# The Search for Di-lepton Signatures from Squarks and Gluinos in $\overline{p}p$ Collisions at $\sqrt{s} = 1.8$ TeV

By

Richard J. Genik II

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Physics and Astronomy

1998

# ABSTRACT

## The Search for Di-lepton Signatures from Squarks and Gluinos in $\overline{p}p$ Collisions at $\sqrt{s} = 1.8$ TeV

By

Richard J. Genik II

A search for Supergravity squark and gluino decays into di-leptons is presented. A novel search strategy of optimizing kinematic thresholds at each point in the three dimensional space of $m_0$-$m_{1/2}$-$\tan\beta$ is employed. The model space is randomly scanned using a parameterized fast Monte Carlo. No events are observed above Standard Model background in 107.6 pb$^{-1}$ of Tevatron data collected by the DØ detector between 1993-96. Exclusion contours are presented in the $m_0$-$m_{1/2}$ plane. At the 95% confidence level, a lower limit is set on the mass of gluinos of 129 GeV/c$^2$ and on the mass of squarks of 138 GeV/c$^2$ for all $\tan\beta < 10$.

To My Family, Richard, Linda, Robert, and Laura Genik.

# ACKNOWLEDGEMENTS

The process of completing a Ph.D. thesis begins with a young idealistic student setting off on a magical journey to make the perfect measurement. Starry-eyed and full of enthusiasm, he will work with reckless abandon to make every detail correct, and to push the limits of what others before him have accomplished.

Next comes the idea of finishing. To finish, some of the perfection needs to be compromised toward, but not transformed into, mediocrity. Forcing this compromise is the responsibility of the set of the student's advisors.

I'd like to thank the whole team of advisors who worked tirelessly for 5 years to crush my idealism.

I would be remiss not to mention several people that contributed to this analysis. First, I would like to thank my advisor, Jim Linnemann, for giving me the opportunity to work at Fermilab. In addition, his extensive knowledge and experience provided excellent guidance through many stages of this analysis. John Hobbs deserves the credit for convincing me that there is more than one charged lepton. Many thanks to Sarah Eno and Lee Lueking for their work on QSIM. My appreciation also to Chris Kolda, Steve Mrenna, and Greg Anderson for their patient assistance in matters of theory.

I would also like to thank all of the people I've had the privilege to meet, work, and become friends with during my tenure. A complete listing is not possible, but I'd like to recognize Eric Flattum, John Krane, Ian Adam, Kevin Davis, Jim McKinley, and the members of The Porch Club. Several outstanding senior scientists added positively to my experience at DØ. I would like to single Hugh Montgomery for his generosity and many spirited conversations.

Finally, worthy of mention are those people that helped to keep me sane during graduate school with a warm demeanor and conversations that weren't full of words ending in -ino: Judy, Nancy, Michelle, Chivas, Fred and the gang at the UC; Norb Jillich, Tim Barr, and all of the other dudes and dudettes from the dart leagues; the Unified Fielders championship softball team; the Prerna and Fermions championship soccer teams; Nick and Nancy from the Q. Whether their efforts were successful is a question for the historians.

Particle physics is the Universe's local attempt to figure itself out.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# The Standard Model and Beyond

This chapter contains a short introduction to the Standard Model, and motivation for why it can't be the complete picture. Supersymmetry and Supergravity (SUGRA), extensions to the Standard Model, are described, as well as how SUGRA may produce di-lepton events. Finally, the search strategy is discussed.

Following chapters discuss the experimental apparatus, particle identification, the fast Monte Carlo written for this analysis, calculation of background, and results. Appendix A describes the fast Monte Carlo in detail. Appendix B describes the author's contributions to DØ, and Appendices C to H provide supporting documentation.

## 1.1 The Standard Model

The Standard Model is a relativistic quantum field theory based on the principle of least action and local gauge invariance. [1]

The concept of a field is introduced in classical physics to account for the interaction between two bodies. The interaction between body 1 and body 2 is viewed as body 2 interacting with a field created by body 1. For charged bodies, the field is a three component vector defined at each point in space-time, $\mathbf{E}(\mathbf{x}, t)$.

The concept of a quantum field is introduced to account for production of particles. The basic idea is that quantum mechanical excitations of a field appear as particles of definite mass and spin. For the electric field, these excitations appear as photon exchange between two charged bodies.

A relativistic quantum field contains pairs of particles of the same mass and spin, but opposite electric charge (antiparticles).

A gauge transformation essentially gives an internal (unobservable) phase to a field. A field that remains unchanged under a static gauge transformation is said to be globally gauge invariant. A field that remains unchanged under a space-time dependent gauge transformation is said to be locally gauge invariant.

The Standard Model consists of two areas of study: Quantum Electrodynamics (QED) and Quantum Chromodynamics (QCD). QED studies electromagnetic and weak interactions. QCD studies strong interactions.

Table 1.1: Elementary particles in the Standard Model

| Generation | I | II | III |
|---|---|---|---|
| Quarks | up ($u$)<br>down ($d$) | charm ($c$)<br>strange ($s$) | top ($t$)<br>bottom ($b$) |
| Leptons | electron ($e$)<br>electron neutrino ($\nu_e$) | muon ($\mu$)<br>muon neutrino ($\nu_\mu$) | tau ($\tau$)<br>tau neutrino ($\nu_\tau$) |

Table 1.2: Mediators in the Standard Model

| Mediator | Symbol | Force Carried | Rest Mass |
|---|---|---|---|
| gluon | $g$ | strong nuclear | 0 |
| Vector Bosons | $W^{\pm}, Z$ | weak nuclear | 80 and 92 Gev/c$^2$ |
| photon | $\gamma$ | electromagnetic | 0 |

### 1.1.1 Particles of the Standard Model

The particles of the Standard Model are leptons and quarks, and they interact by exchanging force carrying particles, or *mediators*. There are six "flavors" of quarks and leptons, grouped into three generations (Table 1.1), all possessing spin-$\frac{1}{2}$. Table 1.2 lists the mediators (all with spin-1) and the force each carries.

There is an additional field in the Standard Model that is responsible for creating mass, the scalar Higgs field.

## 1.2 The Allure of New Physics

The Standard Model contains several theoretical problems. First, the Standard Model generates divergent radiative corrections to the scalar Higgs mass which must be compensated with a counterterm,

$$M_h^2 \sim M_{h0}^2 + \frac{\lambda}{4\pi^2}\Lambda^2 + \delta M_h^2$$

where $M_{h0}^2$ is the bare mass, $\lambda$ is the coupling, $\Lambda$ is a cutoff scale, and $\delta M_h^2$ is the compensating term. $\Lambda$ is set at the scale where new interactions are expected to contribute. In the Standard Model, there are no new interactions between the electroweak scale, $M_W \sim 100$ GeV/c$^2$, and the scale where gravitational couplings become comparable to strong and electroweak couplings, the *Planck scale*, $M_{\text{Planck}} \sim 10^{19}$ GeV/c$^2$.[1] Unitarity[1] demands that $M_h$ must be less than around 800 GeV/c$^2$, requiring $\delta M_h^2$ be adjusted to about one part in $10^{16}$. This extreme sensitivity is present at each order in perturbation theory, so one must fine tune the corrections indefinitely; this is known as *the fine tuning problem.*[3]

Second, the Standard Model contains a large scale hierarchy, $M_W \ll M_{\text{Planck}}$, which is unstable; small changes at one scale can lead to large variations at the other scale. Usually, large hierarchies are protected by an intervening approximate symmetry; for example, classical QCD scale invariance protects $m_{\text{proton}} \ll M_{\text{Planck}}$, and chiral symmetry (with $m_e = 0$) protects $m_e \ll M_W$.[2] The Standard Model contains no symmetry protecting $M_W \ll M_{\text{Planck}}$ and suggests an additional undiscovered symmetry.

[1] Unitarity, or the conservation of probability, is imposed as a constraint in the construction of any quantum field theory. This leads to the requirement that all production cross sections must fall as $s^{-1}$ at energies far above the mass of the produced particle.[1] The scattering of longitudinal gauge bosons violates this limit when the Higgs is too heavy.[2]

[2] In the Standard Model, all of the fermion fields are massless. The electron mass is merely used as an example here.

Finally, electroweak symmetry breaking (EWSB) is introduced artificially: the symmetry is spontaneously broken at just the right scale to agree with experiment. The Standard Model contains no hint of the origin of this breaking. Extensions to the Standard Model that try to explain EWSB usually require currently undiscovered particles near the weak scale ($\sim 100$ GeV).

The predictions of the Standard Model have withstood all current precision measurements, indicating the model's validity up to the energies already probed. Occasionally, a result will appear in the literature that seems to favor one extension of the Standard Model over another. Subsequent results usually modify this back to the bare Standard Model. No matter how experimental results drift with time, the aforementioned theoretical problems with the Standard Model persist, *requiring* new physics, usually at or near the weak scale.

## 1.3 Supersymmetry

One way to remove the quadratic divergence in radiative corrections to the Higgs mass is to define the Lagrangian such that exact cancellation occurs in the perturbation expansion. The one-loop corrections to the mass become

$$M_h^2 \sim M_{h0}^2 + \frac{g_F^2}{4\pi^2}\left(\Lambda^2 + m_F^2\right) - \frac{g_S^2}{4\pi^2}\left(\Lambda^2 + m_S^2\right) + ...$$

where the generic coupling $\lambda$ has been replaced with $g_{F,S}$ (for fermion and scalar fields) and terms include the mass of the particles responsible for the corrections.

The omitted terms either diverge only logarithmically or are uninteresting. To ensure the relative minus sign, one of the fields must obey Fermi-Dirac statistics, while the other must obey Bose-Einstein statistics; thus, for every Dirac particle in the Standard Model, a scalar partner must exist, and for every boson, a Dirac partner must exist. The two fields (particle doublets) are combined into a *superfield* containing particles and their superfield partners, *sparticles.* For example, quarks would partner with a scalar quark, or *squark.* The spin of a sparticle differs by one-half unit from its partner, but otherwise has the same quantum numbers. This symmetry between fermions and bosons is called *Supersymmetry* (SUSY). Table 1.3 presents the complete SUSY particle spectrum. By examining the above equation, one can see that SUSY models implicitly solve the fine tuning problem, provided $|g_F| = |g_S|$ and $m_F^2 \sim m_S^2$.[3] In addition, SUSY provides an origin for EWSB and solves the hierarchy problem by protecting the scale difference with the chiral symmetry of the superfield.[4]

### 1.3.1 Minimal SUSY

The SUSY extension to the Standard Model which is complete and adds the least number of undiscovered particles is called *minimal SUSY.* To construct minimal SUSY, special attention must be paid to the Higgs sector. The Standard Model

[3] Exact cancellation occurs when the masses of the partners are equal; an intrinsic feature of unbroken supersymmetry. SUSY must be a broken symmetry, at scale $M_{\rm SUSY} \sim 10^{16}$ GeV/c$^2$,[3] else the electron would have a scalar partner with the same mass and quantum numbers. Attempts to quantify how similar particle and sparticle masses must be result in a general mass limit of about a TeV for sparticles.

[4] A chiral theory is one in which the gauge bosons couple differently to left- and right-handed fermion states.

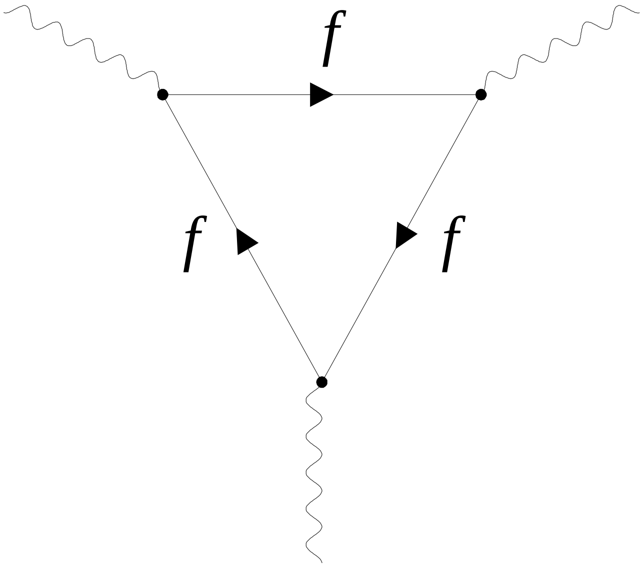


Figure 1.1: Example of triangle diagram which produces a gauge anomaly. There exists such a diagram for every fermion within a model.

contains a single $SU(2)_L$ Higgs doublet consisting of 2 complex scalar fields, requiring 4 real scalar parameters. Three of these 4 degrees of freedom give mass to the $W$ and $Z$ bosons, so the Standard Model contains one physical scalar Higgs. When a fermion partner is added for the Higgs, the problem of triangle *gauge anomalies* is revived.

Figure 1.1 is an example of a gauge anomaly. Such an interaction exists for every fermion in the Standard Model. Because the left and right fermion couplings are unequal in standard electroweak theory, the individual fermion contributions become infinite in an awkward way that upsets standard renormalization. To free the theory of anomalies, the net contribution of these diagrams must be zero. This is accomplished in the Standard Model by requiring that generations be complete; the contributions from $u$ and $d$ quarks exactly cancel that from the electron.[5]

With the addition of a fermion partner to the Higgs, the exact cancellation of gauge anomalies is upset; at least one more fermion must be added to cancel its contribution. Within the framework of SUSY, the minimal option is to add an additional superfield. Therefore, minimal SUSY contains two Higgs doublets instead of a single one as in the Standard Model. Each particle doublet contains two complex elements and thus eight real scalars. Three of these scalars give mass to the $W$ and $Z$ bosons; hence, minimal SUSY contains 5 physical Higgs bosons: two charged, two neutral, and one neutral pseudoscalar.

### 1.3.2 R-parity and the LSP

A new quantum number, $R$, is introduced in SUSY models. $R = +1$ for particles and $R = -1$ for sparticles, and the total R of a state is calculated by multiplying all of the individual particle and sparticle R's. A SUSY model can be classified as R-parity conserving, or R-parity violating.

In all current colliders, the initial state is assumed to only contains particles, so its R value is $+1$; therefore, in R-parity conserving models, the final state must contain an even number of sparticles, usually two. Each SUSY model of this type

[5] This feature of the Standard Model required the existence of the top quark to complete the third generation.[4] The GIM mechanism also requires complete generations, but for different reasons. (The GIM mechanism rotates quark mass eigenstates to weak isospin eigenstates. This formalism defines the Kobayashi-Maskawa mixing matrix.[1])

Table 1.3: The minimal SUSY particle spectrum. The gauginos mix according to charge; hence, charginos are mixtures of charged higgsinos and winos, and neutralinos are mixtures of neutral higgsinos, zinos, and photinos. For the squarks and sleptons a subscript (L or R) is used denoting to which particle chiral state the sparticle is a partner. Alternate notations exist for some sparticles.

| **Particles** | | **Sparticles** | | |
|---|---|---|---|---|
| *Name* | *Symbol* | *Name* | *Symbol* | *Alternate* |
| gluon | $g$ | gluino | $\tilde{g}$ | |
| charged higgs<br>W boson | $H^{\pm}$<br>$W^{\pm}$ | charginos | $\widetilde{W}^{\pm}_{1,2}$ | $\tilde{\chi}^{\pm}_{1,2}$, $\tilde{C}_{1,2}$ |
| light higgs<br>heavy higgs<br>pseudoscalar higgs<br>Z boson<br>photon | $h$<br>$H$<br>$A$<br>$Z$<br>$\gamma$ | neutralinos | $\tilde{Z}_{1-4}$ | $\tilde{\chi}^{0}_{1-4}$, $\widetilde{N}_{1-4}$ |
| graviton | G | gravitino | $\tilde{G}$ | |
| quarks | $q_{L,R}$ | squarks | $\tilde{q}_{L,R}$ | |
| leptons | $l_{L,R}$, $\nu_L$ | sleptons | $\tilde{l}_{L,R}$, $\tilde{\nu}_L$ | |

contains a *Lightest Supersymmetric Particle* (LSP) which cannot decay. Much like neutrinos, the LSPs are assumed neutral and weakly interacting, and they provide a candidate for Cold Dark Matter. The general SUSY interaction of this type can be described

$$\overline{p}p \longrightarrow 2\,\mathrm{LSP} + X$$

where $X$ represents collection of Standard Model particles such as leptons or jets from hadronization of quarks. The presence of LSPs in an event is inferred by an unbalance of net momentum transverse to the beam direction ($\not{E}_T$, Section 2.2.1).

### 1.3.3 Sparticles and Nomenclature

The general SUSY naming convention is to precede a scalar partner's name with an "s", giving us squarks and sleptons generically, and sneutrinos and selectrons specifically.[6] The spin one-half gauge fermions are named by adding the suffix "ino" to the Standard Model name; therefore, the gauginos are called photinos, winos (pronounced wee′nos), zinos (zee′nos), gluinos, and Higgsinos. The electroweak gaugino fields mix according to electromagnetic charge and are more properly referred to as charginos and neutralinos. Finally, the spin 2 graviton superpartner is the spin 3/2 gravitino. Table 1.3 shows the complete minimal SUSY particle spectrum.[7]

## 1.4 Minimal Supergravity

The exact mechanism of breaking Supersymmetry is currently unknown. In minimal SUSY, there are a number of soft breaking mass parameters. "Soft" means that they break the mass degeneracy between particles and sparticles without reintroducing quadratic divergences, and respect the gauge invariance of the theory. These soft parameters are extra mass terms for sparticles, and trilinear scalar couplings (the soft SUSY breaking terms).

This analysis assumes the existence of extra superfields that couple universally to minimal SUSY particles through gravitational interactions at the scale of $M_{\mathrm{SUSY}}$.

[6] Terms such as "stop squark" are redundant; one should refer to a stop quark, a top squark, or just a stop.

[7] While somewhat whimsical, the SUSY naming convention is extremely concise.

This scale is not experimentally reachable and the extra fields are called the *hidden sector*. Interaction terms between the hidden sector and particle superfields break SUSY. This scenario is minimal Supergravity (SUGRA). [5]

The number of additional free parameters in minimal SUSY exceeds 100. These parameters include sparticle masses, sparticle and particle couplings to the assumed SUSY breaking mechanism, and the soft SUSY breaking terms. In SUGRA, the universal coupling of sparticles and particles to the extra superfields reduces this number considerably. In addition, sparticle masses are defined to common values at $M_{\mathrm{SUSY}}$, and Renormalization Group Equations are used to run their masses down to the weak scale (where different Yukawa couplings with particles and soft breaking terms have removed the degeneracy).

The SUGRA parameters, now a total of 4 plus a sign, are

$m_0$ **and** $m_{1/2}$ In SUGRA, all scalar sparticles have a common mass at $M_{\mathrm{SUSY}}$, as do all gauginos. $m_0$ is the common scalar mass and $m_{1/2}$ is the common gaugino mass.

$A_0$ This is a common soft trilinear interaction parameter describing the left and right chiral state mixing of the third generation of sparticles at $M_{\mathrm{SUSY}}$. $A_0$ is set to zero in this analysis since it mainly affects the mass of the light stop, which is not considered.

$\tan\beta$ **and sign(**$\mu$**)** $\tan\beta$ is the ratio of the Higgs vacuum expectation values, and is a soft breaking term. The soft Higgsino mass parameter, $\mu$, is determined

Table 1.4: Production of squarks and gluinos.

| | |
|---|---|
| pair production | $q\bar{q} \to \tilde{q}\bar{\tilde{q}}$, $gg \to \tilde{q}\bar{\tilde{q}}$, $q\bar{q} \to \tilde{g}\tilde{g}$, and $gg \to \tilde{g}\tilde{g}$ |
| gluino-squark production | $qg \to \tilde{q}\tilde{g}$ |
| gaugino associated production | $qg \to \tilde{q}\tilde{W}_i$ and $qg \to \tilde{q}\tilde{Z}_i$ |

(up to a sign) by requiring radiative EWSB occurs at the correct scale. The sign of $\mu$ will affect the mass differences of the gauginos. Positive $\mu$ will lead to smaller mass differences than negative $\mu$. These smaller mass differences result in leptons that are below the identification threshold at DØ; hence, $\mu$ is set negative in this analysis.

In SUGRA, the LSP is the lightest neutralino, $\tilde{Z}_1$.[8]

## 1.5 Leptonic SUGRA Signatures

The SUGRA sparticle masses are dominantly functions of $m_0$ and $m_{1/2}$. High values result is high sparticle mass. This analysis searches for squarks and gluinos (Table 1.4) decaying to a final state including two light leptons ($e$ or $\mu$), $\not{E}_T$, and a minimum of two jets. Leptonic final states necessarily include a chargino or neutralino in the decay chain. For positive $\mu$, the mass differences between the gauginos become smaller and produced leptons are softer, making them difficult to identify with the DØ detector. This analysis focuses specifically on $\mu < 0$.

[8] There is a small corner of theoretically allowed space where the LSP is the sneutrino. This corner is ruled out by LEP I.[6] (LEP I is the generic term used to cite combined results of $Z$-pole collisions recorded at ALEPH, DELPHI, L3, and OPAL).

The source of charginos and neutralinos are gaugino associated production, and decays of squarks. Gluinos decay into squark-quark pairs, and the squarks again are a source.

$\tilde{Z}_2 \rightarrow l\bar{l} + \tilde{Z}_1$ is the major source of lepton pair final states. Decay of two $\tilde{W}_1$ (via $\tilde{W}_1 \rightarrow l\nu_l + \tilde{Z}_1$) is the major source of $e\mu$ final states, and contributes to lepton pair final states.

Production and decay rates vary greatly with model parameters. Of particular note is that for $\tan\beta > 2.5$, mass couplings begin to favor leptonic decay of gauginos toward the $\tau$ and $q\bar{q}$ channels. Coupling to $e$ or $\mu$ is almost zero for $\tan\beta \sim 6$. In addition, hadronic or neutrino channels can be favored over charged leptons, depending on the masses of daughter particles. In these regions of space, the jets + $\not{E}_T$ signature will dominate. That signature is, however, rather unspecific to SUGRA and leptonic final states possess greater potential for discovery.

A complete enumeration of all possible production and decay modes is not possible in a reasonable treatise. Instead, consider the point $m_0 = 140$ GeV, $m_{1/2} = 90$, $\tan\beta = 2.0$, and $\mu$ negative. Sparticle masses are shown in Table 1.5, and the important decays are shown in Table 1.6. Charm and strange squarks have the same mass and decays as $\tilde{u}$ and $\tilde{d}$. Due to Yukawa couplings with particles, sbottom and stop are lighter.[5]

The example point is the typical case. The majority of leptonic final states arise from left-handed squark production and gluino decays through the light sbottom. Right-handed squarks add some to the lepton signature, but usually account for jet

Table 1.5: Masses of some sparticles for example point. The $\tilde{Z}_1$ (LSP) mass is 40 GeV/c$^2$.

| Sparticle | $\tilde{u}_L$ | $\tilde{u}_R$ | $\tilde{d}_L$ | $\tilde{d}_R$ |
|---|---|---|---|---|
| Mass (GeV/c$^2$) | 263 | 258 | 270 | 260 |
| Sparticle | $\tilde{b}_1$ | $\tilde{g}$ | $\tilde{W}_1$ | $\tilde{Z}_2$ |
| Mass (GeV/c$^2$) | 235 | 270 | 87 | 88 |

Table 1.6: Partial decay table for example point.

| $\tilde{u}_L$ | | $\tilde{u}_R$ | | $\tilde{d}_L$ | | $\tilde{d}_R$ | |
|---|---|---|---|---|---|---|---|
| Decay | % | Decay | % | Decay | % | Decay | % |
| $\tilde{W}_1 d$ | 67 | $\tilde{W}_1 d$ | - | $\tilde{W}_1 u$ | 62 | $\tilde{W}_1 u$ | - |
| $\tilde{Z}_1 u$ | 6 | $\tilde{Z}_1 u$ | 98 | $\tilde{Z}_1 d$ | 0 | $\tilde{Z}_1 d$ | 98 |
| $\tilde{Z}_2 u$ | 27 | $\tilde{Z}_2 U$ | 2 | $\tilde{Z}_2 d$ | 37 | $\tilde{Z}_2 d$ | 2 |
| $\tilde{b}_1$ | | $\tilde{g}$ | | $\tilde{W}_1$ | | $\tilde{Z}_2$ | |
| Decay | % | Decay | % | Decay | % | Decay | % |
| $\tilde{Z}_2 b$ | 99.8 | $\tilde{b}_1 b$ | 64 | $\tilde{Z}_1 e\nu$ | 15 | $\tilde{Z}_1 ee$ | 22 |
| | | $\tilde{u}_R u$ | 7.8 | $\tilde{Z}_1 \mu\nu$ | 15 | $\tilde{Z}_1 \mu\mu$ | 22 |
| | | $\tilde{d}_R d$ | 5.2 | $\tilde{Z}_1 \tau\nu$ | 15 | $\tilde{Z}_1 \tau\tau$ | 22 |
| | | $\tilde{u}_L u$ | 2.6 | $\tilde{Z}_1 qq'$ | 56 | $\tilde{Z}_1 q\overline{q}$ | 19 |
| | | | | | | $\tilde{Z}_1 \nu\overline{\nu}$ | 15 |

and $\not{E}_T$ production.

## 1.6 Search Strategy

Known processes can mimic the signal one expects from new physics, called the background to the signal. The ideal search would be free from background and accept all signal events. This is rarely the case. The usual strategy is to generate signal and background events and optimize thresholds to yield high signal efficiency and accept little background.

There is a problem with using the usual strategy when searching for Supergravity: the optimum thresholds change as a function of the input parameters. Consider the signature of two electrons, two jets, and $\not\!\!E_T$. Figure 1.2 shows the ratio of the visible cross section[9] for three jets versus two jets (top) and for higher thresholds on the leading jet and $\not\!\!E_T$ versus a nominal one (bottom). Requiring more jets, or higher energy jets and $\not\!\!E_T$, will decrease the background. Clearly, when the ratio of the stricter requirements to the looser ones is near unity, the stricter requirements are more optimized (less background, same signal). However, there are models where the stricter requirements greatly reduce the signal efficiency, leaving the looser requirements as the only viable option.

The novel strategy employed in this analysis is to define a large number of signatures and thresholds to try to optimize the search channel at each point in model space. In general, each channel will require 2 leptons (electron or muon), two or more jets, and $\not\!\!E_T$. Exact definition of these *kinematic channels* will be given in Chapter 4.

[9] This is the detectable cross section after including corrections for branching ratio, acceptance, and particle identification efficiencies.

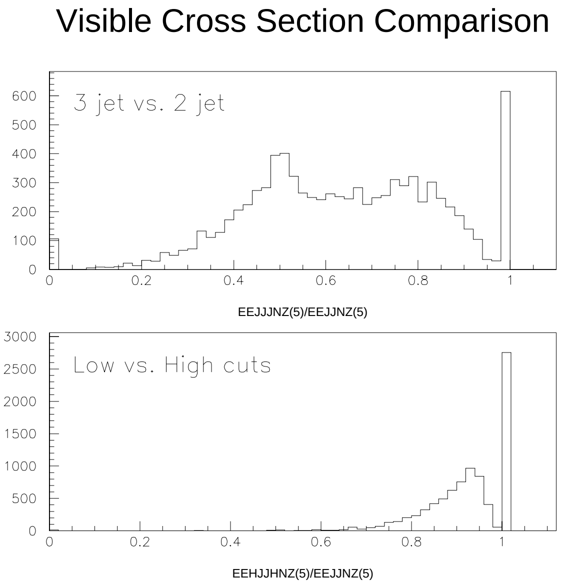


Figure 1.2: Ratio of visible cross sections for about 8000 Supergravity models. Top plot: EEJJJNZ signature means 2 electrons, 3 jets, and exclude di-electron pairs with an invariant mass around the $Z$ boson; EEJJNZ is the same, except only two jets are required. Bottom plot: EEHJJHNZ is the two-jet di-electron signature with higher thresholds on the leading jet and $\not{E}_T$.

# Chapter 2

# Fermilab and the DØ Detector

This chapter provides an overview of the experimental apparatus. This material has been described in several publications and theses, and is only included for completeness. Reference [7] is considered the official DØ documentation. References [8] and [9] are excellent descriptions of accelerator and detector physics and hardware.

## 2.1 The Fermilab Accelerator

Figure 2.1 shows the location of the various subsystems in the Fermilab accelerator complex, and the location of the two main collision detectors, DØ and CDF. Five accelerators are used to obtain the final $\overline{p}p$ center of mass collision energy of 1.8 TeV. These five machines, much like the transmission in a car, operate most efficiently in different energy bands.

The first stage of proton acceleration uses a commercial Cockroft-Walton to ac-

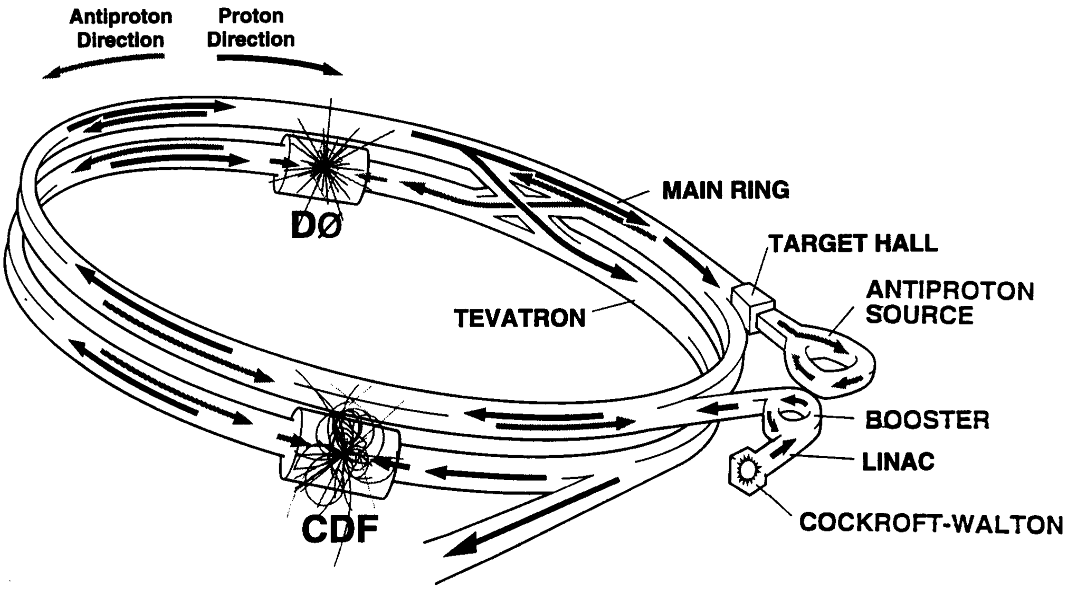


Figure 2.1: The Fermilab accelerators and experimental hall locations.

celerate $H^-$ ions to 750 keV. These ions are injected into the 150 m Linac (for Linear Accelerator) and accelerated to 400 MeV. Emerging from the Linac, the ions are passed through a carbon foil to strip their electrons, leaving bare protons. Protons entering the Booster are accelerated to 8 GeV and injected into the fourth stage, the Main Ring, housed in the 6.28 km underground tunnel. The Main Ring, composed of 1000 conventional copper coil magnets, increase the beam energy to 150 GeV. Finally, the Tevatron accelerates the beam to 900 GeV, using 1000 liquid helium cooled, superconducting magnets.

The antiproton acceleration necessarily begins with $\overline{p}$ production. Protons are accelerated in the Main Ring to 120 GeV and fired onto a nickel target. The resulting particles contain a fraction of antiprotons that pass through a momentum selector (tuned to 8 GeV) and into the Debuncher ring. The Debuncher reduces the longitudinal and transverse profile of the beam. The "cooled" antiprotons are then added to any antiprotons already stored in the Accumulator ring (the antiproton "stack") for later injection into the Tevatron via the Main Ring.

In colliding mode, the Tevatron contains "bunches" of counter-rotating protons and antiprotons. The normal configuration for Run 1 was six-on-six bunches travelling at an energy of 900 GeV, producing a crossing time of 3.5 $\mu$s. The beams are kept apart with electrostatic separators, but allowed to collide at the DØ and BØ interaction points.[1] These collision points correspond to the nominal centers of the DØ and CDF detectors.

## 2.2 The DØ Detector

The DØ detector consists of a cylindrical calorimeter surrounding a central tracker and enclosed by a toroidal muon spectrometer (Figure 2.2). This section includes some definitions and an overview of each major subsystem. Use of these detectors to identify particles is discussed in the Chapter refch:pid.

### 2.2.1 Some Definitions

This section discusses some standard detector quantities, the coordinate system, and the data sets used in the analysis.

[1] The six interaction points are named AØ - FØ. This is the origin of the name of the DØ detector.

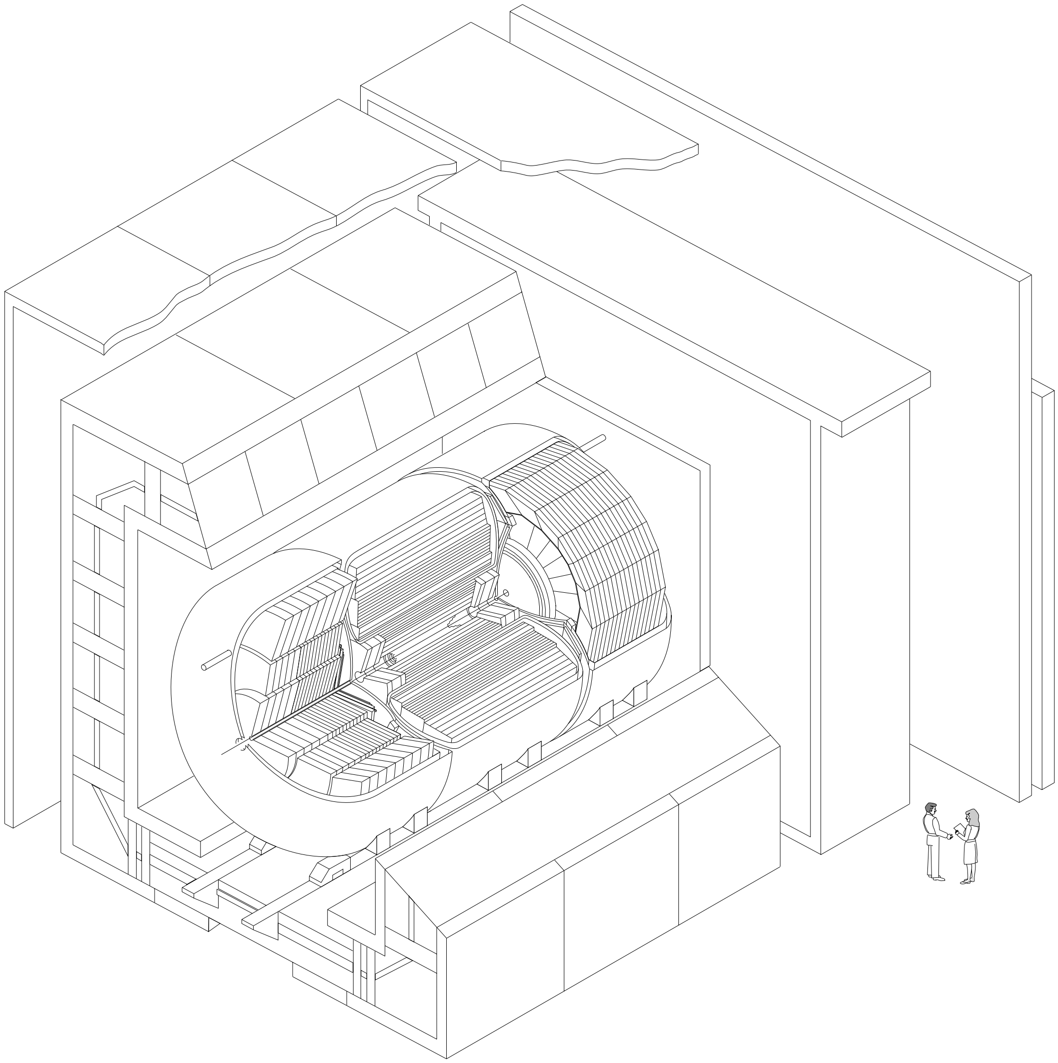


Figure 2.2: The DØ detector.

### Detector Quantities and Coordinates

Collider experiments make use of two specialized quantities: $\eta$ and $E_T$. $\eta$ is the pseudorapidity

$$\eta = -\log \tan \frac{\theta}{2} \tag{2.1}$$

where $\theta$ is the polar angle between a particles direction and the proton beam axis. Use of pseudorapidity rather than $\theta$ greatly simplifies the mathematics of Lorentz transformations. While use of pseudorapidity is strictly valid only for massless particles, it is an excellent approximation for highly relativistic particles, as are seen at DØ, because their mass energy is only a small contribution to their total energy.

The transverse energy is defined

$$E_T = E \sin \theta$$

where $E$ is a particle's total energy. Hard scattering interactions, those most likely to produce new particles, are characterized by particles with high momentum transverse to the initial beam direction. $E_T = p_T$ for massless or highly relativistic particles (in units where $\hbar = c \equiv 1$, used throughout this analysis).

The vector sum of all $E_T$ in an event should be zero in the case of an ideal detector. In a real detector, there is some finite resolution that will produce net $\vec{E_T}$. In addition, particles that pass through the detector without interacting, neutrinos or LSPs, will lead to a substantial amount of net $\vec{E_T}$ opposite to their direction. This "missing" $E_T$, needed to balance the detected $E_T$, is denoted $\not{E}_T$.

The DØ coordinate system is defined first as right-handed Cartesian, with $+z$ along the proton beam direction, and $+x$ pointing toward the Tevatron center ($+y$ then points vertically upward). A modified spherical system is defined using ($E_T$, $\eta$, $\phi$), where $\phi = 0$ along the $x$-axis and $\eta$ calculated from Equation 2.1.

One final question is the definition of $z$ from which to calculate $\eta$. The hard scattering vertex rarely occurs at the nominal center of the detector ($z = 0$), and this gives rise to two definitions: *physics* and *detector* $\eta$. Physics $\eta$ is defined using the hard scattering vertex. Detector $\eta$, $\eta_d$, is defined using $z = 0$. $\eta_d$ will prove quite useful for making fiducial cuts, since it defines a constant position within the detector.

### Data Sets and Files

The Tevatron was run in 1800 GeV colliding mode for three distinct time periods, denoted Run 1a, Run 1b, and Run 1c. Run 1a represents data collected between August 1992 and May 1993. Run 1b represents data collected from January 1994 through July 1995. Run 1c represents data collected between November 1995 and February 1996. Collectively, these data sets are referred to as Run 1.

There are four distinct file types of each event recorded. RAW data streams contain the unprocessed, high-precision data readout of the DØ detector, along with online information. Raw events were processed offline to produce standard output tapes (STAs) and data summary tapes (DSTs). DSTs underwent compression and elimination of redundant information to produce micro-DSTs. The micro-DSTs were

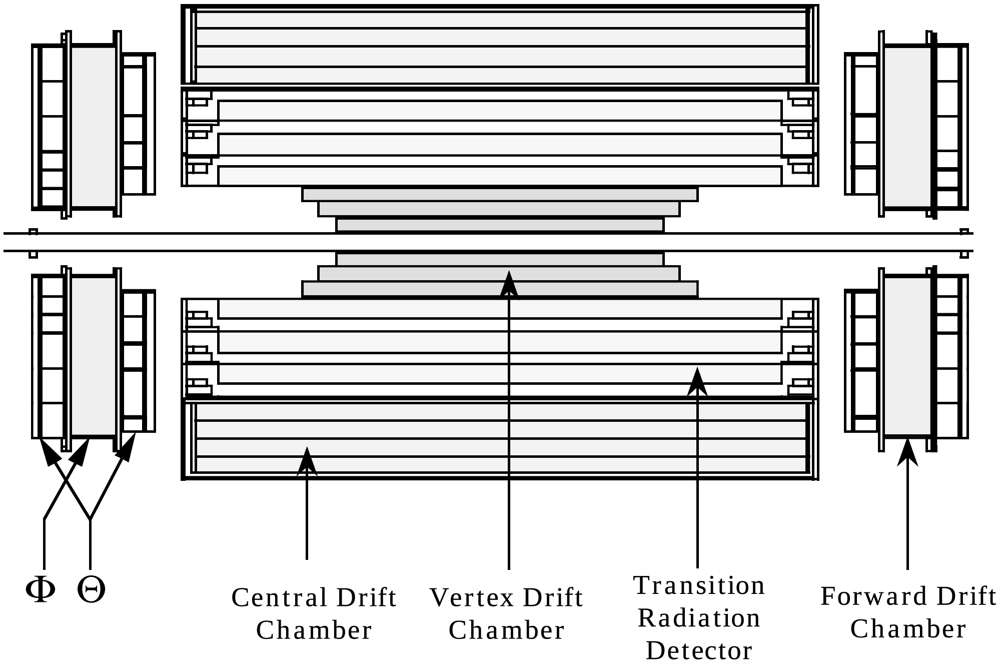


Figure 2.3: The four systems that comprise the central detector.

made available on disk for further analysis.

This analysis incorporates the Run 1b micro-DSTs and the Run 1c DSTs. Run 1a data, although useful for many other analyses, contains several different online configurations that addled the fast parameterization technique used in the current analysis (Chapter 4).

### 2.2.2 The Central Tracker

The central tracker is used to detect charged particles with minimal effect on their momentum. Ionization traces determine spatial orientation (vertex chamber, central and forward drift chambers) and transition radiation provides discriminating power between different particles (transition radiation detector). These four systems

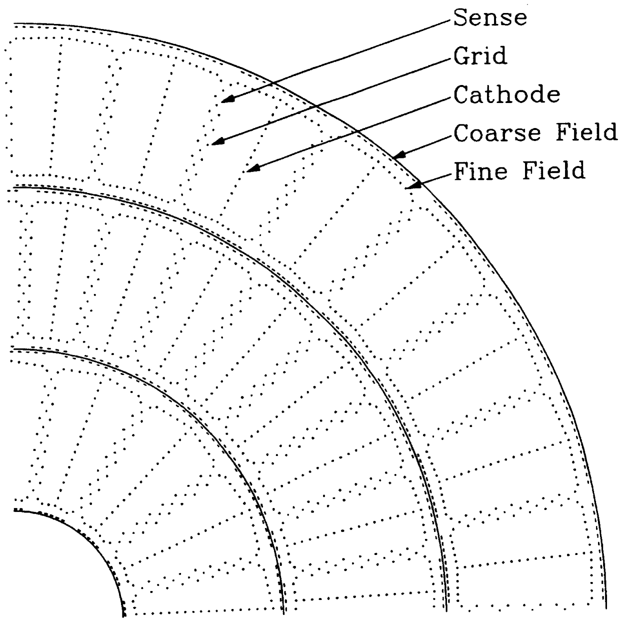


Figure 2.4: R-$\phi$ view of one quadrant of the VTX chamber.

(Figure 2.3) will each be discussed in turn.

### The Vertex Chamber

When a charged particle traverses a gaseous medium, it will ionize the gas through collisions with shell electrons. An electric field forces these ions to collect on a wire, producing a pulse. The pulse has a definite orientation in time, used to determine the intial spatial orientation of the ions, and an area, used to determine the amount of ionization produced by the particle. Such a system is know as a *drift chamber.*

The vertex chamber (VTX) is a cylindrical instrument with an inner radius of 3.7 cm, an outer radius of 16.2 cm, and a length of approximately 110 cm. The VTX consists of three concentric drift chambers with wires parallel to the beam axis

(Figure 2.4). The innermost chamber is divided into 16 cells, while the outer two are divided into 32. The cells between layers are staggered in $\phi$ to avoid uninstrumented regions and improve pattern recognition. Within each cell, adjacent sense wires are staggered 100 $\mu$m to eliminate left-right ambiguity.[2] The *active medium* (the gas ionized) is $CO_2$(95%)-$C_2H_6$(5%) doped with a small amount of $H_2O$. The water helps stabilize the detector against radiation damage.[10, 11] The sense wires operate at an electrical potential of +2.5 kV, above the threshold to cause electron cascades; hence, no information is derived from the pulse areas in the VTX.

### The Transition Radiation Detector

A highly relativistic particle traversing the boundary between two media of differing dielectric constants will emit X-rays of *transition radiation.*[12, 13] The energy of these X-rays increases with Lorentz $\gamma$, so mass and emitted energy are inversely proportional. Electrons will emit more transition radiation than heavier particles, such as pions. Measurement of X-ray energy produced may be used to distinguish between electrons and other particles.

The transition radiation detector (TRD) [14] consists of three layers, each containing a radiator (layered polypropylene foil) and an X-ray detection chamber (a proportional wire chamber, PWC). Figure 2.5 shows the layout of a TRD layer. X-rays are produced in the radiator stack, convert in the gas of the PWC (91% Xe, 7%

[2] The conversion of time to spatial orientation depends on the drifting of charge over a distance at a known velocity (the *drift velocity*). This only yields an absolute impact parameter with respect to the wire position. Staggering the wires removes this ambiguity.

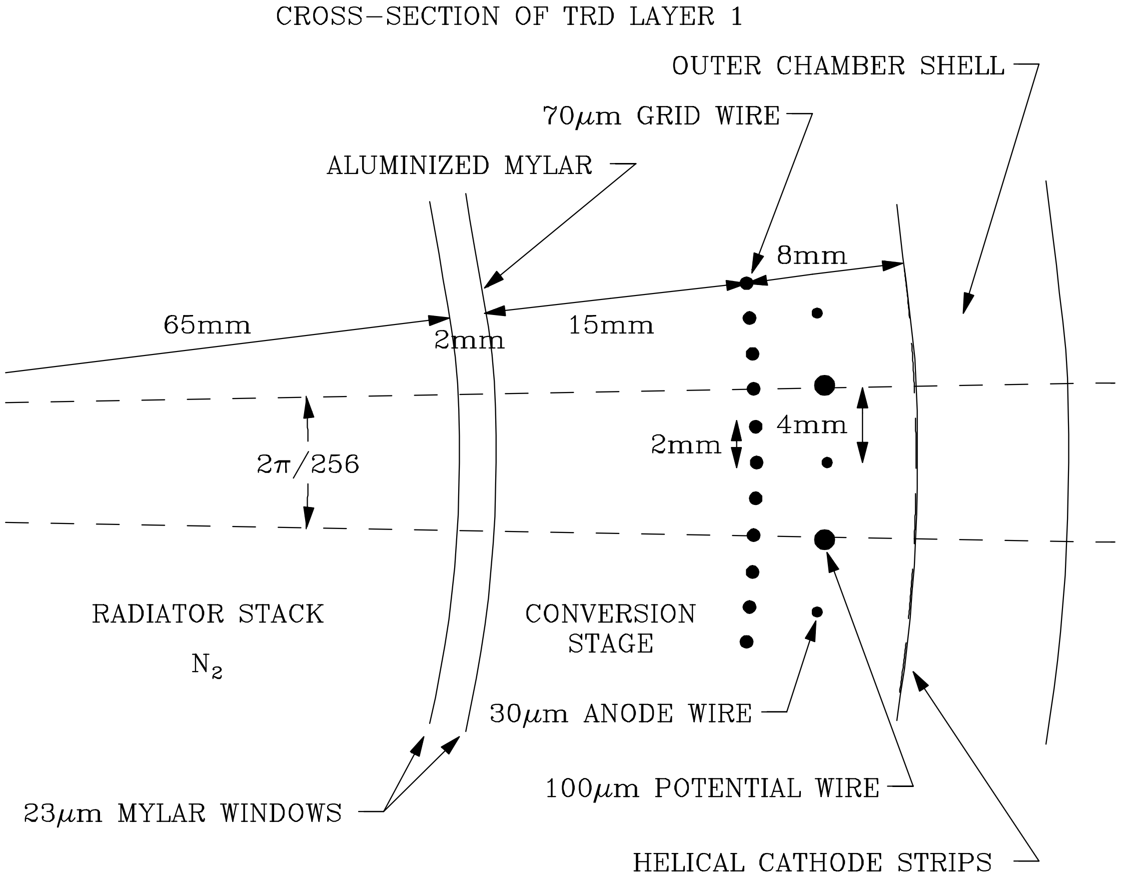


Figure 2.5: A cross section of the TRD.

$CH_4$, 2% $CH_2$), and charge drifts radially outward and is amplified before reaching the sense wires.

