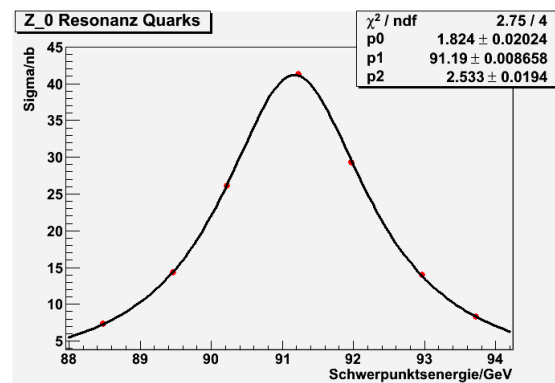


Abbildung 25: Breit-Wigner für Quark-Ereignisse

Ereignis	Γ_Z/GeV	$\sigma_{\Gamma_Z}/\text{GeV}$	M_Z/GeV	σ_{M_Z}/GeV	$\text{red.}\chi^2$
Elektronen	2,501	$\pm 0,3$	91,3	$\pm 0,12$	0,325
Myonen	2,533	$\pm 0,07$	91,2	$\pm 0,04$	0,694
Tau	2,612	$\pm 0,07$	91,21	$\pm 0,04$	1,398
Quarks	2,533	$\pm 0,02$	91,196	$\pm 0,01$	0,688

Tabelle 9: Messparameter für Z_0 aus den Breit-Wigner-Fits

Nach Bildung des gewichteten Mittels für die Z⁰-Parameter erhalten wir somit folgende Messergebnisse:

	Messergebnis	$\sigma_{\text{Messergebnis}}$	Literaturwert
M _Z	91,23	$\pm 0,033 \text{ GeV}/c^2$	$91,187 \pm 0,0021 \text{ GeV}/c^2$
Γ_Z	2,545	$\pm 0,079 \text{ GeV}$	$2,4952 \pm 0,0023 \text{ GeV}$
Γ_e	0,08314	$\pm 0,0067 \text{ GeV}$	$0,08391 \pm 0,00012 \text{ GeV}$
Γ_μ	0,08565	$\pm 0,0038 \text{ GeV}$	$0,08399 \pm 0,00018 \text{ GeV}$
Γ_τ	0,09624	$\pm 0,0047 \text{ GeV}$	$0,08408 \pm 0,00022 \text{ GeV}$
Γ_q	1,824	$\pm 0,02 \text{ GeV}$	$1,7444 \pm 0,002 \text{ GeV}$

Tabelle 10: Endergebnisse

Unsere Werte für Γ_Z , Γ_e und Γ_μ liegen innerhalb der Standardabweichung an den Literaturwerten, auch der Wert für M_Z liegt noch in einer 2- σ -Umgebung um den erwarteten Wert. Allerdings weichen Γ_τ und Γ_q stark um 3 σ vom Literaturwert ab. Der schlechte Wert für Γ_τ ist keine Überraschung, da es schwer war für τ geeignete Schnitte zu finden. Der hohe Wert für Γ_q liegt womöglich an unserem zu niedrigem Wert für Γ_e , der sich auf Grund des höheren Wirkungsquerschnitts stärker bei den Quark-Ereignissen auswirkt.

3.4.3 Leptonenuniversalität

Laut Theorie sollen alle Leptonen die gleiche Zerfallsbreite haben. Zur Überprüfung bildet man den Quotient von dem jeweiligen Γ_l mit Γ_e :

$$V_\mu = \frac{\Gamma_\mu}{\Gamma_e} = (1,03 \pm 0,095) \quad V_\tau = \frac{\Gamma_\tau}{\Gamma_e} = (1,158 \pm 0,11)$$

V_μ bestätigt die Theorie gut, auch V_τ ist innerhalb der Standardabweichung gerade noch mit 1 verträglich.

3.4.4 Anzahl Neutrino-Familien

Aus unseren berechneten Ergebnissen für die verschiedenen Zerfallsbreiten erhalten wir einen Wert für die Gesamtzerfallsbreite der Neutrinos Γ_ν . Daraus ergibt sich dann die Anzahl der Neutrinofamilien:

$$\Gamma_\nu = \Gamma_Z - \Gamma_e - \Gamma_\mu - \Gamma_\tau - \Gamma_q = 0,456$$

$$\Rightarrow N_{\text{Familien}} = \frac{\Gamma_\nu}{\Gamma_{\nu, \text{theo.}}} = 2,8$$

mit $\Gamma_{\nu,theo.} = 0,165$ GeV, der vorausgesagten Neutrinozerfallsbreite.
Der Fehler berechnet sich aus

$$\sigma_{N_{Fam.}} = \frac{1}{\Gamma_{\nu,theo.}} \sqrt{\sigma_{\Gamma_Z}^2 + \sigma_{\Gamma_e}^2 + \sigma_{\Gamma_\mu}^2 + \sigma_{\Gamma_{\tau\mu}}^2 + \sigma_{\Gamma_q}^2} = 0,5$$

Mit der Standardabweichung bestätigt unser Wert die theoretisch vorhergesagten 3 Neutrinofamilien.