### The Central Drift Chamber

The central drift chamber (CDC) is a cylindrical annulus, 184 cm in length, with inner and outer radii of 49.5 and 74.5 cm, respectively. The CDC has 4 concentric layers of 32 azimuthal cells each. One quadrant of the CDC is shown in Figure 2.6. Each cell contains 7 sense wires (readout at one end) and two delay lines (readout at both ends). Sense wires are staggered in $\phi$ by $\pm 200$ $\mu$m to remove left-right

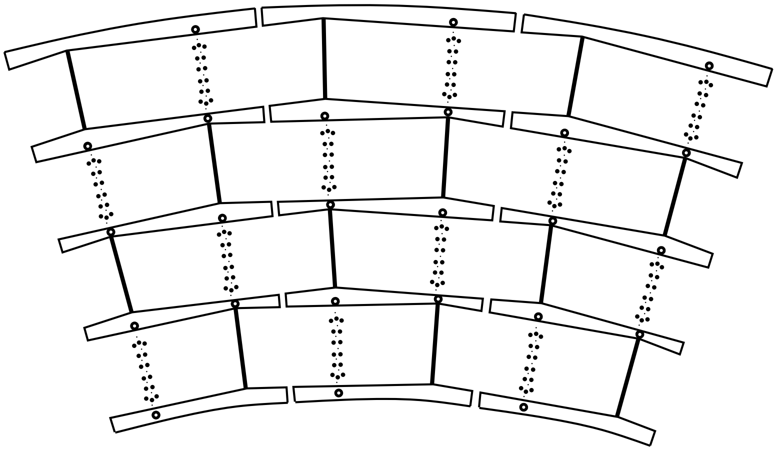


Figure 2.6: R-$\phi$ view of one quadrant of the CDC chamber.

ambiguities. The active medium is a gas mixture of 92.5% Ar, 4% $CH_4$, 3% $CO_2$, and 0.5% $H_2O$. Cells are staggered between layers to aid in pattern recognition. The $\phi$-coordinates of hits on a track are determined from the sense wires, and the $z$-coordinates are determined by the delay lines.[15]

### The Forward Drift Chamber

The forward drift chambers (FDC), one at each end of the CDC, consist of three layers: a $\Phi$ module sandwiched between two $\Theta$ modules (Figure 2.7). The $\Theta$ modules are rotated 45 degrees relative to each other maximize position resolution. The cells of each module contain 16 sense wires and use the same active medium as the CDC.[16]

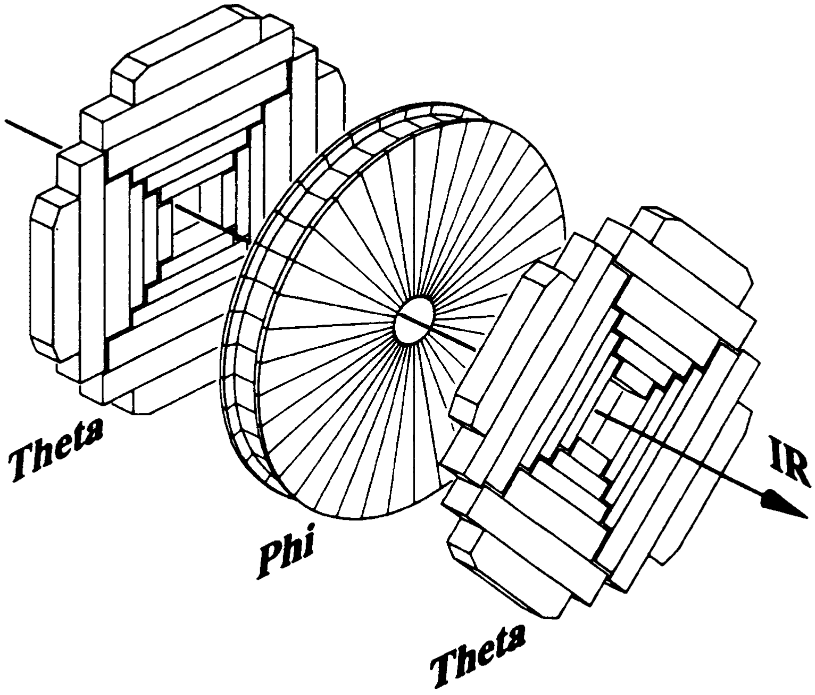


Figure 2.7: Exploded view of one of the forward drift chambers.

### 2.2.3 Calorimetry

The DØ detector employs *sampling calorimetry* [17] to determine the energy of particles produced in an inelastic collision. In a sampling calorimeter, the unit cell consists of an absorbing layer (the *passive medium*, usually a dense metal), an ionizing layer (the active medium), and a readout instrument. Particles interact with the passive material to produce secondaries (showering). Charged secondaries ionize the active medium for readout, as with drift chambers. Radial layers of unit cells sample the shower.[3] For the DØ calorimeter, liquid argon is the active medium, and the readout instrument is a copper pad between two insulating G10 boards (Figure 2.8).

[3] A calorimeter could consist entirely of active material, at the cost of greatly increasing the overall size.

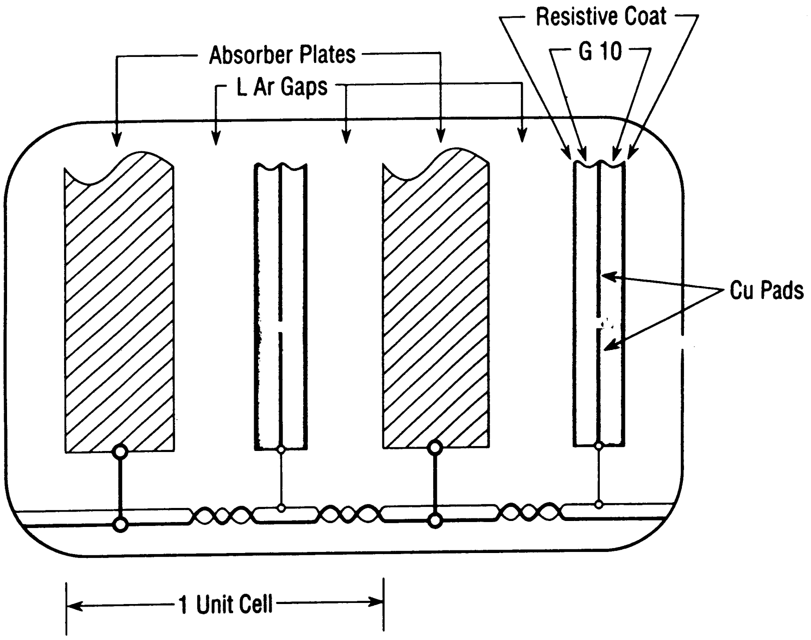


Figure 2.8: Schematic view of a typical uranium liquid argon readout cell.

The DØ calorimeter [18, 19] has three sections: the central calorimeter (CC) and two end calorimeters (EC north and EC south). Each section has three classes of readout cells, ordered by their distance from the interaction point: electromagnetic (EM), fine hadronic (FH), and course hadronic (CH). The passive medium is depleted uranium for EM cells, uranium-niobium (2%) alloy for FH cells, copper for CC CH cells, and stainless steel for EC CH cells. The EM layers total 21 radiation lengths, and all layers total a minimum of 6 interaction lengths.

Segmentation of the readout cells forms a pseudo-projective tower geometry, seen in Figure 2.9. Towers subtend an angle of 0.1 by 0.1 in $\eta \times \phi$ space out to $|\eta_d| = 3.2$ (the last few towers are larger). There are four cells per tower in the third EM layer

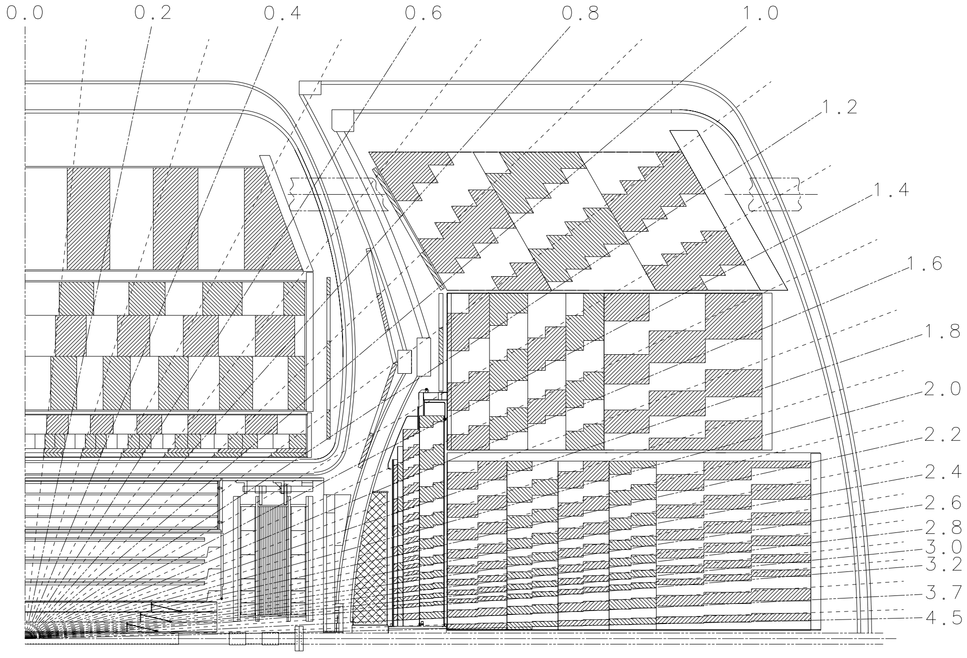


Figure 2.9: One quadrant of the DØ calorimeter and tracking system.

(EM3 is where the maximum energy is deposited by electromagnetic showers) to maximize position resolution.

In addition to the absorber plates, showering also occurs in the CC EM and FH support structure, and in the walls of the cryostats. This "uninstrumented" region is roughly $0.8 \leq |\eta_d| \leq 1.4$. To measure the showering in the CC support structure, readout boards, of the same construction as for the unit cell, are located near the inside wall of the CC cryostat. To measure the showering in the EC cryostat wall, readout boards are mounted on the inside face of the EC hadronic layers. These additional readouts are called massless gaps (MG). Finally, showering in the CC cryostat wall is detected with scintillating tile arrays, mounted on the outside wall of the EC cryostat (the intercryostat detector, or ICD). The readout pad sizes for both the ICD and MGs correspond to the tower geometry (again, Figure 2.9), and they provide a measure of particle energies that might otherwise be lost [20].

### 2.2.4 The Muon Spectrometer

The DØ detector uses five iron toroidal magnets surrounded by proportional drift tubes (PDTs) to detect muons. The central toroid (CF) spans the region $|\eta_d| \leq 1.0$, two end toroids (EF) span the region $1.0 < |\eta_d| \leq 2.5$, and two small angle toroids span the region $2.5 < |\eta_d| \leq 3.6$ (Figure 2.2). A magnetic field of about 2 T runs azimuthally around the square annuluses, producing a bend in the $r - z$ plane used to measure momentum.

The wide angle muon system (WAMUS) PDTs are arrayed in three layers around the CF and EF toroids. The A layer is just before the toroid and consists of four planes of staggered cells. The B and C layers are placed after the toroid (separated by 1-3 m) and consist of three planes of staggered cells. Cathode pads (copper clad Glasteel) are located at the top and bottom of each cell, and an anode wire is located in the middle. The active medium is 90% Ar, 5% $CF_4$, and 5% $CO_2$. The coordinate along the wire direction (also the bend direction) is measured by a combination of cathode pad signals and timing information from the anode wires.

The small angle muon system (SAMUS) PDTs are also arrayed in three layers: the A station precedes the toroid and the B and C stations are after. Each station consists of three doublets of chambers, oriented in the $x$, $y$, and $u$ direction ($u$ is at 45 degrees with respect to $x$, $y$). The active medium is a gas mixture of 90% $CF_4$, 10% $CH_4$. SAMUS muons are not used in this analysis.

## 2.3 Online Data Acquisition

During Run 1, the Tevatron delivered around 300,000 collisions per second at each interaction point. The DØ detector can save 2-3 of these *events* per second. The decision of which events to save is taken by the DØ trigger system.

The DØ trigger consists of three levels of event filtering: Level Ø, Level 1, Level 2. Level Ø determines if an inelastic collision occurred during the beam crossing. The programmable Level 1 hardware trigger filters events based on coarse information from Level Ø, the calorimeter, and the muon system. The Level 2 software trigger, consisting of 50 parallel workstations analyzing the entire high-precision data readout, makes the final decision.

### 2.3.1 Level Ø

Scintillating hodoscope arrays, located between the CC and each EC, identify (>99% efficient) inelastic collisions when both arrays detect charged particles within a time interval near the beam crossing time. In addition, Level Ø quickly calculates a vertex position (*fast-z*), based on the difference in arrival time at the hodoscopes, that is available for use at Level 1. A more accurate slower calculation of the vertex (*slow-z*), using test stand and in situ determined corrections, is available for use in Level 2. Finally, the Level Ø detectors also serve as luminosity monitors.

### 2.3.2 Level 1

Level 1 consists of a programmable two dimensional AND-OR array of 256 *terms*, containing calorimeter and muon system information (the Level 1 information). These terms can be combined into 32 non-exclusive *specific triggers.* Many Level 1 triggers accomplish their decision between beam crossings and thus incur no deadtime. Some triggers analyze more complex combinations of Level 1 information and require many beam crossings. These slower triggers are called Level 1.5. The output rate of Level 1 is about 100 Hz.

The Level 1 calorimeter information consists of ganged readout towers forming *trigger towers.* A trigger tower is 0.2 by 0.2 in $\eta \times \phi$ and contains EM or total $E_T$.[4] There are 1280 trigger towers of each type. Level 1 terms contain an $E_T$ threshold and sometimes a maximum $|\eta_d|$ for a tower. EM(1,12,2.6), for example, requires a single EM trigger tower, $|\eta_d| < 2.6$, with $E_T > 12$ GeV. Level 1.5 terms can apply thresholds to clustered trigger towers. EX(1,12,0.85), for example, requires a 1x2 tower cluster above 12 GeV $E_T$, and EM/total $E_T$ above 0.85.

The Level 1 muon information is a single latch bit for each of the approximately 11,400 drift cells in WAMUS. This information gives the bend coordinate of hits and is used to calculate a centroid with a resolution of 5 cm. Hits in the B and C

[4] Here, $E_T$ is calculated with respect to fast-z from Level Ø.

layers are used as seeds for a Level 1 pattern lookup comparison with data in the A layer. For tracks passing Level 1, a slower more accurate determination of the bend is calculated at Level 1.5 using the centroids of all three layers.

### 2.3.3 Level 2

An event that passes Level 1 triggers readout of the entire detector for analysis at Level 2. Level 2 consists of software units called *tools* that are combined into *filter scripts* (both written in Fortran). Tools perform specific jobs: identifying electrons, jets, or muons; calculating $\not\!\!E_T$ or scalar $E_T$. Scripts control which tools are called, the calling sequence, and what parameters are passed to each tool (e.g. an $E_T$ threshold). Each script is associated with a specific trigger, and more than one script can be associated with a given trigger. There are a total of 128 possible scripts. At the end of Run 1, the typical configuration used about 85 scripts (hereafter, these will be referred to as Level 2 filters).

The time budget for Level 2 is 200 ms per event, to limit deadtime to less than 2% [21]. Substantial additional reconstruction and identification is done offline (see Chapter 3).

# Chapter 3

# Particle Identification

## 3.1 Introduction

The DØ detector is well suited for particle identification and discrimination. Electrons and photons are characterized by narrow depositions of energy in the first few layers of the calorimeter (*electromagnetic* or *EM layers*), and can be differentiated from one another by the presence or absence of a track in the central tracking chambers. Jets, essentially broad sprays of hadrons from quark or gluon fragmentation, deposit energy throughout the calorimeter layers, their showering dominated by nuclear interactions with the passive calorimeter material rather than electromagnetic interactions with nuclei and shell electrons (also in the passive material). All three types of showers may be characterized by their pattern of energy deposition and their fraction of energy in the EM layers of the calorimeter. Muons, more massive

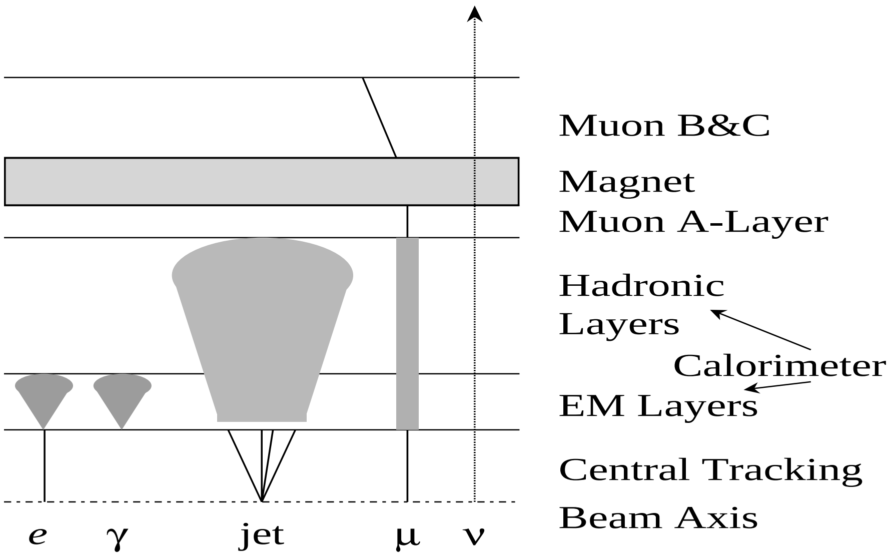


Figure 3.1: Cartoon representation of how different particles deposit energy in the DØ detector.

than electrons, tend not to bremsstrahlung in the uranium; they leave a MIP[1] trace of ionization energy in the central tracker and calorimeter, and are detected by the muon tracking chambers, where a momentum measurement is performed. The presence of a neutrino (or other proposed weakly interacting neutral particles, such as neutralinos from supersymmetry) is inferred by unbalanced or "missing" transverse energy, denoted $\not\!\!E_T$. Finally, heavy flavor decays are often accompanied by a soft lepton ($p_T \geq 4$ GeV/c) within the cone that defines the jet. These leptons tag jets from $b$ quarks, although the current analysis doesn't make use of tagged jets. Figure 3.1 illustrates the particle interactions in the detector and Figure 3.2 is an example of an actual event containing an electron, a muon, and a jet.

[1] Minimum Ionizing Particle.

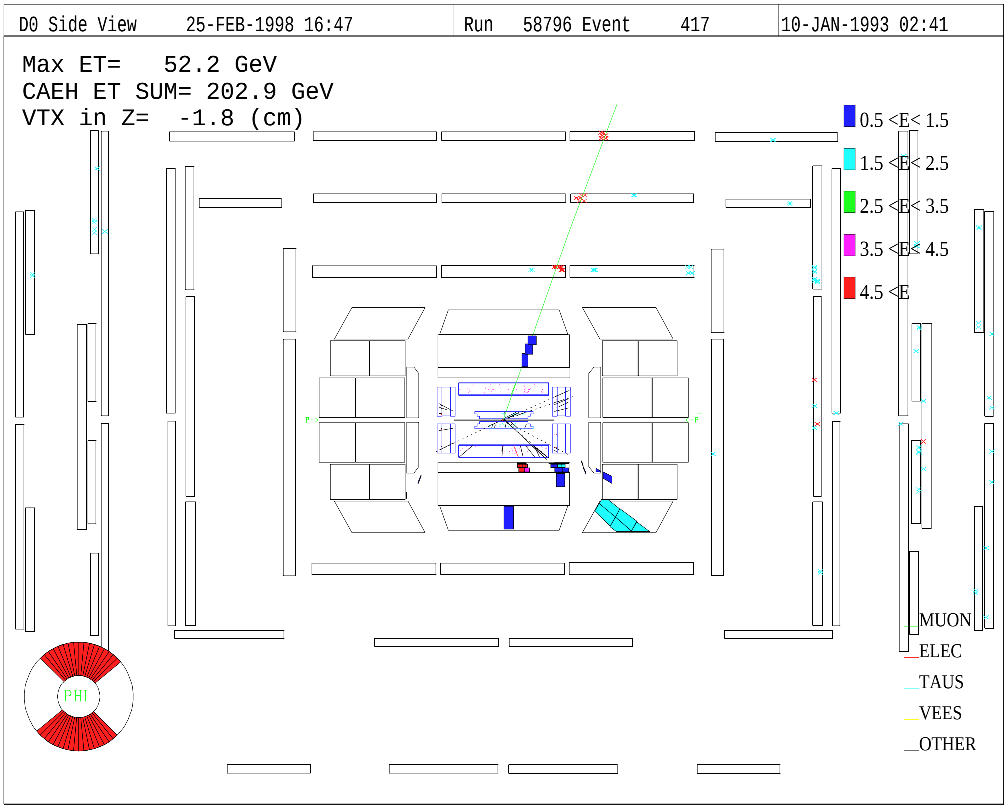


Figure 3.2: Event display containing an electron, a muon and a jet. The electron, located just right of bottom-center in the calorimeter illustrates the narrow energy deposition of electromagnetic showers. There is no dotted line for the electron track extrapolated to the primary vertex because the track was reconstructed as a CDC only track. The jet, a spray of energy to the right of the electron, exhibits a wider shower shape and penetrates well past the EM layers of the calorimeter. Finally, there is a three layer track for the muon matching a calorimeter MIP trace.

The following sections discuss jets, electrons, muons, and $\not\!\!E_T$. Each section includes a description object finding in the DØ reconstruction program (RECO) and offline identification. Selection efficiency and rejection of background ("fakes") is also described.

## 3.2 Jets

### 3.2.1 Candidate Jets

Many hard collisions involve the production of colored particles, quarks and gluons, as tree level partons. Color confinement requires that these partons fragment into a spray of colorless hadrons through a process called *hadronization*, and the physical manifestation of this spray of hadrons is called a jet. The algorithm for detecting jets depends on the jet definition. This analysis uses $\mathcal{R} = 0.5$ cone jets. Cone jets are groups of readout towers within a "cone" of fixed radius $\mathcal{R} = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ about a jet axis. Construction of cone jets proceeds in a three step process: preclustering of readout towers about seed towers, using the preclusters as seeds to form a cone jet, and splitting and merging of jets.

Preclustering begins with seed towers, defined as a readout tower with $E_T > 1$ GeV. The highest $E_T$ tower (the *hottest* tower) is preclustered with all towers that are within $\pm 3$ units in $\phi$ and $\eta$ adjacent to the seed. When a tower is included in a precluster, a flag word is set. With the first precluster complete, the next hottest unclustered tower is considered as a possible seed to form the next precluster, and

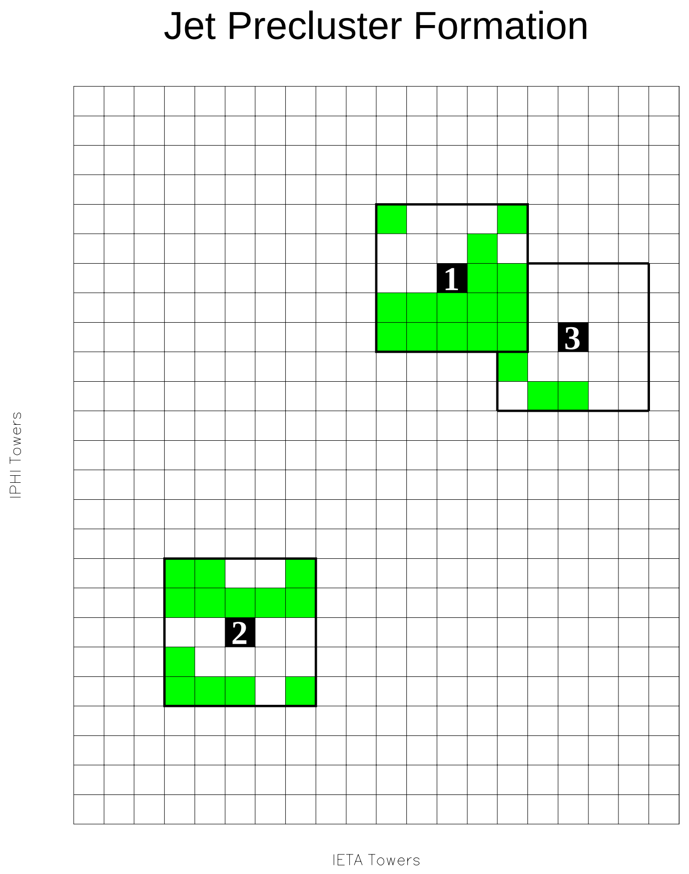


Figure 3.3: Example of precluster formation where the geometry has been projected onto a two dimensional grid in $(\eta, \phi)$ space. Towers above seed threshold are shaded grey or black. The hottest tower is numbered **1** and towers within ±3 units in $\phi$ and ±3 units in $\eta$ are added to the precluster. **2** is then the hottest tower not currently flagged as used and the second precluster is formed. Finally, **3** collects the rest of the seeds for the final precluster.

so on, until all towers have been examined. Figure 3.3 shows the process in a simple two-dimensional projection of readout towers where two jets lead to the formation of three preclusters. Towers used to form the preclusters are shaded black, and numbered in their order of construction. Towers above seed threshold are shaded grey. Towers without shading may possess non-zero $E_T$, but they are irrelevant to the description of precluster and cone jet construction.

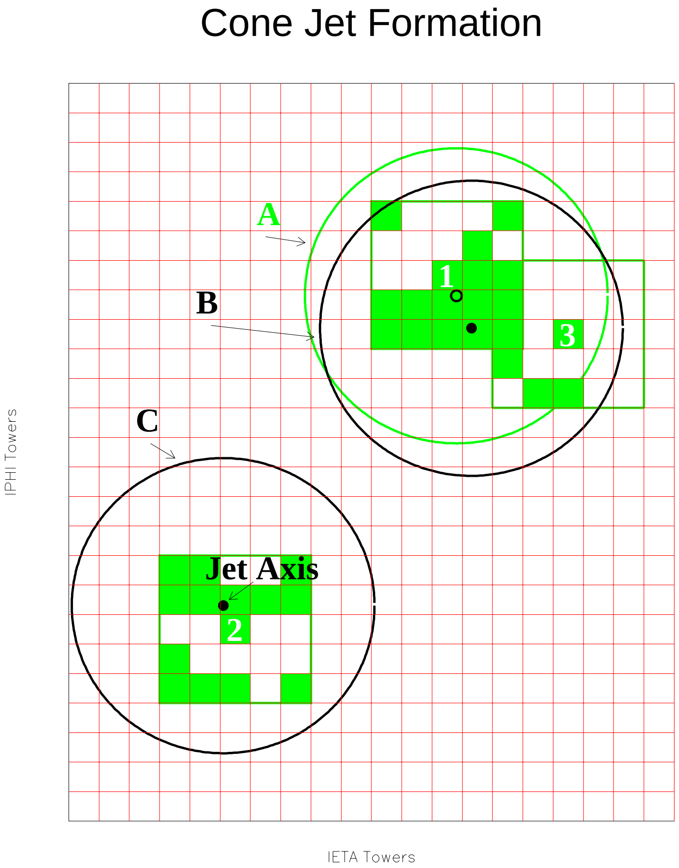


Figure 3.4: Cone jet formation from preclusters.

Cone jet formation proceeds by considering preclusters of decreasing ordered $E_T$. The *jet axis*, defined as the $E_T$ weighted $(\eta, \phi)$ centroid, is determined from the precluster and all towers within radius $\mathcal{R}$ are assigned to the cone jet (Figure 3.4, cone **A**). The jet axis is recalculated including all towers in the cone, and the cone re-defined around the new axis (Figure 3.4, cone **B**). The process iterates until the jet axis stabilizes (axis movement of less than 0.01 in $\eta \times \phi$ space, or a maximum number of iterations have occurred).[2] In Figure 3.4, the cone jet formed from precluster **1** stabilizes after one iteration, and cone **B** is the final candidate cone. If the resulting

[2] The maximum number of iterations is introduced to prevent the rare case of a bi-stable solution causing an infinite loop.

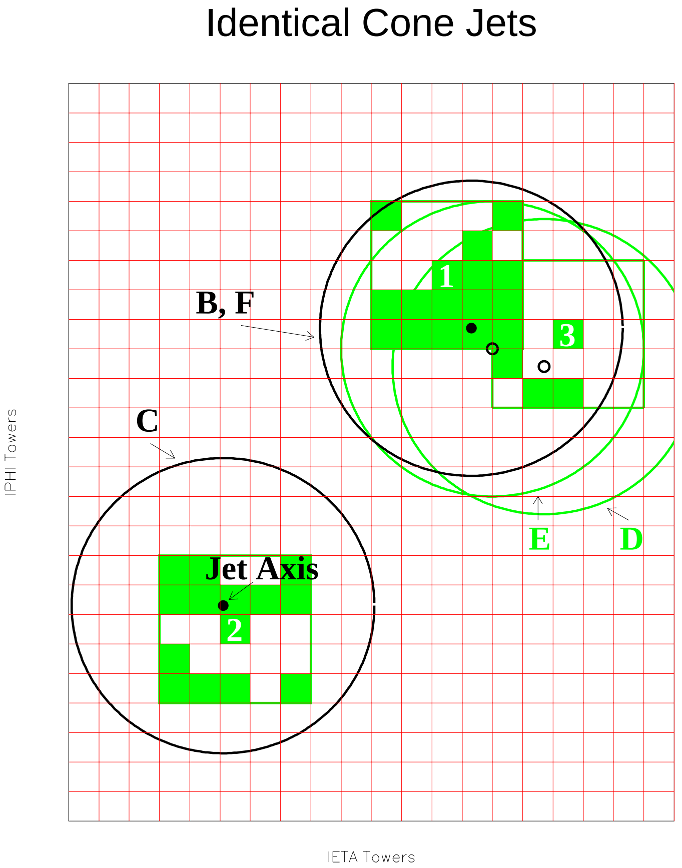


Figure 3.5: Example of discarding duplicate jets during splitting and merging.

cone jet has $E_T > 8.0$ GeV, it is stored for splitting/merging.

The next hottest precluster is selected, and cone jet formation continues with this precluster. A tower may belong to more than one cone jet before splitting/merging. As new jets are formed, a check is performed to determine the number of towers in common with previous jets. When towers are shared and the jet axes differ by less than 0.01 in $\eta \times \phi$ space from a previous jet, the jet is discarded as duplicate. This duplicate jet formation is shown in Figure 3.5, where the cone jet formed from precluster **3** iterates from cone **D**, through cone **E**, and finally stabilizes at cone **F**, which is exactly cone **B** from precluster **1**.

If the two jets sharing towers differ by more than 0.01 in their axes, the split/merge ratio is calculated, $f_{\mathrm{SM}}$, defined by

$$f_{\mathrm{SM}} = \frac{E_T^{\mathrm{shared}}}{E_T^{\mathrm{min}}} \tag{3.1}$$

where $E_T^{\mathrm{min}}$ is the lesser of the two jet $E_T$s. For $f_{\mathrm{SM}} \leq 0.5$, the two jets are split, and contested cells (not towers) are assigned to the jet with the closest axis. For $f_{\mathrm{SM}} > 0.5$, the two jets are merged into a single jet with all towers assigned to it, and the jet axis is recalculated without iteration. Jets that have been split or merged are the only type of cone jet where the allowed jet shape is not azimuthally symmetric about the jet axis. When splitting and merging for a jet is complete, cone clustering proceeds to the next lowest $E_T$ precluster until all preclusters have been examined, and all found jets have been compared with previously found jets.

Once clustering is complete, the kinematic variables for each jet are calculated and stored in the JETS bank.[3] The energy components of the jet are defined by

$$E_i^{\mathrm{jet}} = \sum_{\mathrm{towers}\ k} E_i^k \tag{3.2}$$

where $i = x$, $y$, $z$, or total. The three-vector components of each tower are defined with respect to the primary vertex found by RECO. The jet $E_T$ is defined as the scalar sum of the tower $E_T$'s

$$E_T^{\mathrm{jet}} = \sum_{\mathrm{towers}\ k} E_T^k. \tag{3.3}$$

[3] DØ uses a storage utility called ZEBRA for all data. The basic unit of storage is the bank [22].

The jet angles are defined from the three-vector components as

$$\phi^{\rm jet} = \arctan\left(\frac{E_y}{E_x}\right) \tag{3.4}$$

$$\theta^{\rm jet} = \arccos\left(\frac{E_z}{\sqrt{E_x^2 + E_y^2 + E_z^2}}\right) \tag{3.5}$$

$$\eta^{\rm jet} = -\ln\left[\tan\left(\frac{\theta^{\rm jet}}{2}\right)\right]. \tag{3.6}$$

### 3.2.2 Jet Identification and Efficiency

The DØ calorimeter has a minimum of 7.2 nuclear absorbtion lenghts at $\eta = 0$ and thus one expects that every high-$E_T$ parton produced in a hard collision will deposit nearly all of its energy in a localized region that can be reconstructed as a jet. Successful reconstruction requires a single tower above the seed threshold mentioned in the previous section. For 0.5 cone jets with $E_T > 20$ GeV, the seed tower efficiency is consistent with 100%.[23]

Reconstructed jet candidates result primarily from a hard interaction; however, there are numerous fake processes that must be removed in order to correctly determine the actual number of jets and $\not\!\!E_T$ from the hard scatter. Sources of these fake jets include: cosmic ray bremsstrahlung, protons escaping the Main Ring, beam-gas interactions, and certain hardware malfunctions (baseline subtractor failure, or high-voltage discharges resulting in a hot channel).

Fake jets from these sources can be removed from an event by requiring mild jet quality cuts. Such requirements will naturally remove some good jets. This efficiency

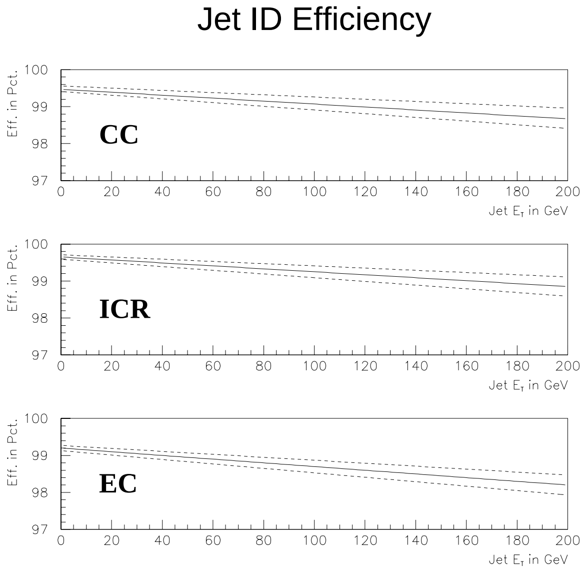


Figure 3.6: Efficiency of jet identification cuts as a function of $E_T$ in the three regions of the calorimeter.

has been evaluated elsewhere in detail; the results are summarized here [24].

Three variables are used to determine jet quality: EMF, CHF, and HCF. Each will be discussed in turn. The discussion assumes the definition of three regions of the calorimeter, depending on detector eta:

$$\begin{array}{ll} 0.0 \le |\eta_d| < 1.0 & \text{Central Region (CC)} \\ 1.0 \le |\eta_d| < 1.5 & \text{Intercryostat Region (ICR)} \\ 1.5 \le |\eta_d| < 3.0 & \text{Forward Region (EC)} \end{array}$$

EMF The electromagnetic fraction is defined as the ratio of $E_T$ deposited in the EM layers of the calorimeter to the total $E_T$ of the jet. The quality cut is imposed $0.05 < \text{EMF} < 0.95$ in the CC and EC. The lower limit is removed in the ICR since no EM layers are present in $1.2 < |\eta_d| < 1.5$. This cut removes

| Region | $p_1 \pm \delta p_1$ (%) | $p_2 \pm \delta p_2$ (%/GeV) |
|---|---|---|
| CC | $99.47 \pm 0.08$ | $-0.004 \pm 0.001$ |
| ICR | $99.65 \pm 0.06$ | $-0.004 \pm 0.001$ |
| EC | $99.20 \pm 0.07$ | $-0.005 \pm 0.001$ |

Table 3.1: Fit parameters for jet identification efficiencies. The efficiency is fit to the function $\epsilon = p_1 + p_2 \cdot E_T$.

noisy cells that occur in either the EM or hadronic layers. It also is removes cosmic brems in the hardronic calorimeter, which deposit energy in a few cells since each hadronic layer comprises several radiation lengths.

CHF The coarse hadronic fraction is defined as the fraction of $E_T$ deposited in the coarse hadronic layers of the calorimeter. The cut is $CHF < 0.4$ in the CC and EC, and $CHF < 0.6$ in the ICR. This cut removes noisy cells in the coarse hadronic layers, and jets produced by Main Ring particles.

HCF The hot cell fraction is defined as the ratio of the $E_T$ of the second hottest cell to the hottest cell in a jet. For all regions, the cut is $HCF > 0.05$, meaning that the hottest cell in a jet contains less than twenty times the transverse energy of the next hottest cell. This cut is designed to remove noisy cells, as normal jets spread their energy over many layers and towers.

The efficiency is fit to a straight line in each region of the calorimeter (Table 3.1 and Figure 3.6).

Finally, it is almost certain that electron showers will produce a JETS bank in addition to a PELC bank. These duplicate entries must be removed to avoid double counting an electron as a jet. This is accomplished by requiring that the distance in $\eta \times \phi$ space between a jet and an identified electron is greater than 0.2.

## 3.3 Electrons

### 3.3.1 Candidate Electrons

Electron candidates are constructed in three step process: clustering of EM towers using a nearest neighbor algorithm, applying loose electromagnetic shower criteria, and creation of electron (PELC) or photon (PPHO) banks to store the candidates for further analysis.

Calorimeter cluster, or CACL banks, are constructed from the EM tower information. Only towers with greater than 50 MeV of energy are considered. Clustering begins with the first tower above threshold, and it is connected with the highest energy neighboring tower, where a neighboring tower is defined as physically adjacent or immediately catercorner in $\eta \times \phi$ space. If there are no adjacent towers above threshold, the tower is paired with itself. The paired towers naturally form a set of unique, non-overlapping clusters of energy. Each cluster is stored in a CACL bank.

Once clustering is complete, a first pass over all CACL banks drops those that fail the following criteria:

- The total energy in the cluster $E_{\text{total}} > 1.5$ GeV
- The total transverse energy in the cluster $E_T > 1.5$ GeV

- The electromagnetic fraction of the cluster $E_{\mathrm{EM}}/E_{\mathrm{total}} > 0.9$

A second pass over CACLs calculates two more electromagnetic shower variables: the cluster energy not contained in the $\eta_d$ phi-slice of the hottest tower, called the shower transverse profile $E_{\mathrm{trans}}$, and the cluster energy deposited in FH1. If $E_{\mathrm{trans}}/E_{\mathrm{total}} > 0.6$ or $E_{\mathrm{FH1}} > 999$ GeV, the CACL is dropped.[4] PELC and PPHO banks are made from this final sample of clusters. The PELC candidate requires a track in a wide road between the shower centroid in the EM3 layer and the primary vertex. If multiple vertices are found, a road is constructed to each secondary vertex and tracks found are recorded; a PPHO with a track to a secondary vertex will not change identity. For the current analysis, only PELC candidates are considered as possible electrons.

### 3.3.2 Electron Identification Variables

The majority of electron candidates are not real electrons and substantial further identification is required. The main backgrounds for real electrons are photon conversions in the TRD and *hadron overlaps*, where a photon or photon pair from a $\pi^0$ decay is overlapped with a random track. Random tracks may even be very low energy pions, due to the absence of a central magnetic field in the DØ detector.

For each CC (EC) electron candidate, five (four) quantities are calculated. The first is the fraction of electromagnetic energy and is simply $E_{\mathrm{EM}}/E_{\mathrm{total}}$. The second is the H-matrix chi-squared ($\chi^2_{\mathrm{hm}}$), derived from shower shape. The third is $dE/dx$,

[4] The transverse profile cut can be interpreted as requiring 40% of the energy deposited in one phi slice.

caculated from the pulse area of hits in the tracking chambers. The fourth is the track match significance ($\sigma_{\rm trk}$) and is determined by how well the electron track points to the cluster. The fifth value is available only for CC electrons and is the TRD-epsilon ($\epsilon_{\rm TRD}$), calculated from the energy deposited in the three layers of the transition radiation detector. Additionally, the degree of isolated of the electron from other energy in the calorimeter is recorded; however, the expected isolation depends on the physics source of the electron. For this reason, generic electron identification requires only a loose isolation cut, leaving tight isolation cuts to individual analyses.

A short discussion of each variable follows; detailed information on construction of each variable may be found in [25].

### Cluster Energy and Electromagnetic Fraction

The final energy of the cluster includes an energy scale correction [26]. The electromagnetic energy fraction is calculated from the uncorrected energies in the CACL banks, EMF $= E_{\rm EM}/E_{\rm total}$.

### H-Matrix

Electromagnetic showers are well collimated depositions of energy compared to the lumpy shape of jets. On average, the shower shape is the well known teardrop pattern [17]; fluctuations cause the energy deposition to vary from the average in a correlated fashion. This correlation, for example, demands that a shower that fluctuates high in one layer will tend to exhibit low fluctuations in other layers.

Such patterns are well modeled by a full detector simulation such as GEANT [27]. To determine the “electron-ness” of a shower, covariance matrices are constructed using Monte Carlo electrons of various energies (10 to 150 GeV) and position (37 matrices are constructed, one for each $\eta_d$ in the positive $\eta$ half of the calorimeter; the negative $\eta$ side is handled with reflective symmetry.) For a sample of N electrons

$$M_{ij} = \frac{1}{N}\sum_{n=1}^{N}(x_i^n - \overline{x_i})(x_j^n - \overline{x_j})$$

where $x_i^n$ is the $i$th observable of the $n$th electron in the sample, and $\overline{x_i}$ is the mean of the $i$th observable. There are 41 quantities used in the covariance matrix: the fraction of energy in layers 1, 2, and 4 of the EM calorimeter, the fraction of energy in each of the cells in a 6x6 square of EM3 cells centered on the hottest tower,[5] the logarithm of the cluster energy, and the vertex position.

For each electron shower, the covariance parameter

$$\chi^2 = \sum_{i,j=1}^{41}(x_i' - \overline{x_i})H_{ij}(x_j' - \overline{x_j})$$

where $x_i'$ is the $i$th observable and $H = M^{-1}$, is a measure of how consistent the shower shape is with an electromagnetic shower. This covariance parameter is called the *full H-matrix chi-squared.* The observables, in general, are not normally distributed and thus the H-matrix chi-squared does not follow a $\chi^2$ distribution.

Because $H$ is symmetric, it can be diagonalized with a unitary transformation.

[5] Since the EM3 layer has 4 cells per tower, this can be thought of as an area of 3x3 towers centered on the hottest tower.

The $\chi^2$ can now be written

$$\chi^2 = yH'y^T$$

where $H' = U^THU$ and $y$ is the vector of linearly uncorrelated variables. If the eigenvalues are unusually large, small differences between shower shapes in data and MC can lead to wide variations in the full H-matrix chi-squared; to account for this, any one eigenvalue is limited to be 100,000 (chosen to optimize efficiency and rejection power). This final covariance variable is formally called the *truncated H-matrix chi-squared*, or simply chi-squared as it is the only covariance variable in common use.

**Determination of dE/dx**

The amount of charge collected on a sense wire is directly proportional to the pulse area and the constant of proportionality is called the *gain*. The gain of a sense wire is a function of the high voltage, gas mixture, and amount of radiation damage to the chambers; these factors all varied slowly with time during data taking. The simple explanation of determining dE/dx is that during reconstruction, the mean pulse area for all hits is determined and stored at regular intervals. Since the vast majority of all tracks are from MIPs, the ratio of the area of a pulse to that of the mean is a measure of how much charge was collected relative to the charge left by a MIP. For electrons, this value should be near one, while it will be lower for soft random tracks and near two for photon conversions where the two tracks are not resolved.

### Track Match Significance

The track from the central tracking chambers is a 3-dimensional vector. The EM3 layer lies at radius $R_{\mathrm{CC}}$ (for the CC), or at $z = z_{\mathrm{EC}}$ (for the EC). The track vector is propagated to the EM3 layer and since EM showers tend to be azimuthally symmetric, the vector should coincide with the measured shower center for real electrons. Due to the finite resolution of both measurements, a significance of the match is calculated for the CC as

$$\sigma_{\mathrm{trk}} = \sqrt{\left(\frac{\Delta z}{\delta z}\right)^2 + \left(\frac{\Delta \phi}{\delta \phi}\right)^2}$$

where $\delta z$ and $\delta\phi$ are the measured resolutions; a similar expression is used for the EC using $(\Delta R, \Delta\phi)$.

### Determining $\epsilon_{\mathrm{TRD}}$

The value $\epsilon_{\mathrm{TRD}}$ is a derived quantity between zero and one constructed such that its distribution is flat for real electrons. It is peaked at the ends for background. The construction uses a sample of $W \rightarrow e\nu$ events with electrons identified by other variables. The variable used is the truncated mean energy

$$E_{\mathrm{trunc}} = E_1 + E_2 + E_3 - \max(E_1, E_2, E_3)$$

where $E_i$ is the energy deposited in the $i$th layer of the TRD. Removing the maximum energy reduces the effect of Landau fluctuations. $\epsilon_{\mathrm{TRD}}$ is simply the cumulative

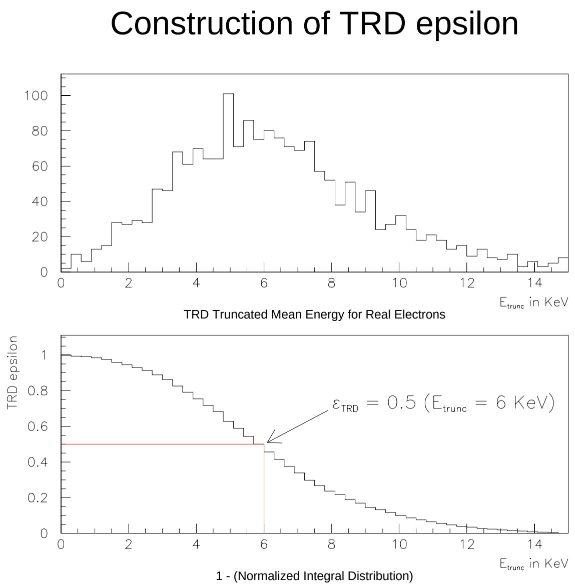


Figure 3.7: Construction of $\epsilon_{\mathrm{TRD}}$. A sample of real electrons is identified with other ID variables and the $E_{\mathrm{trunc}}$ distribution is plotted (top plot). That distribution is then integrated, normalized, and subtracted from 1 (bottom plot). For a given $E_{\mathrm{trunc}}$, $\epsilon_{\mathrm{TRD}}$ is then the value read off of the $y$-axis of this curve.

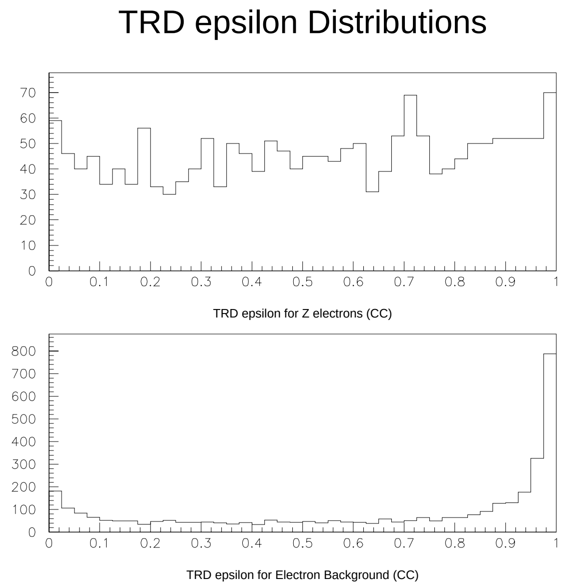


Figure 3.8: Distributions of $\epsilon_{\mathrm{TRD}}$ for electrons and background.

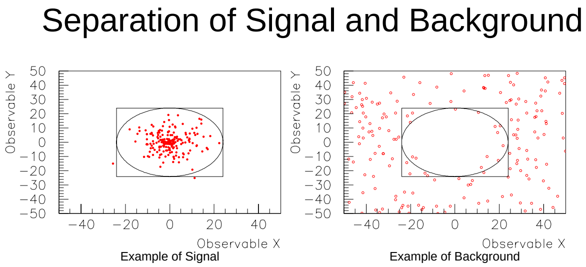


Figure 3.9: Example of independent (square) and correlated cuts on two observables. The signal distribution shows that the two cuts have the same efficiency, while the background distribution shows that the combined cut has about 30% better rejection.

probability of the measured truncated energy,

$$\epsilon_{\mathrm{TRD}}(E_{\mathrm{trunc}}) = \frac{\int_{E_{\mathrm{trunc}}}^{\infty} \frac{\delta N}{\delta E'} dE'}{\int_0^{\infty} \frac{\delta N}{\delta E'} dE'}$$

where $\frac{\delta N}{\delta E'}$ is the distribution for real electrons. Figure 3.7 demonstrates the construction of $\epsilon_{\mathrm{TRD}}(E_{\mathrm{trunc}})$. Figure 3.8 shows the $\epsilon_{\mathrm{TRD}}$ distribution for real electrons and background.

### Construction of 5-variable Likelihood

The simplest use of the available electron ID variables consists of independently cutting on each variable. This does not yield the optimum rejection for a given signal efficiency. Consider, for example, two observables $X$ and $Y$. Figure 3.9 shows a possible signal and background distribution. Selecting $|X|, |Y| < 25$ or $\sqrt{X^2 + Y^2} < 25$ give the same efficiency. However, the combined cut rejects 30% more background. Clearly, the combined cut is more optimized. When more variables are included in a combined cut, a graphical scan of signal and background becomes unwieldly. If the probability distributions $p(x)$ are known that a vector of observables, $x$, was produced by signal (electrons, $e$) or background (hadron overlaps, $h$, or conversions, $ee$), there exists a uniformly most powerful test for the two simple hypotheses, the Neyman-Pearson test:

$$\mathcal{R} = \frac{p(x|b)}{p(x|e)}$$

A cut on $\mathcal{R}$ defines the optimal volume of hyperspace $x$ for a given signal efficiency. Here $x$ is the vector of five (four) ID variables for the CC (EC) The background is modeled as dominated by $ee$ and $h$

$$p(x|b) = f_h p(x|h) + f_{ee} p(x|ee)$$

where the fractional contents $f_h + f_{ee} = 1$. This leads to

$$elike(f_h) = \frac{f_h p(x|h) + (1 - f_h) p(x|ee)}{p(x|e)}$$

where $f_h$ has been shown to be around 0.5.[28] The probability distributions for the individual variables are assumed uncorrelated and $p(x)$ is thus factorized

$$p(x|H) = p_1(\mathrm{EMF}|H) \times p_2(\chi^2|H) \times p_3(dE/dx) \times p_4(\sigma_{\mathrm{trk}}|H) \times p_5(\epsilon_{\mathrm{TRD}}|H)$$

The $p_i(x|H)$s are determined from pure samples of $H = e$, $h$, $ee$.

**Electron Isolation**

Electron isolation is an event topological cut with distribution depending on physics event type and therefore doesn't belong in a generic multivariable electron likelihood; instead, it is cut on separately for each analysis. The isolation is calculated from the electromagnetic energy in a *core cone* of $\mathcal{R} = 0.2$, and the total energy in an *isolation cone* of $\mathcal{R} = 0.4$

$$\text{isolation} = \frac{E_{\text{tot}}^{0.4} - E_{\text{EM}}^{0.2}}{E_{\text{EM}}^{0.2}}$$

Electron showers subtend a cone smaller than 0.2 giving isolation near zero for real isolated electrons. The current analysis requires isolation $< 0.3$.

### 3.3.3 Electron Efficiency

The efficiency for high $E_T$ electron ID is determined from a sample of $Z \to ee$ events. Micro-DSTs from the Run 1b $Z \to ee$ stream were selected requiring two em objects (one a PELC) with $E_T > 15$ GeV.[6] The online trigger requirements for this stream have been shown to be 100% efficient for $Z$ boson decays.[26] This is referred to as the *streamed sample.*

To determine the efficiency for a given cut relative to another cut, two samples are created, a *parent* and a *daughter*, before and after the cut, respectively. A cut is only applied to one of the em objects, yielding single electron efficiency. The number of Z events is determined in both the parent and daughter sample and their ratio

[6] One may worry that a systematic error is introduced by requiring one PELC in this sample. This error scales like $(1 - \epsilon_{\text{trk}})^2$. With $\epsilon_{\text{trk}} \sim 85\%$, the maximum systematic is 2%.

represents the relative efficiency of the cut under study.

$$\mathrm{eff}_{\mathrm{par}\to\mathrm{dau}} = \frac{n^Z_{\mathrm{par}}}{n^Z_{\mathrm{dau}}}$$

For example, to calculate the tracking efficiency for leading electrons, the parent sample is the streamed sample and the daughter sample is events with the leading electron candidate a PELC.

To determine the number of Z events in a sample, a side-band method is used on the invariant mass spectrum of the em pair. The process begins by counting the number of events in three ranges, (72,82), (82,102), and (102,112) GeV/c$^2$.[7] A histogram is filled with the invariant mass spectrum.

The estimation of $n^Z$ is done using the equivalent of a linear fit to the background. The number of background events in the Z-band is the average of the high-band and the low-band, normalized by the appropriate scaling factor: in this case, where the two side bands are 10 GeV/c$^2$ and the Z-band is 20 GeV/c$^2$, the factor is exactly one, so $n_{\mathrm{back}} = n_{\mathrm{high}} + n_{\mathrm{low}}$.

As a cross check, an exponential fit is performed on the histogram. This is accomplished by giving zero weight to events within the Z-bin and fitting using a simple exponential function between 72 and 112 GeV. The background in the

[7] This counting of the unbinned data removes the (small) systematic error associated with finite binning.

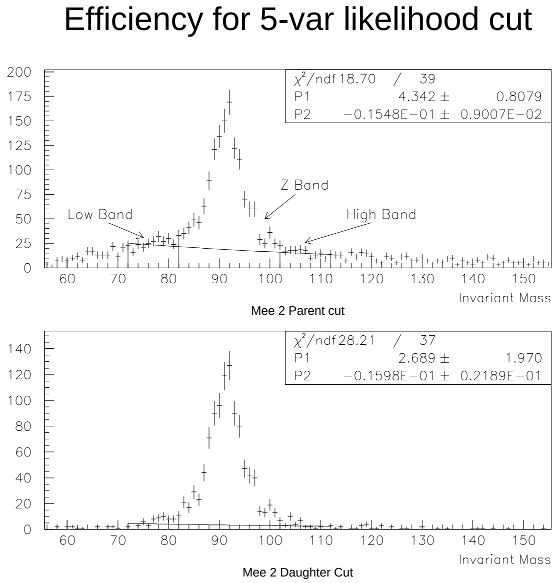


Figure 3.10: Example of determining the efficiency. The high and low side bands are indicated with vertical lines. The exponential fit is also shown. Note that the number of degrees of freedom quoted is 21 too high, as 21 bins are removed from the fit by exploding their errors prior to the fit.

distribution is then just

$$n_{\mathrm{back}} = \int_{82}^{102} e^{p_1 + p_2 x}\, dx = \frac{e^{p_1}}{p_2}\left(e^{102 p_2} - e^{82 p_2}\right)$$

where $p_i$ are the fit parameters.

Once the estimated number of background events is determined, it is subtracted from the total number of events in the Z-band to obtain the number of Z's in both the parent and daughter distributions. An example of the process is shown in Figure 3.10. The efficiency is taken from the exponential fit for high statistics cases, and the linear fit for low statistics cases. The statistical error on the efficiency is determined using binomial errors. This leads to the formula

$$\text{stat err} = \sqrt{\frac{\text{eff}(1-\text{eff})}{N_{\text{par}}}}$$

where the $N_{\text{par}}$ is the number of $Z$s in the parent sample.

The systematic error for a given parent sample is calculated by repeating the efficiency determination for the trailing (lower $E_T$) electron instead of the leading electron. The final efficiency number is taken as the average of the leading and trailing determinations, and a systematic error of one-half the difference is added in quadrature with the statistical error.

### 3.3.4 Calculating the Fake Electron Probability

Jet fluctuations can produce a shower shape and track imitating an electron. Two samples were obtained from the JET_30 filter stream requiring 3 $E_T > 15$ GeV jets ($|\eta_d| \leq 2.5$), but rejecting events consistent with a $W$ or $Z$ boson decay.[8] The first sample had no additional requirements and is called the QCD sample. The QCD sample is used to determine the jet fake probability, the probability that a generic jet imitates an electron. The second sample added the requirement of one PELC with $E_T > 15$ GeV and is called the 1PELC sample. The 1PELC sample is used to determine the fake PELC probability, the probability that a generic PELC imitates an electron.

[8] The removal of the $W$ and $Z$ bosons reduced the fake probability by about 2%.

The QCD fake probability is determined by constructing an $E_T$ spectrum of all jets, and of jets passing an electron ID criterion. The ratio of the two histograms is the QCD fake probability as a function of $E_T$. The ratio spectrum is fit to a constant used as the fake probability. The fit is performed in a range of $E_T$ from 15 GeV, the intended electron minimum $E_T$ for the current study, and 50 GeV, where the fake spectrum runs out of statistics. The error of the fit is used as the error on the fake probability.

The same procedure is used to determine the fake PELC rate, with the modifications that only PELCs are counted in the jet spectrum, and the 1PELC sample is used instead of the QCD sample. When the QCD sample is used to determine the fake PELC rate, similar numbers are observed but with higher errors.

To construct the spectra, one *could* loop over all jets and electron candidates and enter their $E_T$ into the appropriate histogram. However, the actual procedure is somewhat more complicated in its details to avoid several problems. First, not all em candidates have a corresponding jet; for example, PARTICLE_SELECT removes jets matching a tight electron. Second, the trigger jet must to be removed to avoid trigger turn-on bias. Finally, leading electron candidates matching the trigger jet must also be excluded.

### 3.3.5 Results for Electron Efficiency and Fake Probability

The efficiency and fake probability were calculated for the 5 variable likelihood in the CC, and the 4 variable likelihood in the EC. Cut values 2, 1, 0.5, 0.25 on elike were chosen to generate rejection versus efficiency curves.[9] The parent and daughter sample used for the efficiency determination requires $z_{\rm vtx} < 60$ cm, the trailing electron candidate $E_T > 25$ (20) GeV for the CC (EC), and is inclusive with respect to additional jets. All electrons are required to pass an isolation cut of 0.3. The results are given in Table 3.2 and Figure 3.11. The efficiency of various likelihood cuts is presented in Tables 3.3 and 3.4, and Figure 3.12.[10]

There is a possible systematic error due to the jet multiplicity effect on either the tracking efficiency or the likelihood variable. No trend seen in the tracking efficiency as a function of the number of jets in an event (Figure 3.13), when the tracking efficiency is defined as identifying the electron as a PELC.[11] There is, however, a trend toward lower efficiency for the likelihood variable as a function of jet multiplicity (Figure 3.14). This type of degradation is consistent with a slightly poorer tracking resolution in high multiplicity events. Table 3.5 shows the final efficiencies used in FMCØ for signal and background determination.

[9] The term *rejection* is used here interchangeably with fake probability. A low fake probability is a high rejection of background.

[10] All efficiency and rejection numbers were recalculated for CAFIX high and CAFIX low corrections to the two data samples; the results were indistinguishable from the nominal corrections.

[11] This definition is slightly different from another common definition of tracking efficiency where a track is said to be found only if a track match significance cut is passed.

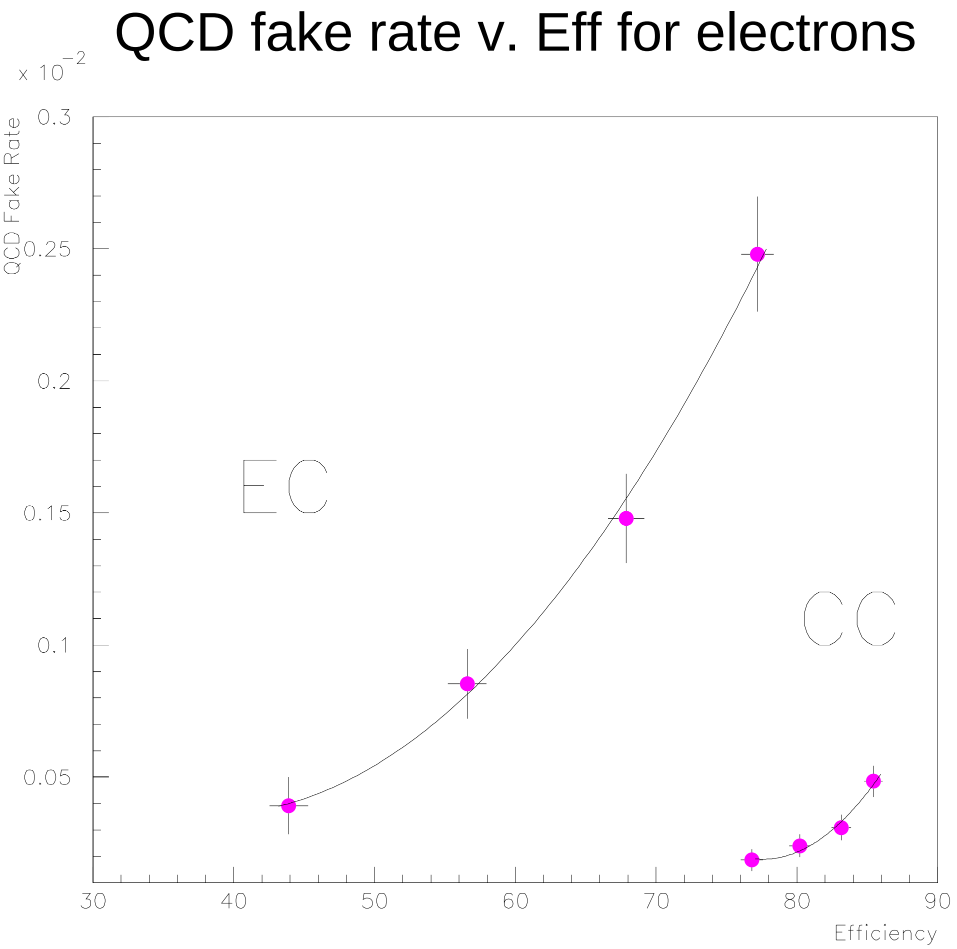


Figure 3.11: Rejection versus efficiency for 5 (4) variable likelihood cuts of 2.0, 1.0, 0.5, and 0.25 on CC (EC) electron candidates. The errors overlap between some points because the samples are correlated.

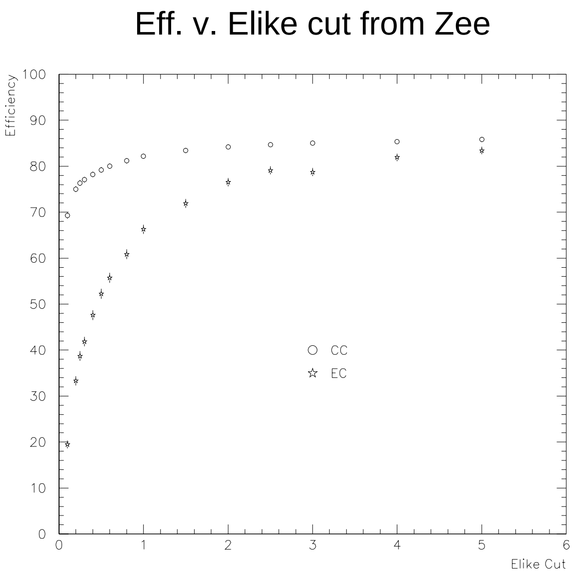


Figure 3.12: Efficiency spectra of electron likelihood cut. CC is the 5 variable, and EC is the 4 variable.

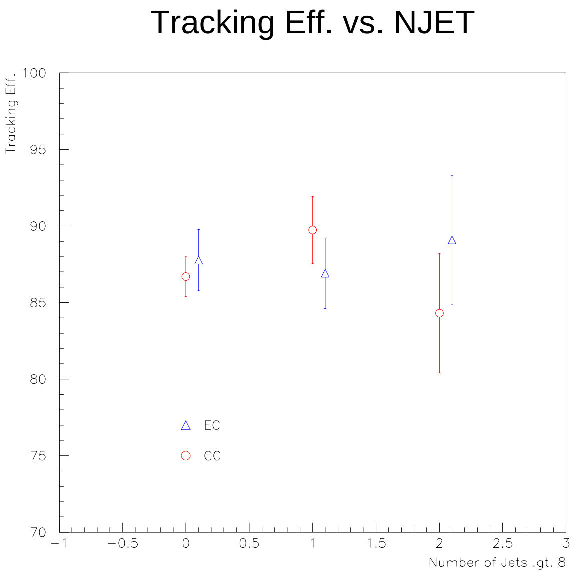


Figure 3.13: Tracking efficiency verses number of jets with $E_T$ greater than 8 GeV.

Table 3.2: Rejection and efficiency numbers for CC and EC electrons. The efficiencies quoted here are from the inclusive $Z$ boson sample, and include the tracking efficiency.

| elike | Jet Prob. | chisq | PELC Prob. | chisq | Cal | eff |
|---|---|---|---|---|---|---|
| 2.00 | $4.84 \pm .599$E-04 | 1.52 | $1.81 \pm .128$E-02 | 2.67 | CC | $85.44 \pm 0.66$ |
| 1.00 | $3.09 \pm .484$E-04 | 1.25 | $1.26 \pm .106$E-02 | 1.98 | CC | $83.15 \pm 0.70$ |
| 0.50 | $2.40 \pm .431$E-04 | 0.63 | $8.27 \pm .861$E-03 | 1.39 | CC | $80.19 \pm 0.75$ |
| 0.25 | $1.86 \pm .418$E-04 | 0.61 | $4.26 \pm .621$E-03 | 1.12 | CC | $76.81 \pm 0.79$ |
| 2.00 | $2.48 \pm .218$E-03 | 1.08 | $5.74 \pm .302$E-02 | 2.21 | EC | $77.19 \pm 1.16$ |
| 1.00 | $1.48 \pm .169$E-03 | 1.16 | $3.66 \pm .238$E-02 | 1.45 | EC | $67.88 \pm 1.29$ |
| 0.50 | $8.53 \pm 1.32$E-04 | 0.76 | $2.01 \pm .176$E-02 | 1.09 | EC | $56.59 \pm 1.37$ |
| 0.25 | $3.92 \pm 1.08$E-04 | 0.76 | $8.43 \pm 1.21$E-03 | 0.91 | EC | $43.91 \pm 1.37$ |

## 3.4 Muons

### 3.4.1 Candidate Muons

Muon candidates are constructed in four steps: hit sorting, track finding, quality determination, and global quality determination. Hit sorting takes the raw data from the MUD1 banks and converts hits into points in the DØ coordinate system. Pattern recognition algorithms find hits due to the passage of single particles through the muon chambers. A flag word (IFW2) records quality of the projection of the track to the vertex, and quality of the fit based on position and timing information from the hits. Other flag bits are set if the track is missing an A-layer hit, or a BC-layer hit.

These track candidates are processed into PMUO banks. MIP traces are searched for in the hadronic layers of the calorimeter in a wide road to the primary vertex. If layers do not contain energy, roads are formed to secondary vertices. For tracks

Table 3.3: Raw numbers used to calculate efficiencies and errors for CC electrons. Shown are the number of Z events found in the parent and daughter distributions, along with the calculated backgrounds in the Z-bin.

| Elike-5 Cut | Eff. | Err. | Z Par. | Z Dau. | Back Par. | Back Dau. |
|---|---|---|---|---|---|---|
| 5.00 | 86.56 | 0.64 | 2841 | 2459 | 650 | 240 |
| 4.00 | 86.21 | 0.65 | 2841 | 2449 | 650 | 238 |
| 3.00 | 85.86 | 0.65 | 2841 | 2439 | 650 | 228 |
| 2.50 | 85.44 | 0.66 | 2841 | 2427 | 650 | 222 |
| 2.00 | 85.44 | 0.66 | 2841 | 2427 | 650 | 195 |
| 1.50 | 84.57 | 0.68 | 2841 | 2402 | 650 | 190 |
| 1.00 | 83.15 | 0.70 | 2841 | 2362 | 650 | 182 |
| 0.80 | 81.83 | 0.72 | 2841 | 2325 | 650 | 177 |
| 0.60 | 80.82 | 0.74 | 2841 | 2296 | 650 | 174 |
| 0.50 | 80.19 | 0.75 | 2841 | 2278 | 650 | 172 |
| 0.40 | 79.27 | 0.76 | 2841 | 2252 | 650 | 162 |
| 0.30 | 77.64 | 0.78 | 2841 | 2206 | 650 | 153 |
| 0.25 | 76.81 | 0.79 | 2841 | 2182 | 650 | 148 |
| 0.20 | 75.54 | 0.81 | 2841 | 2146 | 650 | 144 |
| 0.10 | 69.29 | 0.87 | 2841 | 1968 | 650 | 150 |

confirmed by the calorimeter, a road is formed in the central tracking chambers to search for matching tracks. The combined information on the vertex, the best matching central track, the muon track in the calorimeter, and the tracks in the muon layers are used in a global fit.

Muon momentum is determined by a least squares method that uses a seven parameter fit on the sixteen data points associated with each track [25].

### 3.4.2 Muon Identification

Muon identification uses software (MUJETS_MU_SELECT) developed for top physics. For muons below 10 GeV/c, SOFT_MUON is required, while for others, ISOL_MUON

Table 3.4: Raw numbers used to calculate efficiencies and errors for EC electrons. Shown are the number of Z events found in the parent and daughter distributions, along with the calculated backgrounds in the Z-bin.

| Elike-4 Cut | Eff. | Err. | Z Par. | Z Dau. | Back Par. | Back Dau. |
|---|---|---|---|---|---|---|
| 5.00 | 83.86 | 1.01 | 1316 | 1103 | 481 | 208 |
| 4.00 | 81.92 | 1.06 | 1316 | 1078 | 481 | 193 |
| 3.00 | 79.39 | 1.12 | 1316 | 1045 | 481 | 181 |
| 2.50 | 79.35 | 1.12 | 1316 | 1044 | 481 | 159 |
| 2.00 | 77.19 | 1.16 | 1316 | 1016 | 481 | 147 |
| 1.50 | 73.03 | 1.22 | 1316 | 961 | 481 | 143 |
| 1.00 | 67.88 | 1.29 | 1316 | 893 | 481 | 123 |
| 0.80 | 64.79 | 1.32 | 1316 | 852 | 481 | 116 |
| 0.60 | 60.08 | 1.35 | 1316 | 790 | 481 | 92 |
| 0.50 | 56.59 | 1.37 | 1316 | 745 | 481 | 82 |
| 0.40 | 51.69 | 1.38 | 1316 | 680 | 481 | 77 |
| 0.30 | 46.67 | 1.38 | 1316 | 614 | 481 | 69 |
| 0.25 | 43.91 | 1.37 | 1316 | 578 | 481 | 48 |
| 0.20 | 36.55 | 1.33 | 1316 | 481 | 481 | 47 |
| 0.10 | 22.01 | 1.14 | 1316 | 290 | 481 | 29 |

is required. Efficiencies were calculated using a combination of $Z$ boson and $J/\Psi$ meson data and are a function of position and momentum [29].

Table 3.5: Final electron efficiencies used in FMCØ for likelihood cut of 1.0.

| $N_{\text{jets}}$ | CC (%) | EC (%) |
|---|---|---|
| 0 | $83.46 \pm 1.29$ | $68.64 \pm 2.60$ |
| 1 | $80.86 \pm 4.04$ | $61.43 \pm 3.59$ |
| 2 | $77.98 \pm 5.85$ | $62.68 \pm 7.55$ |

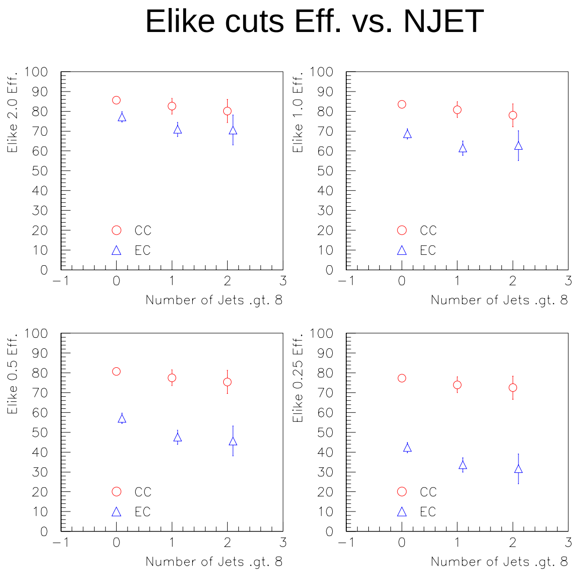


Figure 3.14: Tracking efficiency verses number of jets with $E_T$ greater than 8 GeV.

## 3.5 $\not{E}_T$

The first step in defining the missing $E_T$ is to determine the hard scattering vertex for the event. The hard scattering vertex is defined as the $z$-position at the beam axis of the primary leptons in the event.[12] An electron or muon gives two three-dimensional points: the central tracking center-of-gravity, and the calorimeter cluster center-of-gravity in the EM3 layer for electrons, or the track center-of-gravity position in the A-layer of the muon tracking chambers. These two points are extrapolated to the beam axis to determine the hard scattering vertex.

The fiducial $z_{\rm vtx}$ region of ±60 cm is required for all events. This cut is determined

[12] Here a distinction is made between the hard scattering vertex and the primary vertex, that found by RECO.

to be 98.1±0.8% efficient from the $Z \rightarrow ee$ sample, and the same efficiency is used for muon determination since the extrapolation resolution is similar (2.0 cm for electrons, and 1.5 to 2.5 cm for muons). The efficiency was determined by counting $Z$ bosons before and after the cut on the hard scattering vertex. A binomial statistical error is used (0.3%). The systematic error is the difference in the efficiency calculation between using the hard scattering or primary vertex to determine the invariant mass (0.7%). The two errors are added in quadrature.

The calorimeter missing $E_T$ is determined by CAFIX and corrected for muons that pass SOFT_MUON. A final correction is applied based on the new hard scattering vertex (jet and muon $E_T$'s are from the primary vertex in CAFIX).

# Chapter 4

# Monte Carlo and Analysis Simulation

## 4.1 Introduction

The understanding of any physical process is limited by one's ability to predict the outcome of an experiment. The first step in predicting an outcome is to select a theoretical model, including particles of the theory and their interactions. A *physics generator*, programmed with the theoretical model and the accelerator parameters, selects production and decay of particles from tables of cross sections and branching ratios. The output of the generator is the final state four-vectors of all undecayed particles in an event, the *event record.* The event record may be stored in a common block for further processing or written to a file.

The next step is to simulate the interactions of the final state particles in the experimental apparatus with a *detector simulation.* This program translates the event record into raw data from the detector, suitable for processing by a standard offline reconstruction program (RECO). RECO output can be analyzed with the same methods as the collected data, predicting the outcome of the experiment.

Simulating the outcome of the experiment involves many internal steps that require the generation of uniform random numbers; for example, to choose the decay of a particle from several possible decays weighted by their respective branching ratios. This random number generation is akin to a rolling of dice and this analogy lead to the term *Monte Carlo* to describe the overall process.

## 4.2 Fast Monte Carlos

Several theoretical motivations for new physics at the Electroweak scale prompt searches for undiscovered particles (Chapter 1). A typical search analysis relies heavily on the generation of signal Monte Carlo events (MC) for the particular extension to the Standard Model under study. Usually, one uses a physics generator, such as ISAJET or PYTHIA, to produce events, processes those events through a detector simulation (GEANT) and runs DØ RECO to obtain a final signal sample, and applies analysis cuts to determine a number of expected events from a given model.

Each model possesses free parameters which vary in some *allowed space.* With more than two free parameters, reasonable sampling of the allowed space requires unreasonable CPU cycles. Supergravity models (SUGRA or CMSSM) require four parameters and one free sign. Sampling 10 points in each parameter requires generation of 20,000 models with approximately 20,000 events each.

Processing this many events at the rate of one event per few minutes of CPU, the typical time required to completely simulate the detector and reconstruct the output, would require roughly 50,000 CPU-days; even with a dedicated 100 processor farm this task would take a few years to complete. If a new model suddenly sparks wide interest, or bugs are found in the current physics generator, the whole generation process must be repeated. The solution to the time problem is to develop a program that can quickly take events from a physics generator and determine the probability of finding a given set of objects in the event after full simulation. This can be accomplished if energy resolution and detection efficiency are well understood. For example, after years of studying electrons, one should be able to determine the probability that given electron will

1. Leave energy in a fiducial region of the detector.

2. Pass a given trigger threshold.

3. Pass an offline electron identification cut.

In addition, one should be able to assign a likely offline energy measurement given a known resolution. Finally, one would like the program to loop over a set of models and return, for each model, the only important number in the simulation: $\sigma \cdot BR \cdot$

$a \cdot e$,[1] where $\sigma$ is the production cross section of the process under study, $BR$ is the theoretical branching ratio to the signature under study, $a$ is the detector acceptance (fiducial cuts), and $e$ is the net efficiency for reconstructing and identifying all of the required objects. The general program written to fulfill this purpose is FMCØ.

## 4.3 FMCØ: The General Simulation Program

FMCØ is a complete Monte Carlo signal analysis package. It includes a SUGRA model generator (Michmodel), a MC signal generator (SPYTHIA 2.10), a jet clustering package (DØPJET), a fast detector simulation (QSIM), and an event analysis package. DØPJET operates on the individual final state four-vectors and clusters them into 0.5 cone jets. QSIM operates on the generator level four-vectors for leptons and the DØPJET output replacing the GEANT simulation with fast parameterized smearing. The event analysis package operates on the smeared objects and replaces DØRECO.

The event analysis package is split into three parts: RECO simulation (RSIM), signal simulation (SSIM), and trigger simulation (TSIM). RSIM and TSIM are interface packages that call utility routines from the QSIM library or use hard coded efficiencies, as in the current study. SSIM is a bookkeeping package that analyzes each event as it is produced and processed, keeping track of the number of events produced, the number of events that pass a *kinematic channel* (see next section), and

[1] This final number is multiplied by the luminosity to predict the number of events in the data sample.

each event's weight as determined by TSIM and RSIM.[2] The sum of these weights is used to determine the visible cross section

$$\sigma_{\text{vis}} = \sigma_{\text{tot}} \cdot BR \cdot a \cdot e = \frac{\sigma_{\text{tot}}}{N} \sum_{i=1}^{N} w_i^{\text{QSIM}} w_i^{\text{TSIM}} w_j^{\text{RSIM}}$$

where $N$ is the number of events produced.

The entire package is controlled with various RCP files. The user sets up FMCØ to study a portion of SUGRA parameter space, and FMCØ writes an ntuple that one can use to predict the number of expected signal events at several sample points within the studied region. The complete program manual is included in Appendix A. Details of the program important for the current analysis are documented in the following sections.

## 4.4 Kinematic Channels

The current analysis studies three signatures: $ee + jets$, $e\mu + jets$, and $\mu\mu + jets$. All signatures also require $\not\!\!E_T$. Within each signature, one can vary the lepton, jet, and $\not\!\!E_T$ thresholds. A set of object requirements and thresholds is called a kinematic channel. A kinematic channel can include additional event topology requirements, such as a cut on the invariant mass of the two leading electrons.

The name of a kinematic channel includes three fields: the lepton field, the jet and $\not\!\!E_T$ field, and the topological field. [3] The first two fields are called the object

[2] The number of events that pass a kinematic channel is also a weight, 1.0 or 0.0, from QSIM.

[3] Kinematic channel names were limited to eight alphanumeric characters by the program used

Table 4.1: Kinematic channel name definition: lepton field $E_T$ thresholds (in GeV). "T" was selected to stand for 10.

| field | $e_1$ | $e_2$ | $\mu_1$ | $\mu_2$ |
|---|---|---|---|---|
| $ee$ | 17 | 15 | | |
| $em$ | 17 | | 4 | |
| $e7m$ | 17 | | 7 | |
| $eTm$ | 17 | | 10 | |
| $mm$ | | | 20 | 10 |

fields, described in Tables 4.1 and 4.2.

There are two topological requirements. The first is a cut on the invariant mass of the two leading electrons ($80 \leq M_{ee} \leq 105$ GeV/c$^2$), designed to remove $Z$ boson events in the $ee$ signature. $Nz$ (for "No Z") in the topological field means the mass window has been excluded; "z" means no mass cut.

The second topological cut is on the phi direction of the $\not\!\!E_T$, $|\phi_\mu - \phi_{MET}| >$ 10 degrees. The leading (two leading) muons are considered in the $e\mu$ ($\mu\mu$) signature. No characters are present in the topological field for these cuts since they are applied to all kinematic channels in the muon signatures.

The FMCØ SSIM framework allows for simultaneous analysis of many kinematic channels at once. Different kinematic channels within a signature can produce better results in various regions of model space. Table 4.3 is a partial enumeration of the kinematic channels considered. Not shown are "sister" channels in the $ee$ signature ($z$ replaced by $Nz$), and sister channels in muon signatures ($ee$ replaced by $em$, $e7m$,

to analyse the results.

Table 4.2: Kinematic channel name definition: jet and missing $E_T$ field thresholds (in GeV). The "H" was chosen to signal a "higher" threshold; its placement tells whether it modifies the leading jet or missing $E_T$ cut. The "U" was chosen to imply "ultimate" missing $E_T$.

| field | $j_1$ | $j_2$ | $j_3$ | $\not{E}_T$ |
|---|---|---|---|---|
| $jj$ | 20 | 20 |  | 20 |
| $Hjj$ | 45 | 20 |  | 20 |
| $jjH$ | 20 | 20 |  | 30 |
| $HjjH$ | 45 | 20 |  | 30 |
| $jjU$ | 20 | 20 |  | 40 |
| $HjjU$ | 45 | 20 |  | 40 |
| $jjj$ | 20 | 20 | 20 | 20 |
| $Hjjj$ | 45 | 20 | 20 | 20 |
| $jjjH$ | 20 | 20 | 20 | 30 |
| $HjjjH$ | 45 | 20 | 20 | 30 |
| $jjjU$ | 20 | 20 | 20 | 40 |
| $HjjjU$ | 45 | 20 | 20 | 40 |

$eTm$, or $mm$). There is no invariant mass cut in $\mu\mu$ signatures.[4] There are a total of 52 kinematic channels.

## 4.5 TSIM: Fast Trigger Simulation

The TSIM package returns the probability that an event passed online thresholds. Each trigger is a set of online objects and the net probability is the product of the individual object probabilities, modified by combinatorics. The parameterizations are necessarily a function of offline kinematic quantities since QSIM only returns

[4] Due to poor muon momentum resolution, the $U$ channels instead provide the lowest $Z$ boson background.

Table 4.3: List of some of the kinematic channels used in the present analysis. All thresholds listed are in units of GeV $E_T$.

| Name | $e_1$ | $e_2$ | $j_1$ | $j_2$ | $j_3$ | $\mu_1$ | $\mu_2$ | $\not{E}_T^{cal}$ | $\not{E}_T^{\mu}$ | $M_{ee}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $eejjNz$ | 17 | 15 | 20 | 20 | - | - | - | 20 | - | y |
| $eejjz$ | 17 | 15 | 20 | 20 | - | - | - | 20 | - | n |
| $eejjjz$ | 17 | 15 | 20 | 20 | 20 | - | - | 20 | - | n |
| $eeHjjz$ | 17 | 15 | 45 | 20 | - | - | - | 20 | - | n |
| $eejjHz$ | 17 | 15 | 20 | 20 | - | - | - | 30 | - | n |
| $eeHjjjz$ | 17 | 15 | 45 | 20 | 20 | - | - | 20 | - | n |
| $eejjjHz$ | 17 | 15 | 20 | 20 | 20 | - | - | 30 | - | n |
| $eeHjjHz$ | 17 | 15 | 45 | 20 | - | - | - | 30 | - | n |
| $eeHjjjHz$ | 17 | 15 | 45 | 20 | 20 | - | - | 30 | - | n |
| $emjj$ | 17 | - | 20 | 20 | - | 4 | - | 20 | 20 | - |
| $em7jj$ | 17 | - | 20 | 20 | - | 7 | - | 20 | 20 | - |
| $emTjj$ | 17 | - | 20 | 20 | - | 10 | - | 20 | 20 | - |
| $mmjj$ | - | - | 20 | 20 | - | 20 | 10 | 20 | 20 | - |
| $mmjjU$ | - | - | 20 | 20 | - | 20 | 10 | 40 | 40 | - |
| $mmHjjU$ | - | - | 45 | 20 | - | 20 | 10 | 40 | 40 | - |
| $mmjjjU$ | - | - | 20 | 20 | 20 | 20 | 10 | 40 | 40 | - |
| $mmHjjjU$ | - | - | 45 | 20 | 20 | 20 | 10 | 40 | 40 | - |

smeared offline variables.[5]

Two triggers are used in the current analysis: ELE_JET_HIGH for $ee$ and $e\mu$ signatures, and MU_JET_XXX for the $\mu\mu$ signature. ELE_JET_HIGH parameterization will be discussed in detail, and a short description of the muon parameterization follows.

[5] Full detector simulation (GEANT) allows calculation of online kinematics. For speed reasons, FMCØ is designed with all parameterizations in terms of offline quantities.

Table 4.4: Online Requirements for ELE_JET_HIGH.

| Level 1 $E_T$'s | Level 2 $E_T$'s |
|---|---|
| $1\ e > 12$, $1\ j > 5$ | $1\ e > 15$, $1\ j > 10$, $\not\!\!E_T > 14$ |

### 4.5.1 ELE_JET_HIGH Parameterization

Online $E_T$ requirements for ELE_JET_HIGH are listed in Table 4.4. The minimal selection criteria always require two offline jets above 20 GeV $E_T$ and offline $\not\!\!E_T$ above 20 GeV. The efficiency for these cuts will be returned to later; the first consideration in TSIM is the 15 GeV $E_T$ Level 2 electron turn-on.

#### Level 2 Electron Turn-on

The usual process for determining the Level 2 turn-on for a given electron threshold is to produce an offline good electron $E_T$ spectrum, where the Level 2 threshold was far enough below the threshold under study that it can be assumed 100% efficient. Another $E_T$ spectrum is produced from the same data sample but with the additional requirement of a matching Level 2 electron above the given threshold under study. The ratio of the two spectra is the turn-on curve. Figure 4.1 shows spectra for Level 2 $E_T$ and the offline $E_T$. The ratio distribution is fit to the function

$$\text{efficiency} = \frac{1}{2}par(3)\left[1.0 + erf\left(\frac{x - par(1)}{\sqrt{2}par(2)}\right)\right]$$

where $erf$ is the error function and $par(i)$ represents the three parameters of the fit. This form represents a step function convolved with a gaussian resolution: $par(1)$

is the step threshold, $par(2)$ is the resolution, and $par(3)$ is the plateau efficiency. This is appropriate if the resolution of Level 2 electrons with respect to offline can be described as a gaussian of constant width in the region of the threshold. The fitted function is the turn-on curve, or simply the turn-on. Figures 4.2 and 4.3 show the determination of a 13 GeV $E_T$ threshold turn-on using data from the MU_ELE trigger (which required a 7 GeV $E_T$ electron at Level 2). Offline electrons were selected by $elike < 1.0$, for electron clusters confirming the primary vertex. In addition, the primary vertex was restricted to ±60 cm of $z = 0$.

Unfortunately, there is no readily available data set to determine the Level 2 electron turn-on of ELE_JET_HIGH (15 GeV $E_T$) by the usual method. Instead, a scaling method is employed. Scaling assumes that a plot of $E_T(L2)/E_T(\mathit{off})$ is nearly independent of $E_T(\mathit{off})$. Using this approximate relation, one can make parent and daughter distributions of $E_T \equiv \frac{1}{q}E_T^0(\mathit{off})$ using the Level 2 threshold $E_T(L2) \equiv qE_T^0(L2)$, where $q$ is the scaling factor. $par(1)$, $par(2)$, and $par(3)$ are determined in this manner for $q = 1.3$ ("30% scaling"). This method allows the determination of the 15 GeV turn-on from a 15 GeV trigger sample, and is only based on the assumption that the curves don't change much over the range of a few GeV. This procedure is expected to work because the dominant terms affecting the Level 2 versus offline $E_T$ resolution should obey linear scaling. These effects are:

**Vertex Determination** Level 2 uses the Slow-Z vertex position while the offline vertex is determined by RECO. The relative vertex position provides an event dependent, but $E_T$ independent, correction and thus a plot of $E_T(L2)/E_T(\mathit{off})$

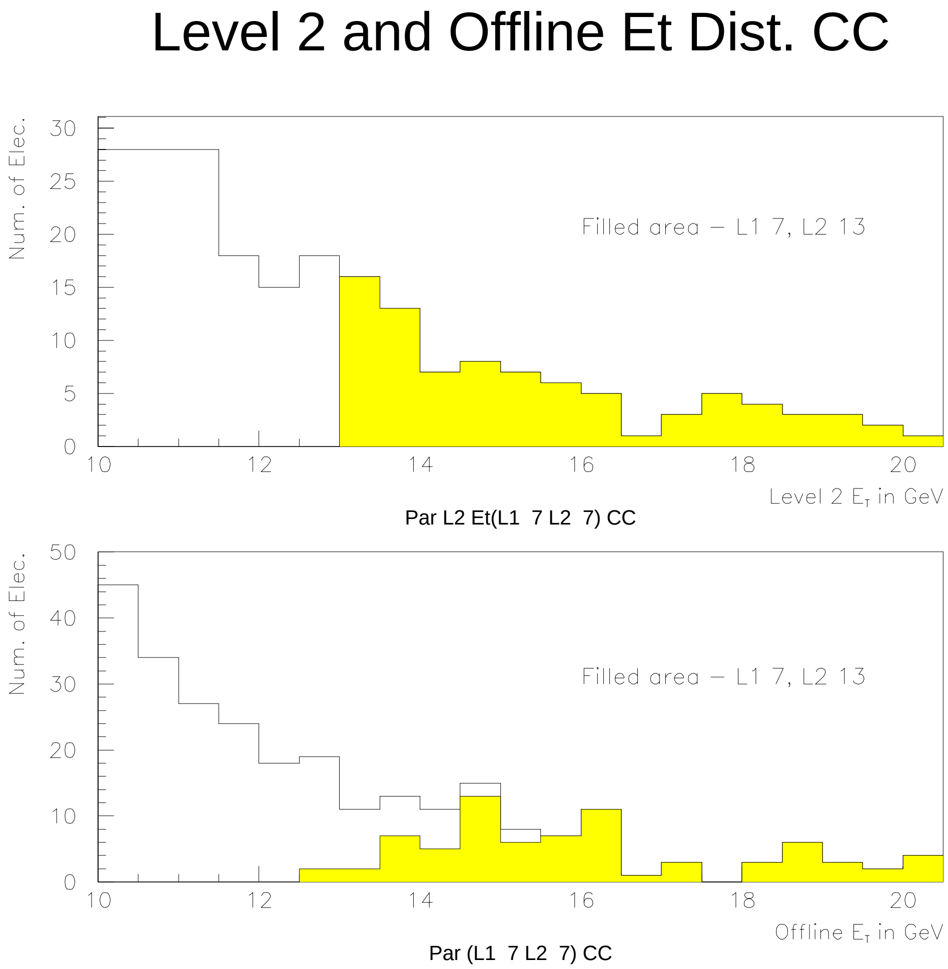


Figure 4.1: Example of Level 2 threshold cut on offline $E_T$ distribution. The spreading of the turn-on is due to the relative resolution between Level 2 and offline.

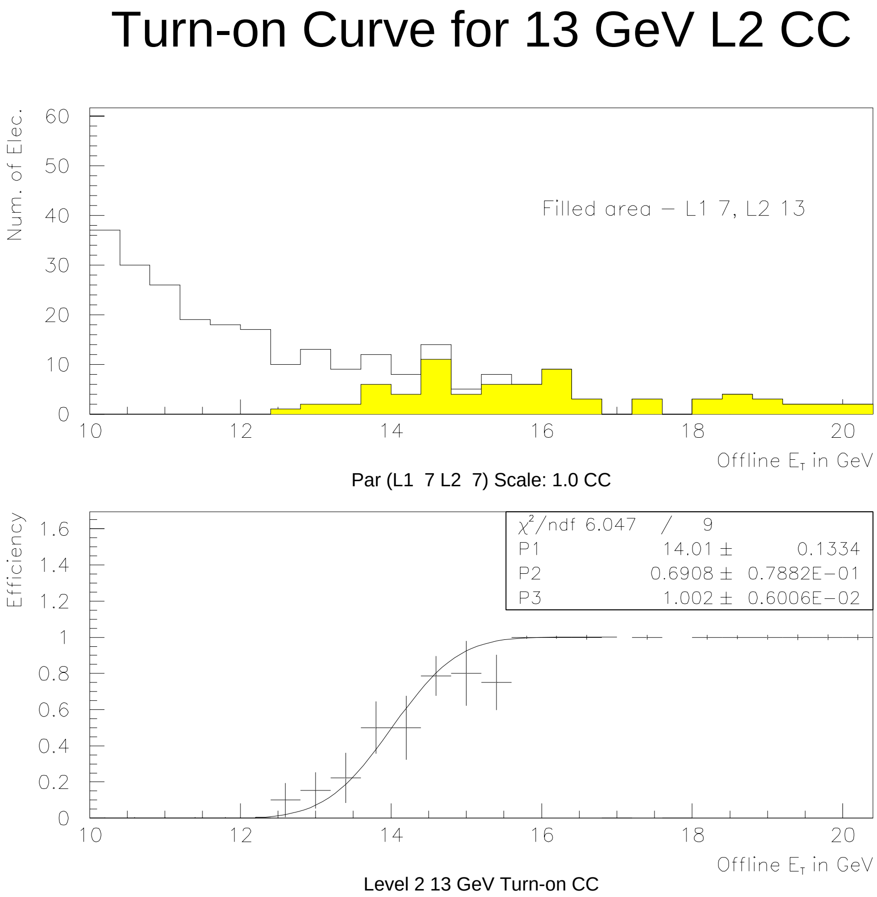


Figure 4.2: 13 GeV Level 2 turn-on for the central calorimeter. The parent offline $E_T$ distribution is the open histogram while the daughter (hatched region) is the offline $E_T$ distribution after application of a 13 GeV Level 2 threshold. The turn-on is determined by a bin-by-bin ratio of the two histograms and is shown in the bottom plot, along with the fit to an error function.

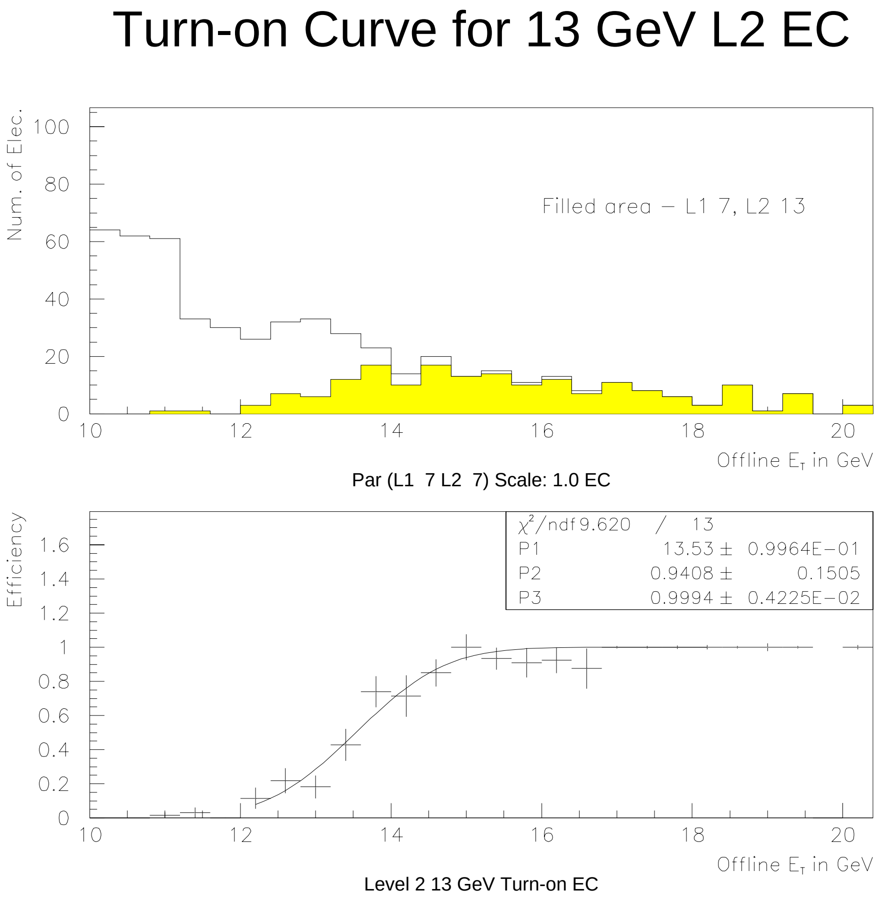


Figure 4.3: 13 GeV Level 2 turn-on for the end cap calorimeters.

is independent of $E_T(\mathit{off})$ and therefore linear scaling will be valid. This is a dominant correction.

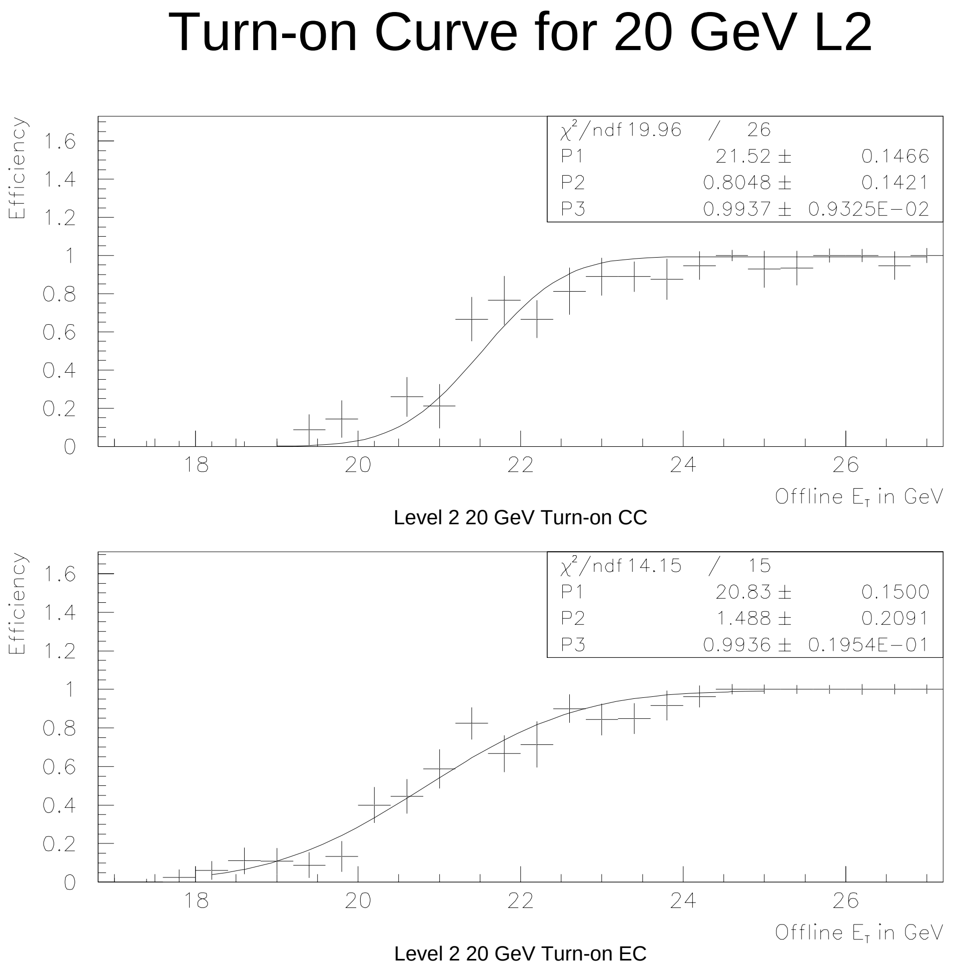
**Calibration** The online and offline energy calibration constants differed, but there should be a nearly constant ratio of online and offline energy to first order. Over a small range, one expects this correction to obey linear scaling.