3.4.5 Vorwärts-Rückwärts-Asymmetrie

Um die Vorwärts-Rückwärts-Asymmetrie der Streuung der Myonen zu bestimmen und damit den Weinbergwinkel, trugen wir die Myonereignisse auf den Winkel auf und fitteten folgende Funktion an:

$$y = C \cdot [F_1(1 + x^2) + 2F_2 \cdot x]$$

mit $C = \frac{\alpha^2 N_c^f}{4s}$ und $x = \cos\theta$.

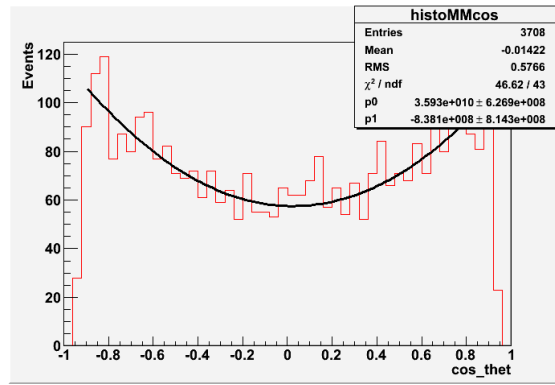


Abbildung 26: Fit der Myon-Ereignisse für die Vorwärts-Rückwärts-Asymmetrie

Die Vorwärts-Rückwärts-Asymmetrie A_{FB} berechnet sich dann aus:

$$A_{FB} = \frac{3}{4} \cdot \frac{F_2}{F_1}$$

und $\sigma_{A_{FB}} = \frac{3}{4} \cdot \frac{F_2}{F_1} \sqrt{\left(\frac{\sigma_{F_2}}{F_2}\right)^2 + \left(\frac{\sigma_{F_1}}{F_1}\right)^2}$

Bei der Strahlenergie der Z⁰-Resonanz gilt für den Weinbergwinkel $\sin^2\theta_W$:

$$\sin^2\theta_W = \frac{1}{4} \cdot \left(1 - \sqrt{\frac{A_{FB}}{3}}\right)$$

$$\sigma_{\sin^2\theta_W} = \frac{\sigma_{A_{FB}}}{8\sqrt{3} \cdot A_{FB}}$$

Wir erhalten für 91,23 GeV Strahlenergie folgende Werte

$$A_{FB} = -0,0175 \pm 0,013 \quad \text{und} \quad \sin^2\theta_W = 0,231 \pm 0,007$$

Unser Wert liegt also sehr gut innerhalb von 1- σ am Literaturwert von $\sin^2\theta_W = 0,23120$. Für die anderen beiden Strahlenergien nahe des Resonanzmaximums erhielten wir folgende Werte:

$$A_{FB, 91,97 \text{ GeV}} = 0,0031 \pm 0,0138 \quad \text{und} \quad \sin^2\theta_W = 0,242 \pm 0,018$$

$$A_{FB, 90,23 \text{ GeV}} = -0,1192 \pm 0,0511 \quad \text{und} \quad \sin^2\theta_W = 0,2 \pm 0,011$$

Man sieht, dass 90,23 GeV schon zu weit vom Resonanzmaximum entfernt ist, so dass die Näherung unserer Formle keine guten Ergebnisse mehr liefert.

4 Zusammenfassung

Aus unserer Analyse der LEP-Daten erhielten wir für die Masse M_Z und die Zerfallsbreite Γ_Z des Z⁰ folgende Werte:

$$M_Z = 91,23 \pm 0,033 \text{ GeV}/c^2$$

$$\Gamma_Z = 2,545 \pm 0,079 \text{ GeV}$$

Unsere Ergebnisse stimmen innerhalb der Fehler recht gut mit den Literaturwerten überein.