**Cone Size** The Level 2 energy is determined within a 3x3 readout tower, while the offline energy is determined by a more inclusive clustering algorithm. These *out-of-cone* corrections to the energy vary according to shower profile, which varies logarithmically with $E_T$. Over any range, linear scaling will systematically underestimate the relative resolution. This is a second order correction, whose irrelevance is demonstrated by good $par(2)$ determination (see below).

**Noise and Underlying Event** Noise and underlying event corrections to the online and offline energy will be a weak function of energy and are not expected to obey linear scaling. Rather, the effects give an approximate offset ($\delta$), so linear scaling will underestimate the noise effects in the resolution. This is also a second order correction, and is shown to be small by the good $par(1)$ (threshold) determination (see below).

As proof that scaling works, a 20 GeV turn-on was generated from the ELE_JET_HIGH sample using both the usual and scaling methods, Figures 4.4 and 4.5. The results between the two methods agree well within errors.

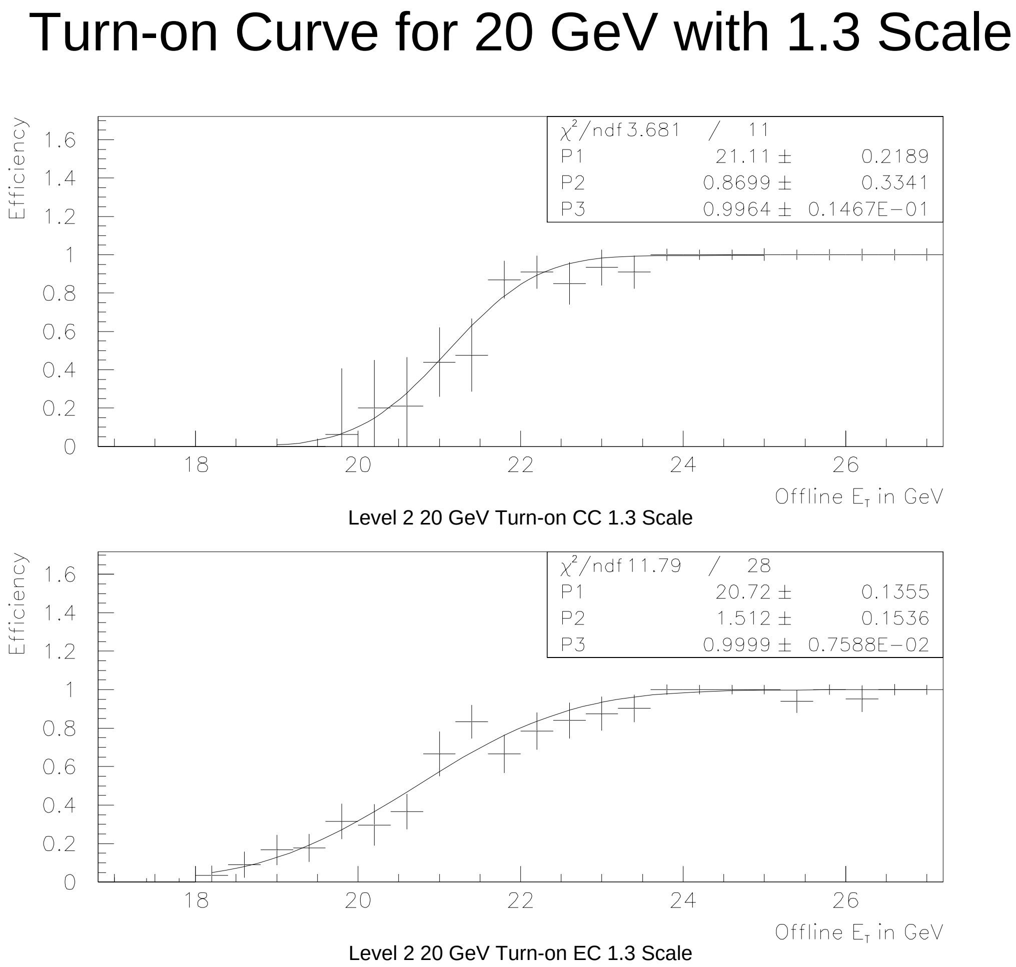
Figure 4.6 shows the results of 30% scaling for the 15 GeV threshold. The



Level 2 20 GeV Turn-on CC

Level 2 20 GeV Turn-on EC

Figure 4.4: 20 GeV Level 2 Threshold without using a scaling factor.

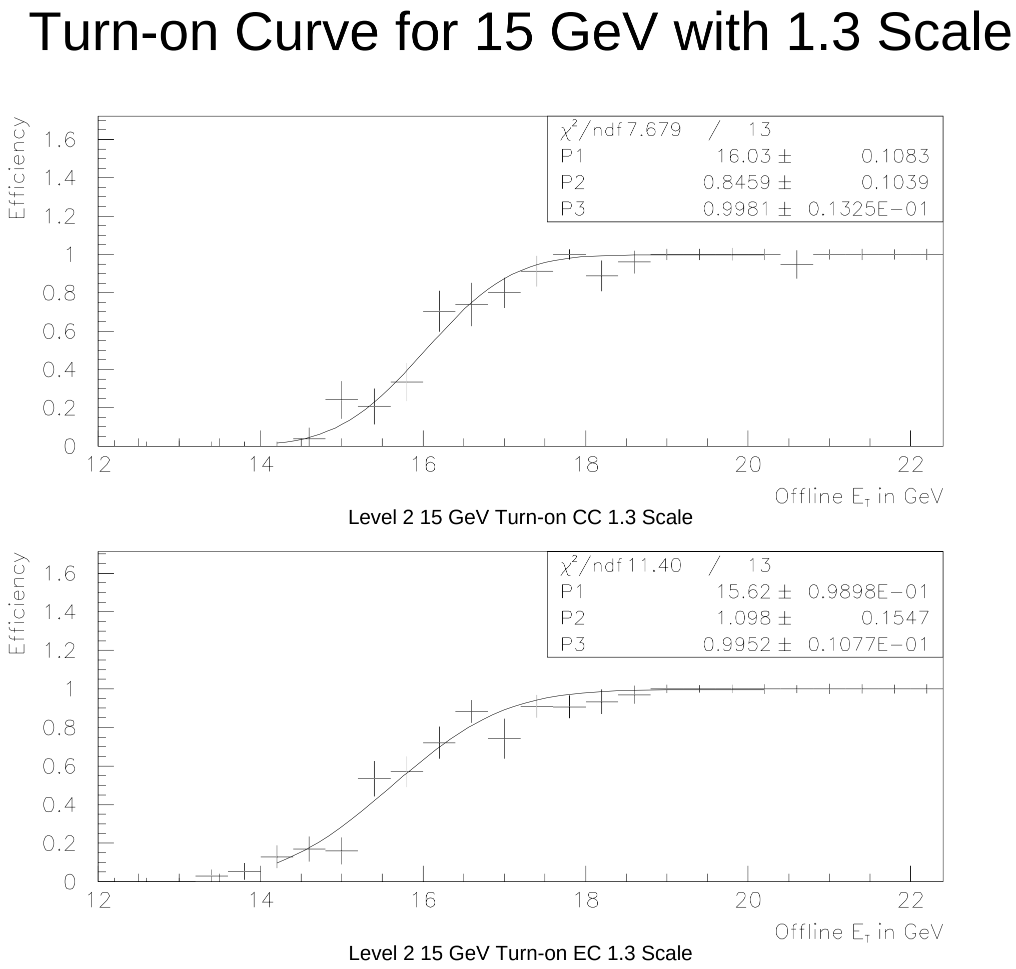


Turn-on Curve for 20 GeV with 1.3 Scale

Level 2 20 GeV Turn-on CC 1.3 Scale

Level 2 20 GeV Turn-on EC 1.3 Scale

Figure 4.5: 20 GeV Level 2 Threshold using a 30% scaling factor. Note that all three parameters are well determined.

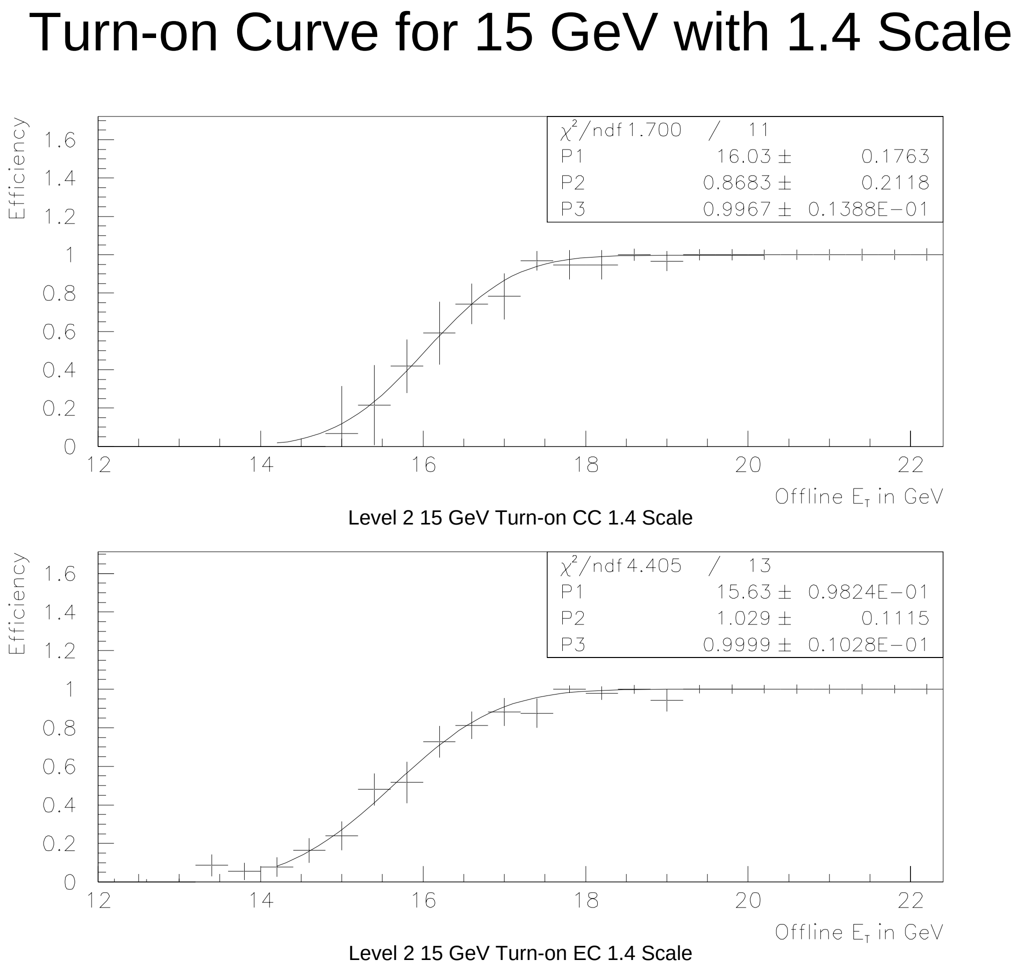


Turn-on Curve for 15 GeV with 1.3 Scale

Level 2 15 GeV Turn-on CC 1.3 Scale

Level 2 15 GeV Turn-on EC 1.3 Scale

Figure 4.6: 15 GeV Level 2 threshold using a 30% scaling factor.



Level 2 15 GeV Turn-on CC 1.4 Scale

Level 2 15 GeV Turn-on EC 1.4 Scale

Figure 4.7: 15 GeV Level 2 threshold using a 40% scaling factor. The procedure was repeated with a 40% scaling factor to show stability in the results.

procedure was repeated with a 40% scaling factor as a cross-check (Figure 4.7), and the results are quite stable. The 15 GeV turn-on curve is then defined as

$$par(1)\ \mathrm{CC\ (EC)} = 16.03 \pm 0.11\ (15.62 \pm 0.10) \quad (4.1)$$

$$par(2)\ \mathrm{CC\ (EC)} = 0.85 \pm 0.10\ (1.10 \pm 0.15) \quad (4.2)$$

$$par(3)\ \mathrm{CC\ (EC)} = 0.998 \pm 0.013\ (0.995 \pm 0.011) \quad (4.3)$$

A portion of the Run 1c data was taken with an increased Level 2 threshold of 17 GeV. The same procedure is used to determine the turn-on. Figure 4.8 shows the 17 GeV turn-on for 30% scaling and Figure 4.9 show the results for 40% scaling.

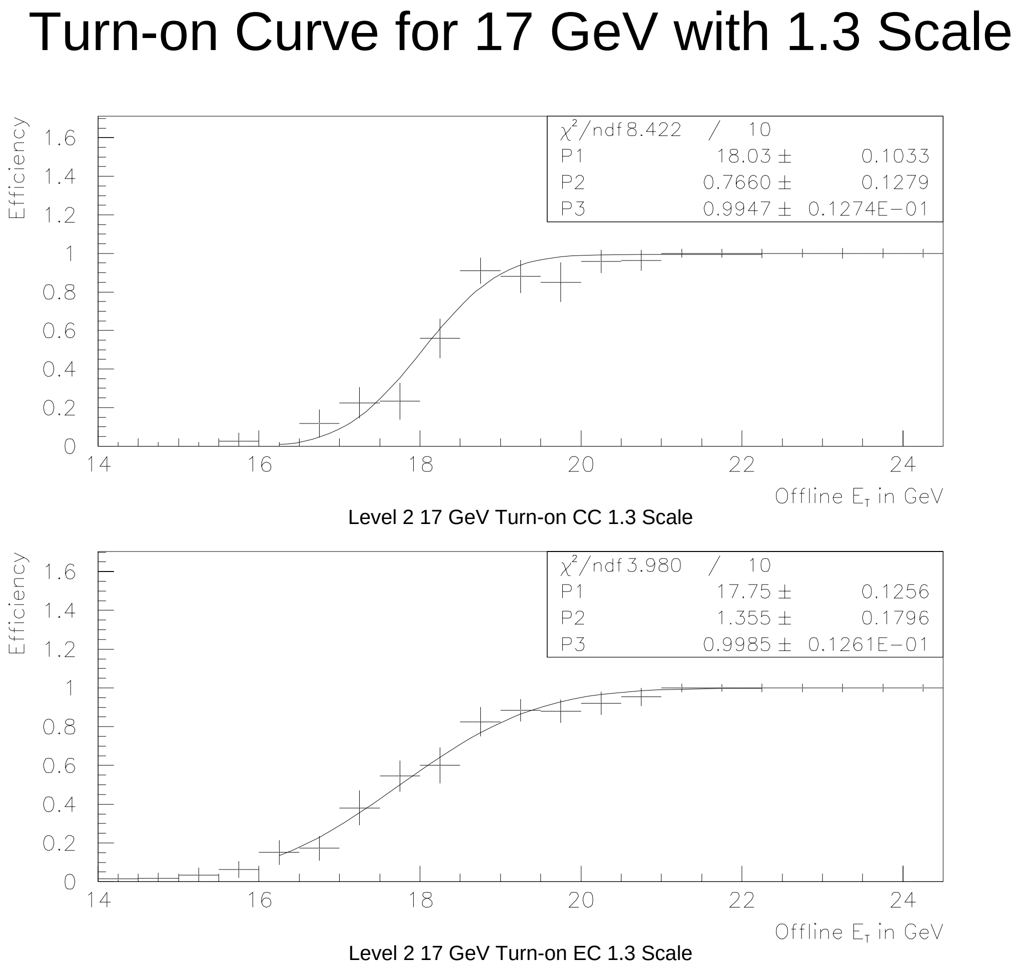
A further cross check is performed by assuming that the fit parameters should vary linearly in the range between 13 and 20 GeV. Figures 4.10, 4.11, and 4.12 compare interpolation and scaling methods. Good agreement is found for both the 15 GeV and 17 GeV parameters.

With $\epsilon_{15}$ and $\epsilon_{17}$ the trigger efficiencies for the two thresholds, the final effective Level 2 electron efficiency is

$$\epsilon = l_{15}\epsilon_{15} + l_{17}\epsilon_{17}$$

where $l_i$ is the relative luminosity for the two thresholds. $l_{15} = 0.9026$ and $l_{17} = 0.0974$ for the data sample. The errors, determined below, are weighted in a similar fashion.[6]

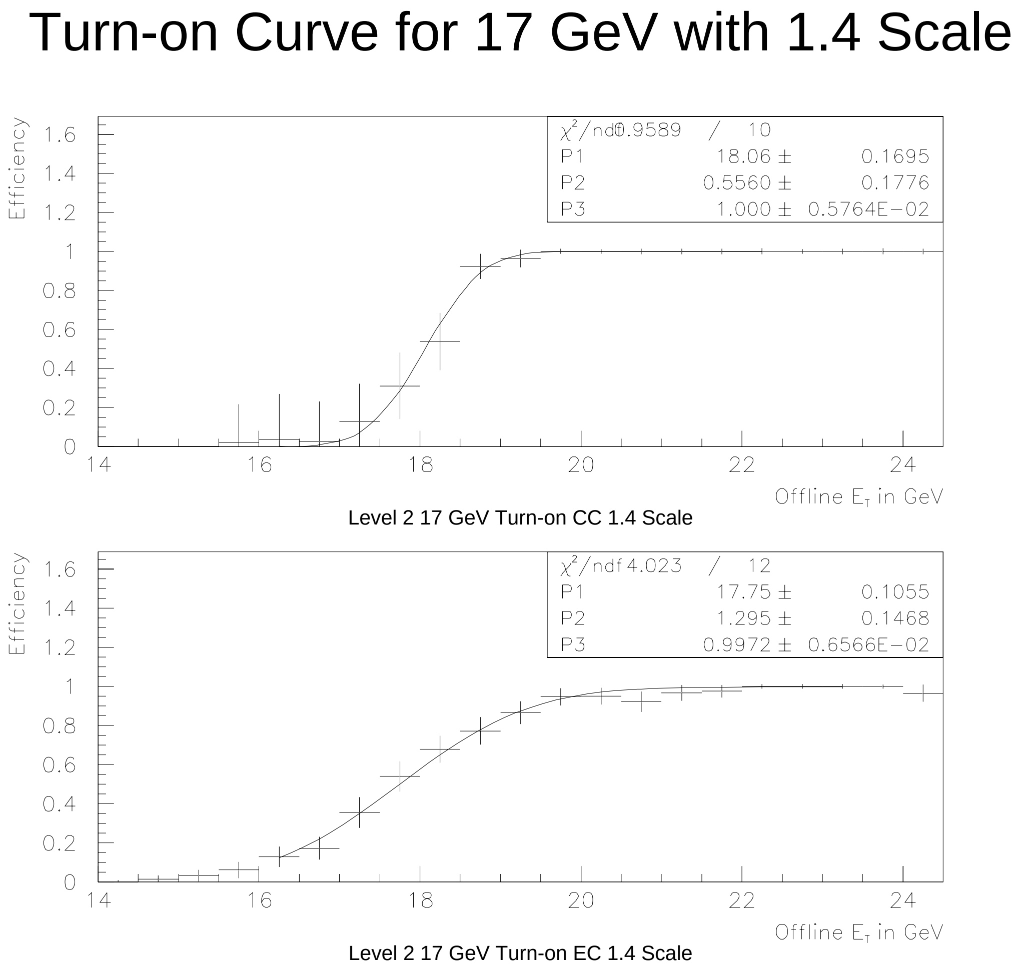
[6] One might worry that weighting the single electron efficiency in this manner will lead to an error in the two electron case. The difference appears in the $\epsilon^2$ term, where the true value is $l_{15}\epsilon_{15}^2 + l_{17}\epsilon_{17}^2$. The method above thus contains a maximum systematic error of $(l_{15} - l_{15}^2)(\epsilon_{15} - \epsilon_{17})^2$, or 2%, when both electrons fall in the range 16 to 18 GeV (the error is much smaller elsewhere).



Level 2 17 GeV Turn-on CC 1.3 Scale

Level 2 17 GeV Turn-on EC 1.3 Scale

Figure 4.8: 17 GeV Level 2 Threshold Determination using a 30% scaling factor.



Turn-on Curve for 17 GeV with 1.4 Scale

Level 2 17 GeV Turn-on CC 1.4 Scale

Level 2 17 GeV Turn-on EC 1.4 Scale

Figure 4.9: 17 GeV Level 2 Threshold Determination using a 40% scaling factor. Again, the results are stable.

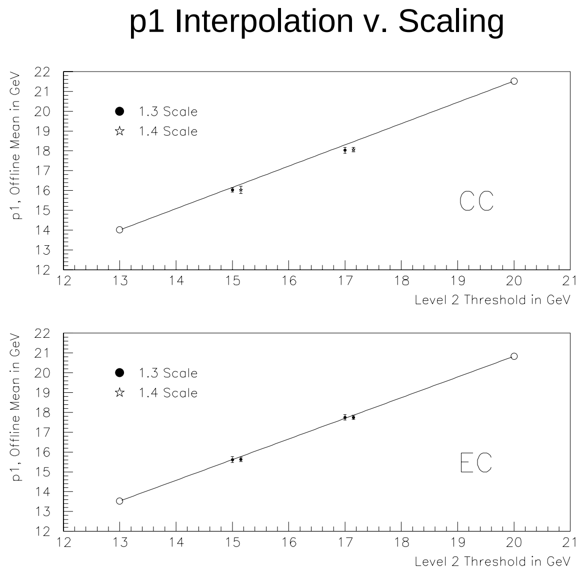


Figure 4.10: Comparison of results for $p_1$ between scaling and a linear interpolation between 13 and 20 GeV thresholds.

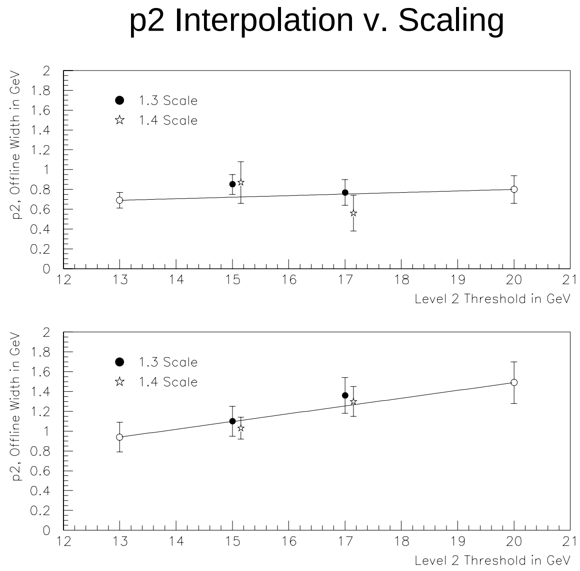


Figure 4.11: Comparison of results for $p_2$ between scaling and a linear interpolation between 13 and 20 GeV thresholds.

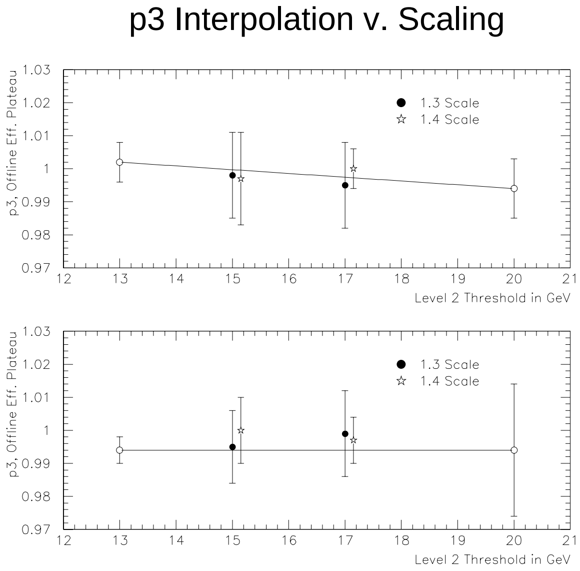


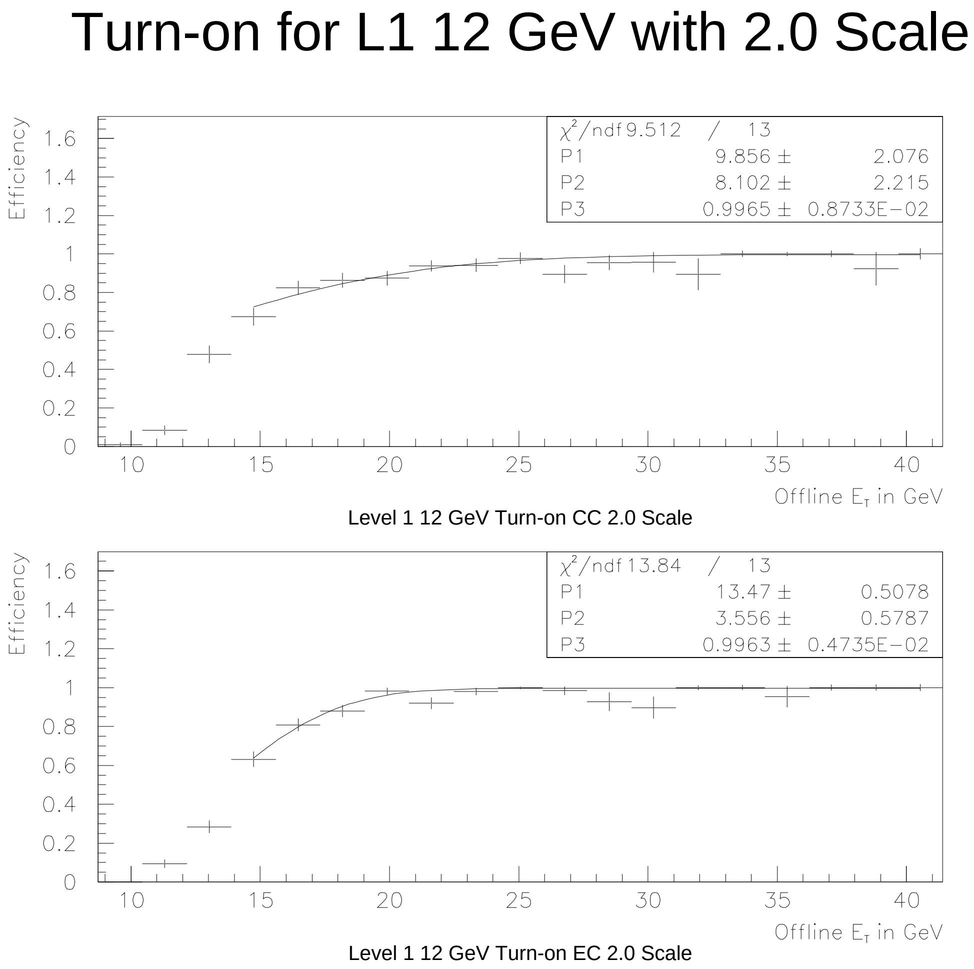
Figure 4.12: Comparison of results for $p_3$ between scaling and a linear interpolation between 13 and 20 GeV thresholds.

### Level 1 Electron Threshold Corrections

What has been obtained so far is the Level 2 threshold turn-on. The total trigger turn-on is a product of the Level 1 turn-on and the Level 2 turn-on.

The Level 1 electron cone of a single trigger tower dominates the resolution of Level 1 compared to offline $E_T$. Because an electron shower may occur at the edge of a trigger tower, the efficiency turns on at about twice trigger threshold. This geometric contribution to the energy seen by Level 1 results in worse energy resolution for Level 1 than Level 2, and thus the slower turn-on.

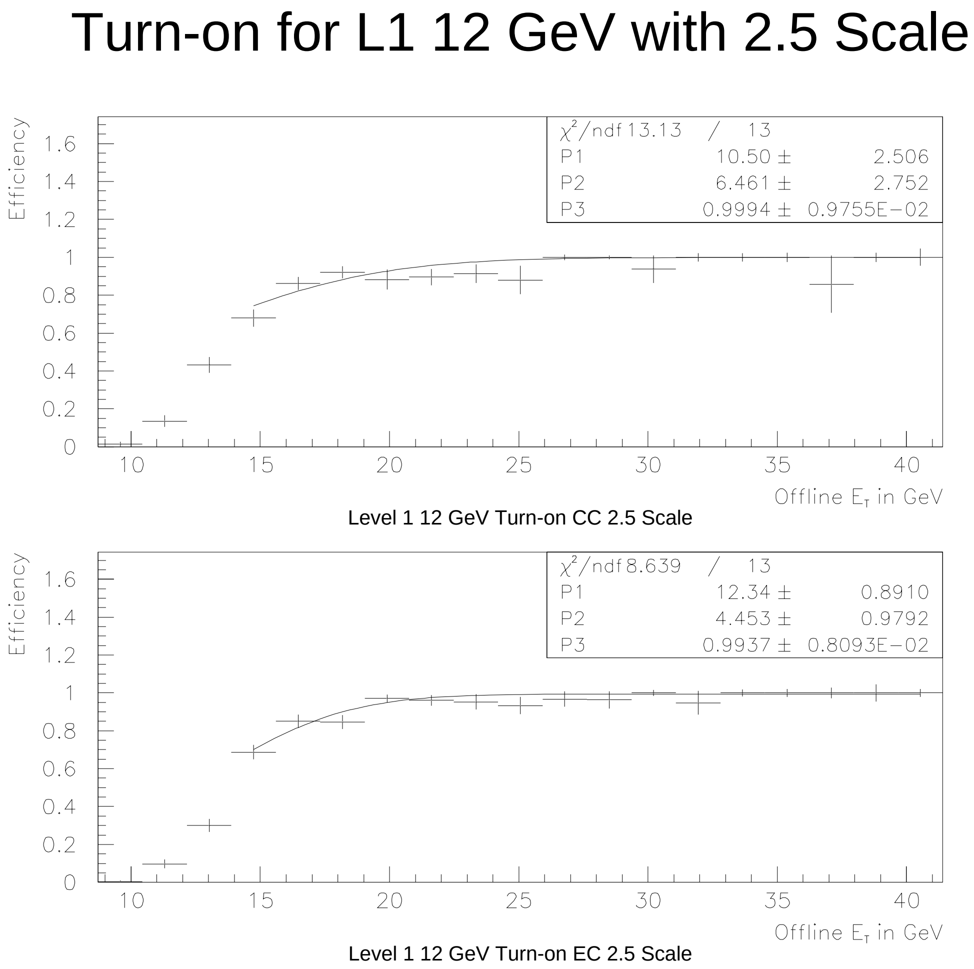
Once again, due to the lack of an appropriate data sample, the Level 1 turn-on will be measured by scaling. The dominant contribution to the resolution is geometric


Turn-on for L1 12 GeV with 2.0 Scale

Level 1 12 GeV Turn-on CC 2.0 Scale

Level 1 12 GeV Turn-on EC 2.0 Scale

Figure 4.13: Determination of the correction to the Level 2 efficiency from Level 1 turn-on using a 2.0 scale factor.



Level 1 12 GeV Turn-on CC 2.5 Scale

Level 1 12 GeV Turn-on EC 2.5 Scale

Figure 4.14: Determination of the correction to the Level 2 efficiency from Level 1 turn-on using a 2.5 scale factor. The average correction of the the two scales is applied to the Level 2 efficiency and half of their difference is used as the error.

($E_T$ independent), and the second order corrections are vertex and calibration effects. Level 1 will therefore obey scaling.

The ELE_JET_HIGH stream was processed to determine the Level 1 12 GeV turn-on. Two scaling factors were employed, 2.0 and 2.5, and the spectra were fit between 15 and 40 GeV with the error function form already presented.[7] The difference in the parameters found with the two scales is small (Figures 4.13 and 4.14). Figure 4.15 shows the two curves on the same plot for comparison. The correction applied to the Level 2 efficiency is the average of the two fits at a given $E_T$, and the error is taken as half of the difference.

### Net Electron Trigger Efficiency and Error

Figure 4.16 shows the single electron trigger efficiency ($\epsilon_{L2} \cdot \epsilon_{L1}$). The trigger efficiency for each of the two leading electrons is determined ($\epsilon_1$ and $\epsilon_2$), and the net efficiency is

$$\epsilon_{trig} = 1 - (1 - \epsilon_1)(1 - \epsilon_2). \tag{4.4}$$

Note that for $e\mu$ signatures, $\epsilon_2 = 0$ and the equation still holds.

The efficiency is re-calculated for each electron with $par(1)$ and $par(2)$ increased by their respective errors (for both Level 1 and Level 2 turn-ons), and the difference with the nominal calculation is added in quadrature with an assumed 1% error on the plateau to yield the final error on the electron trigger efficiency. This error will vary from 1% at plateau, to larger values near the steep portion of the turn-on. The

[7] The minimum of 15 GeV reflects the minimum offline $E_T$ under consideration.

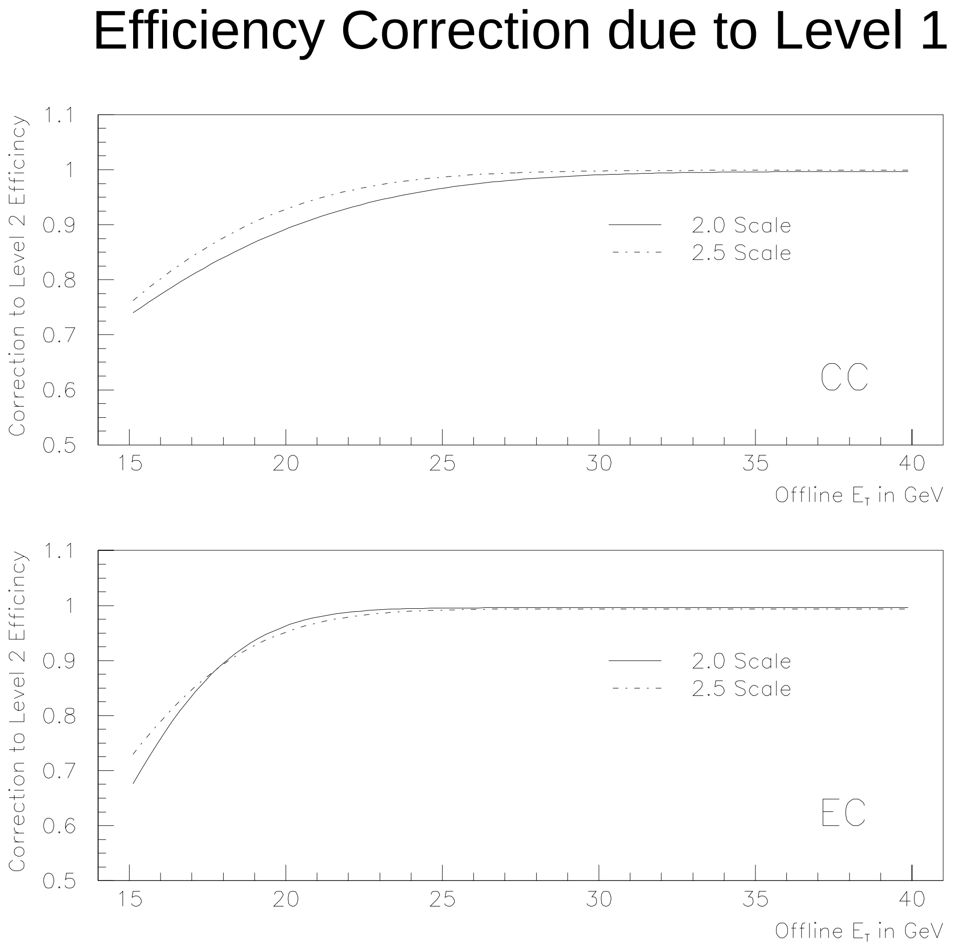


Figure 4.15: The two determinations of the correction to the Level 2 efficiency from scales 2.0 and 2.5.

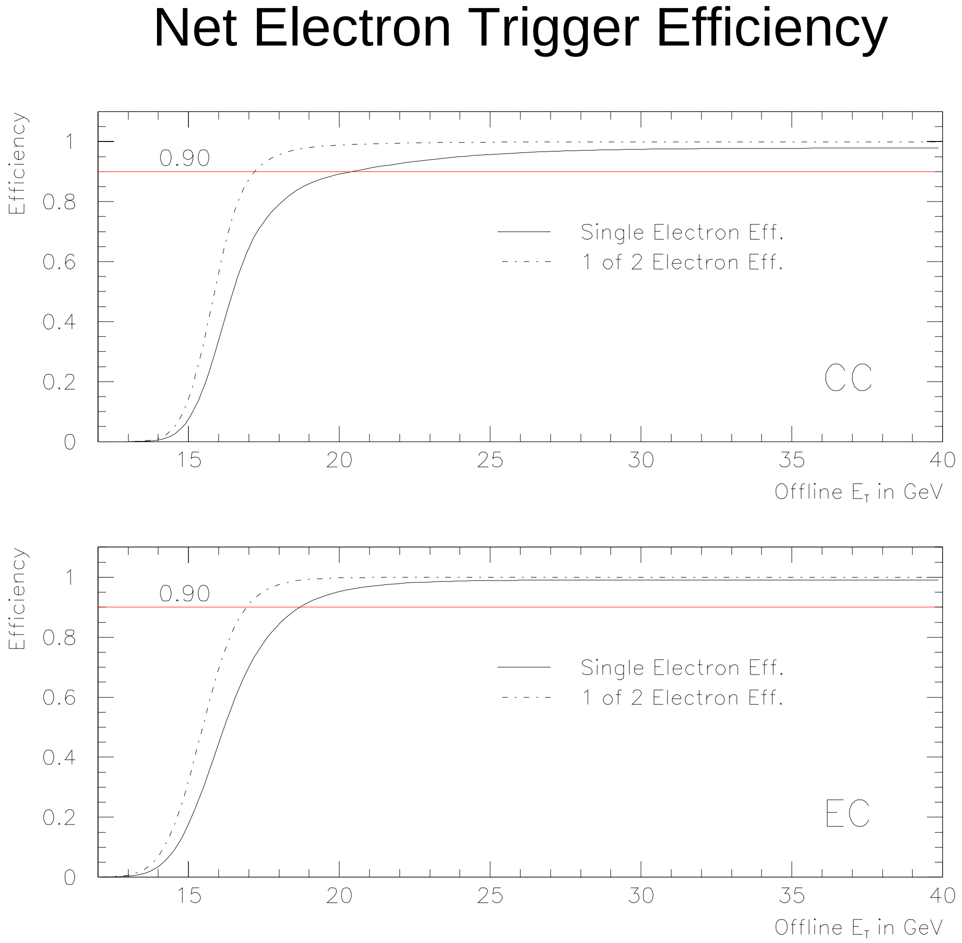


Figure 4.16: Net electron trigger efficiency as a function of offline $E_T$.

error on the electron trigger efficiency is

$$\delta\epsilon_{trig} = \delta\epsilon_1(1 - \epsilon_2) + \delta\epsilon_2(1 - \epsilon_1).$$

Again, the equation holds for the $e\mu$ signature ($\delta\epsilon_2 = 0$). Figure 4.16 also shows a plot of the efficiency for either of two equal $E_T$ electrons to pass the single electron requirement. This curve, with its 90% efficiency at 17 GeV for the CC, determined the leading electron $E_T$ cut.

### Corrections from Jet Turn-on

The complete trigger term for ELE_JET_HIGH is

EM(1,12,<2.6)EM(1,2.5)JT(2,5,<2.0)JT(1,3)

at Level 1, and

EM(1,15,<2.5)JT(2,10,<2.5)MS(14)

at Level 2. The trigger and filter terms describe the required number of objects, an $E_T$ cut, and are sometimes followed by a cut on detector $\eta$. There are thus several ingredients missing from the trigger efficiency

- The turn-on curve for offline jets given the trigger requirements.
- The probability that an offline electron that triggered the EM requirement also triggered the jet requirement. (A triggered electron almost always fires jet terms of lower $E_T$ in the same $\eta$ region.)

- The probability that non-triggered offline electron above 15 GeV $E_T$ fired the jet term. (This term is required to calculate combinatorics of the jet term in the two electron case.)

- The efficiency of the $\not\!\!E_T$ Level 2 term (this is discussed in the next section).

Previous studies of jet triggers computed efficiency as a function of jet multiplicity for a given $E_T$ cut on the jets.[29] This method assumes that a data sample is readily available that will mimic the signal jet spectrum, as $W$ bosons mimic the top signature. In this search, the jet spectrum varies model to model so a full turn-on curve must be constructed. The multiplicity method will serve only as a cross check against the previous studies.

To obtain jets in a similar environment to the signal, a sample of $W$ boson events is obtained from the EM1_EISTRKCC_MS trigger. Standard electron identification cuts are applied, as well as requiring the electron cluster be associated with a primary vertex within 60 cm of $z = 0$. Finally, the electron $E_T$ and $\not\!\!E_T$ are required above 25 GeV. Two samples are selected, with jet $E_T$ above 15 or 20 GeV. For the 15 GeV sample, events with a single jet are used to compute the turn-on curve as a function of offline $E_T$. The 15 GeV sample is produced twice: once with CAFIX and once without. The comparison of the turn-on for uncorrected and corrected jets appears in Figure 4.17.

The multiplicity study for the top search used uncorrected jet $E_T$ and was performed on pre-DØFix data. The method used in that study was repeated and the

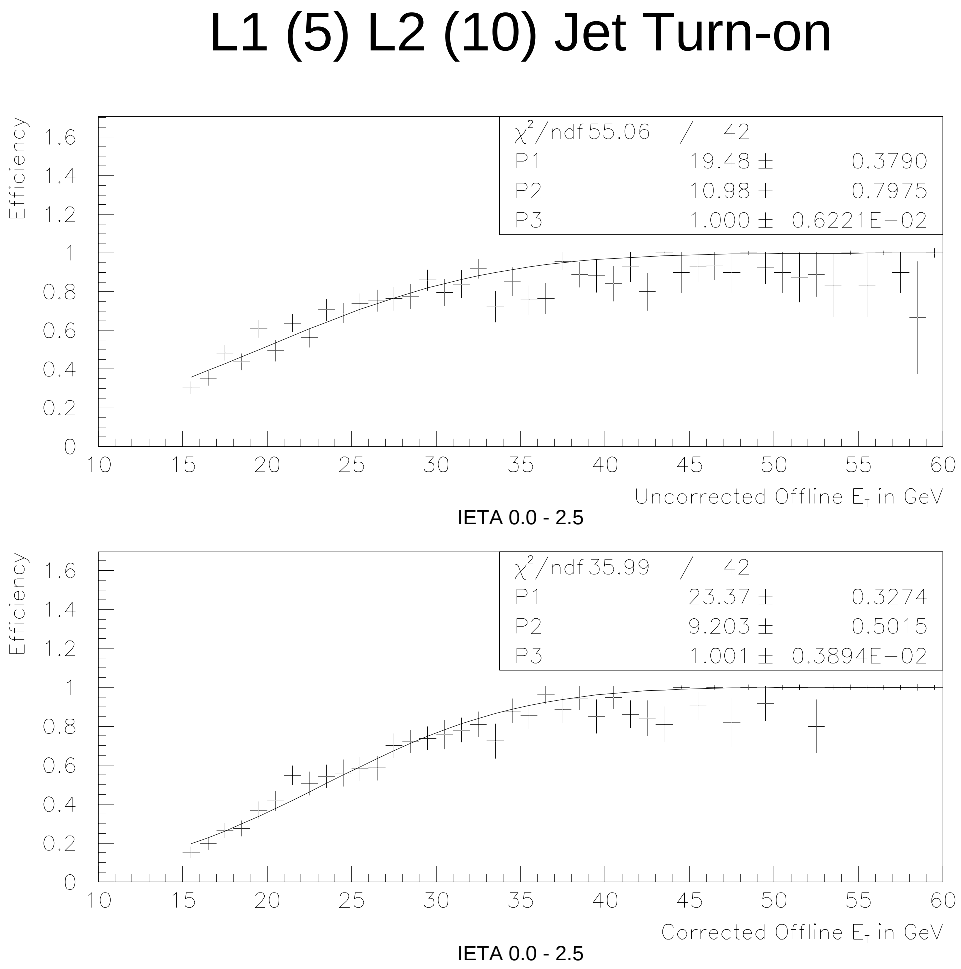


Figure 4.17: Net trigger turn-on for L1JT(5) and L2JT(10) as a function of offline $E_T$. The upper plot is the turn-on for uncorrected jet $E_T$ and the lower plot is for the fully corrected jet $E_T$.

Table 4.5: Comparison of $\not\!\!E_T$trigger term efficiency as a function of jet multiplicity.

| | $E_T > 15$ GeV | | $E_T > 20$ GeV | |
|---|---|---|---|---|
| $N_{jets}$ | top | new | top | new |
| 1 | $60.2 \pm 0.6$ | $63.1 \pm 1.0$ | $77.5 \pm 0.6$ | $77.0 \pm 1.0$ |
| 2 | $84.4 \pm 1.0$ | $85.3 \pm 1.6$ | $95.1 \pm 0.8$ | $92.5 \pm 1.6$ |
| 3 | $94.8 \pm 1.4$ | $95.3 \pm 2.1$ | $99.2 \pm 0.8$ | $100.0 \pm 2.0$ |
| 4 | $96.5 \pm 2.4$ | $100.0 \pm 5.0$ | $100.0 \pm 4.6$ | $100.0 \pm 2.5$ |

comparison with the current data sample is shown in Table 4.5.[8]

The EM and jet triggers overlap due to the absence of a maximum electromagnetic fraction cut for online jets. Ignoring this overlap would significantly underestimate jet efficiency. For example, an event with two high $E_T$ electrons and $\not\!\!E_T$ has a high probability of firing the ELE_JET_HIGH trigger even without additional jets. To determine this probability, a sample of $Z$ boson events is obtained from the EM2_EIS_ELE trigger including the standard electron identification on both electrons above 15 GeV $E_T$, vertex confirmation from one of the electrons, the vertex fiducial cut of 60 cm, and the invariant mass of the two electrons required in the interval (70, 104) GeV. Two sub-samples are obtained: the online electron sample, where an electron passes the ELE_JET_HIGH EM requirements, and the offline electron sample, where this requirement is ignored. The fraction of electrons passing the ELE_JET_HIGH jet requirements for each sample is obtained as a function of $\eta_d$ in two regions: $|\eta_d| \leq 1.2$ and $1.5 \leq |\eta_d| < 2.0$. Electrons above 2.0, due to their narrow shower profile, never pass Level 1 JT($\eta < 2.0$). The distributions are fit to a constant and the results are presented in Table 4.6.

[8] This comparison is done as a cross check on the data sample.

Table 4.6: Probability online or offline electrons pass the ELE_JET_HIGH jet requirements. The errors are the error from the fit.

| $\|\eta_d\|$ | Online (%) | Offline (%) |
|---|---|---|
| $0.0 - 1.2$ | $99.4 \pm 0.4$ | $95.1 \pm 0.7$ |
| $1.5 - 2.0$ | $99.8 \pm 0.8$ | $97.3 \pm 1.3$ |

### Correction from Level 2 $\not\!\!E_T$

Another possible correction to the trigger efficiency is the 14 GeV Level 2 $\not\!\!E_T$ requirement. The top analysis showed that this Level 2 trigger term is fully efficient at 20 GeV offline $\not\!\!E_T$. Since the data samples differed due to DØFix, the study was repeated with available statistics to confirm consistency with the top analysis. The data sample was obtained from the ELE_1_MON trigger, with standard electron identification above 25 GeV $E_T$, and vertex fiducial and confirmation cuts. The electron was required to pass the ELE_JET_HIGH online requirements, and the sample divided into $W + 1$ or more jets ($E_T^j > 20$ GeV) and $W + 2$ or more jets. The daughter sample required that Level 2 $\not\!\!E_T$ be above 14 GeV. Figure 4.18 shows that the offline 20 GeV cut is fully efficient for the two jet sample.

### Overall Trigger Efficiency

With all of the turn-ons for the trigger terms in place, the final trigger efficiency is determined. Since the trigger is based on electron and jet showers deposited in the calorimeter ($\not\!\!E_T$ is 100% efficient), there are 2 terms to be considered, $EM$ and $JT$. $EM$ is determined from a logical OR of the two leading electron $E_T$s and the net

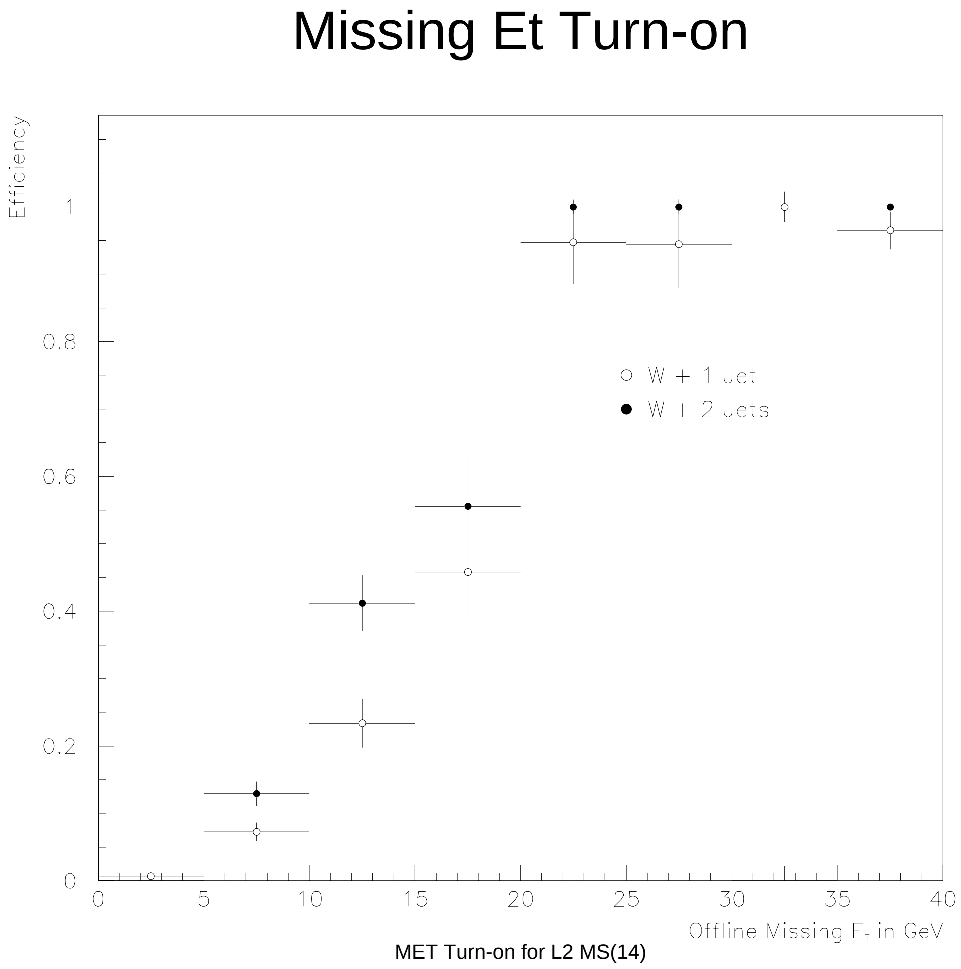


Figure 4.18: Efficiency of the Level 2 $\not{E}_T$ 14 GeV term from $W$ boson events.

final electron efficiency curves (Equation 4.4).

$JT$ is determined from the probability that the collection of electrons and jets in the event passed two jet terms (each term includes Level 1 and Level 2). FMCØ examines the two leading electrons and the five leading jets to determine trigger probabilities $p_{1-7}$:

$$p_1 \equiv e_{trig}, \quad p_2 \equiv e_{off}, \quad \text{and} \quad p_{3-7} \equiv j_{1-5}$$

where $e_{trig}$ is the probability that the trigger electron fired the jet term, $e_{off}$ is the probability that a non-trigger electron, if present, fired the jet term, and $j_{1-5}$ are the probabilities the the five leading jets fired the jet term (for less than five jets, the trailing probabilities are zero). From these seven probabilities, the probability that two jet terms are satisfied can be determined. This is most easily evaluated by considering the probability that six terms didn't fire

$$P(2 \text{ of } 7) = 1 - \overline{P(6 \text{ of } 7)}$$

and

$$\overline{P(6 \text{ of } 7)} = P(0 \text{ of } 7) + P(1 \text{ of } 7)$$

so

$$JT = P(2 \text{ of } 7) = 1 - \prod_{i=1}^{7}(1 - p_i) - \sum_{j=1}^{7} p_j \prod_{i \neq j}(1 - p_i).$$

The final probability the the trigger fired is now determined. TSIM returns this as a weight to assign to the event

$$w^{\mathrm{TSIM}} = EM \times JT.$$

The error on $w^{\mathrm{TSIM}}$ is the quadrature sum of the error on $EM$ and the error on $JT$. The error on $JT$ is determined by varying all the parameters of the jet terms by one sigma low and high, repeating the $JT$ calculation in each case, and taking half of the difference as the error.

### 4.5.2 MU_JET_XXX Parameterization

The $\mu\mu$ signature uses a set of triggers known collectively as MU_JET_XXX. This set is the logical OR of the triggers MU_JET_HIGH, MU_JET_CENT, MU_JET_CAL, and MU_JET_CENCAL. The parameterization was done for the top cross section analysis[29], and pre-existing software that calculates the trigger probability was employed.

## 4.6 RSIM: Simulating Standard Event Reconstruction

As with the trigger efficiency, the major consideration in the object identification is lepton identification. The electron efficiency is obtained in Chapter 4. The net lepton efficiency is defined as the product of the required lepton efficiencies. For example, in an electron-muon channel, only the efficiency of the leading electron and the leading muon are considered.

The muon efficiency is taken from QSIM_PARAM_MUO which returns an efficiency based on $\eta$, $\phi$, $z$, and $p_t$. The fractional error is 7%. The final efficiency is weighted by pre-zap and post-zap fractional luminosities of 51.91% and 48.09%, respectively.[9]

The net event efficiency is the lepton efficiency times the jet efficiency. The jet efficiency includes combinatorics; the event is analyzed to determine the probability that each jet passes each required jet term in the kinematic channel (KC): $P(i,j)$ for jet $i$ passing KC jet threshold $j$. The probability is taken from a table of jet efficiency as a function of $E_T$ and $\eta_d$, and multiplied by a step function at the kinematic threshold of requirement $j$ (jets above threshold have the normal jet ID efficiency, and jets below threshold have probability zero). The jet probability is based on all possible combinations of $n$ jets passing $m$ requirements, given $P(i,j)$ with $i = 1,n$ and $j = 1,m$. The jet probability is multiplied by the lepton efficiency to produce the net RSIM event weight, $w^{\mathrm{RSIM}}$.

The calculation is repeated with all turn-on parameters varied by ±1 sigma. The difference between the high and low values is taken as the error on the weight, $\delta w^{\mathrm{RSIM}}$.

[9] Outgassing of an organic binding agent in the muon cathode boards deposited a insulating film on the sense wires in the early part of Run 1b (colloquially known as the "green slime problem"). The coating was removed in the early part of 1995 by subjecting the wires to a short, high current impulse. The process is known as "zapping" the chambers.

## 4.7 SSIM: Analysis Simulation

SSIM keeps track of the sum of the weights for each event for each kinematic channel. For each model, there are six values recorded per each kinematic channel:

$BR(1)$ The fraction of unsmeared events passing a kinematic channel.

$BR(2)$ The fraction of smeared events passing a kinematic channel.

$BR(3)$ The sum of the weights returned by TSIM.

$BR(4)$ The error on $BR(3)$.

$BR(5)$ The visible fraction of the total cross section

$$BR(5) = W = \frac{1}{N}\sum_{i=1}^{N} w_i^{\text{TSIM}} w_i^{\text{RSIM}}$$

with $N$ the number of events produced. The weights from TSIM and RSIM are zero for smeared events not passing a kinematic channel. The number of events used to calculate $W$ is $N \cdot BR(2)$.

$BR(6)$ The error on $BR(5)$.

The bookkeeping of event weights is described in detail in Appendix A. Discussed below is determination of $BR(6)$ and $BR(4)$.

The net error for the event is set by adding the relative errors from RSIM and TSIM in quadrature. Thus, for event $i$, the error on the event weight is

$$\delta w_i = w_i \sqrt{\left(\frac{\delta w_i^{\mathrm{TSIM}}}{w_i^{\mathrm{TSIM}}}\right)^2 + \left(\frac{\delta w_i^{\mathrm{RSIM}}}{w_i^{\mathrm{RSIM}}}\right)^2}$$

Since $w_i = w_i^{\mathrm{TSIM}} w_i^{\mathrm{RSIM}}$, the error is

$$\delta w_i = \sqrt{(\delta w_i^{\mathrm{TSIM}} w_i^{\mathrm{RSIM}})^2 + (\delta w_i^{\mathrm{RSIM}} w_i^{\mathrm{TSIM}})^2}$$

Because the identification efficiencies are 100% correlated event to event, the overall error on the returned event weights is set to the average error on each signal event.

$$\delta W = \frac{1}{N} \sum \delta w_i$$

where $N$ is the number of events generated. This total error replaces the calculation in Appendix A to determine $BR(6)$. The TSIM error, also 100% correlated, is calculated

$$\delta W^{\mathrm{TSIM}} = \frac{1}{N} \sum \delta w_i^{\mathrm{TSIM}}$$

and this value replaces the $BR(4)$ determination in Appendix A.

# Chapter 5

# Analysis

## 5.1 Background Calculation

The various kinematic channels are divided into three sets of signatures: electron-electron, electron-muon, and muon-muon. The calculations are the same for any kinematic channel within these signatures, as only the thresholds change. Details of each background calculation are presented, and results are summarized in the next chapter.

### 5.1.1 Electron-Electron Signatures

There are five backgrounds to consider for $ee$ signatures: $Z \rightarrow ee + 2jets$, $Z \rightarrow \tau\tau + 2jets$, $W \rightarrow e\nu_e + 3jets$ where one of the jets fakes an electron, $t\bar{t} \rightarrow ee\nu_e\nu_e + 2jets$, and QCD multi-jet events where two of the jets fake electrons. For Z decaying to electrons, no real $\not\!\!E_T$ is present and the 2 jets fake a $\not\!\!E_T$ signal through mismeasurement or fluctuations.

### Background from $Z/\gamma* \rightarrow ee + 2jets$

Two kinds of checks were performed on the FMCØ generation of $Z$ boson events to check that 1) the returned cross sections from PYTHIA were correct, and 2) the jet spectra produced by PYTHIA and LUJETS were correct.

First, the Z production cross section was checked. Twenty-five thousand[1] $Z$ events were produced in the s- ($f_i\overline{f_i} \rightarrow (\gamma*/Z)$) and t-channel ($f_ig \rightarrow f_i(\gamma*/Z)$).[2] The decay of the $Z$ boson was limited to the electron channel at the generator level. PYTHIA quoted a cross section of 0.2382 nb, in good agreement with the DØ measured value of 0.235 ± 0.02 nb. In addition, the total cross section was checked. The two channels were generated separately with all decays turned on: PYTHIA quoted 5.306 nb for the s-channel and 1.732 nb for the t-channel, for a total of 7.029 nb. Assuming a SM branching ratio of 3.366%, the corresponding DØ measured value is $6.98 \pm 0.6$ nb. As an additional check on the $Z/\gamma*$ cross section, 8 million events were produced with the $Z$ boson decay limited to the electron channel. This sample is used to predict the number of events in the Run 1b $ee$ data sample. Using BR(5) from the $Z$ kinematic channel, 4780 ± 212 events are predicted to be found in the inclusive $ee$ sample of 97 $\mathrm{pb}^{-1}$(error from BR(6)). Requiring the same kinematics on the data sample, 4663 events are found. The PYTHIA cross sections

[1] The number of events to produced for a sample was selected such that statistical errors were smaller than systematic errors.

[2] The minimum $p_T$ for the $2 \rightarrow 2$ interaction, `CKIN(3)`, was set to 10 GeV/c for all $Z$ boson studies. In this first study, $\gamma*$ production was turned off.

Table 5.1: Kinematic channel definitions for $Z$ boson production. A negative threshold means that the object is required absent from the event.

| Sample | name | $e$ req. | $j1$ | $j2$ | $j3$ | Comment |
|---|---|---|---|---|---|---|
| $Z$ | $Z$ | $2 > 25$ | - | - | - | Write Key |
| $Z + 1j$ | $Zj15$ | $2 > 25$ | 15 | - | - | |
| $Z + 2j$ | $Zjj15$ | $2 > 25$ | 15 | 15 | - | |
| $Z + 1j$ | $Zj20$ | $2 > 25$ | 20 | - | - | |
| $Z + 2j$ | $Zjj20$ | $2 > 25$ | 20 | 20 | - | |
| $Z$ | $Zx$ | $2 > 25$ | -15 | - | - | |
| $Z + 1j$ | $Zj15x$ | $2 > 25$ | 15 | -15 | - | |
| $Z + 2j$ | $Zjj15x$ | $2 > 25$ | 15 | 15 | -15 | |
| $Z + 1j$ | $Zj20x$ | $2 > 25$ | 20 | -20 | - | |
| $Z + 2j$ | $Zjj20x$ | $2 > 25$ | 20 | 20 | -20 | |

Table 5.2: Comparison of jet multiplicity in Z events with a cut on $E_T^j$.

| Njets | MC% $E_T^j > 15$ | Data% $E_T^j > 15$ | MC% $E_T^j > 20$ | Data% $E_T^j > 20$ |
|---|---|---|---|---|
| $\geq 0$ | 100.0 | 100.0 | 100.0 | 100.0 |
| $= 0$ | $85.0 \pm 0.9$ | $85.6 \pm 1.9$ | $89.0 \pm 0.9$ | $91.1 \pm 2.0$ |
| $= 1$ | $13.25 \pm 0.33$ | $12.0 \pm 0.56$ | $9.3 \pm 0.3$ | $7.7 \pm 0.4$ |
| $= 2$ | $2.275 \pm 0.13$ | $2.14 \pm 0.2$ | $1.18 \pm 0.10$ | $1.06 \pm 0.16$ |

are therefore correct within the error associated with TRSIM. The PYTHIA cross sections were used in various cross checks of the jet spectra where only t-channel events, the dominant source of 2 jet events, were generated.

To check that the jet spectra are correct, 5 million events were generated with the above PYTHIA setup and kinematic channels shown in Table 5.1. Using the exclusive channels, those with suffix "$x$", the ratio of events with a specific number of jets can be predicted: $BR(5)Zj15x/BR(5)Z$ in the $Z + 1jet > 15$ channel, for example. This is shown and compared with data in Table 5.2.

The statistical error is taken as the square root of the number of unweighted passing signal events. The systematic error is from $BR(6)$.

To show that $\gamma*$ production can be neglected relative to $Z$ boson production, 13 million events were generated with $\gamma*/Z$ turned on.[3] Comparison of events passing the $eejjNz$ channel with $eejjz$ showed that $Z$ boson production is a factor of 15 larger in contribution to background (2 events versus 29 events).

Since $\gamma*$ can be ignored and the PYTHIA jet spectrum is in agreement with data, only $Z$ bosons are produced to estimate this background, and the production cross section is taken as the DØ measured value. Five million events were produced and the estimated background for various kinematic channels is summarized in Table 5.3.

As a final cross check on $Z + 2jet$ production, events were produced in the t-channel. To compensate for the lost ISR jets from the s-channel, non-perturbative $Z+$ gluon production is turned on ($f\overline{f} \rightarrow gZ/\gamma*$), as the PYTHIA manual recommends for jet studies. Five million events were produced; the results were consistent with the combined s- and t-channel $Z$ boson production.

### Background from $Z \rightarrow \tau\tau$ in the $ee$ channel

The $Z \rightarrow \tau\tau + 2jets$ background is straightforward to calculate using FMCØ. The default kinematic channels are analyzed and the cross section used is the DØ measured value for $Z \rightarrow ee$ inclusive. Five million events were generated and the resulting

[3] The $z/\gamma*$ cross section is 42 nb, or 4 million events for 100 $\text{pb}^{-1}$. To simulate this well, with 10 times the data statistics, would require 10 days of CPU on the fastest available processor.

Table 5.3: Background in the $ee$ signature from $Z \rightarrow ee + 2jets$. Production cross section was set at the DØ measured electron value of $0.235 \pm 0.02$ nb. The first error is due to the statistics of the MC, and the second error is due to the uncertainty in the cross section, and the final is that returned from TRSIM.

| Channel | Num. in 107.6 $\text{pb}^{-1}$ |
|---|---|
| $eejjNz$ | $0.49 \pm 0.050 \pm 0.042 \pm 0.102$ |
| $eejjz$ | $6.16 \pm 0.173 \pm 0.524 \pm 1.254$ |
| $eejjjz$ | $1.89 \pm 0.096 \pm 0.160 \pm 0.487$ |
| $eeHjjz$ | $4.74 \pm 0.152 \pm 0.404 \pm 1.020$ |
| $eejjHz$ | $1.39 \pm 0.082 \pm 0.118 \pm 0.296$ |
| $eeHjjHz$ | $0.98 \pm 0.070 \pm 0.083 \pm 0.226$ |
| $eeHjjjz$ | $1.61 \pm 0.089 \pm 0.137 \pm 0.428$ |
| $eejjjHz$ | $0.37 \pm 0.043 \pm 0.032 \pm 0.105$ |
| $eeHjjjHz$ | $0.33 \pm 0.041 \pm 0.028 \pm 0.096$ |
| $eejjHNz$ | $0.49 \pm 0.050 \pm 0.042 \pm 0.102$ |
| $eejjjNz$ | $0.13 \pm 0.026 \pm 0.011 \pm 0.031$ |
| $eeHjjNz$ | $0.37 \pm 0.044 \pm 0.031 \pm 0.080$ |
| $eeHjjHNz$ | $0.07 \pm 0.019 \pm 0.006 \pm 0.024$ |
| $eeHjjjNz$ | $0.10 \pm 0.023 \pm 0.009 \pm 0.025$ |
| $eejjjHNz$ | $0.02 \pm 0.011 \pm 0.002 \pm 0.004$ |
| $eeHjjjHNz$ | $0.02 \pm 0.009 \pm 0.001 \pm 0.004$ |

prediction is shown in Table 5.4.

### Background from top in the $ee$ channel

One hundred thousand top events were generated with FMCØ. $W$ boson decays were limited to the electron channel. Using the DØ measured top cross section of $5.5 \pm 1.8$ pb and the SM branching ratio of 10.8% for $W \rightarrow e\nu_e$, the production cross section for $t\bar{t} \rightarrow ee + jets$ is set to $0.064 \pm 0.021$ pb. Table 5.5 summarizes the top backgrounds in the various $ee$ signatures.

One might worry about the contribution due to one $W$ boson decaying to an

Table 5.4: Background in the $ee$ signature from $Z \to \tau\tau$. Production cross section was set at the DØ measured electron value of 0.235 ± 0.02 nb. The first error is due to the statistics of the MC, the second is due to the uncertainty in the $Z$ boson production cross section, and the last error is that returned from TRSIM.

| Channel | Num. in 107.6 $\text{pb}^{-1}$ |
|---|---|
| $eejjNz$ | $0.63 \pm 0.040 \pm 0.053 \pm 0.106$ |
| $eejjz$ | $0.64 \pm 0.040 \pm 0.054 \pm 0.107$ |
| $eejjjz$ | $0.14 \pm 0.019 \pm 0.012 \pm 0.032$ |
| $eeHjjz$ | $0.34 \pm 0.030 \pm 0.029 \pm 0.062$ |
| $eejjHz$ | $0.32 \pm 0.029 \pm 0.028 \pm 0.053$ |
| $eeHjjHz$ | $0.21 \pm 0.023 \pm 0.018 \pm 0.037$ |
| $eeHjjjz$ | $0.10 \pm 0.016 \pm 0.009 \pm 0.023$ |
| $eejjjHz$ | $0.08 \pm 0.015 \pm 0.007 \pm 0.017$ |
| $eeHjjjHz$ | $0.07 \pm 0.013 \pm 0.006 \pm 0.013$ |
| $eejjHNz$ | $0.63 \pm 0.040 \pm 0.053 \pm 0.106$ |
| $eejjjNz$ | $0.14 \pm 0.018 \pm 0.012 \pm 0.032$ |
| $eeHjjNz$ | $0.34 \pm 0.029 \pm 0.029 \pm 0.061$ |
| $eeHjjHNz$ | $0.21 \pm 0.023 \pm 0.018 \pm 0.037$ |
| $eeHjjjNz$ | $0.10 \pm 0.015 \pm 0.008 \pm 0.023$ |
| $eejjjHNz$ | $0.08 \pm 0.015 \pm 0.007 \pm 0.017$ |
| $eeHjjjHNz$ | $0.07 \pm 0.013 \pm 0.006 \pm 0.013$ |

electron, and the chain $b \to e$ producing a second isolated electron. As a cross check, approximately 250 thousand top events were produced where the $W$ boson was allowed to decay in all channels. No difference was found in the predicted background for the $ee$ channel.

### Background from QCD and $W$ bosons in the $ee$ channel

The QCD multi-jet and $W + 3jet$ backgrounds are combined and calculated using a data-based method. These backgrounds are combined into one estimate since one cannot separate the events in the data. The fake probability for jets imitating an

Table 5.5: Background in the $ee$ signature from $t\bar{t} \rightarrow ee + jets$. Production cross section was set at the DØ measured electron value of $0.064 \pm 0.021$ pb. The first error quoted is from statistics, the second error quoted is due to the uncertainty in the top production cross section, and the final error is from TRSIM.

| Channel | Num. in 107.6 pb$^{-1}$ |
|---|---|
| $eejjNz$ | $1.12 \pm 0.013 \pm 0.368 \pm 0.246$ |
| $eejjz$ | $1.38 \pm 0.014 \pm 0.454 \pm 0.304$ |
| $eejjjz$ | $0.45 \pm 0.008 \pm 0.149 \pm 0.128$ |
| $eeHjjz$ | $1.21 \pm 0.013 \pm 0.400 \pm 0.279$ |
| $eejjHz$ | $1.24 \pm 0.014 \pm 0.411 \pm 0.276$ |
| $eeHjjHz$ | $1.10 \pm 0.013 \pm 0.364 \pm 0.255$ |
| $eeHjjjz$ | $0.41 \pm 0.008 \pm 0.134 \pm 0.119$ |
| $eejjjHz$ | $0.41 \pm 0.008 \pm 0.135 \pm 0.116$ |
| $eeHjjjHz$ | $0.37 \pm 0.007 \pm 0.123 \pm 0.109$ |
| $eejjHNz$ | $1.12 \pm 0.013 \pm 0.368 \pm 0.246$ |
| $eejjjNz$ | $0.36 \pm 0.007 \pm 0.120 \pm 0.103$ |
| $eeHjjNz$ | $0.98 \pm 0.012 \pm 0.323 \pm 0.226$ |
| $eeHjjHNz$ | $0.89 \pm 0.012 \pm 0.295 \pm 0.206$ |
| $eeHjjjNz$ | $0.33 \pm 0.007 \pm 0.108 \pm 0.096$ |
| $eejjjHNz$ | $0.33 \pm 0.007 \pm 0.109 \pm 0.094$ |
| $eeHjjjHNz$ | $0.30 \pm 0.007 \pm 0.099 \pm 0.088$ |

electron is derived in Chapter 3. An electron + jets sample is used to calculate a weight per event based on all combinations of one of the jets faking a second electron

$$w_s = \sum_{i=1}^{Njets} f_i$$

where $f_i$ is either the CC or EC fake probability for a jet. For each combination, the jets in the event (excluding the one assumed to fake an electron) are required to pass the thresholds of the kinematic channel. These weights are summed to predict the number of $ee$ events expected.

Table 5.6: Background in the $ee$ signature from QCD and $W$ bosons. The first error quoted is that from the uncertainty in the electron fake rate, the second error quoted is from the statistics of the calculation, and the final error is due to the uncertainty in the energy scale.

| Channel | Num. in 107.6 $\text{pb}^{-1}$ |
|---|---|
| $eejjNz$ | $2.70 \pm 0.405 \pm 0.065 \pm 0.324$ |
| $eejjz$ | $2.70 \pm 0.405 \pm 0.065 \pm 0.324$ |
| $eejjjz$ | $0.66 \pm 0.099 \pm 0.034 \pm 0.079$ |
| $eeHjjz$ | $1.41 \pm 0.211 \pm 0.041 \pm 0.169$ |
| $eejjHz$ | $1.86 \pm 0.278 \pm 0.054 \pm 0.223$ |
| $eeHjjHz$ | $1.03 \pm 0.155 \pm 0.036 \pm 0.124$ |
| $eeHjjjz$ | $0.42 \pm 0.063 \pm 0.026 \pm 0.050$ |
| $eejjjHz$ | $0.44 \pm 0.066 \pm 0.027 \pm 0.053$ |
| $eeHjjjHz$ | $0.44 \pm 0.066 \pm 0.027 \pm 0.053$ |
| $eejjHNz$ | $1.86 \pm 0.278 \pm 0.054 \pm 0.223$ |
| $eejjjNz$ | $0.66 \pm 0.099 \pm 0.034 \pm 0.079$ |
| $eeHjjNz$ | $1.41 \pm 0.211 \pm 0.041 \pm 0.169$ |
| $eeHjjHNz$ | $1.03 \pm 0.155 \pm 0.036 \pm 0.124$ |
| $eeHjjjNz$ | $0.42 \pm 0.063 \pm 0.026 \pm 0.050$ |
| $eejjjHNz$ | $0.44 \pm 0.066 \pm 0.027 \pm 0.053$ |
| $eeHjjjHNz$ | $0.44 \pm 0.066 \pm 0.027 \pm 0.053$ |

The entire disk sample[4] of ELE_JET_HIGH micro-DSTs was streamed by requiring one PELC with $E_T > 12$ GeV. This sample is dominated by $W + 1jet$ events, but includes fake electrons from QCD multi-jet events. The total weight is calculated and scaled up to the luminosity of the total sample. The error on the fake probability is 10-15%; therefore, a 15% error is assigned to this background calculation.

To determine the error due to energy scale, the total weight calculated for the $eejjNz$ channel was done with CAFIX high and low corrections: this yielded a 13% difference in the predicted background. This error is used as the relative error on all

[4] The disk sample corresponds to a luminosity of 92 $\text{pb}^{-1}$. The remaining micro-DSTs are available on tape.

channels due to energy scale uncertainty. The results are summarized in Table 5.6.

### 5.1.2 Electron-Muon Signatures

There are four main sources of background for $e\mu$ signatures: $t\bar{t} \rightarrow e\mu + jets$, $Z \rightarrow \tau\tau + jets$, $W \rightarrow \mu + 3\,jets$ where one of the jets fakes an electron, and QCD multi-jet events that include heavy flavor decays to muons, and where one of the jets fakes an electron. All found muons are considered real and originating from the hard scattering vertex. Cosmic muons that coincide with the the hard scattering vertex are rare and are accounted in the QCD estimate.

#### Background from top production in the $e\mu$ channel

One hundred thousand top events were produced with FMCØ. The top mass was set at 175 GeV, and the $W$ boson decay was limited to either the electron or muon channel. Table 5.7 summarizes the results.

#### Background from $Z$ boson production in the $e\mu$ channel

Five million $Z \rightarrow \tau\tau$ events were produced with FMCØ. $Z$ boson decays were limited to the $\tau\tau$ channel. Table 5.8 summarizes the results.

#### Background from QCD and $W$ bosons in the $e\mu$ channel

The number of isolated muons originating from QCD interactions (dominantly heavy flavor decays, but cosmic muons are included by this method) and $W \rightarrow \mu + 3j$ are

Table 5.7: Background in the $e\mu$ signature from top production, $\sigma_{top} = 0.257 \pm 0.084$ pb. The errors are from statistics, $\delta\sigma_{top}$, and TRSIM.

| Channel | Num. in 107.6 $\text{pb}^{-1}$ |
|---|---|
| $emjj$ | $1.31 \pm 0.013 \pm 0.432 \pm 0.339$ |
| $emHjj$ | $1.15 \pm 0.012 \pm 0.381 \pm 0.315$ |
| $emjjH$ | $1.18 \pm 0.012 \pm 0.389 \pm 0.306$ |
| $emHjjH$ | $1.04 \pm 0.012 \pm 0.344 \pm 0.285$ |
| $emjjj$ | $0.33 \pm 0.007 \pm 0.109 \pm 0.109$ |
| $emHjjj$ | $0.30 \pm 0.006 \pm 0.097 \pm 0.102$ |
| $emjjjH$ | $0.30 \pm 0.006 \pm 0.098 \pm 0.098$ |
| $emHjjjH$ | $0.27 \pm 0.006 \pm 0.088 \pm 0.092$ |
| $e7mjj$ | $1.30 \pm 0.013 \pm 0.430 \pm 0.338$ |
| $e7mHjj$ | $1.15 \pm 0.012 \pm 0.379 \pm 0.313$ |
| $e7mjjH$ | $1.17 \pm 0.012 \pm 0.388 \pm 0.305$ |
| $e7mHjjH$ | $1.04 \pm 0.012 \pm 0.342 \pm 0.284$ |
| $e7mjjj$ | $0.33 \pm 0.007 \pm 0.108 \pm 0.108$ |
| $e7mHjjj$ | $0.29 \pm 0.006 \pm 0.097 \pm 0.101$ |
| $e7mjjjH$ | $0.30 \pm 0.006 \pm 0.098 \pm 0.097$ |
| $e7mHjjjH$ | $0.27 \pm 0.006 \pm 0.088 \pm 0.091$ |
| $eTmjj$ | $1.29 \pm 0.013 \pm 0.424 \pm 0.333$ |
| $eTmHjj$ | $1.13 \pm 0.012 \pm 0.374 \pm 0.309$ |
| $eTmjjH$ | $1.16 \pm 0.012 \pm 0.382 \pm 0.301$ |
| $eTmHjjH$ | $1.02 \pm 0.011 \pm 0.338 \pm 0.279$ |
| $eTmjjj$ | $0.32 \pm 0.007 \pm 0.107 \pm 0.106$ |
| $eTmHjjj$ | $0.29 \pm 0.006 \pm 0.095 \pm 0.099$ |
| $eTmjjjH$ | $0.29 \pm 0.006 \pm 0.096 \pm 0.096$ |
| $eTmHjjjH$ | $0.26 \pm 0.006 \pm 0.086 \pm 0.089$ |

determined with a data-based method. All muons are considered real if they pass the SOFT_MUON criteria from the PARTICLE_SELECT and MUJETS_MU_SELECT packages. The background from both sources is a real muon, measured missing $E_T$, and a jet faking an electron.

A weight is calculated based on the fake electron probability in a similar manner to that used for the in the $ee$ channel. The DØFix disk sample was processed and

Table 5.8: Background in the $e\mu$ signature from $Z \to \tau\tau$, $\sigma_Z = 0.235 \pm 0.020$ nb. The errors are that from statistics, $\delta\sigma_Z$, and TRSIM.

| Channel | Num. in 107.6 pb$^{-1}$ |
|---|---|
| $emjj$ | 0.56 ± 0.033 ± 0.048 ± 0.109 |
| $emHjj$ | 0.32 ± 0.026 ± 0.027 ± 0.068 |
| $emjjH$ | 0.31 ± 0.026 ± 0.026 ± 0.061 |
| $emHjjH$ | 0.20 ± 0.022 ± 0.017 ± 0.043 |
| $emjjj$ | 0.08 ± 0.013 ± 0.007 ± 0.023 |
| $emHjjj$ | 0.07 ± 0.013 ± 0.006 ± 0.019 |
| $emjjjH$ | 0.05 ± 0.011 ± 0.004 ± 0.015 |
| $emHjjjH$ | 0.04 ± 0.010 ± 0.004 ± 0.013 |
| $e7mjj$ | 0.45 ± 0.030 ± 0.038 ± 0.088 |
| $e7mHjj$ | 0.25 ± 0.023 ± 0.022 ± 0.055 |
| $e7mjjH$ | 0.26 ± 0.024 ± 0.022 ± 0.053 |
| $e7mHjjH$ | 0.18 ± 0.020 ± 0.015 ± 0.038 |
| $e7mjjj$ | 0.07 ± 0.012 ± 0.006 ± 0.020 |
| $e7mHjjj$ | 0.06 ± 0.012 ± 0.005 ± 0.017 |
| $e7mjjjH$ | 0.05 ± 0.010 ± 0.004 ± 0.013 |
| $e7mHjjjH$ | 0.04 ± 0.010 ± 0.003 ± 0.012 |
| $eTmjj$ | 0.36 ± 0.027 ± 0.031 ± 0.069 |
| $eTmHjj$ | 0.20 ± 0.021 ± 0.017 ± 0.044 |
| $eTmjjH$ | 0.20 ± 0.021 ± 0.017 ± 0.039 |
| $eTmHjjH$ | 0.13 ± 0.017 ± 0.011 ± 0.029 |
| $eTmjjj$ | 0.06 ± 0.011 ± 0.005 ± 0.017 |
| $eTmHjjj$ | 0.05 ± 0.011 ± 0.004 ± 0.014 |
| $eTmjjjH$ | 0.04 ± 0.010 ± 0.003 ± 0.011 |
| $eTmHjjjH$ | 0.03 ± 0.009 ± 0.003 ± 0.011 |

the results were scaled up to the total luminosity. The process was repeated with CAFIX high and low corrections to determine a systematic error from energy scale uncertainties. The results for $e\mu$ kinematic channels are given in Table 5.9.

Table 5.9: Background from QCD and W bosons in the $e\mu$ channel. The errors are from the electron fake rate (FR), statistics, and energy scale (ES).

| Channel | <events> (FR) (ES) (Stat) |
|---|---|
| $emjj$ | $4.54 \pm 0.681 \pm 0.194 \pm 0.545$ |
| $emHjj$ | $2.13 \pm 0.320 \pm 0.114 \pm 0.256$ |
| $emjjH$ | $1.43 \pm 0.215 \pm 0.111 \pm 0.172$ |
| $emHjjH$ | $0.77 \pm 0.115 \pm 0.069 \pm 0.092$ |
| $emjjj$ | $1.37 \pm 0.205 \pm 0.094 \pm 0.164$ |
| $emHjjj$ | $0.94 \pm 0.140 \pm 0.076 \pm 0.112$ |
| $emjjjH$ | $0.47 \pm 0.071 \pm 0.058 \pm 0.056$ |
| $emHjjjH$ | $0.40 \pm 0.061 \pm 0.055 \pm 0.048$ |
| $e7mjj$ | $1.66 \pm 0.249 \pm 0.118 \pm 0.199$ |
| $e7mHjj$ | $0.93 \pm 0.139 \pm 0.075 \pm 0.111$ |
| $e7mjjH$ | $0.60 \pm 0.089 \pm 0.072 \pm 0.072$ |
| $e7mHjjH$ | $0.39 \pm 0.059 \pm 0.049 \pm 0.047$ |
| $e7mjjj$ | $0.54 \pm 0.081 \pm 0.059 \pm 0.065$ |
| $e7mHjjj$ | $0.42 \pm 0.063 \pm 0.051 \pm 0.050$ |
| $e7mjjjH$ | $0.28 \pm 0.042 \pm 0.045 \pm 0.034$ |
| $e7mHjjjH$ | $0.26 \pm 0.039 \pm 0.045 \pm 0.031$ |
| $eTmjj$ | $0.90 \pm 0.135 \pm 0.086 \pm 0.108$ |
| $eTmHjj$ | $0.48 \pm 0.071 \pm 0.054 \pm 0.057$ |
| $eTmjjH$ | $0.35 \pm 0.052 \pm 0.055 \pm 0.042$ |
| $eTmHjjH$ | $0.22 \pm 0.032 \pm 0.037 \pm 0.026$ |
| $eTmjjj$ | $0.30 \pm 0.046 \pm 0.044 \pm 0.037$ |
| $eTmjjjH$ | $0.13 \pm 0.019 \pm 0.028 \pm 0.015$ |
| $eTmHjjj$ | $0.24 \pm 0.036 \pm 0.039 \pm 0.029$ |
| $eTmHjjjH$ | $0.12 \pm 0.019 \pm 0.031 \pm 0.015$ |

### 5.1.3 Muon-Muon Signatures.

The backgrounds for $\mu\mu$ are calculated exactly the same as for the $ee$ channel with the exception of the QCD/$W$ background. Jets don't fake muons as they do in the electron channel. Instead, a real muon is produced from a heavy flavor decay and there is a measurable probability that the muon appears as an isolated muon [29]. This probability is a function of the number of jets in the event. For 2 jet events,

Table 5.10: Background in the $\mu\mu$ signature from $t\bar{t} \rightarrow \mu\mu + jets$. Production cross section was set at the DØ measured muon value of $0.064 \pm 0.021$ pb. The first error quoted is statistical, the second error quoted is due to the uncertainty in the top production cross section, and the final error is that returned from TRSIM.

| Channel | Num. in 102.9 $\text{pb}^{-1}$ |
|---|---|
| $mmjj$ | $0.40 \pm 0.007 \pm 0.132 \pm 0.120$ |
| $mmHjj$ | $0.36 \pm 0.006 \pm 0.118 \pm 0.112$ |
| $mmjjH$ | $0.36 \pm 0.006 \pm 0.120 \pm 0.109$ |
| $mmHjjH$ | $0.32 \pm 0.006 \pm 0.106 \pm 0.102$ |
| $mmjjj$ | $0.08 \pm 0.003 \pm 0.025 \pm 0.029$ |
| $mmHjjj$ | $0.07 \pm 0.003 \pm 0.022 \pm 0.027$ |
| $mmjjjH$ | $0.07 \pm 0.003 \pm 0.023 \pm 0.026$ |
| $mmHjjjH$ | $0.06 \pm 0.003 \pm 0.020 \pm 0.024$ |
| $mmjjU$ | $0.32 \pm 0.006 \pm 0.107 \pm 0.098$ |
| $mmHjjU$ | $0.29 \pm 0.006 \pm 0.096 \pm 0.092$ |
| $mmjjjU$ | $0.06 \pm 0.003 \pm 0.020 \pm 0.022$ |
| $mmHjjjU$ | $0.05 \pm 0.002 \pm 0.018 \pm 0.021$ |

the probability is 0.068 ± .005 (0.171 ± .027) for CF (EF) muons, while for 3 jet events, the probability is $0.057 \pm 0.009$ ($0.097 \pm 0.041$). Events are selected from the MU_JET_XXX trigger stream. One isolated muon is required with $p_T > 10$ GeV/c, and a second muon with $p_T > 20$ GeV. The jet spectrum and missing $E_T$ requirements are imposed and each event obtains a weight based on the fake isolated muon probability. All of the backgrounds for the muon signatures are summarized in Tables 5.10 - 5.13.

Table 5.11: Background in the $\mu\mu$ signature from $Z \rightarrow \tau\tau$. Production cross section was set at the DØ measured electron value of $0.235 \pm 0.02$ nb. The first error is due to the statistics of the MC, the second is due to the uncertainty in the $Z$ boson production cross section, and the last error is that returned from TRSIM.

| Channel | Num. in 102.9 pb$^{-1}$ |
|---|---|
| $mmjj$ | $0.06 \pm 0.011 \pm 0.005 \pm 0.018$ |
| $mmHjj$ | $0.04 \pm 0.009 \pm 0.004 \pm 0.014$ |
| $mmjjH$ | $0.03 \pm 0.008 \pm 0.003 \pm 0.011$ |
| $mmHjjH$ | $0.03 \pm 0.007 \pm 0.002 \pm 0.009$ |
| $mmjjj$ | $0.00 \pm 0.002 \pm 0.000 \pm 0.002$ |
| $mmHjjj$ | $0.00 \pm 0.001 \pm 0.000 \pm 0.001$ |
| $mmjjjH$ | $0.00 \pm 0.001 \pm 0.000 \pm 0.002$ |
| $mmHjjjH$ | $0.00 \pm 0.001 \pm 0.000 \pm 0.001$ |
| $mmjjU$ | $0.03 \pm 0.008 \pm 0.002 \pm 0.009$ |
| $mmHjjU$ | $0.02 \pm 0.007 \pm 0.002 \pm 0.008$ |
| $mmjjjU$ | $0.00 \pm 0.000 \pm 0.000 \pm 0.000$ |
| $mmHjjjU$ | $0.00 \pm 0.000 \pm 0.000 \pm 0.000$ |

Table 5.12: Background in the $\mu\mu$ signature from $Z \rightarrow \mu\mu + 2jets$. Production cross section was set at the DØ measured electron value of $0.235 \pm 0.02$ nb. The first error is due to the statistics of the MC, and the second error is due to the uncertainty in the cross section, and the final is that returned from TRSIM.

| Channel | Num. in 102.9 pb$^{-1}$ |
|---|---|
| $mmjj$ | $0.90 \pm 0.057 \pm 0.077 \pm 0.206$ |
| $mmHjj$ | $0.64 \pm 0.049 \pm 0.054 \pm 0.158$ |
| $mmjjH$ | $0.23 \pm 0.030 \pm 0.020 \pm 0.054$ |
| $mmHjjH$ | $0.18 \pm 0.026 \pm 0.015 \pm 0.045$ |
| $mmjjj$ | $0.18 \pm 0.025 \pm 0.015 \pm 0.067$ |
| $mmHjjj$ | $0.16 \pm 0.024 \pm 0.014 \pm 0.062$ |
| $mmjjjH$ | $0.04 \pm 0.011 \pm 0.003 \pm 0.020$ |
| $mmHjjjH$ | $0.04 \pm 0.011 \pm 0.003 \pm 0.020$ |
| $mmjjU$ | $0.12 \pm 0.023 \pm 0.011 \pm 0.023$ |
| $mmHjjU$ | $0.09 \pm 0.020 \pm 0.007 \pm 0.017$ |
| $mmjjjU$ | $0.01 \pm 0.008 \pm 0.001 \pm 0.006$ |
| $mmHjjjU$ | $0.01 \pm 0.008 \pm 0.001 \pm 0.006$ |

Table 5.13: Background from QCD and W bosons in the $\mu\mu$ channel for various thresholds. The three errors quoted are the error on the muon fake rate (FR), error caused by energy scale (ES), and statistical error from the number of events used to determine the total weight.

| Channel | <events> (FR) (ES) (Stat) |
|---|---|
| $mmjj$ | $0.41 \pm 0.030 \pm 2.356 \pm 0.049$ |
| $mmHjj$ | $0.34 \pm 0.025 \pm 2.150 \pm 0.041$ |
| $mmjjH$ | $0.14 \pm 0.010 \pm 1.360 \pm 0.016$ |
| $mmHjjH$ | $0.14 \pm 0.010 \pm 1.360 \pm 0.016$ |
| $mmjjj$ | $0.11 \pm 0.018 \pm 0.850 \pm 0.014$ |
| $mmHjjj$ | $0.11 \pm 0.018 \pm 0.850 \pm 0.014$ |
| $mmjjjH$ | $0.06 \pm 0.018 \pm 0.425 \pm 0.007$ |
| $mmHjjjH$ | $0.06 \pm 0.009 \pm 0.601 \pm 0.007$ |
| $mmjjU$ | $0.07 \pm 0.005 \pm 0.962 \pm 0.008$ |
| $mmHjjU$ | $0.07 \pm 0.005 \pm 0.962 \pm 0.008$ |
| $mmjjjU$ | $0.06 \pm 0.009 \pm 0.601 \pm 0.007$ |
| $mmHjjjU$ | $0.06 \pm 0.009 \pm 0.601 \pm 0.007$ |

## 5.2 Setting Limits

### 5.2.1 Introduction

The process of setting a cross section limit using the background estimates, analyzes the data, and generates signal samples, is a non-trivial task filled with subtle systematic effects. In addition, it is sought to generate an arbitrary number of models, and be able to analyze a significant number of kinematic channels. The standard DØ prescription for setting limits is employed.[30] Two methods are used: one which will provide a model independent limit on the visible cross section, and another that provides a limit given a point in SUGRA space.

In what follows, $\mathcal{L}$ is luminosity, $b$ is the number of expected background events,

$\epsilon$ is the net signal acceptance which includes the branching ratio into the kinematic channel and all associated efficiencies, $k$ is the number of events found in the data sample, $\sigma$ is the signal production cross section, and $I$ represents all prior information in the problem. All of these quantities are assumed known and have associated errors $\delta X$. As an example, consider a single fictional kinematic channel, where $\mathcal{L} = 90.0 \pm 2.7$ pb, $b = 5.3 \pm 1.1$, and $k = 3$.

### 5.2.2 The Model Independent Limit

The standard recipe can be summarized as the calculation of the integral

$$P(\sigma|k,I) = \int_0^\infty d\mathcal{L} \int_0^1 d\epsilon \int_0^\infty db \frac{e^{-\mu}\mu^k}{k!\sigma_{max}} \mathcal{G}(\mathcal{L},\delta\mathcal{L})\mathcal{G}(\epsilon,\delta\epsilon)\mathcal{G}(b,\delta b)$$

where $\mathcal{G}$ is a standard gaussian, $\mu = \mathcal{L}\epsilon\sigma + b$, and $\sigma_{max}$ is chosen large enough that the likelihood function is negligible for $\sigma > \sigma_{max}$. This form neglects correlations between measured values.[5] $P(\sigma|k,I)$ is integrated to a confidence limit to obtain a cross section limit. For the 95% C.L.

$$0.95 = \frac{\int_0^{\sigma_{limit}} d\sigma \, P(\sigma|k,I)}{\int_0^{\sigma_{max}} d\sigma \, P(\sigma|k,I)}$$

and one solves for $\sigma_{limit}$.

The main difference among models and kinematic channels is $\epsilon$ and $\delta\epsilon$. As a starting point, 95% C.L. cross section limits are generated for a range of expected

[5] Strong correlations can improve the limit. This “simple” form is thus conservative.

$\epsilon$ values using a *fixed* relative error of 5%. A function can then be fit, $\sigma_{limit}(\epsilon)$, over the generated range. Using the assumption that $(\sigma\epsilon)_{limit} = constant$, one now has a model independent limit on the visible cross section. This assumption will be examined for validity.

For this and all subsequent limit setting, the heart of the calculation is the same. $P(\sigma|k, I)$ is integrated numerically using adaptive quadrature. The limits on the integration are $\pm 10\delta X$ in each dimension (truncated by 0 from below for all dimensions, and at 1 above for $\epsilon$). $P(\sigma|k, I)$ is calculated for 100 $\sigma_i$ points in the interval 0.0 to 2.5 pb, $I(\sigma_i)$, and normalization is determined by integrating using the trapezoidal rule ($\mathcal{N}$). The normalized 100 point running sum as a function of $\sigma_i$ is found

$$S(\sigma_i) = \frac{1}{2\mathcal{N}} \sum_{j=1}^{i} (I(\sigma_j) + I(\sigma_{j-1})) \cdot \Delta\sigma$$

where $\Delta\sigma$ is the step size, 0.025 pb here. To calculate a cross section limit, the two bins to either side of 0.95 are used in a linear interpolation. Since the sum is slowly varying over this region, the center of the bin method is used in this interpolation.

### $\sigma_{limit}$ as a function of $\epsilon$

To determine the function $\sigma_{limit}(\epsilon)$, $\delta\epsilon/\epsilon$ is fixed at 5% and the limit calculating procedure is run at 18 points, from $\epsilon = 5\%$ to $\epsilon = 90\%$, in steps of 5%. For this run, by trial and error $\sigma_{max}$ for each $\epsilon$ value was determined such that the unnormalized $P(\sigma_{max}|k, I) < 10^{-6}$ ($P(\sigma = 0|k, I) = 0.12$), and 250 points were sampled from each distribution in the range 0 to $\sigma_{max}$. The results are presented in Figure 5.1, where the fit has a $\chi^2$ consistent with zero. What this means is that $\sigma_{limit}\epsilon$ = constant, 0.0536 in this case. A given model can be excluded if $\sigma_s\epsilon_{fmc0} \geq (\epsilon\sigma)_{limit} = 0.0536$ pb, provided that the error on $\epsilon_{fmc0}$ is less than the set value used in determining the limit, and that enough signal statistics are generated to ignore the statistical error.

### Method to Determine the Limit

A fit over a range of $\epsilon$ is performed with $\delta\epsilon/\epsilon$ set at a reasonable value to obtain $(\sigma\epsilon)_{limit}$. From studies of the background, it is found that the relative error on BR(5) is consistently below 15%. Guided by the 10% error caused by energy scale uncertainty in the various QCD estimates, 10% is used as the maximum energy scale error.[6] The relative error on $\epsilon$ is therefore set at 20%. $(\sigma\epsilon)_{limit}$ is found for each kinematic channel and thus a set of model independent limits on the visible cross section is obtained.

### Choosing the Best Channel

To determine if a model is excluded, one must first choose the best final state before applying a limit.[7] The channel with the *best significance* (defined below) can be determined from the *expected* background events and the expected signal events.[31, 32, 33] The analysis of this best channel is then used to determine whether a model is excluded or not. One may also select the best kinematic channel within each

[6] The nature of the QCD $E_T$ spectrum is rapidly falling ( $E_T^{-6}$), while it is expected that the signal spectrum will be nominally flat in the region near threshold and therefore will have a considerably smaller energy scale error.

[7] Applying the limits from all kinematic channels essentially chooses those channels where the background fluctuated low in the data.