Für die verschiedenen fermionischen Zerfallsbreiten erhielten wir:

$$\Gamma_e = 0,08314 \pm 0,0067 \text{ GeV}$$

$$\Gamma_\mu = 0,08565 \pm 0,0038 \text{ GeV}$$

$$\Gamma_\tau = 0,09624 \pm 0,0047 \text{ GeV}$$

$$\Gamma_q = 1,824 \pm 0,02 \text{ GeV}$$

Γ_τ und Γ_q weichen etwas stärker von den Literaturwerten ab. Der Grund ist, dass es schwer ist für τ -Ereignisse gute Schnitte zu finden. Auch die s-t-Trennung gelingt mit unseren Mitteln nur grob, dadurch wirken sich Abweichungen vom Literaturwert für Γ_e auch auf die anderen Werte aus.

Wir können innerhalb der Fehler die Leptonuniversalität und die vorhergesagte Anzahl Neutrinofamilien bestätigen. Die Abweichungen der Zerfallsbreiten wirken sich natürlich auch hier negativ auf die Genauigkeit unseres Ergebnisses aus.

Aus der Vorwärts-Rückwärts-Asymmetrie am Resonanz-Maximum erhielten wir für den Weinbergwinkel θ_W :

$$\sin^2\theta_W = 0,231 \pm 0,007$$

Auch dieser Wert bestätigt innerhalb der Fehler den Literaturwert.

Literatur

- [1] Versuchsskript, Analyse von Z⁰-Zerfällen.
- [2] Versuchsskript, Fortgeschrittenen Praktikum - Z⁰.
- [3] Christoph Berger, „Elementarteilchenphysik“, 2.Auflage, Springer 2006.

A C++ Quellcode für ROOT

```
void effiautoalle() //Berechnung der Effizienzmatrix
```

```
    TFile* files[4];
    files[0] = new TFile("~/ee.root");
    files[1] = new TFile("~/mm.root");
    files[2] = new TFile("~/tt.root");
    files[3] = new TFile("~/qq.root");

    Int_t nacceptedEE=0;
    Int_t nacceptedEM=0;
    Int_t nacceptedET=0;
    Int_t nacceptedEQ=0;
    Int_t nacceptedME=0;
    Int_t nacceptedMM=0;
    Int_t nacceptedMT=0;
    Int_t nacceptedMQ=0;
    Int_t nacceptedTE=0;
    Int_t nacceptedTM=0;
    Int_t nacceptedTT=0;
    Int_t nacceptedTQ=0;
    Int_t nacceptedQE=0;
    Int_t nacceptedQM=0;
    Int_t nacceptedQT=0;
    Int_t nacceptedQQ=0;

    for (Int_t i=0; i<4; i++)
        gROOT->Reset();
        files[i]->cd();
        TTree *tree = (TTree*)files[i]->Get("h3");

    Float_t run;
    Float_t event;
```

```

Float_t Ncharged;
Float_t Pcharged;
Float_t E_ecal;
Float_t E_hcal;
Float_t E_lep;
Float_t cos_thru;
Float_t cos_thet;

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

Int_t nentries = tree->GetEntries();

for (Int_t k=0; k<nentries;k++)
    tree->GetEntry(k);
    if ( Ncharged < 5&& E_ecal > 80 )

        if (i==0)nacceptedEE++;
        if (i==1)nacceptedME++;
        if (i==2)nacceptedTE++;
        if (i==3)nacceptedQE++;


for (Int_t j=0; j<nentries;j++)
    tree->GetEntry(j);
    if ( Pcharged > 80 && E_ecal < 10 )
        if (i==0)nacceptedEM++;
        if (i==1)nacceptedMM++;
        if (i==2)nacceptedTM++;
        if (i==3)nacceptedQM++;

```

```

for (Int_t j=0; j<nentries;j++)
  tree->GetEntry(j);
  if ( Pcharged < 70 && Pcharged > 0 && Ncharged<7 && E_ecal < 70 )
    if (i==0)nacceptedET++;
  if (i==1)nacceptedMT++;
  if (i==2)nacceptedTT++;
  if (i==3)nacceptedQT++;

```

```

for (Int_t j=0; j<nentries;j++)
  tree->GetEntry(j);
  if ( Ncharged > 7 )
    if (i==0)nacceptedEQ++;
  if (i==1)nacceptedMQ++;
  if (i==2)nacceptedTQ++;
  if (i==3)nacceptedQQ++;