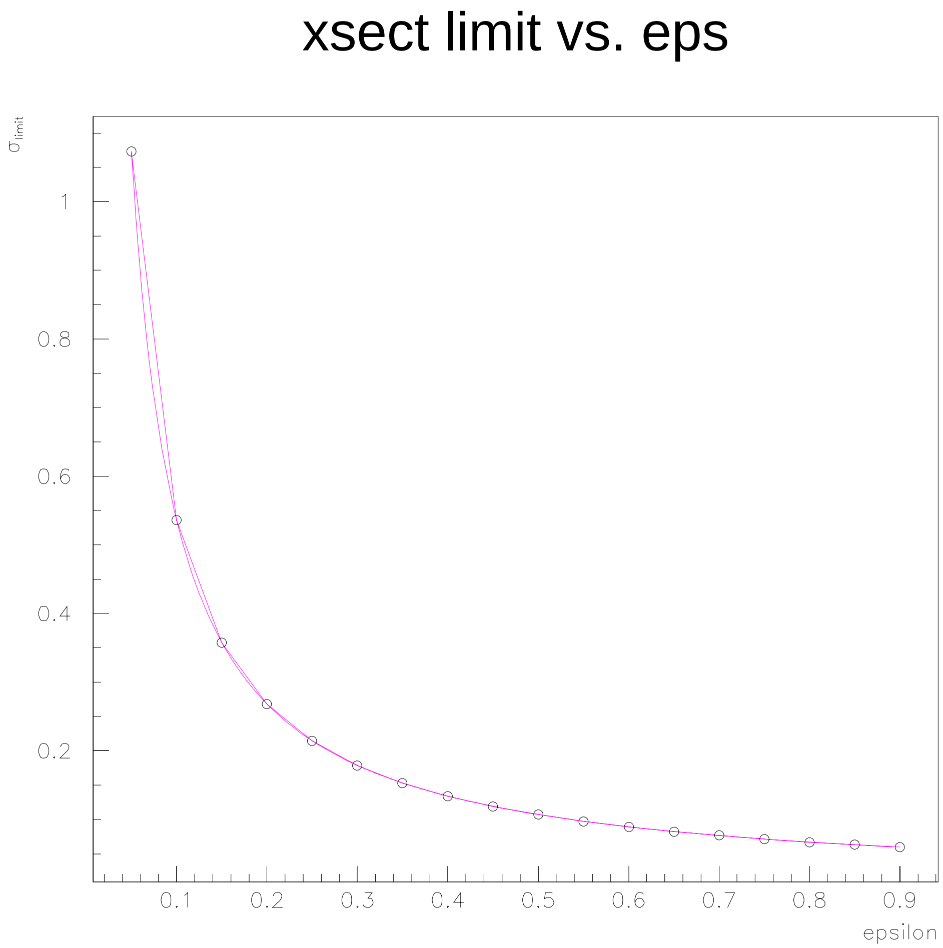


Figure 5.1: Obtained $\sigma_{limit}$ as a function of $\epsilon$. The fitting function is $y = p_1/x$. The obtained $\chi^2$ was consistent with zero.

signature and apply all three limits since the signatures are independent.

The probability that the background fluctuates to give $N$ or more observed data events is

$$F(N|b) = \sum_{k=N}^{\infty} P(k|b)$$

where $P(k|b)$ is the Poisson probability for seeing $k$ events when $b$ events are expected,

$$P(k|b) = \frac{b^k e^{-b}}{k!}$$

The fluctuation probability can be written as an equivalent number of Gaussian standard deviations, or *significance*, $S(N|b)$, as

$$F(N|b) = \frac{1}{\sqrt{2\pi}} \int_{S(N|b)}^{\infty} e^{-t^2/2}\, dt$$

For $s$ expected signal events, the *expected significance*, $\overline{S}(s,b)$ (the mean value that would be found if the experiment were to be repeated many times) is then

$$\overline{S}(s,b) = \sum_{N=0}^{\infty} P(N|s+b) \cdot S(N|b)$$

The kinematic channel where the expected significance is maximum is then defined as the channel with the best significance. This channel is the one where the current model would have the least chance of “hiding”. Note that the channel is chosen based on the signal and background estimates only, and no information is used from the data to bias this selection.

### 5.2.3 Model Dependent Limits

Using the procedure outlined above for setting a limit given a specific $\epsilon$ and $\delta\epsilon$, it is now possible to determine a model dependent limit for each point in SUGRA space. For each model, the kinematic channel with the best expected significance within each signature is found. $\sigma_{max}$ is obtained from the results of the model independent limits

$$\sigma_{max} = \frac{2 \times (\sigma\epsilon)_{limit}}{\epsilon_i}$$

where $\epsilon_i$ is BR(5) from kinematic channel $i$.[8] [9] The limit on the total cross section is calculated and the model is excluded appropriately.

Finally, the three best kinematic channels are combined and a limit is calculated and applied in a similar manner but with a few modifications. The maximum cross section is set

$$\sigma_{max} = \frac{2 \times \sum_i (\sigma\epsilon)^i_{limit}}{\sum_i \epsilon_i}$$

and the luminosity is set to the common luminosity of the $ee$ and $e\mu$ channels. To account for the lower luminosity of the $\mu\mu$ channel, the net $\epsilon$ value for $\mu\mu$ channels, when included, is reduced by the ratio of the luminosities

$$\epsilon_{\mu\mu} = \epsilon^{fmc0}_{\mu\mu} \frac{\mathcal{L}_{\mu\mu}}{\mathcal{L}_{ee}}$$

[8] The factor of two was set somewhat arbitrarily. $(\sigma\epsilon)_{limit}$ is an estimate of 95% of the integrated probability. Since this tail of the distribution is falling as a gaussian tail, multiplying it by two assures that $P(\sigma_{max}|k, I)$ is negligible.

[9] $\sigma_{max}$ in this expression should be very close to twice the predicted $\sigma_{limit}$ from the model independent limit. $P(\sigma|k, I)$ falls of as a gaussian tail. Setting $\sigma_{max}$ to twice the predicted limit assures $P(\sigma_{max}|k, I)$ is negligible.

The channels chosen are either 2 channel combinations or all three channels, depending on the expected significance (only the combination with the maximum significance is used).

Setting a limit at each point in space is quite time consuming at about thirty seconds per model. To reduce the time to run over the entire set of models, channels where the number of expected events from signal is less than 2.5 are considered non-excludable.

# Chapter 6

# Results

## 6.1 Background and Data Summary Tables

Table 6.1 summarizes the results from the previous chapter for electron-electron signatures, Table 6.2 summarizes the electron-muon signatures, and Table 6.3 summarizes the muon-muon signatures. Tables 6.4 - 6.6 enumerate the total expected background in each kinematic channel, compares them with data, and shows the 95% confidence limits on visible cross section.

No significant excess is seen above background. There are six channels that "look" like there might be something there. These are listed in Table 6.7 where the Poisson probability of the expected background to fluctuate to the observed events is shown. The two fluctuations above two sigma are best viewed in the context of best channels (Section 5.2.2).

Across all kinematic channels, there will be one with the best expected significance. About six thousand models were generated, and Table 6.8 shows the breakdown of the number of times a kinematic channel producing more than 2.5 signal events was best, and whether the visible cross section is excluded. Di-electron signatures dominate the best channels due to higher identification efficiency. Note that the two channels where the "excess" is seen, $mmjj$ and $mmHjj$, exclude models where these signatures are expected to dominate.

Not all channels appear in Table 6.8. This indicates that enough channels were studied. If all channels appeared, one would want to further subdivide the different kinematic cuts to achieve full optimization. Furthermore, these "extra" channels serve as a cross check on the analysis as a whole: they show that the background is very well modeled.

For di-electron signatures, there are two channels that dominate best channels when the expected significance is between 2 and 3 sigma (*i.e.* near where an exclusion contour would be drawn). $eeHjjHNz$ dominates the region below $m_0$ of 150 GeV and $eeHjjjNz$ dominates the region above $m_0$ of 150 GeV.

For electron-muon signatures, two jet fields dominate the near exclusion: $Hjj$ and $HjjH$. For $m_0$ below 50 GeV, $e7m$ is important, while $em$ takes over out to $m_0$ of 150 GeV. Above that, electron-muon alone can't exclude models; however, $eTmHjjj$ does help out in the combined limit.

Similarly, $mmHjjU$ is important near exclusion. $mmjjjU$ helps out the combined limits above $m_0$ of 150 GeV.

Table 6.1: Background breakdown for $ee$ signatures. The error is the quadrature sum of the various errors.

| Channel | QCD / $W \to e$ | $Z \to \tau\tau$ | $Z \to ee$ | $t\bar{t} \to ee$ |
|---|---|---|---|---|
| $eejjNz$ | $2.70 \pm 0.52$ | $0.63 \pm 0.13$ | $0.49 \pm 0.12$ | $1.12 \pm 0.45$ |
| $eejjz$ | $2.70 \pm 0.52$ | $0.64 \pm 0.13$ | $6.16 \pm 1.41$ | $1.38 \pm 0.55$ |
| $eejjjz$ | $0.66 \pm 0.13$ | $0.14 \pm 0.04$ | $1.89 \pm 0.53$ | $0.45 \pm 0.20$ |
| $eeHjjz$ | $1.41 \pm 0.27$ | $0.34 \pm 0.08$ | $4.74 \pm 1.14$ | $1.21 \pm 0.49$ |
| $eejjHz$ | $1.86 \pm 0.36$ | $0.32 \pm 0.07$ | $1.39 \pm 0.34$ | $1.24 \pm 0.50$ |
| $eeHjjHz$ | $1.03 \pm 0.20$ | $0.21 \pm 0.05$ | $0.98 \pm 0.26$ | $1.10 \pm 0.45$ |
| $eeHjjjz$ | $0.42 \pm 0.08$ | $0.10 \pm 0.03$ | $1.61 \pm 0.47$ | $0.41 \pm 0.18$ |
| $eejjjHz$ | $0.44 \pm 0.09$ | $0.08 \pm 0.02$ | $0.37 \pm 0.12$ | $0.41 \pm 0.18$ |
| $eeHjjjHz$ | $0.44 \pm 0.09$ | $0.07 \pm 0.02$ | $0.33 \pm 0.11$ | $0.37 \pm 0.17$ |
| $eejjHNz$ | $1.86 \pm 0.36$ | $0.63 \pm 0.13$ | $0.49 \pm 0.12$ | $1.12 \pm 0.45$ |
| $eejjjNz$ | $0.66 \pm 0.13$ | $0.14 \pm 0.04$ | $0.13 \pm 0.04$ | $0.36 \pm 0.16$ |
| $eeHjjNz$ | $1.41 \pm 0.27$ | $0.34 \pm 0.08$ | $0.37 \pm 0.10$ | $0.98 \pm 0.40$ |
| $eeHjjHNz$ | $1.03 \pm 0.20$ | $0.21 \pm 0.05$ | $0.07 \pm 0.03$ | $0.89 \pm 0.36$ |
| $eeHjjjNz$ | $0.42 \pm 0.08$ | $0.10 \pm 0.03$ | $0.10 \pm 0.04$ | $0.33 \pm 0.15$ |
| $eejjjHNz$ | $0.44 \pm 0.09$ | $0.08 \pm 0.02$ | $0.02 \pm 0.01$ | $0.33 \pm 0.15$ |
| $eeHjjjHNz$ | $0.44 \pm 0.09$ | $0.07 \pm 0.02$ | $0.02 \pm 0.01$ | $0.30 \pm 0.13$ |

Table 6.2: Background breakdown for $e\mu$ signatures. The error is the quadrature sum of the various errors.

| Channel | QCD / $W \to \mu$ | $Z \to \tau\tau$ | $t\bar{t} \to ee$ |
|---|---|---|---|
| $emjj$ | $4.54 \pm 0.89$ | $0.56 \pm 0.13$ | $1.31 \pm 0.55$ |
| $emHjj$ | $2.13 \pm 0.43$ | $0.32 \pm 0.08$ | $1.15 \pm 0.50$ |
| $emjjH$ | $1.43 \pm 0.30$ | $0.31 \pm 0.07$ | $1.18 \pm 0.50$ |
| $emHjjH$ | $0.77 \pm 0.16$ | $0.20 \pm 0.05$ | $1.04 \pm 0.45$ |
| $emjjj$ | $1.37 \pm 0.28$ | $0.08 \pm 0.03$ | $0.33 \pm 0.15$ |
| $emHjjj$ | $0.94 \pm 0.20$ | $0.07 \pm 0.02$ | $0.30 \pm 0.14$ |
| $emjjjH$ | $0.47 \pm 0.11$ | $0.05 \pm 0.02$ | $0.30 \pm 0.14$ |
| $emHjjjH$ | $0.40 \pm 0.10$ | $0.04 \pm 0.02$ | $0.27 \pm 0.13$ |
| $e7mjj$ | $1.66 \pm 0.34$ | $0.45 \pm 0.10$ | $1.30 \pm 0.55$ |
| $e7mHjj$ | $0.93 \pm 0.19$ | $0.25 \pm 0.06$ | $1.15 \pm 0.50$ |
| $e7mjjH$ | $0.60 \pm 0.14$ | $0.26 \pm 0.06$ | $1.17 \pm 0.50$ |
| $e7mHjjH$ | $0.39 \pm 0.09$ | $0.18 \pm 0.05$ | $1.04 \pm 0.45$ |
| $e7mjjj$ | $0.54 \pm 0.12$ | $0.07 \pm 0.02$ | $0.33 \pm 0.15$ |
| $e7mHjjj$ | $0.42 \pm 0.10$ | $0.06 \pm 0.02$ | $0.29 \pm 0.14$ |
| $e7mjjjH$ | $0.28 \pm 0.07$ | $0.05 \pm 0.02$ | $0.30 \pm 0.14$ |
| $e7mHjjjH$ | $0.26 \pm 0.07$ | $0.04 \pm 0.02$ | $0.27 \pm 0.13$ |
| $eTmjj$ | $0.90 \pm 0.19$ | $0.36 \pm 0.08$ | $1.29 \pm 0.54$ |
| $eTmHjj$ | $0.48 \pm 0.11$ | $0.20 \pm 0.05$ | $1.13 \pm 0.49$ |
| $eTmjjH$ | $0.35 \pm 0.09$ | $0.20 \pm 0.05$ | $1.16 \pm 0.49$ |
| $eTmHjjH$ | $0.22 \pm 0.06$ | $0.13 \pm 0.04$ | $1.02 \pm 0.44$ |
| $eTmjjj$ | $0.30 \pm 0.07$ | $0.06 \pm 0.02$ | $0.32 \pm 0.15$ |
| $eTmHjjj$ | $0.13 \pm 0.04$ | $0.05 \pm 0.02$ | $0.29 \pm 0.14$ |
| $eTmjjjH$ | $0.24 \pm 0.06$ | $0.04 \pm 0.02$ | $0.29 \pm 0.14$ |
| $eTmHjjjH$ | $0.12 \pm 0.04$ | $0.03 \pm 0.01$ | $0.26 \pm 0.13$ |

Table 6.3: Background breakdown for $\mu\mu$ signatures. The error is the quadrature sum of the various errors.

| Channel | QCD / $W \to \mu$ | $Z \to \tau\tau$ | $Z \to \mu\mu$ | $t\bar{t} \to \mu\mu$ |
|---|---|---|---|---|
| $mmjj$ | 0.41 ± 0.06 | 0.06 ± 0.02 | 0.90 ± 0.23 | 0.40 ± 0.18 |
| $mmHjj$ | 0.34 ± 0.05 | 0.04 ± 0.02 | 0.64 ± 0.18 | 0.36 ± 0.16 |
| $mmjjH$ | 0.14 ± 0.02 | 0.03 ± 0.01 | 0.23 ± 0.07 | 0.36 ± 0.16 |
| $mmHjjH$ | 0.14 ± 0.02 | 0.03 ± 0.01 | 0.18 ± 0.06 | 0.32 ± 0.15 |
| $mmjjj$ | 0.11 ± 0.03 | 0.00 ± 0.00 | 0.18 ± 0.07 | 0.08 ± 0.04 |
| $mmHjjj$ | 0.11 ± 0.03 | 0.00 ± 0.00 | 0.16 ± 0.07 | 0.07 ± 0.03 |
| $mmjjjH$ | 0.06 ± 0.03 | 0.00 ± 0.00 | 0.04 ± 0.02 | 0.07 ± 0.03 |
| $mmHjjjH$ | 0.06 ± 0.01 | 0.00 ± 0.00 | 0.04 ± 0.02 | 0.06 ± 0.03 |
| $mmjjU$ | 0.07 ± 0.01 | 0.03 ± 0.01 | 0.12 ± 0.03 | 0.32 ± 0.15 |
| $mmHjjU$ | 0.07 ± 0.01 | 0.02 ± 0.01 | 0.09 ± 0.03 | 0.29 ± 0.13 |
| $mmjjjU$ | 0.06 ± 0.01 | 0.00 ± 0.00 | 0.01 ± 0.01 | 0.06 ± 0.03 |
| $mmHjjjU$ | 0.06 ± 0.01 | 0.00 ± 0.00 | 0.01 ± 0.01 | 0.05 ± 0.03 |

Table 6.4: Background in the $ee$ signature. The error is the quadrature sum of the various errors.

| Channel | Total Background | Data | 95% C.L. in fb |
|---|---|---|---|
| $eejjNz$ | 4.84 ± 0.70 | 5 | 67 |
| $eejjz$ | 10.67 ± 1.57 | 10 | 85 |
| $eejjjz$ | 3.08 ± 0.57 | 2 | 42 |
| $eeHjjz$ | 7.56 ± 1.25 | 5 | 58 |
| $eejjHz$ | 4.72 ± 0.69 | 2 | 43 |
| $eeHjjHz$ | 3.26 ± 0.55 | 2 | 45 |
| $eeHjjjz$ | 2.49 ± 0.50 | 0 | 31 |
| $eejjjHz$ | 1.28 ± 0.23 | 1 | 42 |
| $eeHjjjHz$ | 1.18 ± 0.21 | 0 | 31 |
| $eejjHNz$ | 4.02 ± 0.59 | 1 | 37 |
| $eejjjNz$ | 1.27 ± 0.21 | 1 | 40 |
| $eeHjjNz$ | 3.03 ± 0.49 | 3 | 60 |
| $eeHjjHNz$ | 2.16 ± 0.41 | 2 | 53 |
| $eeHjjjNz$ | 0.93 ± 0.17 | 0 | 31 |
| $eejjjHNz$ | 0.86 ± 0.17 | 1 | 44 |
| $eeHjjjHNz$ | 0.80 ± 0.16 | 0 | 31 |

Table 6.5: Background in the $e\mu$ signature. The error is the quadrature sum of the various errors.

| Channel | Total Background | Data | 95% C.L. in fb |
|---|---|---|---|
| $emjj$ | $6.30 \pm 1.04$ | 9 | 105 |
| $emHjj$ | $3.54 \pm 0.65$ | 6 | 92 |
| $emjjH$ | $2.87 \pm 0.57$ | 6 | 99 |
| $emHjjH$ | $1.97 \pm 0.47$ | 3 | 65 |
| $emjjj$ | $1.75 \pm 0.31$ | 1 | 41 |
| $emHjjj$ | $1.27 \pm 0.24$ | 1 | 42 |
| $emjjjH$ | $0.80 \pm 0.17$ | 0 | 31 |
| $emHjjjH$ | $0.70 \pm 0.16$ | 0 | 31 |
| $e7mjj$ | $3.36 \pm 0.64$ | 7 | 107 |
| $e7mHjj$ | $2.29 \pm 0.53$ | 5 | 91 |
| $e7mjjH$ | $1.99 \pm 0.51$ | 4 | 79 |
| $e7mHjjH$ | $1.57 \pm 0.45$ | 2 | 54 |
| $e7mjjj$ | $0.93 \pm 0.19$ | 1 | 44 |
| $e7mHjjj$ | $0.76 \pm 0.17$ | 1 | 45 |
| $e7mjjjH$ | $0.61 \pm 0.15$ | 0 | 31 |
| $e7mHjjjH$ | $0.56 \pm 0.14$ | 0 | 31 |
| $eTmjj$ | $2.50 \pm 0.57$ | 3 | 61 |
| $eTmHjj$ | $1.79 \pm 0.49$ | 3 | 67 |
| $eTmjjH$ | $1.67 \pm 0.49$ | 1 | 42 |
| $eTmHjjH$ | $1.35 \pm 0.44$ | 1 | 43 |
| $eTmjjj$ | $0.68 \pm 0.17$ | 1 | 45 |
| $eTmHjjj$ | $0.46 \pm 0.14$ | 1 | 46 |
| $eTmjjjH$ | $0.56 \pm 0.15$ | 0 | 31 |
| $eTmHjjjH$ | $0.41 \pm 0.13$ | 0 | 31 |

Table 6.6: Background in the $\mu\mu$ signature. The error is the quadrature sum of the various errors.

| Channel | Total Background | Data | 95% C.L. in fb |
|---|---|---|---|
| $mmjj$ | $1.74 \pm 0.30$ | 5 | 90 |
| $mmHjj$ | $1.35 \pm 0.24$ | 5 | 96 |
| $mmjjH$ | $0.75 \pm 0.17$ | 2 | 60 |
| $mmHjjH$ | $0.65 \pm 0.16$ | 2 | 60 |
| $mmjjj$ | $0.37 \pm 0.09$ | 2 | 63 |
| $mmHjjj$ | $0.34 \pm 0.08$ | 2 | 63 |
| $mmjjjH$ | $0.16 \pm 0.05$ | 1 | 49 |
| $mmHjjjH$ | $0.16 \pm 0.04$ | 1 | 49 |
| $mmjjU$ | $0.53 \pm 0.15$ | 1 | 49 |
| $mmHjjU$ | $0.46 \pm 0.14$ | 1 | 49 |
| $mmjjjU$ | $0.13 \pm 0.03$ | 1 | 50 |
| $mmHjjjU$ | $0.12 \pm 0.03$ | 1 | 50 |

Table 6.7: Poisson probability that the expected background fluctuated to the observed number of events.

| Channel | Expected Background | Observed | Poisson Prob. |
|---|---|---|---|
| $emjj$ | 6.3 | 9 | 18.0% |
| $emHjj$ | 3.5 | 6 | 15.0% |
| $emjjH$ | 2.9 | 6 | 7.2% |
| $e7mjj$ | 3.3 | 7 | 5.3% |
| $mmjj$ | 1.8 | 5 | 3.2% |
| $mmHjj$ | 1.4 | 5 | 1.3% |

Table 6.8: Breakdown of best channels. "Excluded" here means that the visible cross section exceeds the model independent limit.

| Channel | # Times Best | # Excluded |
|---|---|---|
| $eejjNz$ | 35 | 29 |
| $eejjz$ | 12 | 0 |
| $eeHjjz$ | 9 | 0 |
| $eejjHz$ | 7 | 0 |
| $eeHjjHz$ | 42 | 11 |
| $eeHjjjz$ | 1 | 0 |
| $eejjjNz$ | 4 | 0 |
| $eeHjjNz$ | 131 | 108 |
| $eeHjjHNz$ | 288 | 167 |
| $eeHjjjNz$ | 158 | 143 |
| $eejjjHNz$ | 1 | 1 |
| $eeHjjjHn$ | 5 | 4 |
| $emjj$ | 2 | 0 |
| $emHjj$ | 3 | 0 |
| $emjjH$ | 1 | 0 |
| $emHjjH$ | 11 | 3 |
| $e7mjj$ | 1 | 0 |
| $e7mHjj$ | 4 | 0 |
| $e7mjjH$ | 2 | 1 |
| $e7mHjjH$ | 12 | 1 |
| $eTmjj$ | 1 | 1 |
| $eTmHjj$ | 10 | 2 |
| $eTmjjH$ | 1 | 1 |
| $eTmHjjH$ | 4 | 3 |
| $mmjj$ | 1 | 1 |
| $mmHjj$ | 2 | 2 |
| $mmjjH$ | 4 | 3 |
| $mmHjjH$ | 3 | 0 |
| $mmjjj$ | 2 | 1 |
| $mmjjjH$ | 1 | 0 |
| $mmjjU$ | 11 | 7 |
| $mmHjjU$ | 16 | 3 |
| Totals | 785 | 492 |

## 6.2 Excluded Models

Some of the model space has been excluded previous to this analysis. There are theoretical constraints on SUGRA models, namely, requiring EWSB and disallowing tachyons. LEP I has also presented model independent mass limits on various sparticles. The most important of these to SUGRA are limits on $\tilde{W}_1$ and a light higgs. The exclusion contours, for $\tan\beta$ values of 2.0, 3.0, 4.0, 6.0, and 10.0 are shown in Figure 6.1. In subsequent figures, only the LEP I limits will be shown, as they naturally also exclude disallowed regions. These regions of space are excluded by FMCØ at the time of model generation.[1]

There are many ways to plot excluded models. The one chosen here is to make plots in the $m_{1/2}$-$m_0$ plane within a range of $\tan\beta$. Figures 6.2 to 6.6 show excluded and not excluded models. The space below and to the left in all plots is either theoretically excluded, or excluded from LEP I experimental constraints.

For $\tan\beta \sim 2$, there is a "spoiler mode" that turns on around $m_0 = 80$ GeV/c$^2$, namely, the three body decay of the $\tilde{Z}_2$ into neutrinos. This reduces both the dilepton and jet production (Table 6.9), and accounts for the "dip" seen in those plots.

This invisible region becomes visible with an increase in $\tan\beta$ to around 4.0. The gaugino mass couplings increase, causing $\tilde{Z}_2$ decays to jets to increase by a factor of ten (to a bit over 50%). This is at the expense of 1) charged lepton decays (now

[1] Points in model space are chosen randomly within a specified range. When a constructed model fails experimental of theoretical constraints, it is discarded before event generation begins. See Section A.5 for more details on the constraints.

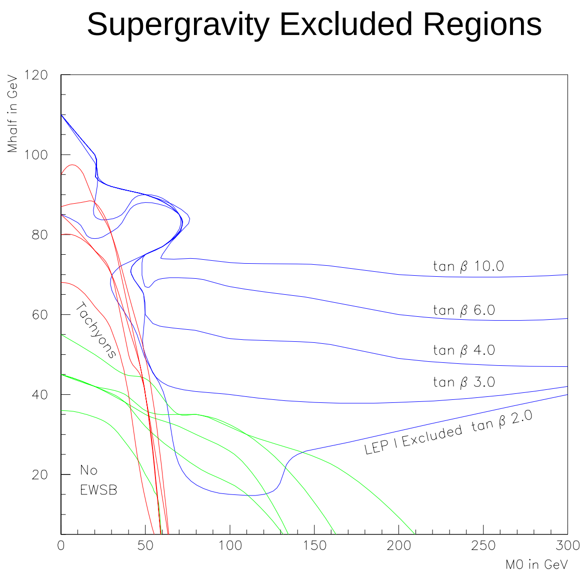


Figure 6.1: Disallowed and previously excluded regions in the $m_0$-$m_{1/2}$ plane for $\mu < 0$, $A_0 = 0$, and various $\tan\beta$.

8% each), which doesn't matter so much because the $\tilde{Z}_2$-$\tilde{Z}_1$ mass difference reduces lepton $E_T$'s below threshold, and 2) the neutrino-LSP mode, which is reduced to around 25% (compare these numbers to $m_0 = 75$ GeV/c$^2$ in Table 6.9).

In addition, the chargino and neutralino masses are decreased in this region. This opens up squark decay modes to $\tilde{Z}_3$ and $\tilde{Z}_4$, which in turn dominantly decay to left-handed sneutrinos. Furthermore, sneutrinos in this region decay into a chargino and a charged lepton about 50% of the time (versus 9% for $\tan\beta = 2$), and into $\tilde{Z}_2$ and a neutrino about 25% of the time (versus 7%).

The effect of these factors is to increase the net branching ratio of signatures in the current analysis up to $\tan\beta$ values of around 7.0, where decays into light charged

Table 6.9: Spoiler mode decay of the $\tilde{Z}_2$. Branching ratio for the dominant decays of $\tilde{Z}_2$ for $m_{1/2} = 60$, $\tan\beta = 2$, $\mu < 0$, and $A_0 = 0$.

| $m_0$ (GeV/$c^2$) | 75 | 80 | 85 |
|---|---|---|---|
| BR($\tilde{Z}_2 \to ee$) (%) | 19.5 | 22.5 | 23.8 |
| BR($\tilde{Z}_2 \to \mu\mu$) (%) | 19.5 | 22.5 | 23.8 |
| BR($\tilde{Z}_2 \to \tau\tau$) (%) | 23.4 | 25.8 | 26.5 |
| BR($\tilde{Z}_2 \to \nu\nu$) (%) | 31.3 | 20.4 | 15.1 |
| BR($\tilde{Z}_2 \to q\overline{q}$) (%) | 6.2 | 8.7 | 10.9 |

leptons are reduced by increased mass coupling.

There is some interleaving between the maximum excluded and the minimum not excluded. This is caused by statistical fluctuations in signal production, and the accuracy of the predicted total cross section. The exclusion contour is defined as the average of the highest point excluded and the lowest point not excluded. Figure 6.7 show the exclusion contour for $\tan\beta$ between 1.5 and 2.5 along with the LEP 1 previous limit. Figure 6.8 shows the exclusion regions as $\tan\beta$ varies between 1.5 and 10.0.

Finally, the excluded models can be used to set limits on sparticle masses for the region of parameter space studied ($20 < m_0 < 300$, $20 < m_{1/2} < 110$, $1.1 < \tan\beta < 10.$, $A_0 = 0$, and $\mu$ negative). All gluinos below 129 GeV/c$^2$ and all squarks below 138 GeV/c$^2$ are excluded at the 95% confidence level by this analysis.

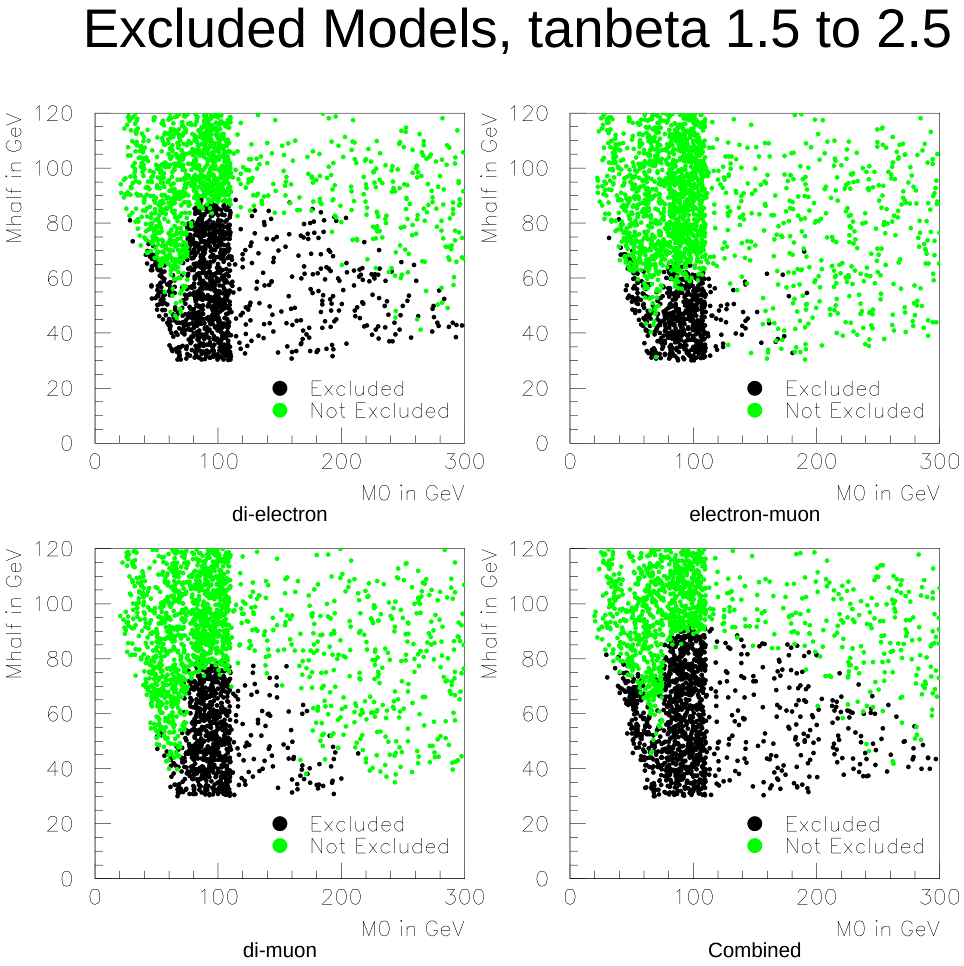


Figure 6.2: Excluded models in the $\tan\beta$ region 1.5 to 2.5.

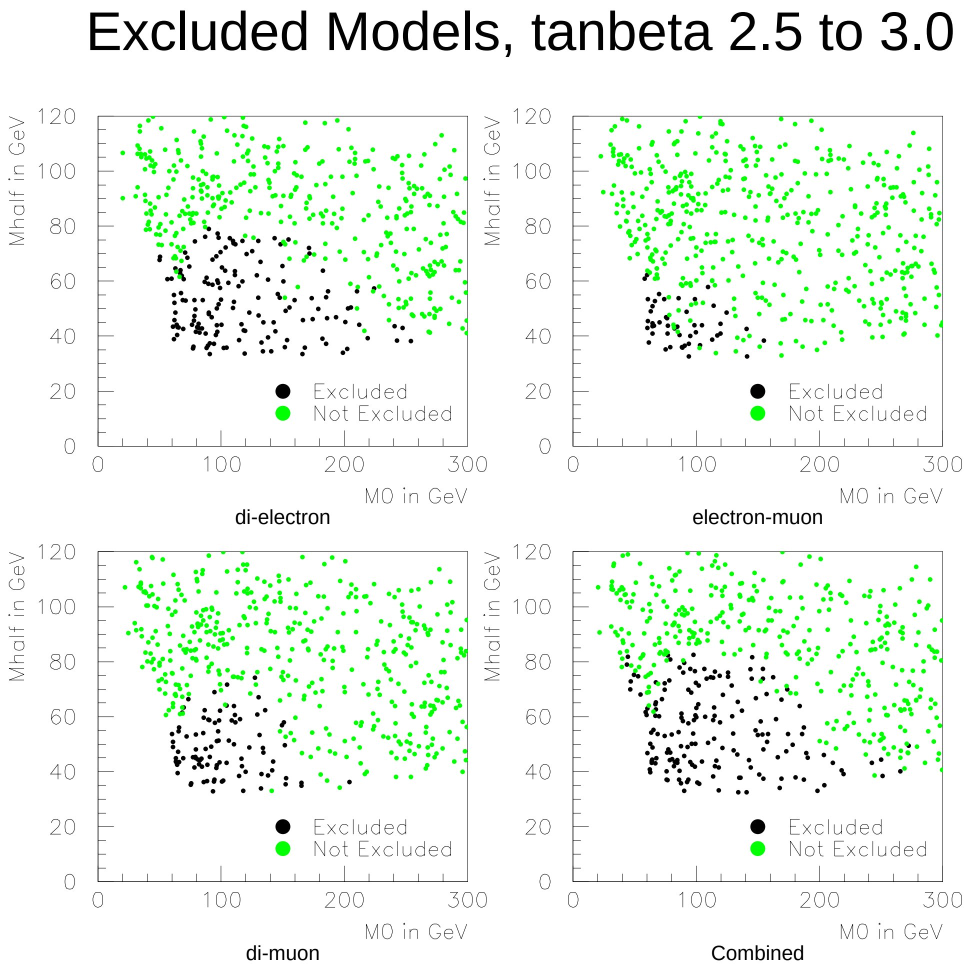


Figure 6.3: Excluded models in the $\tan\beta$ region 2.5 to 3.0.

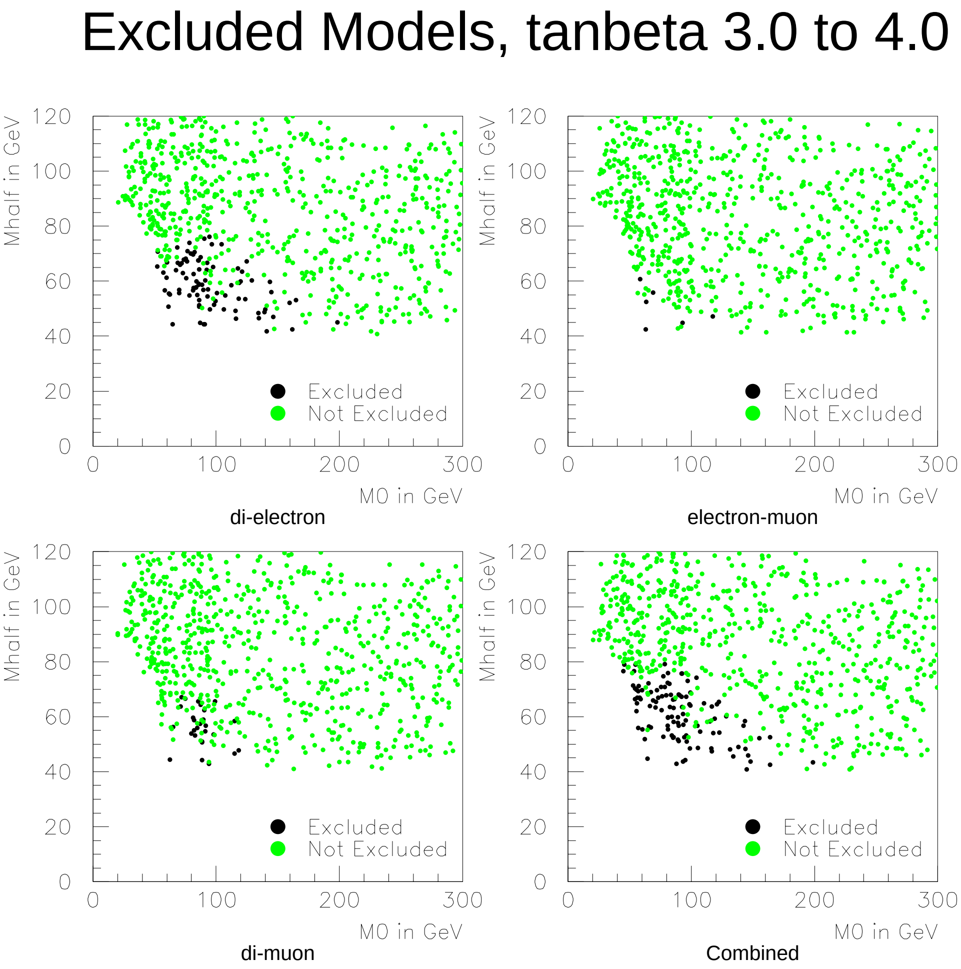


Figure 6.4: Excluded models in the $\tan\beta$ region 3.0 to 4.0.

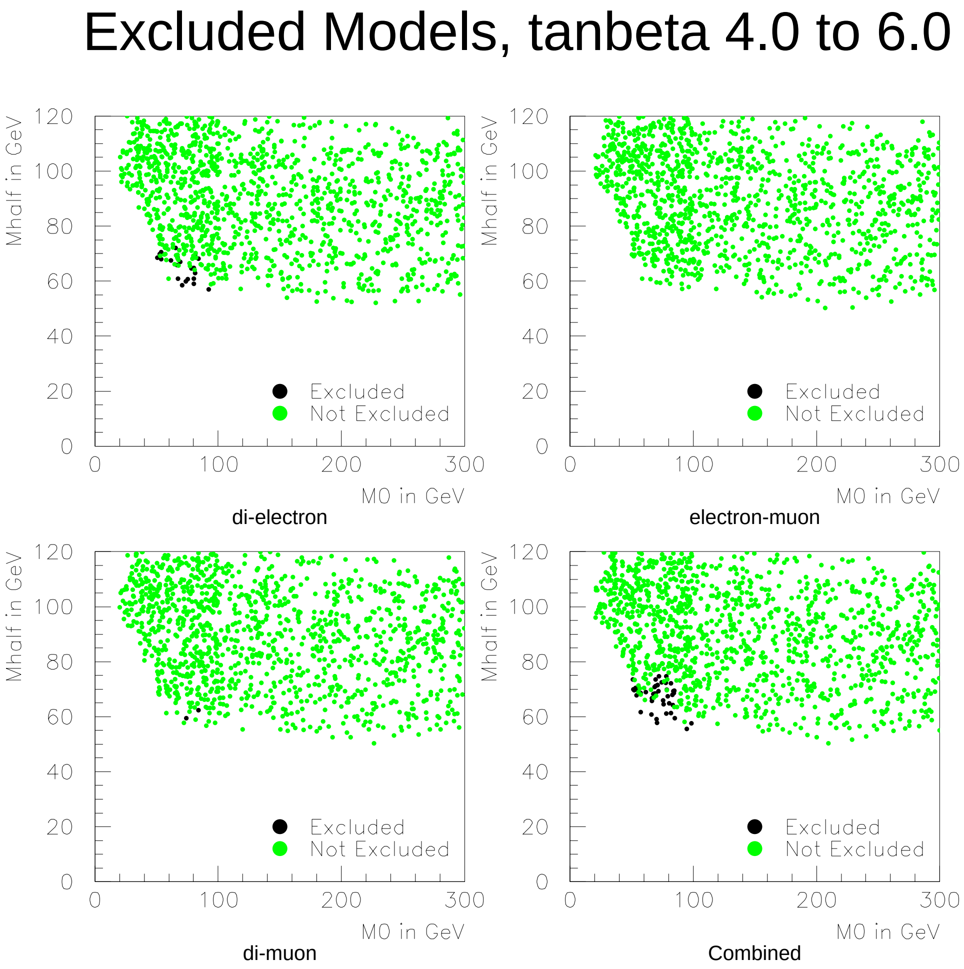
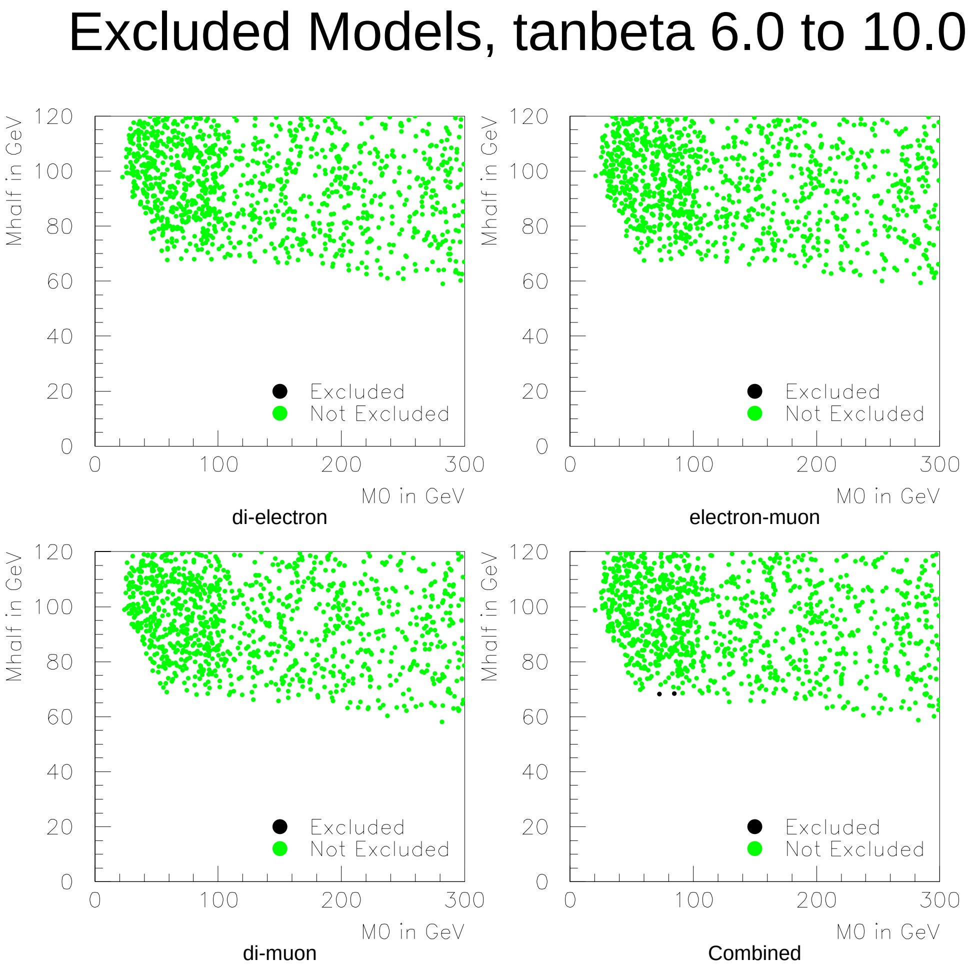


Figure 6.5: Excluded models in the $\tan\beta$ region 4.0 to 6.0.



di-electron

electron-muon

di-muon

Combined

Figure 6.6: Excluded models in the tan $\beta$ region 6.0 to 10.0.

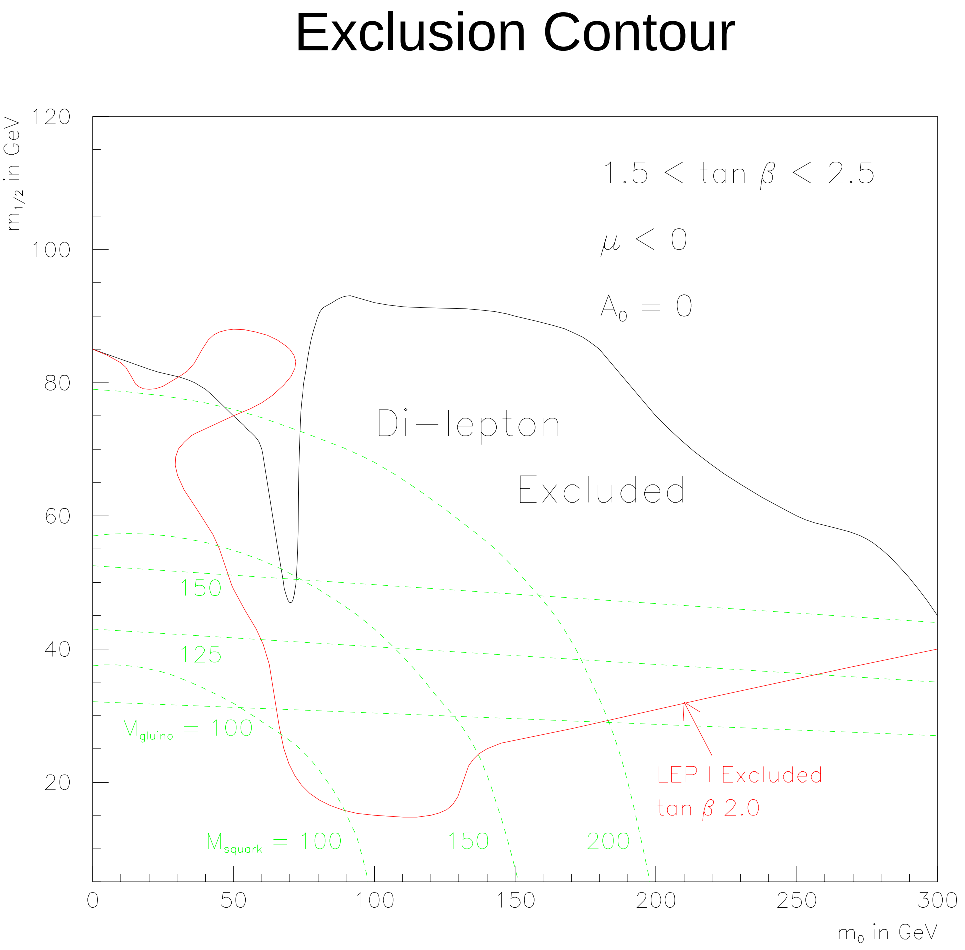


Figure 6.7: 95% exclusion contour in the $\tan\beta$ region 1.5 to 2.5.

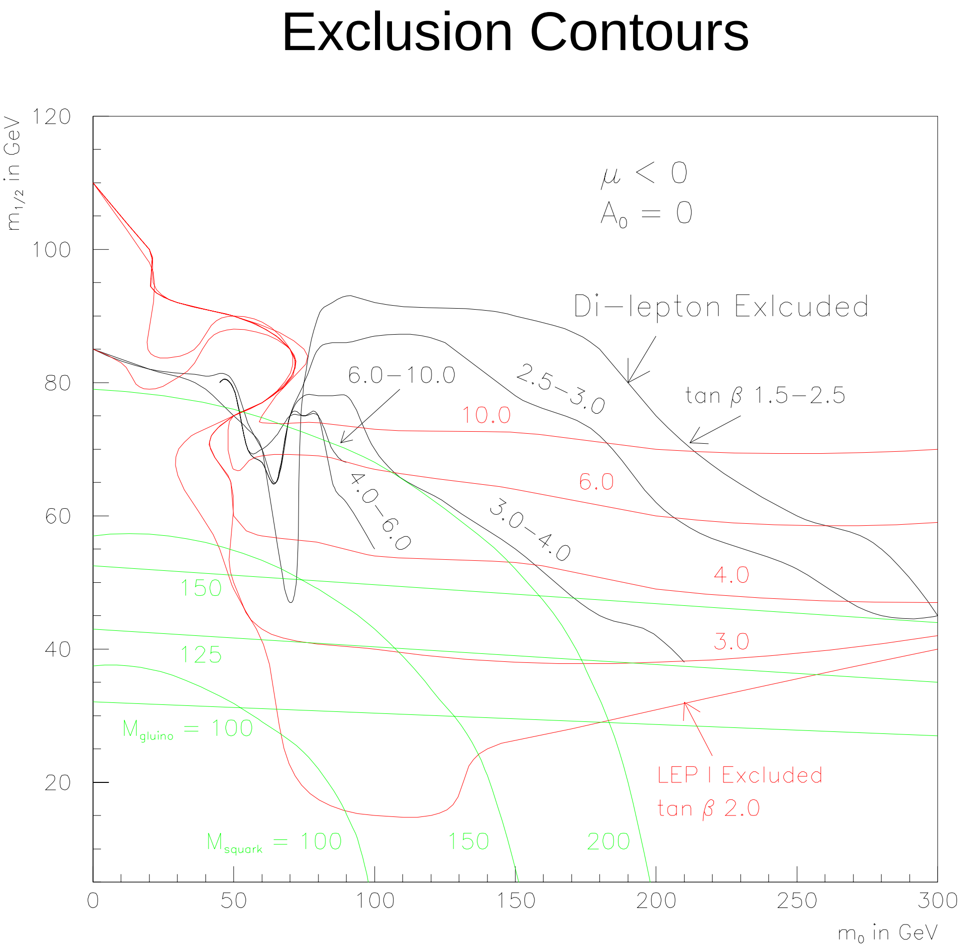


Figure 6.8: 95% exclusion contours for various $\tan\beta$ ranges.

## 6.3 Conclusion

A search for SUGRA squarks and gluinos decaying into di-leptons has been presented. Using a search strategy of optimizing kinematic cuts at each point in model space, significant additional exclusion has been achieved above LEP 1 limits for low $\tan\beta$. In addition, all gluinos below 129 GeV/c$^2$ and all squarks below 138 GeV/c$^2$ have been excluded for $\tan\beta < 10$.

# Appendix A

# FMCØ Program Manual

## A.1 Introduction

FMCØ is a complete Monte Carlo signal analysis package. It includes a SUGRA model generator (Michmodel), a MC signal generator (SPYTHIA 2.10), a fast detector simulation (QSIM), and an event analysis package.[1] The event analysis package can be thought of as split into three parts: RECO simulation (RSIM), signal simulation (SSIM), and trigger simulation (TSIM). RSIM and TSIM are interface packages that call utility routines from the QSIM library. SSIM is a bookkeeping package that analyzes each event as it is produced and processed, keeping track of the number of events produced, the number of events that pass a *kinematic channel* (see below and Ref. [35]), and each event's weight as determined by RSIM and TSIM. The entire

[1] The original version of QSIM was written by Lee Lueking. Michmodel is a Supergravity model generator written by Chris Kolda, of the Institute for Advanced Study at Princeton, and the author. SPYTHIA is a Supersymmetric extension to PYTHIA 5.7 written by Steve Mrenna of Argonne National Lab (see also Ref. [34]). [Both Drs. Kolda and Mrenna are former members of Gordy Kane's group at the University of Michigan.]

package is controlled with various RCP files. In short, the user sets up FMCØ to study a portion of SUGRA parameter space, and FMCØ outputs an ntuple that one can use to predict the number of expected signal events at several sample points within the studied region.

We begin with a background section for readers not familiar with PYTHIA or the FMCØ programming philosophy. We then include detailed sections on the structure of FMCØ, package initialization considerations, various running options, specifics regarding the SPYTHIA setup, the loading of the `RMSS` array from the Michmodel output, adding analysis cuts, random number generation, and infinite loop detection. We conclude with sections describing how to setup and use FMCØ under VMS and a look toward Run II. Throughout, we define normal running as using the program to generate SUGRA models for analysis. Documentation of QSIM, RSIM, and TSIM is available elsewhere [36].

## A.2 Background Information

This section describes the necessary prerequisite knowledge for understanding the terms and ideas presented in other sections of this note. We include sections on running PYTHIA, the FMCØ programming philosophy, and the definition of kinematic channels.

### A.2.1 Running PYTHIA

PYTHIA is a general high energy physics event generator. Its running is controlled by a set of common blocks with integer flags and adjustable integer and real values. These flags and values number in the thousands and we will not attempt to describe all possible running options; rather, we will try and give a flavor of how one runs standard PYTHIA in the DØ software environment. We refer to this set of flags and values as the PYTHIA parameters.

The PYTHIA parameters completely describe the process to be generated, all possible decays and fragmentations, and possible output options. For example, the `PYSUBS` common block contains the integer flags `MSUB(200)` and `MSEL` (in the SPYTHIA distribution of PYTHIA 5.7, this array is declared with 400 elements). The `MSUB` array controls which of the 200 possible production processes, called *subprocesses*, are allowed. `MSUB(ISUB)=1` will turn on the `ISUB` process, while `MSUB(ISUB)=0`, the default, turns it off. `MSEL` is an integer value, introduced for convenience, that will turn on a set of subprocesses that usually are studied concurrently; for example, `MSEL=15` will turn on subprocesses 22, 23, 25, 69, and 70, the group of all $W/Z$ boson pair production subprocesses. Another example is the `PYPARS` common which contains the integer array `MSTP(200)`. Each of these 200 values controls some aspect of the interactions; the choice of structure function, for example, is governed by the values `MSTP(51)` and `MSTP(52)`.

The parameters can be set up by directly accessing the common blocks or by issuing a command to change a paramater via `LUGIVE`. `LUGIVE`, the standard method of choice at DØ, accepts a single character string of the syntax "param=value." This method combines well with the SRCP package as all setup commands can be placed in a character array in an RCP file. The standard PYTHIA_RCP contains two arrays of commands: a user array that sets up the process to be studied and a default array that sets advanced options to values appropriate for the DØ running environment. Also included in PYTHIA_RCP are four parameters that describe the Tevatron as a center-of-mass collider of protons and anitprotons operating at $\sqrt{s} = 1800$ GeV; these parameters are passed to the PYTHIA initialization routine `PYINIT` after setup and before looping over the generation routine `PYEVNT`. PYTHIA uses implicit variable declaration throughout, and the user should pass values appropriately to `LUGIVE`. For access to a specific common block, the user may include the file `FMCO_xxx.INC`, where xxx is replaced with the common block name. These files declare all of the variables appropriately and are compatible with an `IMPLICIT NONE` statement.

Another important common block is `LUJETS`, which stores the current event.[2] The reader should consult sections 5.2 and 5.3 of Ref [37] for a complete description of the event record. Briefly, for each particle, `LUJETS` contains an integer array (`K(n,5)` where `n` is the ordinal entry in the event record) describing particle status, identification, and decay information. In addition, two real arrays (`P(n,5)` and `V(n,5)`) contain four-momentum and particle position information. The first several entries contain a summary of the primary interaction followed by the event in its entirety. A simple example of generating 100 events of $W/Z$ pairs is

[2] PYTHIA 5.7 is actually a combination of the two programs PYTHIA and JETSET 7.4. JETSET, originally conceived separately by members of the Lund theory group, controls the jet fragmentation and event storage. All subroutines and common blocks associated with the PYTHIA portion of the program begin with the two letters `PY`, while all of the JETSET program elements begin with `LU`. This separation has been eliminated in PYTHIA 6.0 in which all elements begin with `PY`.

```
      PROGRAM PY_EXAMPLE
C----------------------------------------------------------------
C-   Purpose and Methods : Example of PYTHIA event generation.
C----------------------------------------------------------------
C
      CALL LUGIVE('MSEL=15')             ! W/Z pair production
      CALL PYINIT('CMS','P','PBAR',1800.)! initialize PYTHIA
C
      DO I = 1,100
        CALL PYEVNT                      ! generate an event
        CALL USER_PROG                   ! user analysis
      ENDDO
C
      END
```

The *user_prog* would need to call PYTEFL to transfer the event record from LUJETS to ZEBRA banks.

### A.2.2 FMCØ Programming Philosophy

FMCØ was designed using the *object-inspired* modular style of programming. In the object-inspired approach, subroutines know as little about the calling routine as possible. For example, the routine which fills electrons into the EVENT_VALUES array doesn't know whether the electrons have been smeared or not, or even whether any are required for the event under study; it only knows what electrons are and where to find them. In addition, the routine that requests the electrons to be filled doesn't know or care what an electron is, merely that it has to make a cut relative to some value on a set of objects returned from a set of object construction and filling routines.

Designing FMCØ with objects in mind will facilitate the eventual translation into C++. In addition, the modularity restrictions provide some insulation against the propagation of changes into external routines, and with extensive internal error checking, unexpected consequences of alterations should be easier to locate.

A true object-oriented design is not possible with Fortran 77, so in some routines, the object-inspired protocol has been violated. The routine that returns jets, for example, has to know about the returned electrons because the jet constructor, DØPJET, does not make a distinction.

In addition to the programming style protocol of the object-inspired approach, an analysis protocol is also present. Each event is completely described by a set of objects, and each object is allowed to have one and only one real value associated with it. The set of these values is stored in the two arrays `EVENT_VALUES` and `EVENT_VALUES_S`, where the latter contains the object values after the event has been smeared. For physics objects such as electrons or jets, the object value is the $E_T$. Given the limitation of one real value per object, the object constructor and filling routines take the decision whether an object should be included in `EVENT_VALUES`; an electron that is found in the ICR, for example, will be skipped by the electron filling routine.

Finally, part of the FMCØ programming philosophy must include mention of speed. The original design set a goal of four events per second with circa 1996 CPUs.[3] This number was chosen so that one processor could generate roughly one million events per CPU-month. A single I/O cycle requires on the order of 100-200

[3] We specify 1996 CPUs because it would be cheating to create an inefficient program and wait for Moore's Law to help meet the design charge.

milliseconds for typical hardware environments; if one desires to read events from a file, the I/O limitation sets a hard maximum of roughly 10 events per second, regardless of how fast a CPU is running. FMCØ was designed to be run in a mode in which the events are discarded after analysis, avoiding I/O cycles altogether. The only important number for a given model is the final $\sigma \cdot BR$ after full simulation.

### A.2.3 Kinematic Channels

We define a kinematic channel as a set of 22 parameters that completely describe the signal. Thirteen of these parameters are for object $E_T$ cuts on the leading 3 electrons, 5 jets, 3 muons, and $\not\!\!E_T$with and without muons. If an object parameter is exactly zero, that object is not required in an event. If an object parameter is positive, we require the object $E_T$ be greater than or equal to the parameter to pass. If an object parameter is negative, we require that $E_T$ be less than or equal the magnitude of the parameter to pass. An event will pass the object requirements if it passes all thresholds.

The other 9 parameters that define a kinematic channel denote 7 special cuts; for example, SPC_1 determines whether the leading two electrons lie in the Z mass window. The 9 parameters also contain a string that defines the name of the channel and a string that is passed to TSIM to define the trigger used in the channel. The ordering of this set of parameters is (name, e1, e2, e3, jet1, jet2, jet3, jet4, jet5, mu1, mu2, mu3, metc, metmu, spc1, spc2, spc3, spc4, spc5, spc6, spc7, trig) where metc is calorimeter only $\not\!\!E_T$, and metmu includes muons. SPC_1 is defined above, while the other 6 cuts are reserved for future implementation or user-defined cuts.

In object-inspired terms, the 20 real values describe a kinematic channel that is compared to the set of object values for a given event. The comparison protocol allows a user to require, for a given kinematic channel, that an object exists above threshold, that it does not exist above threshold, or that the channel should ignore the presence of the object.

Once the event is generated, filling and object constructor routines, operating with information from the ZEBRA banks, place the 20 object values in `EVENT_VALUES`; `FMCO_FILL_ELEC` and `FMCO_SPC1`, for example. The event is then smeared with QSIM and the results overwrite the ZEBRA banks. The same routines fill `EVENT_VALUES_S`. We then loop over all defined kinematic channels and increment a separate counting array for the smeared and unsmeared event values. If the smeared event passes a given channel, we increment the real arrays for RSIM and TSIM using the appropriate returned weights.

When the event loop terminates, we scale the counting arrays according to the number of events generated in that model. For the production process under study, we now have four branching ratios per kinematic channel: one for generator level events, one for smeared events, one for smeared events weighted by a trigger efficiency, and one for smeared events weighted by the product of the trigger and reconstruction efficiency. This final branching ratio can be used with the total production cross section to directly determine the number of expected events in a data set.

## A.3 Structure of FMCØ

FMCØ was designed to produce and analyze Supergravity signal events for six input parameters ($m_0$, $m_{1/2}$, $\tan\beta$, $A_0$, sign($\mu$), and $m_{top}$). FMCØ first produces a valid SUGRA model from either randomly generating a set of input parameters or reading them from a file, then the event loop begins. For each event, the generator level four-vectors are analyzed to determine whether the event passes the set of defined kinematic channels. The event is then smeared with QSIM and the kinematic requirements are applied again. If the smeared event passes a given kinematic channel, it obtains a weight from TSIM and RSIM, which could be zero in some cases (*e.g.* an electron is lost in a phi module crack); otherwise, the weight is set directly to zero. FMCØ terminates the event generation loop in one of four ways:

1. A sufficient number of signal events have been generated. This condition is controlled by a threshold number, set in FMC0_RCP, of unsmeared or smeared events passing a set of enumerated kinematic channels. For example, one can ask that 300 smeared events pass the $eejjNz$ channel from Ref. [35]. This number is termed the *requested statistics.*

2. The maximum number of events have been generated. FMCØ will terminate event generation at `NUMBER_OF_EVENTS` from SPYTHIA_RCP.

3. The generation is going poorly. Every 5000 events, FMCØ will calculate the ratio of the current number of signal events to the requested statistics, $R_S$, and the ratio of generated events to the maximum allowed, $R_{total}$. If $R_S < 0.02 * R_{total}$, generation is terminated. In addition, the current estimate of $\sigma \cdot BR$ is calculated; if it is less then 0.01 pb$^{-1}$, then FMCØ also terminates event generation.

4. An error exit is requested from `FMCO_PYEVNT`, the shell routine which call `PYEVNT`. This routine keeps track of the number of serious errors (*e.g.* infinite loops) encountered during event generation. When 10 consecutive events or 50 non-consecutive events within a given model fail, `FMCO_PYEVNT` requests an error

exit, and the results of the current model are discarded.

Once event generation is successfully completed, an ntuple entry is filled with the following information:

**Model Parameters** All 94 parameters generated from Michmodel are placed in the ntuple. In addition, the routine `FMCO_TRANSMICH_MSBAR` obtains the physical gluino and degenerate squark masses. For normal running, the Michmodel squark masses do not include D terms, and the masses that SPYTHIA used during event generation may not exactly match, so all SUSY particle masses from the SPYTHIA `PMAS` array are included also.

**Cross Sections** The total production cross section, the production cross section for each included subprocess, a cross section read from a table for squark-squark, squark-gluino, and gluino-gluino production,[4] and the equivalent 3 cross sections from SPYTHIA are included in the ntuple.

**Kinematic Channel Production Ratios** Two numbers per kinematic channel, the ratio of unsmeared events passed to total events generated, and the ratio of smeared events passed to total events generated, are placed in the ntuple.

**Weighted Net Branching Ratios** Two numbers per channel, the ratio of the sum of the weights from TSIM to the total number of events generated, and the ratio of the sum of the products $w_i^{RSIM} w_i^{TSIM}$ to the total number of events generated, are included in the ntuple.

[4] This table was produced a few years ago using a program provided by H. Baer.

**Errors on the Weighted Sums** Two values per kinematic channel, calculated as follows, are included in the ntuple.

$$\delta R_{TSIM} = \frac{1}{nevents}\sqrt{\sum_{events}(\delta w_i^{TSIM})^2}$$

and

$$\delta R_{TSIM*RSIM} = \frac{1}{nevents}\sqrt{\sum_{events}\frac{(\delta w_i^{RSIM})^2}{(w_i^{TSIM})^2}+\frac{(\delta w_i^{TSIM})^2}{(w_i^{RSIM})^2}}$$

where the error on each weight was calculated in RSIM and TSIM.

After writing out the ntuple entry, FMCØ gets another model and repeats the process until `MAXMODELS` from FMC0_RCP is reached.

## A.4 Initialization and Loops

Each package included in FMCØ needs specialized initialization, additional re-initialization for each model, and sometimes initialization for each event. This initialization is the most abstruse function performed by FMCØ, and a detailed account is presented. In addition, the model and event loops are described in the context of this initialization procedure.

The first initialization involves the general FMCØ setup, in which FMC0_RCP and SPYTHIA_RCP are loaded, and some of the general control parameters are read. Next, SPYTHIA is initialized for the type of models to be produced. This initialization includes the setup of the `IMSS` array[34], production process setup (`MSEL` determination or `MSUB` array manipulation), and general PYTHIA options such as choice of structure function. For SUGRA, `IMSS`, the SPYTHIA control parameter array, is set up for general MSSM simulation; the internal SPYTHIA SUGRA generator is an approximation of the full Michmodel calculation which is not valid in all cases, and at best, is only accurate to 10-15% in any parameter. (See RMSS initialization for further details of which of the parameters SPYTHIA or Michmodel generate.) At this stage of initialization, all SPYTHIA_RCP flags have been set and all commands from that file have been passed to LUGIVE. The remaining values to initialize before calling PYINIT are all model dependent; we call this the *boot-up* state of SPYTHIA. The commands in SPYTHIA_RCP *define* the boot-up state.

PYTHIA was not intended to be run with more than one model in the same PROGRAM as is done in FMCØ. Many of the PYTHIA program elements rely on BLOCK DATA statements for initialization, and a few use the IF(FIRST) technique. Changing the model without returning PYTHIA to its boot-up state will result in several physics errors, the most serious of which is that not all of the model branching ratios are recalculated between models.

To facilitate returning SPYTHIA to its boot-up state, it is necessary to copy all relevant common blocks to a secure storage location. This copying is performed by a call to FMCO_LUDATA which uses the IF(FIRST) technique to trigger storage of the boot-up state; subsequent calls will re-initialize the PYTHIA commons.[5]

To take care of the PYTHIA and SPYTHIA IF(FIRST) initializations, a central routine, FMCO_REINIT, switches the value of FIRST back to true in all relevant locations. This routine required modifications to the PYTHIA and SPYTHIA source code to place the flags into various common blocks. These flags are kept track of by hand and are not in an include file. Currently, three routines have been identified

[5] This techniques is probably overkill, as the "I've been initialized" keys most likely are only a handful of values in a few of the blocks rather than the tens of thousands of words which we reset; however, finding ALL of those keys by hand would require an unacceptable amount of time for a non-PYTHIA author.

as requiring this type of re-initialization: PYBBN, PYTBBC, and PYMSIN. We use the naming convention routine_FIRST for the common block names (*e.g.* PYBBN_FIRST). This initialization is also an integral part of returning SPYTHIA to its boot-up state.

We refer to the procedures that allow FMCØ to return SPYTHIA to its boot-up state as *setting up the model loop*; after this step, it is possible to loop over models in the allowed space. The model loop consists of a repeated sequence of calls to FMCO_LUDATA, FMCO_REINIT, FMCO_RUN_SPYTHIA (this routine contains the event loop), and FMCO_NT_FILL.

FMCO_RUN_SPYTHIA first calls FMCO_MODEL to generate a valid model. We define a *submodel* as a generated model that has not yet been constrained to be valid. FMCO_MODEL generates a model with Michmodel, returns to the PYTHIA boot-up state (redundant on the first submodel), loads the appropriate RMSS values[34], calls PYINIT to initialize SPYTHIA and calculate the decay table, and then constrains the model within theoretical and LEP I experimental limits.[6] Michmodel was designed to be self-initializing, so no separate consideration is required while looping to find a valid model. Invalid submodels will not increment the counter that loops until MAXMODELS have been generated; however, if 1000 consecutive submodels fail constraint, an error message is issued to the screen and the program performs an orderly exit.

PYINIT is checked for successful completion of three routines, PYMSIN, PYCJDC, and PYNJDC, if CONSTRAIN is set true. When these three routines are successful [7] and both versions of the model (the SUGRA model generated by Michmodel and its properly

[6] It is dangerous to set CONSTRAIN false in FMC0_RCP, because this will assume that the user is supplying valid model input parameters. No check is done on the generated model before proceeding with this flag turned off. See **CONSTRAIN** in the section on various running options for specific information on the constrains used.

[7] Success of these three routines is determined by including the Michmodel failure flag common block and setting the flag true when an error is encountered. These three routines were determined necessary to check by the trial and error method.

RMSS loaded SPYTHIA counterpart) pass the constraint, SPYTHIA is assumed to have been initialized successfully. This assumption has not been proven to be true in all cases; some explanation is required as to why two model versions are required.

SPYTHIA 1.x was designed to run with a generator very similar to Michmodel (perhaps due to Mrenna's lineage). The 100 variable returned array from Michmodel contained the same parameters and ordering as the PARSUSY common block used for calculating the SUSY decay table. With the upgrade to version 2.x, the initialization was changed to provide several running options. One of those options is the SUGRA approximate formulae mentioned above. Another is the general MSSM, which requires 21 parameters from RMSS; the other 70 or so can be calculated using generic MSSM formulae. The RMSS array is loaded with the appropriate values from Michmodel and the general MSSM option in SPYTHIA is chosen. Several issues arose because of this initialization alteration:

**The Stop Sector** The stop mass matrix definition varies between what SPYTHIA expects and what Michmodel generates. SPYTHIA is set up to calculate $\tilde{t}_1$ and $\tilde{t}_2$ from the left and right stop components, which it expects to be passed without D-term contributions. Michmodel includes the D-term contributions by default.

**The Higgs Sector** SPYTHIA uses a full two-loop corrected Higgs potential from $H_A$ and the Higgs mixing angle, whereas Michmodel only uses the one-loop corrections.

**The Squark Sector** SPYTHIA assumes the squark masses are passed without D-term contributions for all squarks.

**The Slepton Sector** SPYTHIA assumes that the slepton masses are passed with

D-term contributions.

A flag that controls whether Michmodel returns its fully corrected particle masses or particle masses appropriate for input to SPYTHIA. A message is printed when the D-term contributions are not included in Michmodel.

After a valid model is generated and SPYTHIA has been initialized, a call to FMCO_KCUTS_INIT (for kinematic cuts) initializes SSIM. The requested event analysis to be done is read from FMC0_RCP. Finally, within SSIM and during event smearing, calls are made to NPSMEAR_RUN_INIT and QSIM_PARAM_RUN_INIT once per program run, and NPSMEAR_EVENT_INIT and QSIM_PARAM_EVENT_INIT once per event.

## A.5 Various Running Options

We include here an exhaustive functional description of each running parameter in the order they appear in FMC0_RCP. Attempts have been made to protect the user from turning on incompatible options, but exhaustive testing has not been completed as of the writing of this note.

**MODELS_ONLY** If set to true, event generation is skipped and FMCØ becomes a shell for running Michmodel. Also, when true, the KCUTS block of the ntuple is not created. See also MODCOMP and EX_DTERMS for running in this mode. This flag should be false for normal running.

**WRITE_EVENTS** If set true, unsmeared *signal* events are written to a file called SPY_RUN-xx.SPY, where xx is the current run number. Essentially, FMCØ will become a shell for running SPYTHIA or PYTHIA with this option (see also SUSY_MC below). A signal event is defined by **FMC0_WRITE**. In normal running, this flag should be set false.

**BEG_RUN_NUM** Each model is assigned a unique run number, incremented by one for each model passing constraints if appropriate, beginning with this number.

**SUSY_MC** If this flag is set false, FMCØ will skip model generation via Michmodel and not create (most of) the sparticle mass parameter blocks in the ntuple. FMCØ becomes a shell for running PYTHIA in this mode. Also, with this flag false, only one run will be executed as it is assumed that the user is merely generating Standard Model events. This flag is useful with the **STORE_SMEARED** flag below. This flag should be set true for normal running.

**SUGRA** This flag allows FMCØ to loop over models that are not generated from Michmodel. When set false with **SUSY_MC** true, the **NS_FORMAT** and **NS_PARAM** arrays are read (see below). Running in this non-SUGRA mode, no model constraints are applied. This flag should be set true for normal running.

**CONSTRAIN** When true, general model and experimental constraints from LEP I are applied before proceeding with event generation. Note that this cut on the model parameters is executed after Michmodel successfully completes and SPYTHIA is initialized. The sparticle masses are read from the `PMAS` array during normal running or from the Michmodel output when MODELS_ONLY is set true. The gluino mass read is the physical mass from SPYTHIA or the $\overline{DR}$ running mass from Michmodel. The squark mass is defined as the average of the 4 first generation squarks ($\tilde{u}_{L,R}$ and $\tilde{d}_{L,R}$). The model is skipped if any of the below cuts are not satisfied:

*gtop_too_big* $= false$ This is the generic Michmodel error flag and is set true when any of five error conditions are satisfied during model generation: the CERNLIB routine `DDEQMR` fails to produce an accurate answer (the source code for `DDEQMR` was altered to facilitate this), `RTSEC` exceeds the maximum iterations evaluating the effective potential during the running of the Renormalization Group Equations, or any of the third generation Yukawa couplings exceed their theoretical maximum of $3\pi$, indicating that coupling is divergent.[8] In addition, this flag can be set true during SPYTHIA initialization by the routines `PYMSIN`, `PYCJDC`, and `PYNJDC`.

$|\mu_Z| \leq 50$ **TeV** This cut removes grossly unphysical models which Michmodel somehow solved numerically. The $Z$ indicates that the higgsino mass parameter has been evaluated at the electroweak scale (instead of the Planck scale).

$|\mu_X| \geq 0.25$ **GeV and** $h_A > 0$ Michmodel sets the value of the higgsino mass parameter at the Planck scale to zero when $\mu_Z^2 < 0$, which indicates that electroweak symmetry did not break when the higgs potential was minimized. A model with a negative scalar higgs mass indicates that Michmodel found the higgs potential unstable at negative infinity, which occurs when the field theory itself breaks down (minimum at $V = -\infty$) and all states will have infinite energy.

**Zero or Positive Scalar Mass** The physical stop, sbottom, stau, selectron, and sneutrino masses are required to be non-negative. Michmodel will transfer the sign of a sparticle's squared mass to its physical mass; when this sign is negative, the sparticle is tachyonic.

[8] This is the condition that bounds the SUGRA top mass to be less than 206 GeV/c$^2$. This upper bound is a function of $\tan\beta$.

**Neutral Weak LSP** Either the lightest neutralino or the electron sneutrino must be lighter than the physical stop, sbottom, stau, selectron, and lightest chargino.

**$h \geq 47$ GeV, $\tilde{e}_L \geq 45$ GeV, and $\tilde{\nu}_L \geq 37.1$ GeV** Light higgs, selectron, and sneutrino experimental limits from LEP I are imposed.

**$\widetilde{W}_1 > 45.2$, $\widetilde{Z}_1 > 41$ GeV, or $\widetilde{W}_1 > 47$, $\widetilde{Z}_1 < 41$ GeV** The gaugino experimental limits from LEP I are imposed.

This flag is recommended set true for normal running.

**NAT_CCB** If this flag is set true, additional constraints are applied on naturalness and charge-color symmetry breaking. These cuts are still rather controversial and have the effect of bounding SUGRA parameter space from above. Specifically, we require the standard charge and color symmetry breaking condition and set sparticle mass maximums at Anderson gamma less than 10 plus 200 GeV. [leave for a better treatment later...] This flag should be set false unless the user specifically **desires** a 30 minute diatribe with Mont during every presentation of results.

**FILE_READ** If set true, FMCØ will read SUGRA input parameters from a file instead of randomly generating them. The file (specified by **SUGRA_FILE**) is read using free format REAL*4, and the parameter order must be $m_{top}$, $\tan\beta$, $m_0$, $m_{1/2}$, $A_0$, sign $\mu$. Normal operation requires the user to set this flag false. This option can be used to compare results produced with another generator. FMCØ will exit normally when an end-of-file is encountered.

**RANLUX Control** The next set of RCP parameters controls the generation of random points in SUGRA space. **SEQ_NUM** is the seed passed to **RANLUX**, and **LEVEL** is the luxury level (0-4 with the higher numbers "more" random, but slower to generate). The **MINN** and **MAXX** values for each of the six parameters denote the endpoints of uniform random generation. The **M0_BREAK** value will generate half of the models below this value and half above. The endpoints are allowed to be equal. To use the internal random SUGRA point generation option, **FILE_READ** must be false.

**MAX_MODELS** This value dictates the number of valid models to generate. For the **FILE_READ** option, this value should be set equal to or larger than the number of parameter sets in the file to process all points.

**SAVE_SMEAR and SAVE_UNSMEAR** These two flags control an unconditional write to the files **SMEAR_FILE** and **UNSMEAR_FILE** for all events that are loaded into ZEBRA banks (see **CHECKSUM** for an option regarding events that will not be loaded into ZEBRA). Note that the models are not split up into separate files as they are for the **WRITE_EVENTS** flag. Normally, the user should set these flags false, and the events will be discarded after processing.

A dependency exists between the **WRITE_EVENTS** and **SAVE_SMEAR** flags. When **WRITE_EVENTS** is true, only those events that pass channels enumerated in **FMC0_WRITE** (see below) will be written to the smeared stream, allowing the user to write a consistent set of events in both smeared and unsmeared streams. Note that the smeared file is still concatenated if multiple runs are processed. The **SAVE_UNSMEAR** flag has no such dependency.

**FMC0_KCUTS** This array contains the user-defined kinematic channels to be analyzed. Currently, a hard limit of 100 channels maximum is imposed. See the section on kinematic channels for more information.

**NEVENTS_REQ** This integer sets the requested signal statistics to control termination of the event loop. We define a signal event as the logical AND of all kinematic channels in in the **FMC0_REQ** array. One may reference the channel from its ordinal occurrence in the **FMC0_KCUTS** array or by its name. If a partial name is given, FMCØ will key on the first ordinal match of the partial name. The channel names are case insensitive.

**FMC0_WRITE** When **WRITE_EVENTS** is true, any event passing the logical OR of the channels listed in this array will be written out. The channel name is referenced as in the **FMC0_REQ** array.

**USE_SMEARED** When set true, FMCØ uses the smeared event to make the decision to increment the signal event counter and/or write the event. No separate control for each file exists, because such an option would lead to an inconsistent set of events when both smeared and unsmeared streams are written. If this flag is false, the unsmeared event is used to make these decisions.

**DR_EJCUT** This real number controls whether a jet is filled in the `EVENT_VALUES` array. If a PJET is less than this distance away from a selected electron in DR, the PJET is skipped. This method is a terrible way to make this decision, and it it not recommended to set this value to any positive number. When set to zero or a negative number, the electromagnetic fraction of the jet is calculated from referenced the ISP1 banks (from the associated PJPT bank). Jets are excluded if their electromagnetic fraction is less than 5% or greater than 95%. This corresponds to the standard offline jet identification criteria. If the parameter is set exactly to zero, the jet is excluded if an ISP1 bank corresponding to an `EVENT_VALUES` filled electron is present in the jet.

**ETAMAX values** For each object, these values require $|\eta| \leq etamax$ in order to be filled into the `EVENT_VALUES` array. Detector eta is always used.

**MODPHI values** This cut is only currently implemented for electrons. We calculate the value

$$x = \mathrm{mod}\,(\phi, \frac{\pi}{16}) \div \frac{\pi}{16}$$

and require that $modphi \leq x \leq 1 - modphi$ for an electron to be filled into the EVENT_VALUES array. This value represents a percent distance from the crack cut and currently is applied to all electrons in FMCØ. Setting this value to zero is recommended to effectively turn off the cut; QSIM will correctly apply a fiducial cut for the central calorimeter only when the electron is smeared. The parameter is kept active for generator level correlation studies.

**CHECK_EVENT** This flag provides a speed option that is intended cause the program to bail out of models from which sufficient statistics will never be reached within a reasonable amount of time. If true two checks are performed every 5000 events. The current number of signal statistics is divided by requested signal statistics and the program will bail on the model if this value is less that 2% of the ratio of the current number of produced events to the maximum allowed (`NUMBER_OF_EVENTS` from SPYTHIA_RCP). Also, the current number of signal events, with "signal" defined in **FMC0_REQ**, is divided by the total number of produced events and multiplies by the current total production cross section estimate (in effect, $\sigma \cdot BR$). If this value is less than 0.01 pb$^{-1}$, the model is also abandoned with the current statistics. If the number of signal events currently generated is zero, a one is used in the calculation to avoid a systematic loss of models where a rare decay of a high cross section process may produce observable numbers of events.

**CHECKSUM** One of the time consuming operations in FMCØ is the filling of the ZEBRA banks from the LUJETS common block. When set true, this option will check the LUJETS summary section for leptons and require **SUMMARY_NLEP** leptons above **SUMMARY_ET** before proceeding with the ZEBRA filling. If the event fails, the number of tried events is incremented and the program proceeds to the next event. The type of lepton cut on is controlled by **SUMMARY_FLAG**: 'E' for electrons only, 'M' for muons only, or 'EM' for either.

When running with **SUSY_MC** false, and when the number of found leptons in the summary part of LUJETS is exactly zero, the rest of the event is also scanned for leptons. This scanning will allow, for example, one to require $J/\psi$ to decay to electrons; the electrons from such decays do not appear in the summary part of LUJETS.

**ISOLMU** This flag, when true, will exclude all muons from entry into EVENT_VALUES unless they are 0.5 in dR away from all PJET banks. Note that the MUON flag in DOPJET_RCP should be set true to use this option (muons are excluded from jets).

**MC_INPUT** When this flag is set true, FMCØ will skip the event generation and instead read events from the file **MC_FILENAME**. Michmodel and SPYTHIA are still run according to the **FILE_READ** option. This allows the user to use the file read option to reconstruct the model spectrum used to generate the events in the file. It also allows the user to run the FMCØ analysis on events from any physics event generator, provided they are stored in the standard DØ ZEBRA format. DOPJET is not re-run. The ISAE tree is required to process the event. This flag should be set false for normal running.

**NTUPLE_FILENAME** This character value is file name for the FMCØ ntuple.

**NS_FORMAT** Used when the **SUGRA** option is false and the **SUSY_MC** option is true, this array describes the values which will be read from the file **NS_PARAM**. For example, if the **NS_PARAM** array consists of the elements `'RMSS(12)'` `'RMSS(1)'` `'RMSS(2)'`, then FMCØ will read the text file expecting the light stop mass, M1, and M2 for each model. One should use SPYTHIA_RCP to set static `RMSS` values.

This feature is actually more general than it appears. Any set of models that can be constructed with PYTHIA commands to `LUGIVE` theoretically will be allowed. The detailed protocol is that the common block value is read from the format array, an `'='` is added, then the real values read from the parameter file are passed directly to `LUGIVE` as commands. In the above example, if the first line in the parameter file was `100. 20. 20.`, the three commands `'RMSS(12)=100.'`, `'RMSS(1)=20.'`, and `'RMSS(2)=20.'` would be passed to `LUGIVE`. See the PYTHIA manual for more detail on setups.

**ZMASS_LOW and ZMASS_HIGH** These two values control the $Z$ boson mass window for the SPC_1 cut.

**EX_DTERMS** This flag controls whether D terms are excluded from the Michmodel squark mass caclulation. This flag should be set true for normal running, as SPYTHIA expects these terms to be excluded. Setting this flag false without **MODCOMP** true will yield erroneous mass values from SPYTHIA.

**MODCOMP** This flag controls the FMCØ model compare option. When set true, SPYTHIA is initialized with Michmodel masses controlled by **EX_DTERMS**,

Table A.1: FMC∅ Timing Breakdown

| **Function** | **Level 0** | **Level 4** |
|---|---|---|
| PYTHIA | 69.7% | 64.1% |
| **RANLUX** | 9.5% | 20.9% |
| **PYTEFL** | 18.0% | 13.0% |
| SSIM | 3.4% | 2.1% |

and Michmodel will always be re-run including the D terms so that the Michmodel spectrum in the ntuple is always correct. **MODCOMP** can be useful with the **MODELS_ONLY** option true, but it is not required. This flag should be set false for normal running.

**DOTIME** When true, the FMC∅ timing analysis is performed. DT_REPET is a global repeat loop over the scale factors for the various program elements, **RANLUX**, **PYEVNT**, **PYTEFL**, and SSIM. The number of repetitions executed is the product of the global repeat and the various scale factors. Setting this option true will significantly slow down the program.

## A.6 Timing and Performance

There are four main functions of FMC∅ for which timing information makes sense: event generation (PYTHIA), random number generation (**RANLUX**), ZEBRA bank filling (**FMCO_PYTEFL**), and the FMC∅ event analysis (SSIM). Several thousand events were processed and discarded with the timing option activated. FMC∅ was run with two different **RANLUX** levels, zero and four (see [38]). The percentage of time spent performing each function is summarized in Table A.1.

FMC∅ performance has been measured running under AXP OpenVMS. Four machines of various configurations[9] were used to obtain an average generation speed. FMCØ was run in normal mode with the SUMMARY_CHECK flag activated and selecting events with two electrons above 15 GeV $E_T$. In 146 hours of total CPU time, 5.5 million events were processed, an average performance of 10.5 events/second/CPU. Other Alpha processors were tested for comparison; the slowest processed about 6 events/second, while the fastest produced about 25 events/second. PYTHIA is the slowest function in FMCØ, and simpler production processes run faster; $Z$ boson events, for example, can be processed at a rate as high as 40 events/second on the fastest machines.

For both of these performance tests, the full RSIM and TSIM were not yet implemented. Because these functions will consist of simple look-ups, it is not thought that performance will be severely degraded.

## A.7 Setting up SPYTHIA

SPYTHIA uses the same initialization protocol as standard DØ PYTHIA. For the FMCØ implementation, we use the file SPYTHIA_RCP to pass general setup commands. For those unfamiliar with PYTHIA, we include a brief introduction here. Variables not described here should be left alone by the first-time user.

**Maximum Number of Events** The variable NUMBER_OF_EVENTS sets the maximum number of events to generate per model as described in the section on FMCØ structure.

[9] Alpha processors running between 133 to 266 MHz were used in this test. Specifically, DØTNG cluster nodes DATAM2, DASB02, DATT01, and DABR01.

**Printing Events** Setting the flag `DO_PRINT` true will cause SPYTHIA to print various levels of detail of the first `NUMBER_OF_EVENTS_TO_PRINT` events based on the value `LEVEL_OF_PRINT`. At print level 1, a summary of the hard interaction is echoed to the PYTHIA default output device (standard terminal output by default, which can be reset with `MSTU(11)`). Print level 2 will dump the whole event common block. Print level 12 will print the decay table for the current model. The decay table information is useful for obtaining KF codes to turn on and off decay modes (see below). This flag is recommended false for normal running.

**Setting the Parton Distribution Function** With this command, as with the others below, the character entry in the array is passed to `LUGIVE`. `LUGIVE` will allow the user to change any common block variable by passing a character value "varname=value"; for example, to change the PDF from an internal set to a PDFLIB set, one would `CALL LUGIVE('MSTP(52)=2')`. This change tells PYTHIA to get the PDFLIB structure function described by `MSTP(51)`. `MSTP(51)` is a single integer code equal to 1000*Ngroup + Nset; for example, CTEQ3M is group 4 set 30, so `MSTP(51)` should be set to 4030.[10] All PYTHIA functionality is controlled from the common block variables. See the PYTHIA manual for further details.

**Setting the Production Process** Groups of processes can be selected by setting `MSEL`. `MSEL=39` will produce all SUSY particles except for Higgs. Various `MSEL` options are listed in [34]. To get *any* combination of production processes, set `MSEL=0` for direct process control. The array `MSUB(400)` controls whether each of

[10] A list of the group and set codes can be found at

http://wwwcn.cern.ch/asdoc/pdflib/pdflib.ascii

Note that the DØ release of PDFLIB might not contain all of the newest sets listed at this site.

the 400 production processes available in SPYTHIA is turned on. The default value is `MSUB(n)=0`; set `MSUB(n)=1` to turn on a specific process. For example, `MSUB(243)=1` will turn on the process $gg \rightarrow \tilde{g}\tilde{g}$. See [34] for a complete list of the available SUSY production processes.

**Turning off Decay Channels** For studies in which one is looking for a rare decay mode (low branching fraction) of a rather high cross section process, such as W1Z2 decaying to 3 leptons, it may be efficient to turn off various decay modes of certain particles. Each decay mode is assigned a unique IDC code; for example, IDC=231 is the decay $W \rightarrow \overline{d}u$. To turn off a decay mode, issue the command `'MDME(IDC,1)=0'`. The decay table contains the IDC codes for each process and can be printed out with a print level 12 described above. It is recommended to only do this for a single event, single model run.[11] If a range of values is to be turned off, FMCØ contains a protocol to allow this. For example, to turn off all Z boson decays except to electrons, one may issue the commands `'MDME(156:163,1)=0'`, and `'MDME(165:171,1)=0'`. These two commands will turn off all Z decays except KF=164, $Z \rightarrow ee$. (For the B meson $J/\psi$ study, it was quicker to write a translation routine than to try to type in and verify the 200 or so channels that needed to be turned off.) The protocol specifies that all commands containing a colon are processed through `FMCO_LUGIVE`, a routine that translates the specified range into `LUGIVE` commands.

[11] Presently, one can view a decay table at

tmp$root320:[genik.susy.mc.fastmc0]TEST_140_90_TABLE.DAT

in lieu of generating one. Note that the IDC codes are model dependent. Section 14.6 of the Pythia manual has further information.

**NOTE :** PYTHIA, unlike ISAJET, will reduce the production cross section based on the new number of available decay channels. A complete description of this adjustment can be found in section 7.6.2 of the PYTHIA manual. This adjustment is not successful in all cases, particularly when a significant cascade is turned off. For example, when studying $J/\psi$ events from B meson production, the reduced BR for turning off all but $J/\psi \to ee$ did not propagate back up the chain to the meson production cross section. This effect is caused by turning off channels in an order that prohibits their complete propagation to the particles actually being produced. The process of reading and understanding again the order of the calculation from the PYTHIA manual is more efficiently replaced by letting all processes be on for a limited run, finding $\sigma * BR$, and repeating the process with the appropriate channels turned off and comparing $\sigma'$. If the two match for a few models, they should match for all models.

**The IMSS array** All values used in SPYTHIA are documented in the sample RCP file along with their default values. The parameters that must be changed from their defaults are not commented out: for normal running this includes `'IMSS(1)=1'` for a general MSSM simulation and `'IMSS(8)=0'` to let SPYTHIA find the physical stau masses.

Other PYTHIA command are issued in the sample RCP. `MSTP(126)` controls the maximum number of lines in the summary section of the `LUJETS` common; we set it from its default (according to the PYTHIA 5.7 manual) of 20 lines to a maximum of 40, because the SUSY cascades can occasionally exceed 20 total particles. (The SPYTHIA 2.10 distribution defaults to 40.) `CKIN(3)`, the minimum $p_T$ for a $2 \to 2$ interaction, is set to 10 GeV to avoid divergent production cross sections near the origin. `MSTP(81)` is set to zero to disallow multiple interactions.

`MSTU(11)`, the unit number for standard PYTHIA output, is set to 87 from its default of of 6 (the screen). This change limits the number of PYTHIA warning messages that will appear in a log file that may obscure FMCØ error messages. One can expect a number of "cross section violation" and "infinite loop" messages to be generated from PYTHIA. The first of these messages merely indicates that the initial production cross section maximums have been exceeded during event generation, where better estimates are made of the true differential distributions. The latter messages are generated when PYTHIA detects it is having problems. Normally, if a serious error occurred that would make the event not useable, PYTHIA issues a Fortran STOP. The FMCØ treatment of these errors is described in the section on infinite loops. Essentially, FMCØ will allow PYTHIA to try to recover and if it can not, FMCØ will proceed to the next event.

The default list of commands are standard DØ PYTHIA setup commands, except for setting `MSTP(127)=1`; this parameter setting turns off the printing of the header information, which is constant for each model.

**NOTE:** The calculation of $Q^2$ differs between PYTHIA and ISAJET. The calculation is controlled via `MSTP(32)` with a default value of 2. The user may set `MSTP(32)` to 1 to obtain the ISAJET default. See section 9.3 of the PYTHIA manual for more information.

## A.8 Loading the RMSS array

The routine `FMCO_LOAD_RMSS` places the output from Michmodel into the appropriate `RMSS` entry. A key for the complete translation is available from the author's home-

Table A.2: Loading of the RMSS array from the Michmodel output array PASS(94). The * indicates that the value has been modified: for $\mu$ and $h_\alpha$, the signs have been changed. All the trilinear terms have been multiplied by $m_0$. The two parameters $\tilde{u}_R$ and $h_\alpha$ are not used in normal running but are loaded for the user's convenience.

| RMSS entry | PASS entry | Description |
|---|---|---|
| 1 | 51 | bino mass at the electroweak (EW) scale. |
| 2 | 52 | wino mass at the EW scale. |
| 3 | 53 | $\overline{MS}$ gluino mass at the EW scale. |
| 4 | 13* | $\mu$, higgsino mass parameter at the EW scale. |
| 5 | 11 | $\tan\beta$, ratio of the Higgs expectation values. |
| 6 | 34 | $\tilde{e}_L$ mass at the EW scale. |
| 7 | 40 | $\tilde{e}_R$ mass at the EW scale. |
| 8 | 21 | $\tilde{u}_L$ mass at the EW scale. |
| 9 | 28 | $\tilde{d}_R$ mass at the EW scale. |
| 10 | 26 | $\tilde{b}_L$ mass at the EW scale. |
| 11 | 32 | $\tilde{b}_R$ mass at the EW scale. |
| 12 | 31 | $\tilde{t}_R$ mass at the EW scale. |
| 13 | 38 | $\tilde{\tau}_L$ mass at the EW scale. |
| 14 | 44 | $\tilde{\tau}_R$ mass at the EW scale. |
| 15 | 19* | $A_b$, bottom trilinear coupling at the EW scale. |
| 16 | 18* | $A_t$, top trilinear coupling at the EW scale. |
| 17 | 20* | $A_\tau$, tau trilinear coupling at the EW scale. |
| 18 | 50* | $h_\alpha$, the higgs mixing angle. |
| 19 | 47 | $H_A$, pseudoscalar Higgs mass at the EW scale. |
| 20 | 8 | $\alpha_{GUT}$, the GUT scale coupling constant. |
| 22 | 27 | $\tilde{u}_R$ mass at the EW scale. |

page. Two issues are of note on the translation between SPYTHIA and Michmodel. First, we change to the Haber-Kane convention for $\mu$ and $h_\alpha$ by switching the sign on both paramaters, and second, we multiply the trilinear couplings by $m_0$ to obtain their value in GeV/c$^2$. Table A.2 contains the parameters as loaded for each model.

## A.9 Using a Generator Other Than SPYTHIA

SPYTHIA is embedded in the FMC∅ structure and it is a non-trivial task to not only change the generator, but to determine what re-initialization would be required for a different generator, such as ISAJET; however, FMC∅ allows the user to not generate events internally, but instead read them from a single file. The file must contain the complete ISAE structure, including PJET banks. One can run in this mode by setting the flag MC_INPUT to true in FMC0_RCP and identifying the file in the array MC_FILENAME. If desired, one may also turn on the FILE_READ option and indicate a file that contains the SUGRA input parameters used to generate the event file being processed; this will cause FMC∅ to generate the SUGRA model parameters, via Michmodel, for inclusion in the ntuple. It is advisable to turn off all model constraints in FMC0_RCP and to turn off the EX_DTERMS flag to get the squark masses correct.

## A.10 Adding Analysis Cuts

SSIM only examines the `EVENT_VALUES` arrays when determining whether an event passes a kinematic channel. For physics objects, only $E_T$ is specified. Several hooks are allow the user to add *special* cuts. These cuts are the values which will be stored in the SPC_1-7 entries in the `EVENT_VALUES` arrays. For example, SPC_1 is a cut on the invariant mass of the two leading selected electrons to eliminate Z bosons from a signal sample. The protocol for SPC cuts is that they return 2.0 if passed and -2.0 if failed, which fits in well with the SSIM protocol that allows the user to enter 1.0 for requiring the cut and -1.0 for requiring failure of the cut. Frame code for the SPC cuts is available for modification by the user in the routines `FMCO_SPCn.FOR`.

Note that the same routines are, by design, used to determine pass/fail for smeared and unsmeared events. To perform global cuts, one has the option of accessing the FMCØ event common (FMCO_EVENT.INC), which is filled with physics objects entered into the EVENT_VALUES array; the QSIM commons (see QSIM documentation); or the ZEBRA banks directly as these are overwritten with the values returned in the object 4-vector from NPSMEAR. See the section on running FMCØ for specific details regarding how to modify the source code and re-link a local executable.

Missing $E_T$ requires special attention because no ISAZEB equivalent of a PNUT bank exists. To store the missing $E_T$ calculated by QSIM, we have created a new bank, FMC0. This bank hangs off of the ISAC structural link from ISAE, the existing structural link to the pseudo calorimeter from ISAJET. This was done so that a fatal error would most likely occur if a user tries to use both a toy calorimeter and QSIM; such a configuration is tricky because the user might not know whether the values in the ZEBRA banks are smeared or unsmeared. The FMC0 bank structure is described in FMCO.ZEB. This bank contains six real values: $\not{E}_T$, $\not{E}_x$, $\not{E}_y$, and $\phi$ for calorimeter only, and $\not{E}_T$ and $\phi$ for calorimeter plus muons. Refer to QSIM documentation for details regarding these two calculations.