```

```

TMatrixD effmatrix(4,4);
effmatrix(0,0)= nacceptedEE/100000.;
effmatrix(1,0)= nacceptedME/100000.;
effmatrix(2,0)= nacceptedTE/100000.;
effmatrix(3,0)= nacceptedQE/100000.;
  effmatrix(0,1)= nacceptedEM/100000.;
effmatrix(1,1)= nacceptedMM/100000.;
effmatrix(2,1)= nacceptedTM/100000.;
effmatrix(3,1)= nacceptedQM/100000.;
  effmatrix(0,2)= nacceptedET/100000.;
effmatrix(1,2)= nacceptedMT/100000.;
effmatrix(2,2)= nacceptedTT/100000.;
effmatrix(3,2)= nacceptedQT/100000.;
  effmatrix(0,3)= nacceptedEQ/100000.;
effmatrix(1,3)= nacceptedMQ/100000.;

```

```
    effmatrix(2,3)= nacceptedTQ/100000.;
    effmatrix(3,3)= nacceptedQQ/100000.;

    cout<<"Effizienzmatrix:"<<endl;
    effmatrix.Print();

void Energie1() //Anwenden der Cuts
gROOT->Reset();
TFile* daten[6];
daten[0]= new TFile("daten_1.root");

Float_t      run;
Float_t      event;
Float_t      Ncharged;
Float_t      Pcharged;
Float_t      E_ecal;
Float_t      E_hcal;
Float_t      E_lep;
Float_t      cos_thru;
Float_t      cos_thet;

TCanvas *c1 = new TCanvas("c1","drawall",200,10,700,500);

//Anwenden der Cuts und Zählen der beobachteten Ereignisse

TMatrixD supermatrix(4,0);

for (Int_t q=0; q <1; q++)

gROOT->Reset();
TH1F* historealEE = new TH1F("historealEE","",100,0,100);
TH1F* historealMM = new TH1F("historealMM","",100,0,100);
TH1F* historealTT = new TH1F("historealTT","",100,0,100);
TH1F* historealQQ = new TH1F("historealQQ","",100,0,100);

Int_t neventsE = 0;
Int_t neventsM = 0;
Int_t neventsT = 0;
```

```

Int_t neventsQ = 0;
daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

Int_t nentries = tree->GetEntries();
for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Ncharged<5 && E_ecal>80 && E_lep>44 && E_lep<44.5)
    historealEE->Fill(E_ecal);
    neventsE++;

historealEE->GetXaxis()->SetTitle("E_ecal");
historealEE->GetYaxis()->SetTitle("Events");

cout<<"Anzahl Elektronen: "<<neventsE<<endl;

gROOT->Reset();
c1->cd();
daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

```

```
Int_t nentries = tree->GetEntries();

for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Pcharged>80 && E_ecal<10 && E_lep>44 && E_lep<44.5)
    historealMM->Fill(E_ecal);
    neventsM++;

    cout<<"Anzahl Myonen: "<<neventsM<<endl;

gROOT->Reset();
daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

Int_t nentries = tree->GetEntries();

for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Ncharged<7 && Pcharged>0 && Pcharged<70 && E_ecal<70 && E_lep>44 && E_lep<44.5)
    historealTT->Fill(E_ecal);
    neventsT++;

    cout<<"Anzahl Taus: "<<neventsT<<endl;

gROOT->Reset();
daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
```

```

tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

Int_t nentries = tree->GetEntries();

for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Ncharged>7 && E_lep>44 && E_lep<44.5)
    historealQQ->Fill(E_ecal);
    neventsQ++;

    cout<<"Anzahl Quarks: "<<neventsQ<<endl;
Int_t max = 0;
if (historealEE->GetMaximum() > max) max = historealEE->GetMaximum();
if (historealMM->GetMaximum() > max) max = historealMM->GetMaximum();
if (historealTT->GetMaximum() > max) max = historealTT->GetMaximum();
if (historealQQ->GetMaximum() > max) max = historealQQ->GetMaximum();
historealEE->SetMaximum(1.1*max);
historealEE->SetLineColor(kRed);
historealEE->Draw();
historealMM->SetLineColor(kBlue);
historealMM->Draw("same");
historealTT->SetLineColor(kGreen);
historealTT->Draw("same");
historealQQ->SetLineColor(kBlack);
historealQQ->Draw("same");

TLegend *leg = new TLegend(0.652406,0.664021,0.885918,0.873016,"","brNDC");
leg->SetTextSize(0.0478723);
leg->SetLineColor(0);
leg->AddEntry("historealEE","e++e- → e++e-","L");
leg->AddEntry("historealMM","e++e- → μ++μ-","L");
leg->AddEntry("historealTT","e++e- → τ++τ-","L");
leg->AddEntry("historealQQ","e++e- → q#barq","L");
leg->Draw();

c1->Modified();

```



```

c1->cd();
c1->SetSelected(c1);

TMatrixD vector(4,1);
vector(0,0)= neventsE;
vector(1,0)= neventsM;
vector(2,0)= neventsT;
vector(3,0)= neventsQ;
cout<<"Vector"<<endl;
vector.Print();

// include void effiautoalle

//Fehlermatrix

TMatrixD fehlereff(4,4);
for(int w =0; w<4; w++)
    fehlereff(w,0)= sqrt(effmatrix(w,0)*(1- effmatrix(w,0))/100000.);
    fehlereff(w,1)= sqrt(effmatrix(w,1)*(1- effmatrix(w,1))/100000.);
    fehlereff(w,2)= sqrt(effmatrix(w,2)*(1- effmatrix(w,2))/100000.);
    fehlereff(w,3)= sqrt(effmatrix(w,3)*(1- effmatrix(w,3))/100000.);

cout<<"Fehler Eff.Matrix: "<<endl;
fehlereff.Print();

//invertierte Eff.Matrix
TMatrixD Inverse(4,4);
Inverse = effmatrix.Invert();
cout<<"Inverse Effizienzmatrix"<< endl;
Inverse.Print();

//fehler invertiert

TMatrixD fehlerinv(4,4);
for(int v=0; v<4; v++)
    for(int u=0; u<4; u++)

        if(u==v)        fehlerinv(v,u)=fehlereff(v,u)/effmatrix(v,u);
        else             fehlerinv(v,u)=1/(effmatrix(v,v)*effmatrix(u,u))* sqrt((feh

cout<<"Fehler inv.Matrix: "<<endl;
fehlerinv.Print();

```

```
//Berechnung wahre Werte
TMatrixD wahreAnzahl(4,1);
wahreAnzahl=Inverse * vector;
cout<<"Wahre Werte Anzahl Ereignisse:"<< endl;
wahreAnzahl.Print();

//fit st

Float_t      run;
Float_t      event;
Float_t      Ncharged;
Float_t      Pcharged;
Float_t      E_ecal;
Float_t      E_hcal;
Float_t      E_lep;
Float_t      cos_thru;
Float_t      cos_thet;

TCanvas *c2 = new TCanvas("c2","drawall",200,10,700,500);
gStyle->SetOptFit(111);

Int_t neventsEE = 0;

TH1F* histoEE = new TH1F("histoEE","",50,-1,1);

daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);
```

```

Int_t nentries = tree->GetEntries();

for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Ncharged<5 && E_ecal>80 && E_lep>44 && E_lep<44.5)
    histoEE->Fill(cos_thet);
neventsEE++;

    cout<<"Anzahl Elektronen: "<<neventsEE<<endl;

histoEE->GetXaxis()->SetTitle("cos_thet");
histoEE->GetYaxis()->SetTitle("Events");

histoEE->SetLineColor(kRed);
histoEE->Draw();

TLegend *leg = new TLegend(0.652406,0.664021,0.885918,0.873016,"","brNDC");
leg->SetTextSize(0.0478723);
leg->SetLineColor(0);
leg->AddEntry("histoEE","e++e- → e++e-","1");
leg->Draw();

c2->Modified();
c2->cd();
c2->SetSelected(c1);

//TFile* histos = new TFile("Pro.root");
//TGraphErrors *gr = (TGraphErrors*)histos->Get("histoEE");

//gr->SetLineWidth(2);
//gr->SetLineColor(1);

TF1* grfit = new TF1("grfit","[0]*(1+x^2)+[1]*(1/(1-x)^2)",-0.9,0.9);
grfit->SetParameters(1.8,0.4);
//gr->Draw("A*");
histoEE->SetMarkerStyle(20);
histoEE->Fit("grfit","","",-0.9,0.9);
cout << "Params=" << grfit->GetParameter(0)<<endl;
cout << "Paramt=" << grfit->GetParameter(1)<<endl;

```

```

Double_t p0 = grfit->GetParameter(0);
Double_t p1 = grfit->GetParameter(1);
TF1* sfunction = new TF1("sfunction",Form("%f*(1+x*x)",p0),-0.98,0.98);
TF1* tfunction = new TF1("tfunction",Form("%f/((1+x)*(1+x))",p1),-0.98,0.98);

Double_t integral_sfunction, integral_tfunction, integral_komplett;
integral_sfunction = sfunction->Integral(-0.98,0.98);
integral_tfunction = tfunction->Integral(-0.98,0.98);
integral_komplett = grfit->Integral(-0.98,0.98);
cout<<"integral_sfunction = "<<integral_sfunction<<endl;
cout<<"integral_tfunction = "<<integral_tfunction<<endl;
cout<<"integral_komplett = "<<integral_komplett<<endl;
cout<<"N_s/N_s+t = "<<integral_sfunction / integral_komplett<<endl;
cout<<"N_t/N_s+t = "<<integral_tfunction / integral_komplett<<endl;

TMatrixD wahreAnzahlST(4,1);
    wahreAnzahlST(0,0)= (integral_sfunction / integral_komplett)*wahreAnzahl(0,0);
    wahreAnzahlST(1,0)= wahreAnzahl(1,0);
    wahreAnzahlST(2,0)= wahreAnzahl(2,0);
    wahreAnzahlST(3,0)= wahreAnzahl(3,0);
    cout<<"Wahre Werte Anzahl Ereignisse mit st-Korrektur:"<<q+1<< endl;
    wahreAnzahlST.Print();

//Fehler wahreAnzahlST

TMatrixD fehlerwahr(4,1);
for(int k=0; k<4; k++)

    fehlerwahr(k,0)= sqrt((fehlerinv(k,0)*vector(0,0))*(fehlerinv(k,0)*vector(0,0)));

TMatrixD fehlerwahrst(4,1);
for(int k=0; k<4; k++)
    if(k==0)fehlerwahrst(k,0)=sqrt((integral_sfunction / integral_komplett*fehlerwahr(k,0)*fehlerwahr(k,0)));
    elsefehlerwahrst(k,0)=fehlerwahr(k,0);

cout<<"Fehler wahre Werte: "<<endl;

```

```
fehlerwahrst.Print();

TMatrixD korr(4,1);
korr(0,0)=0.09;
korr(1,0)=0.09;
korr(2,0)=0.09;
korr(3,0)=2.0;

TMatrixD energie1(4,1);
energie1=wahreAnzahlST*(1./675.8590) + korr;

cout<<"Ereignisse Energie 88,47: "<<endl;
energie1.Print();

TMatrixD fehlersigma(4,1);
for(int h=0; h<4; h++)
    fehlersigma(h,0)= energie1(h,0)* sqrt((fehlerwahrst(h,0)/wahreAnzahlST(h,0))

cout<<"Fehler Wirkungsquerschnitt: "<<endl;
fehlersigma.Print();


void FIT()          //Breit-Wigner-Fits

TMatrixD ende(4,7);
ende(0,0)=0.2406;
ende(1,0)=0.3029;
ende(2,0)=0.4309;
ende(3,0)=7.35;
ende(0,1)=0.6002;
ende(1,1)=0.7019;
ende(2,1)=0.7373;
ende(3,1)=14.33;
ende(0,2)=1.171;
ende(1,2)=1.262;
ende(2,2)=1.186;
ende(3,2)=26.11;
ende(0,3)=1.972;
```

```

ende(1,3)=1.922;
ende(2,3)=2.054;
ende(3,3)=41.25;
ende(0,4)=1.371;
ende(1,4)=1.419;
ende(2,4)=1.483;
ende(3,4)=29.2;
ende(0,5)=0.7376;
ende(1,5)=0.6362;
ende(2,5)=0.7426;
ende(3,5)=14.01;
ende(0,6)=0.3666;
ende(1,6)=0.4037;
ende(2,6)=0.4248;
ende(3,6)=8.318;

cout<<"ende"<<endl;
ende.Print();

TCanvas *c1 = new TCanvas("c1","Elektronen",200,10,700,500);

TGraphErrors* gr1= new TGraphErrors(7);
float x[7]=88.48,89.47,90.22,91.23,91.97,92.97,93.72;
float y[7]=ende(0,0),ende(0,1),ende(0,2),ende(0,3),ende(0,4),ende(0,5),ende(0,6);
float yerror[7]=0.1348,0.1943,0.2604,0.3004,0.1987,0.07855,0.08094;
for (int ipoint=0; ipoint<7; ipoint++)
    gr1->SetPoint(ipoint, x[ipoint], y[ipoint]);
    gr1->SetPointError(ipoint, 0, yerror[ipoint]);

//gr1->Draw("A*");
//gr1->SetMarkerStyle(20);

c1->Modified();
c1->cd();
c1->SetSelected(c1);

TF1* grfit1 = new TF1("grfit1","12*pi*x*x*[0]*[0]*0.384*10^6/([1]*[1]*((x*x-[1]*[1]
grfit1->SetParameters(0.084,91.08,2.7);

gr1->SetMarkerStyle(20);
gr1->Fit("grfit1","", "",87,96);
gr1->Draw("ap");

```

```

cout << "Gamma_e=" << grfit1->GetParameter(0)<<endl;
cout << "M_z=" << grfit1->GetParameter(1)<<endl;
cout << "Gamma_z=" << grfit1->GetParameter(2)<<endl;

grfit1->SetParName(0,"Gamma_e");
grfit1->SetParName(1,"M_z");
grfit1->SetParName(2,"Gamma_z");

gStyle->SetOptFit(111);
gr1->SetTitle("Z_0 Resonanz Elektronen");
gr1->GetXaxis()->SetTitle("Schwerpunktsenergie/GeV");
gr1->GetYaxis()->SetTitle("Sigma/nb");

TCanvas *c2 = new TCanvas("c2","Mu",200,10,700,500);

TGraphErrors* gr2= new TGraphErrors(7);
float x[7]=88.48,89.47,90.22,91.23,91.97,92.97,93.72;
float y[7]=ende(1,0),ende(1,1),ende(1,2),ende(1,3),ende(1,4),ende(1,5),ende(1,6);
float yerror[7]=0.02888,0.04837,0.07398,0.0356,0.05912,0.0413,0.02402;
for (int ipoint=0; ipoint<7; ipoint++)
    gr2->SetPoint(ipoint, x[ipoint], y[ipoint]);
    gr2->SetPointError(ipoint, 0, yerror[ipoint]);

TF1* grfit2 = new TF1("grfit2","12*pi*x*x*0.08314*[0]*0.384*10^6/([1]*[1]*((x*x-[1]*[1]
grfit2->SetParameters(0.084,91.08,2.7);
gr2->SetMarkerStyle(20);
gr2->Fit("grfit2","", "",87,96);
gr2->Draw("ap");
cout << "Gamma_e=" << grfit2->GetParameter(0)<<endl;
cout << "M_z=" << grfit2->GetParameter(1)<<endl;
cout << "Gamma_z=" << grfit2->GetParameter(2)<<endl;

grfit2->SetParName(0,"Gamma_m");
grfit2->SetParName(1,"M_z");
grfit2->SetParName(2,"Gamma_z");

gStyle->SetOptFit(111);
gr2->SetTitle("Z_0 Resonanz Myonen");

```

```
gr2->GetXaxis()->SetTitle("Schwerpunktsenergie/GeV");
gr2->GetYaxis()->SetTitle("Sigma/nb");
```

```
TCanvas *c3 = new TCanvas("c3","Tau",200,10,700,500);
```

```
TGraphErrors* gr3= new TGraphErrors(7);
float x[7]=88.48,89.47,90.22,91.23,91.97,92.97,93.72;
float y[7]=ende(2,0),ende(2,1),ende(2,2),ende(2,3),ende(2,4),ende(2,5),ende(2,6);
float yerror[7]=0.03507,0.05383,0.08051,0.04022,0.06599,0.04886,0.02705;
for (int ipoint=0; ipoint<7; ipoint++)
    gr3->SetPoint(ipoint, x[ipoint], y[ipoint]);
    gr3->SetPointError(ipoint, 0, yerror[ipoint]);
```

```
TF1* grfit3 = new TF1("grfit3","12*pi*x*x*0.08314*[0]*0.384*10^6/([1]*[1]*((x*x-[1]*
grfit3->SetParameters(0.084,91.08,2.7);
gr3->SetMarkerStyle(20);
gr3->Fit("grfit3","", "",87,96);
gr3->Draw("ap");
cout << "Gamma_e=" << grfit3->GetParameter(0)<<endl;
cout << "M_z=" << grfit3->GetParameter(1)<<endl;
cout << "Gamma_z=" << grfit3->GetParameter(2)<<endl;
```

```
grfit3->SetParName(0,"Gamma_t");
grfit3->SetParName(1,"M_z");
grfit3->SetParName(2,"Gamma_z");
```

```
gr3->SetTitle("Z_0 Resonanz Tau");
gr3->GetXaxis()->SetTitle("Schwerpunktsenergie/GeV");
gr3->GetYaxis()->SetTitle("Sigma/nb");
```

```
TCanvas *c4 = new TCanvas("c4","QUARK",200,10,700,500);
```

```
TGraphErrors* gr4= new TGraphErrors(7);
float x[7]=88.48,89.47,90.22,91.23,91.97,92.97,93.72;
float y[7]=ende(3,0),ende(3,1),ende(3,2),ende(3,3),ende(3,4),ende(3,5),ende(3,6);
float yerror[7]=0.1386,0.2343,0.3896,0.3251,0.3475,0.2162,0.1196;
for (int ipoint=0; ipoint<7; ipoint++)
    gr4->SetPoint(ipoint, x[ipoint], y[ipoint]);
```



```

    gr4->SetPointError(ipoint, 0, yerror[ipoint]);

TF1* grfit4 = new TF1("grfit4","12*pi*x*x*0.08314*[0]*0.384*10^6/([1]*[1]*((x*x-[1]*
grfit4->SetParameters(0.084,91.08,2.7);
gr4->SetMarkerStyle(20);
gr4->Fit("grfit4","", "",87,96);
gr4->Draw("ap");
cout << "Gamma_e=" << grfit4->GetParameter(0)<<endl;
cout << "M_z=" << grfit4->GetParameter(1)<<endl;
cout << "Gamma_z=" << grfit4->GetParameter(2)<<endl;

grfit4->SetParName(0,"Gamma_q");
grfit4->SetParName(1,"M_z");
grfit4->SetParName(2,"Gamma_z");

gr4->SetTitle("Z_0 Resonanz Quarks");
gr4->GetXaxis()->SetTitle("Schwerpunktsenergie/GeV");
gr4->GetYaxis()->SetTitle("Sigma/nb");

void cos5() //Berechnen der Vorwärts-Rückwärts-Asymmetrie
gROOT->Reset();
TFile* daten[6];
daten[0]= new TFile("daten_1.root");

Float_t      run;
Float_t      event;
Float_t      Ncharged;
Float_t      Pcharged;
Float_t      E_ecal;
Float_t      E_hcal;
Float_t      E_lep;
Float_t      cos_thru;
Float_t      cos_thet;

TCanvas *c1 = new TCanvas("c1","drawall",200,10,700,500)

for (Int_t q=0; q <1; q++)

```

```

gROOT->Reset();

//fit st

Float_t      run;
Float_t      event;
Float_t      Ncharged;
Float_t      Pcharged;
Float_t      E_ecal;
Float_t      E_hcal;
Float_t      E_lep;
Float_t      cos_thru;
Float_t      cos_thet;

TCanvas *c2 = new TCanvas("c2","drawall",200,10,700,500);
gStyle->SetOptFit(111);

Int_t neventsMM = 0;

daten[q]->cd();
TTree *tree = (TTree*)daten[q]->Get("h33");

tree->SetBranchAddress("run",&run);
tree->SetBranchAddress("event",&event);
tree->SetBranchAddress("Ncharged",&Ncharged);
tree->SetBranchAddress("Pcharged",&Pcharged);
tree->SetBranchAddress("E_ecal",&E_ecal);
tree->SetBranchAddress("E_hcal",&E_hcal);
tree->SetBranchAddress("E_lep",&E_lep);
tree->SetBranchAddress("cos_thru",&cos_thru);
tree->SetBranchAddress("cos_thet",&cos_thet);

Int_t nentries = tree->GetEntries();

TH1F* histoMMcos = new TH1F("histoMMcos","",50,-1,1);

for (Int_t i=0; i<nentries;i++)
tree->GetEntry(i);
if(Pcharged>80 && E_ecal<10 && E_lep<46 && E_lep>45.9)
    histoMMcos->Fill(cos_thet);
neventsMM++;

```

```
cout<<"Anzahl Myonen: "<<neventsMM<<endl;

histoMMcos->GetXaxis()->SetTitle("cos_thet");
histoMMcos->GetYaxis()->SetTitle("Events");

histoMMcos->SetLineColor(kRed);
histoMMcos->Draw();

TLegend *leg = new TLegend(0.652406,0.664021,0.885918,0.873016,"","brNDC");
leg->SetTextSize(0.0478723);
leg->SetLineColor(0);
//leg->AddEntry("histoEE","e++e- #rightarrow e++e-","1");
//leg->Draw();

c2->Modified();
c2->cd();
c2->SetSelected(c1);

//TFile* histos = new TFile("Pro.root");
//TGraphErrors *gr = (TGraphErrors*)histos->Get("histoEE");

//gr->SetLineWidth(2);
//gr->SetLineColor(1);

TF1* grfit = new TF1("grfit","0.1575*10^(-8)*([0]*(1+x*x)+2*[1]*(x))",-0.9,0.9);
grfit->SetParameters(1000000000,1000000000);
//gr->Draw("A*");
histoMMcos->SetMarkerStyle(20);
histoMMcos->Fit("grfit","","",-0.9,0.9);
cout << "ParamF1=" << grfit->GetParameter(0)<<endl;
cout << "ParamF2=" << grfit->GetParameter(1)<<endl;
```