The FMCØ event common contains physics objects present in the EVENT_VALUES array. Not all particles in an events are incldued; for example, an electron that falls into the ICR will not be filled. Specifically, the include file is

```
      INTEGER MAXENTRY, NCOLMNS, OBJNUM
      INTEGER CHID_ELEC, CHID_JET, CHID_MUON, CHID_MET, CHID_MMET
      PARAMETER( CHID_ELEC=4HELEC, CHID_MUON=4HMUON, CHID_JET=4HJETS,
     +  CHID_MET=4HMSET, CHID_MMET=4HMMET)
      PARAMETER( MAXENTRY = 50 )
      PARAMETER( NCOLMNS  = 11 )
      PARAMETER( OBJNUM   = 5  )
C SORT_LOC is the order in which the objects are filled. They are in
C descending order of Et within each object class. SORT_NUM(OBJNUM+1)
C is the total number of objects stored.
      INTEGER SORT_LOC(OBJNUM),SORT_NUM(OBJNUM+1),SORT_POS(OBJNUM+1)
      DATA SORT_LOC(1)/CHID_ELEC/
      DATA SORT_LOC(2)/CHID_MUON/
      DATA SORT_LOC(3)/CHID_JET/
      DATA SORT_LOC(4)/CHID_MET/
      DATA SORT_LOC(5)/CHID_MMET/
C note: eta is detector eta in range -4.0 to 4.0
C event structure for each entry
C ID,     LINK, Px, Py, Pz, P, Pt, eta, phi, theta, mass
C Hollerith or INT, INT, R*4 ->
      INTEGER IEVENT(NCOLMNS,MAXENTRY)
      REAL REVENT(NCOLMNS,MAXENTRY)
      COMMON /FMCO_EVENT/ REVENT, SORT_NUM, SORT_POS
      EQUIVALENCE (IEVENT(1,1), REVENT(1,1))
      SAVE SORT_LOC
C
C define pointers to the object information
C
      INTEGER KFMCO_ID,KFMCO_LINK,KFMCO_PX,KFMCO_PY,KFMCO_PZ,KFMCO_P,
     +  KFMCO_PT,KFMCO_ETA,KFMCO_PHI,KFMCO_THETA,KFMCO_MASS
      PARAMETER ( KFMCO_ID=1, KFMCO_LINK=2, KFMCO_PX=3, KFMCO_PY=4,
     +   KFMCO_PZ=5, KFMCO_P=6, KFMCO_PT=7, KFMCO_ETA=8, KFMCO_PHI=9,
     +   KFMCO_THETA=10, KFMCO_MASS=11 )
```

The array SORT_POS holds the ordinal starting point of each type of object; if there are zero entries in an object class, the value will be equal to the next starting point.

## A.11 Random Number Generation

We use a replacement to the default PYTHIA RLU which calls RANLUX instead. The PYTHIA manual claims that 30% of the generation time is nominally spent generating random numbers. We find that RANLUX at level 0 does a bit better that this, while RANLUX at level 4 nominally agrees with this. Because the level is set in FMC0_RCP, the user is allowed to trade a speed improvement for lower quality random numbers. See Ref. [38] for further information.

## A.12 Infinite Loop Detection

Unaltered PYTHIA will check periodically during some loops to determine whether no progress is being made toward finishing the event. For example, LUPREP, the routine that rearranges partons along strings and checks flavors (*e.g.* makes sure that color-singlet systems make sense), may find that the seed particle choice disallows the complete evolution of the event to colorless final states; in this case, a new seed particle is selected and the evolution is attempted again. This continues until the current seed changes $4N$ times, where $N$ is the number of particles in the event record; if no proper combination is found, LUERRM issues a message and MSTU(23) is incremented. For this type of infinite loop, the event is never completely generated and should be discarded by the user. LUERRM will issue MSTU(22) such messages in a given run, then MSTU(21) controls whether PYTHIA continues or issues a Fortran

STOP after dumping the last event; the former running option may lead to true infinite loops.

This default method of stopping event generation is not appropriate when running over several models, as is done in FMCØ. We have identified two routines in which true infinite loops may be encountered during SUGRA event generation, LUPREP and LUKFDI. To address this issue, we have modified these two routines, as well as PYEVNT and LUERRM, in addition to writing a shell routine to call PYEVNT that is appropriate for multi-model generation, FMCO_PYEVNT.

The shell routine FMCO_PYEVNT zeroes MSTU(23) before calling PYEVNT and locally keeps track of the number of errors encountered while attempting to generate the current event and the number of errors encountered while generating events within the current model. If either counter exceeds a maximum, a message is issued, and an error exit is requested from FMCO_RUN_SPYTHIA to flag the current model to be discarded. FMCO_PYEVNT will allow up to 10 consecutive tries to generate an event, and up to 50 failed events within a model. With this shell routine protection, the PYTHIA routines were modified as follows.

PYEVNT sets MSTU(24), the code of the last error encountered, to zero before calling LUPREP. If an infinite loop is encountered during evolution[12] the remaining generation is skipped, and we return to FMCO_PYEVNT.

LUPREP can enter an infinite loop while generating a new quark or diquark flavor to combine with an existing flavor to form a hadron (calls to LUKFDI). Usually, this loop is successful after a few calls so no protection is included in default PYTHIA.

[12] Section 14.4 of the PYTHIA manual contains a list of the single digit MSTU(24) error codes. When calling LUERRM, a programmer should add 10 to these codes to indicate that they are errors and not warning messages; warnings are counted separately and cannot stop execution. To immediately stop execution, pass a code larger than 20 to LUERRM.

The modified routine will allow 100 such calls before issuing an infinite loop message via LUERRM (thus flagging the event to be discarded) and returning to PYEVNT. If an infinite loop is encountered in LUKFDI that was not recovered from (determined by return status codes on the success of the quark generation), the event is immediately flagged as bad.

LUKFDI can enter an infinite loop while generating a diquark flavor. Again, the modified routine is allowed 100 attempts to recover before flagging the event as bad and returning.

With these modifications, LUERRM may be called many more times than normal within an event before recovering or flagging the event as bad. To avoid stopping program execution in these instances, LUERRM will issue a “trying to recover” message after MSTU(22) errors, when current event will be dumped and PYTHIA will be silent until 1000*MSTU(22) errors occur. If PYTHIA doesn’t successfully recover, a “could not recover” message is issued, and execution is stopped.

## A.13 Running FMCØ Under VMS

FMCØ and related utilities are not currently released into the DØlibrary. The official area for FMCØ is tmp$root320:[genik.susy.mc.fastmc0], located on the DØ AXP cluster. FMC0_SETUP.COM will define all necessary logicals and the symbol FMC0, which will run the official executable. If in the current default directory there does not exist a file FMC0.RCP, FMC0_SAMPLE.RCP will be copied; the same is true for SPYTHIA.RCP. Immediately issuing the FMC0 command will run a short sample program. The user should modify the RCPs appropriately. Several other RCP files are defined in this setup (*e.g.* those required by QSIM); the user is advised against modifying these files.

If the user needs to add cuts and re-link the program, the command procedure `FMCO_LINK.COM` should be used. The user should include all replacement subroutines in a library called `FMCOUSER.OLB`. The link procedure accepts one parameter to aid the user; if omitted, the program will execute a re-link. The possible options are

**COMP** This option will re-make the library `FMCOUSER.OLB` from all `.FOR` files in the current directory and link a new executable.

**DEBUG** This option will re-make the library `DEB_FMCOUSER.OLB` from all `.FOR` files in the current directory and link a new debug executable.

**DLINK** This option will re-link a debug executable.

**HELP** This option will print a list of these options.

**anything else** If a parameter other than those enumerated above is passed, the **COMP** option is executed.

**NOTE:** This procedure will delete all object files in the current directory. If only a couple of routines in `FMCOUSER.OLB` have changed, it is recommended to interactively compile an appropriate version and use the command

```
LIBRARY/REPLACE FMCOUSER.OLB objects
```

where `objects` is either a comma separated list of object files (no comma for a single object file) or `*.obj` to reference all object files in the current default directory.

## A.14 Expected Error Messages

FMCØ will generate several informational warning messages during generation: the bulk of these are normal. The reason for the messages is the extensive internal error checking installed during program development. Since the program may be considered as constantly under development, these messages have been retained in a limited fashion. Ten expected messages will be printed to the screen for informational before the program goes silent. The user can tell the expected messages when a `LAST WARNING` will is issued. This protocol provides the user with breakpoints to set during debugging to ensure that some code modification did not change the nature of the expected error. We present a list of the expected error messages and their meaning.

FMC0_D0_COORDS... Particle with Zero total P In addition to the obvious hint of a problem, this message will normally be generated when a particle fails a detector fiducial cut. For example, an electron passing into the ICR.

## A.15 Looking Toward Run II

FMCØ is structured such that specific detector configurations are nearly irrelevant. Resolutions, trigger, and identification efficiencies are parameterized from data, or from full MC simulations when data is not yet available. These parameterizations are used by QSIM and no significant upgrade of FMCØ is required to simulate the Run II hardware environment; however, the Run II software environment will require some changes in the future.

Several dependencies exist for FMCØ on Run I software: the jet constructor, ZEBRA storage, and various DØ utilities such as SRCP. As these packages become obsolete, Run II replacements will need to be included in FMCØ. The program was written and designed with this upgrade planned and this accounts for the modularity of the routines that allows a staged upgrade. In addition, PYTHIA 5.7 is no longer supported by its authors and an eventual upgrade to PYTHIA 6.x will be necessary.

## A.16 Conclusions

FMCØ is a complete Monte Carlo signal analysis package that scans a section of model space to produce output that allows one to quickly predict a final $\sigma \cdot BR$ for each model in a variety of channels. The program can also be run as a shell for running any model or set of models allowed in SPYTHIA 2.10, including the Standard Model. In addition, an option is available for producing SUGRA models without generating events. It has been shown that the FMCØ full detector simulation can be run with only a 20% decrease in speed over merely generating events. Finally, the FMCØ framework will provide an essential tool for new physics analyses using the upgraded DØ detector.

# Appendix B

# The Author's Contribution to DØ

In large experiments such as DØ, it is becoming standard to include an Appendix informally describing details of a student's specific contribution. This Chapter serves that function. To facilitate a quick overview, I have included some jargon. Appendix A essentially is the last chapter formally dealing with the analysis. Following Appendices include additional documentation on some of the projects, where I have used the convention "we" to indicate "the scientific community talking to itself": "we did this" indicates work that I did.

### The Di-lepton Analysis

With respect to the di-lepton analysis, there are several subjects. I won't bother with a long list repeating what is contained in earlier sections, except to point out that unless otherwise stated in the text, all of the work described was work that I did. It should be noted specifically that I conceived, designed, and authored FMCØ.

### Model Comparisons and Generation

One of the various cross-checks done was to compare the model spectra between Michmodel and the ISAJET generator SUGRA. I have included this documentation as Appendix C.

Another study I did was to check the dimensionality of SUGRA model space. The idea here was that it was possible that there was some constraining hypersurface that could only be found be generating a large number of models and examining the results. This work involved approximately six months of CPU time and was conducted mostly on a single machine at MSU in the summer of 1994. I generated about 14 million random points throughout all possible values of the SUGRA parameters and ended up with around 1.4 million allowed inputs. This was a tremendous learning experience in getting into the guts of the calculations; but, in the end, I was unable to see any such constraining surface. This study produced several talks at New Phenomena meetings, and added to the institutional awareness of the theory.

### Statistical Comparisons: HDIFFB Through L2_VERIFY

The first project I worked on was to write a bin-by-bin statistical comparator for HBOOK histograms. The detailed write-up of this project is included in Appendix D.

The statistical comparisons used within HDIFFB found application in online trigger rate monitoring. The Global Monitor served as the last line of defense in determining if the data acquisition system was operating as expected. One of this person's responsibilities was to obtain a sample the of the 100 or so active Level 2 filter rates. By hand, each rate was to be compared with a reference number collected at a similar instantaneous luminosity. The problem here was that the task was very tedious and time consuming. The solution was to write an internal routine that compared the number of actual filter firings to a reference value, and return a single character to be placed next to the filter rate indicating the result of the test. A blank indicated that rates were nominal, while other characters indicated the rate was high or low. This effectively reduced the lines of output that required hand scanning to usually less than five. This system ensured that the Global Monitors were efficiently using their time to investigate the most likely place an error may be present, and greatly reduced the chance of operator error in hand scanning so many rates. This routine was implemented starting near the end of Run 1a and continued throughout Run 1b. Although low maintenance, a few hours of work were required to change reference values every time a new trigger list was released.

HDIFFB found a direct application in monitoring of L2EM. Trigger Examine produces a histogram containing what are known as IFAILED codes. These values indicate the specific tool that didn't pass within L2EM. The histogram can be compared with a reference histogram from a previous version of the executable to determine if any functional changes had occurred in the operation of Level 2 between releases. HDIFFB allowed the detection of arbitrarily small variations. I wrote a wrapper package, called Autocompare, that could check multiple run files against a reference, and output the exact location of differences in an easily readable format (Appendix E). This was originally used as a cross check during the intial parameter tuning of L2EM. It was also used to monitor L2EM for stability.

There were a couple of limitations in Autocompare: namely, it could only check one histogram ID within each file, and it wasn't programmed to accept two dimensional histograms. With the presentation of results of the L2EM monitoring project, the Level 2 group requested several extensions of Autocompare's capabilities: check multiple histograms across an arbitrary number of files, allow two dimensional histograms, and allow tuning of the output from a full Autocompare dump, to a single line per run. Given that this request expanded Autocompare from a single program to a couple of dozen routines, the new utility was called Mega Autocompare (MAC).

MAC is a stand alone program that, in some sense, is a translator (as opposed to a compiler). There are 25 commands in the MAC syntax,[1] allowing complete control of an arbitrarily large number of comparisons. I programmed it (via RCP) to check all 125 histograms in the output of Trigger Examine, and dump the exact location of any change from previous executables. With the presentation of results of the Trigger Examine histogram project, the Level 2 group suggested another application.

Every new Level 2 executable required substantial verification prior to release. This entailed several people running VMS_FILTER on reference STAs and checking the output, some 50 pages of text for each STA file, against the prior executable. There existed a utility, GRAND_FSUM, that would compare this information automatically, but it was thought that for safety reasons (letting a new bug slip into the Level 2 executable was not an option), the text output should at least be scanned by hand. Instead of hand scanning, a MAC Trigger Examine comparison seemed a viable option.

I won't go into all of the technical details of what eventually became known as the Level 2 Verification System, but a general overview is appropriate. I wrote a DCL package that would run Trigger Examine and MAC, as well as GRAND_FSUM on reference STAs. The reference STAs were provided by the five physics groups (New Phenomena, Top, b-Physics, W/Z, and QCD at the time), and stored on tape in the FCC vault. A single command would remotely submit the analysis of some

[1] See DØ$UTIL:MAC_COMMANDS_SYNTAX.DOC.

ten thousand events (quite a bit of disk space at the time), and indicate with a single line of text whether the user needs to examine any additional output. (It may of interest to note that anyone at DØ that ever used vaulted tapes and was emailed a "happy bear," is using utilities that were developed for Level 2 verification.)

With Level 2 release verification now a trivial task, I handed the project off to a new MSU post-doc after testing was complete. Run 1b saw the release of some 30 executables: to date, zero bugs have been found that were introduced after the Level 2 verifier was in place.

### Level 1.5 Calorimeter Trigger

I designed and built the Level 1.5 Calorimeter Trigger Readout Control P2 Paddle Board. This board performed a few essential functions within the Level 1.5 framework. Appendix F contains of description of this device.

I did the physics commissioning of the Level 1.5 electromagnetic trigger. This is described in Appendix G. Of additional note (*i.e.* not mentioned in Appendix G) is that the efficiency and algorithm selection was also a concern. Based solely on my work, two algorithms were employed, Em 2x1 and EmFrac, and one algorithm was rejected, EmIso.[48] I provided the W/Z group with efficiency curves showing that the thresholds selected were 100% efficient.

### Low Energy Electrons

I did several studies attempting to identify low energy electrons. Appendix H is a description of one of these. The main thrust of this work was devoted to measuring tracking quantities from high $E_T$ electrons, and using data-overlapped Monte Carlo electrons for calorimeter quantities. This work was dropped in late 1997 in order to focus on high $E_T$ electrons and adding muon channels to the di-lepton analysis.

### Miscellaneous

I was a member of the ad-hoc DEC Alpha group. This group was formed in 1993 to facilitate communication between persons involved in porting code to the DEC AXP platform. My specific contribution was tracking down compatibility problems involving CERNLIB.

There were numerous other projects I contributed to, I will merely mention a few: CAL_OVERLAP, DØFIX micro-DST streaming, fmss script development, Detector shifts, consulting work for new students and post-docs, and maintaining the computer equipment owned by MSU.

In addition to physics, I was involved in a few other noteworthy things. I served for four years (1992-96) on the Users Facilities Advisory Committee (UFAC). My main contributions here were pushing for upkeep of recreational equipment (*e.g.* one of the pool tables in the UC predated my birth, and didn't age as well) and categorically opposing all policies that treated graduate students as a source of income.

In 1994, I created the Fermilab Association of Graduate Students (FLAGS). This organization eventually evolved into the current Graduate Student Association (GSA) when it was pointed out that FLAGS wasn't the proper acronym for the previous name.

Finally, I was one of only a couple of bartenders licensed by the state of Illinois. Thus, at most of the DØ functions since 1992, I served the collaboration by serving the collaboration.

# Appendix C

# Comparison of Supergravity Model Spectra from SUGRA and Michmodel

## C.1 Introduction

The DØ New Phenomena Group has embarked on a project to develop a fast detector simulation which parameterizes our particle detection ability given only Monte Carlo generator level four vectors. In order to assign a systematic error for the generator used, we have compared momentum spectra from ISAJET with that from DØSPYTHIA. We see differences in the initial state parton momenta. To check whether the Supergravity models used in the event generators are equivalent, we compare model mass spectra for a set of equivalent input parameters. We see that the generators agree in almost all particle masses except light Higgs, gluino, and the stop left and right masses. After consultation with the authors of the two generators, we believe we understand all but the light Higgs difference. In the following, we outline the versions of the generators used, present comparative plots, and hypothesize on the origins of the differences.

## C.2 Generators and Constraints

The two generators used for the comparison will be referred to as SUGRA, for the ISAJET 7.21 supergravity model generator[61], and Michmodel, the Constrained Minimal Supersymmetric Model generator[39] used in the DØ implementation of SPYTHIA . For the comparison, we choose a random point[1] in six-space and try and solve the model using Michmodel. If successful, we impose additional experimental and theoretical constraints. Those satisfied, we pass the input parameters to SUGRA for solution. The spectrum of masses generated is written to an HBOOK ntuple and saved for analysis.

The additional experimental constraints imposed are from LEP I and enumerated as follows:

$$\begin{array}{rcl} m_{Z1} > 18.4 \text{ and } \tan\beta \leq 3 & \text{or} & m_{Z1} > 20.4 \text{ and } \tan\beta > 3 \\ m_{W1} > 47 \text{ and } m_{Z1} < 41 & \text{or} & m_{W1} > 45.2 \text{ and } m_{Z1} \geq 41 \\ m_{\tilde{l}} > 45 \text{ and } m_{Z1} < 41 & \text{or} & m_{\tilde{l}} > 0 \text{ and } m_{Z1} \geq 41 \\ m_{\tilde{\tau}_1} > 45 \text{ and } m_{Z1} < 38 & \text{or} & m_{\tilde{\tau}_1} > 0 \text{ and } m_{Z1} \geq 38 \end{array}$$

$$m_{W2} > 99,\ m_{Z2} > 45,\ m_{Z3} > 70,\ m_{Z4} > 108,\ m_{\tilde{\nu}} > 37.1$$

$$m_h > 47$$

[1] To generate uniform random numbers, we use RANLUX, V115 from CERNLIB, at luxury level 4. All bits are guaranteed chaotic.

For the additional theory constraints, we require that the charged sfermion and chargino masses are greater than the Z1 mass. (There are solutions with $m_{\tilde{\nu}} < m_{Z1}$, but these all occur at sneutrino masses of less than 30 GeV/c$^2$which has been excluded experimentally.) Note that this cut also kills off the negative squared scalar fermion masses, since Michmodel returns $m_{\tilde{f}} = SIGN(m^2_{\tilde{f}}) \times \sqrt{|m^2_{\tilde{f}}|}$. In addition, we require that the "traditional" charge and color symmetry breaking constraints are satisfied. Within the Michmodel spectrum, we impose

$$\begin{aligned} A^2_{\tau Z} &\leq 3(m^2_{H1} + \mu^2_Z + m^2_{\tilde{\tau}_R} + m^2_{\tilde{\tau}_L}) \\ A^2_{bZ} &\leq 3(m^2_{H1} + \mu^2_Z + m^2_{\tilde{b}_R} + m^2_{\tilde{b}_L}) \\ A^2_{tZ} &\leq 3(m^2_{H2} + \mu^2_Z + m^2_{\tilde{t}_R} + m^2_{\tilde{t}_L} - 2m^2_t) \end{aligned}$$

which most theorists view as necessary, but not sufficient constraints. The LEP I cuts are used because they are model independent, and the other cuts are considered extremely conservative.

In addition to the above constraints, we require that the model was actually solved. This translates into three conditions: $|\mu_z| > 0.01$, $h_a > 0.0$, and $g_{topX} < \infty$. The first condition is a flag set in Michmodel to indicate that various error conditions were violated, such as unification never being reached. The second condition indicates a breakdown in the field theory, since pseudoscalars cannot have negative mass. The last error condition is explained below.

For other studies, we have imposed constraints based on naturalness. These have not been used in this analysis.

## C.3 Generator Modifications

The original Michmodel generator code (circa 1994) would solve one model given the 6 input parameters. We wrote an interface to pass random points based on parameter limits in a Run Control Parameter (RCP) file.[2] In addition, it was found that Michmodel was susceptible to overflow errors while running the top Yukawa coupling to the GUT scale. This problem was solved by creating an error flag in a common block and inserting a set of cascade returns to the top level routine. The top Yukawa limit was set at $3\pi$ and all models where the running exceeded this limit were thrown away.[3] The limit condition was eventually applied to the bottom and tau Yukawas also. The effect of this change, "short-circuiting" the calculation when an error condition is detected, was to increase the overall speed of the generator by a factor of 10 while retaining all allowed solutions.[4]

Other modifications to the original Michmodel source code are limited to creating include files for the common blocks, correcting some occurrences of REAL*4's being passed into REAL*8's, which the DEC Alpha architecture doesn't automatically correct for, and the gluino physical mass calculation described below.

During testing of our program, we encountered two conditions which cause a crash in SUGRA and related routines (SUGRGE, SURG06, SURG26, SUGFRZ, SSMASS, and CERLIB's RKSTP). The first was a divide by zero error in SSMASS when calculating , $_R$; this was fixed by setting , $_R$ equal $\pi/2$ when the denominator was zero and should have no effect on the calculation. The second involved various history dependent overflows during the running of the Renormalization Group Equa-

[2] This is accomplished via the SRCP package, a standard DØ library utility, which allows one to alter internal parameters without recompiling and relinking, or entering them each run interactively.

[3] This is commonly referred to as the Infrared Fixed Point for the top Yukawa. It is responsible for the upper limit of approximately 206 $GeV/c^2$ on the top mass for all SUGRA possible scenarios.

[4] The maximum generated top Yukawa, after tens of millions of points, is 6.57 so far.

tions (RGEs).[5] This problem was solved via the above mentioned method of cascade returns; however, instead of comparing the numbers to a theoretical limit, the variables were checked to assure that they could be raised to the required power. To find the largest allowed machine dependent real number, we called HMACHI from HBOOK, which puts this number in the variable BIGP in the HCPRIN common block.

Initial runs of the comparison program showed a systematically lower value of the physical gluino mass for Michmodel. Although in agreement at the 12% level, the calculation was investigated for possible differences in the method. Both routines used a variation of the routine SSPOLE, which calculates the physical mass from the MSbar mass, the light squark mass, the two physical stop masses, the left and right sbottom masses, and the passed parameters $Q^2$ and $\alpha_3$. The calculation can be summarized as determination of a correction factor $f(\bar{\tilde{g}}, m_t, \tilde{t}_1, \tilde{t}_2, \tilde{b}_R, \tilde{b}_L, \tilde{q}, Q^2)$, then assignment of

$$\tilde{g}_{phys} = \bar{\tilde{g}}(1.0 + 0.0796\alpha_3 f)$$

where $\bar{\tilde{g}}$ and $\tilde{g}_{phys}$ are the MSbar and physical gluino masses, respectively, and $Q^2$ is the scale at which to perform the calculation. SUGRA uses this routine to calculate the MSbar mass from the physical mass by passing $-\alpha_3$ instead of $\alpha_3$, thereby making the approximation $f(\bar{\tilde{g}}) \approx f(\tilde{g}_{phys})$ and

$$(1.0 - 0.0796\alpha_3 f) \approx \frac{1}{1.0 + 0.0796\alpha_3 f}$$

[5] The condition was shown to be history dependent by saving the input parameters which caused the crash, then inserting them into SUGRA using the VMS debugger. SUGRA always solved the model on the second try. Many hours were spent trying to track down this bug to no avail. It was finally decided to just throw away these models since they constitute less than 5% of the space searched, and the study at hand wasn't affected. The parameter spectrum where these errors occurred is included in the Section C.6.

Michmodel used this call to SSPOLE to kludge INV_SSPOLE, which added to the passed argument list all required parameters previously taken from ISASUSY common blocks (all those mentioned except $\tilde{\bar{g}}$, $Q^2$, and $\alpha_3$), and changed the final correction to

$$\tilde{g}_{phys} = \frac{\tilde{\bar{g}}}{1.0 - 0.0796\alpha_3 f}$$

thereby making only the latter of the above two approximations. These are considered good approximations, but for the sake of this study, we altered SUGRA to call a new INV_SSPOLE, and Michmodel to call a new routine, IND_SSPOLE. INV_SSPOLE correctly inverts the approximation by calculating the correction factor $f$ iteratively with the MSbar mass from the last iteration until the MSbar mass is stable to within 0.5%. IND_SSPOLE (IND for INDependent) is a copy of SSPOLE with the additional required parameters for the calculation passed instead of read from ISAJET common blocks. IND_SSPOLE and INV_SSPOLE should be called with positive $\alpha_3$. These changes had the effect of straightening out the plot of $\tilde{g}_{phys}^{mich}$ verses $\tilde{g}_{phys}^{SUGRA}$, but the systematically lower Michmodel mass was still present.

The final modification to the Michmodel physical gluino mass calculation was that previously we were calling IND_SSPOLE with the calculated alpha(3), whereas SUGRA uses $\alpha_3 \equiv 0.12$. The calculated alpha(3) value nominally was 0.13. We changed it to 0.12 so that the calculation was now done equivalently in both routines, up to the agreement of the stop, sbottom, and squark masses. This had little noticeable effect in the comparison plot.

We also modified some of the DØLibrary source code which generates the random points. We found that a temporary routine which used the trilinear coupling in GeV/c$^2$instead of dimensionless had been mistakenly included. Once this was fixed, the 20% disagreement in the light physical stop mass was reduced to nominal agreement at the 8% level.

## C.4 The Sign of $\mu$

Many months ago, the Michmodel v. SUGRA comparison was done with a couple of points and through trial and error it was determined that the generators were expecting different signs of $\mu$. This has become ambiguous enough that we devote a section in this paper to the convention. In order to prove that the signs should be flipped with respect to one another, we ran 2000 models with the signs the same. A sample of the resultant plots are included in Section C.7. In addition to the relative sign difference in $\mu$, we also see that the output Higgs mass matrix rotation angle also contains a relative factor of -1.

Because of this relative difference, we have developed a local nomenclature: $\mu_G$ and $\mu_H$ for the sign conventions of Michmodel and SUGRA, respectively. $\mu_H$ seems to be more widely used, so we have defined $\mu_G \equiv -\mu_H$. At the theoretical level, the $\mu_G$ convention defines $\mu$ as positive in the neutralino mass matrix, while $\mu_H$, also known as the Haber-Kane convention, defines $\mu$ as negative. To illustrate some of the confusion this may lead to, let us take SPYTHIA as an example, since the author doesn't want to write a separate note on this issue.[6]

In the SPYTHIA 1.x releases, the model was defined by the parameters input to the common block /parsusy/ psusy(100). psusy(2:94) correspond to the output array pass(2:94) from Michmodel. pass(1) from Michmodel is a model identification

[6] This example contains technical detail of concern to persons writing an interface to SPYTHIA 1.x and is not directly related to the comparison at hand. It is included because it is not documented elsewhere.

number set equal to

$$\begin{aligned} pass(1) &= \mathrm{NINT}((\mathrm{SGN}(\mu)+1)\div 2)+10\times\mathrm{NINT}(m_0\div 10) \\ &\quad +10^3\times\mathrm{NINT}(m_{1/2}\div 10)+10^5\times\mathrm{NINT}(10(5-A_0)) \\ &\quad +10^7\times\mathrm{NINT}(m_t-100) \end{aligned}$$

where NINT is the nearest integer. SPYTHIA uses the $\mu_H$ convention internally and in an attempt to be general, keys on $psusy(1) > 70 \times 10^7$ to flip the input sign of $\mu$. This is fine for all $m_t > 170$ $GeV/c^2$, but runs into an ambiguity for lower top masses, such as the DØ or CDF di-lepton fits. The DØ interface to SPYTHIA always uses Michmodel so its smart enough to check that the condition on pass(1) is satisfied before filling the psusy array. SPYTHIA 2.x doesn't perform this internal sign flip so it is up to the user to assure that the Haber-Kane convention is followed for $\mu$ and the Higgs rotation angle, if needed.

## C.5 Results

For the current analysis, we chose 2500 points in six-space uniformly random to the ranges:

$$\begin{array}{lll} m_t = (180,181)\ GeV/c^2 & \tan\beta = (1.5,5) & m_0 = (40,200,500)\ GeV/c^2 \\ m_{1/2} = (40,125)\ GeV/c^2 & A_0/m_0 = (-2,2) & \mathrm{SGN}(\mu) = (-1,1) \end{array}$$

where the middle number in the $m_0$ range indicates that we chose half of the points above and and half below 200 $GeV/c^2$, but otherwise uniform random. The results are presented in HBOOK two-dimensional and profile histograms. Profile histograms are two dimensional plots where the $y$-axis point is the mean of all entries in the $x$-axis bin, and the error bars are the RMS deviation of entries within the $x$-axis bin.[7]

Figures C.1 and C.2 show the agreement between the generators in the squark and slepton spectra. Figure C.3 show that the light Higgs exhibits a systematic difference depending on the sign of $\mu$, while the heavy Higgs particle masses are in agreement to 3.5%. In addition, Figure C.4 shows that the physical stops are in nominal agreement, while the left and right stop masses disagree markedly. The difference in the left and right states is due to the Michmodel inclusion of F- and D-terms in the superpotential and their inclusion in the definition these states. These states aren't physical and their definition is rather arbitrary, so the fact that they differ is of little concern for the study at hand. The D-terms are sometimes not included in the raw stop matrix; rather, they are included during the calculation of the physical states. This appears to be that case with SUGRA. The $\tilde{b}_1$, $\tilde{b}_2$, and $\tilde{b}_R$ comparisons are in agreement.

For the charginos and neutralinos, Figure C.5 shows agreement to within 5%. For the heavy chargino and two heaviest neutralinos, we see some wiggle in the resolution of the agreement around 275 GeV/c$^2$, but this isn't thought to be important for this analysis. Figure C.6 shows MSBar gluino mass agreement within 13% between the generators, and a systematic lowering of the Michmodel physical gluino mass by about 10%. This is due to different approximations and corrections in the two generators; the MSBar mass is what is produced in event generators, so we concentrate on that. We have plotted the mass difference as a function of $\tilde{t}_1$, $\tilde{t}_2$, $\tilde{b}_L$, $\tilde{b}_R$, and $\alpha_3(W)$. We only find a trend in the $\alpha_3(W)$ plot, Figure C.7, and note

[7] We use the spread option in HBPROF.

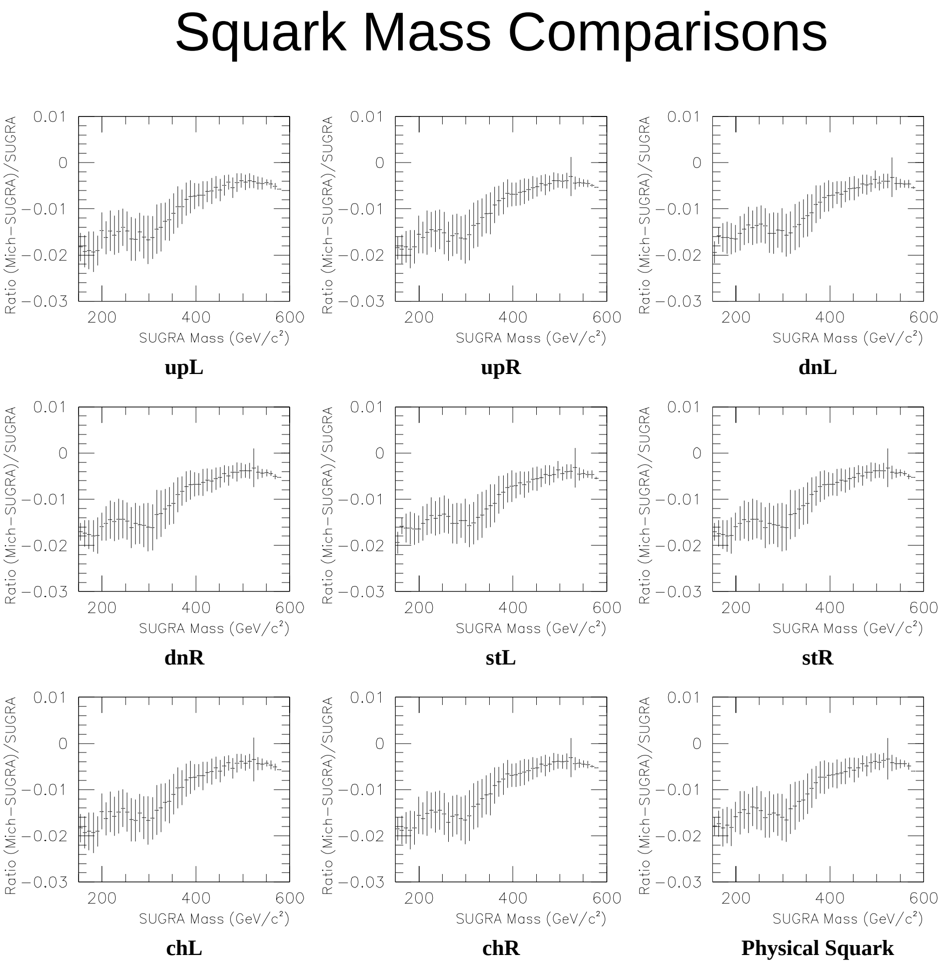
that there is a linear trend downward from +15% to -10% for $\alpha_3(W)$ going from 0.125 to 0.134. We also note that this same linear trend is apparent when comparing $\alpha_3(W)$ to $\tilde{\bar{g}}$. These plots indicate a correlation, although it may be indirect. Finally, we compare the calculated trilinear terms and Higgs rotation angle in Figure C.8. We see at 10% variation in $A_t$ and $A_b$, while only a very small ($< 1\%$) difference in $A_\tau$. All three trilinear terms show a divergence near zero. Once we take the sign difference into account for the Higgs rotation angle, we see agreement at the 4% level. SPYTHIA 2.x uses this angle to construct the Higgs sector for some optional setups. The pseudoscalar Higgs mass can be used to reconstruct the Higgs sector using the full two-loop corrections.

## C.6 SUGRA Failing Models

We show in Figure C.9 the input parameters for models which were solved by Michmodel, but caused overflows in SUGRA. We note that they were history dependent, and occur around $\tan\beta = (1.7, 2.0)$.

## C.7 Models With $\mu_G = \mu_H$

For the confirmation that we need to flip the sign of $\mu$ between the two generators, we added RCP parameter control and re-ran the above mentioned 2000 models. Figure C.10 show a sample of the spectra generated. If the two generators were using the same sign convention, we should see a clustering of the points along the diagonal; instead, we see two groupings of the points, indicating a different sign convention. When the signs are flipped relative to one another, we see the expected



upL

upR

dnL

dnR

stL

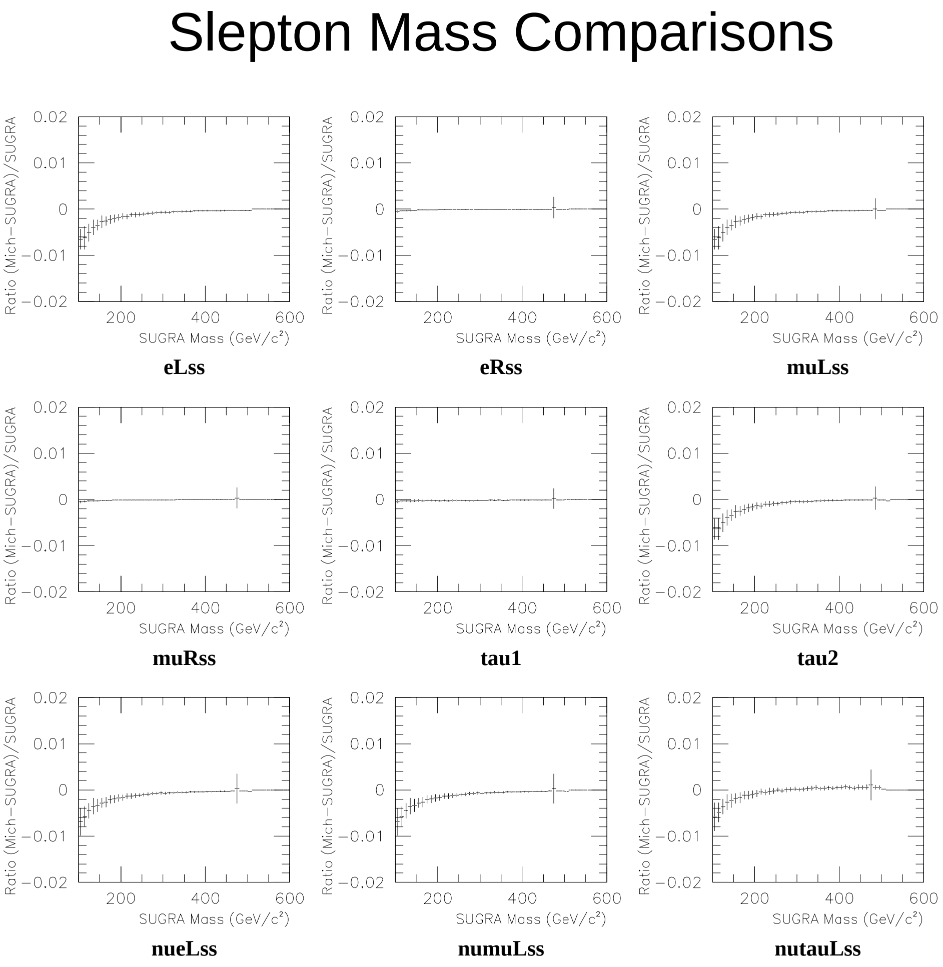
stR

chL

chR

Physical Squark

Figure C.1: Comparison of the mass spectra in the first two generations of squarks. We see agreement between the generators to within 2%.



eLss

eRss

muLss

muRss

tau1

tau2

nueLss

numuLss

nutauLss

Figure C.2: Comparison of the mass spectra for sleptons. We see agreement between the generators to within 1% for the left-handed sleptons, and no detectable difference for the right-handed sleptons.

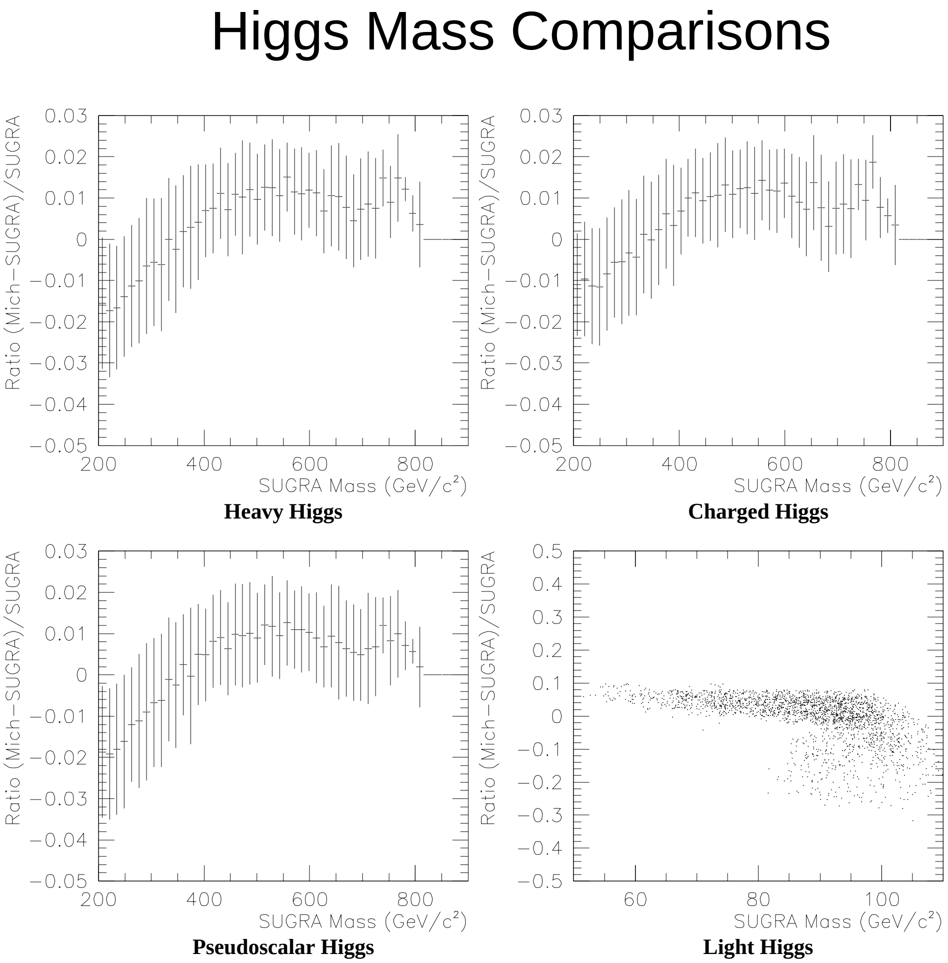
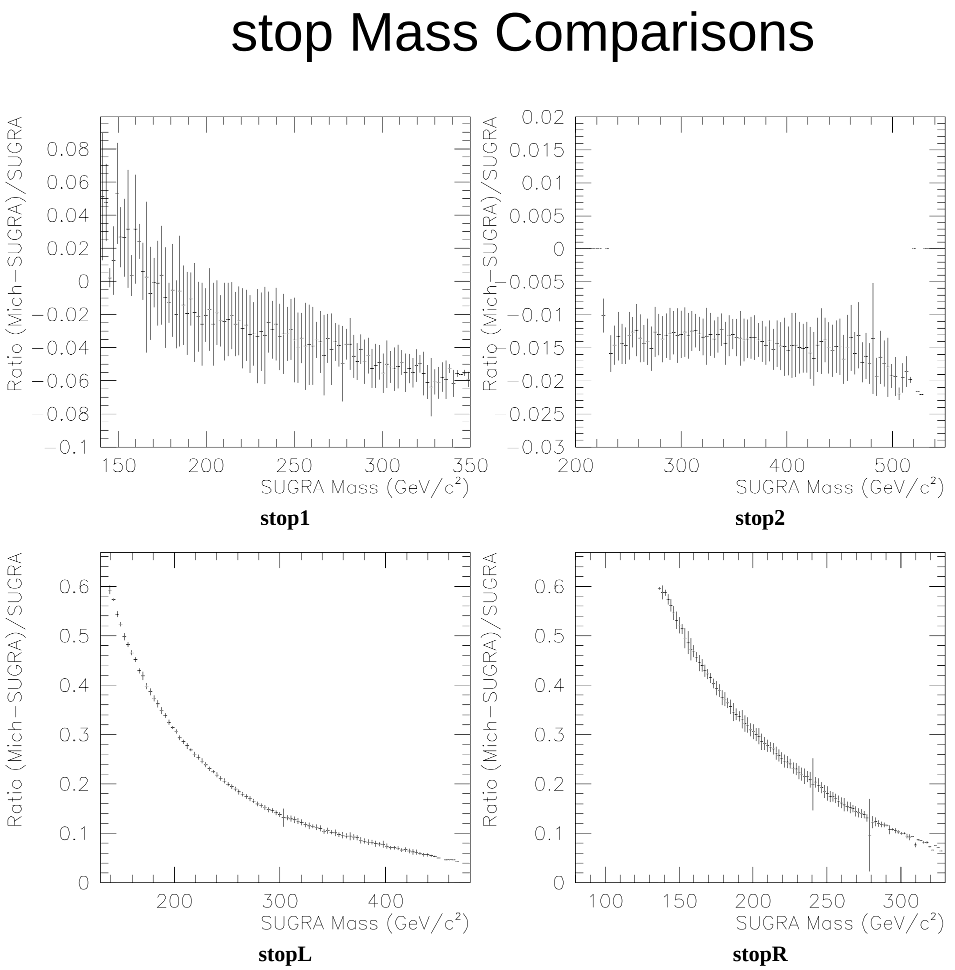


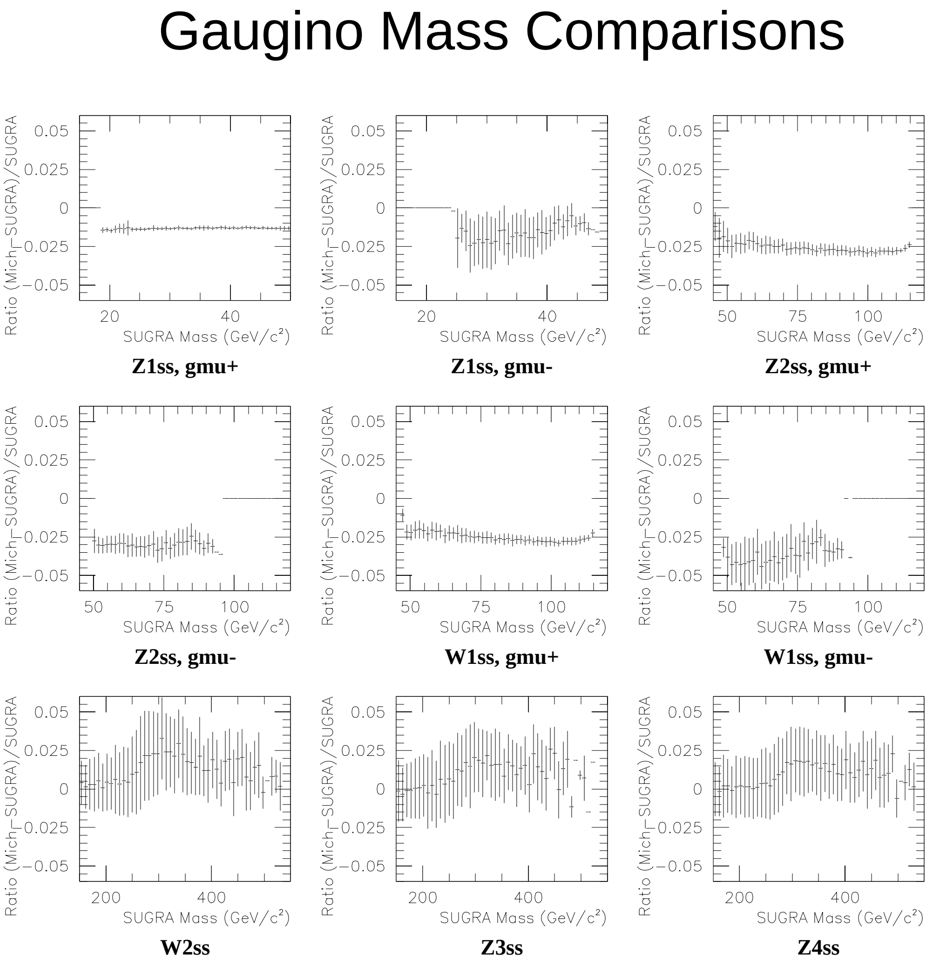
Figure C.3: Comparison of the mass spectra for the Higgs sector. We see agreement between the generators to within 3.5% for the heavy particles, but a large difference in the light Higgs. The two regions on the light Higgs “upside down pipe” plot correspond to different sign $\mu$ inputs, with the barrel of the pipe being $\mu_g < 0$, and the bowl being $\mu_g > 0$.



**stop1** **stop2**

**stopL** **stopR**

Figure C.4: Comparison of the mass spectra for the stop sector. We see agreement in the physical stop mass to within about 5%. The right- and left-handed stop masses disagree due to their respective generator dependent definitions.



Z1ss, gmu+

Z1ss, gmu-

Z2ss, gmu+

Z2ss, gmu-

W1ss, gmu+

W1ss, gmu-

W2ss

Z3ss

Z4ss

Figure C.5: Comparison of the mass spectra for the gauginos. We see agreement for all of the charginos and neutralinos within 5%. The agreement for the light gauginos, $\mu_g > 0$, is better.

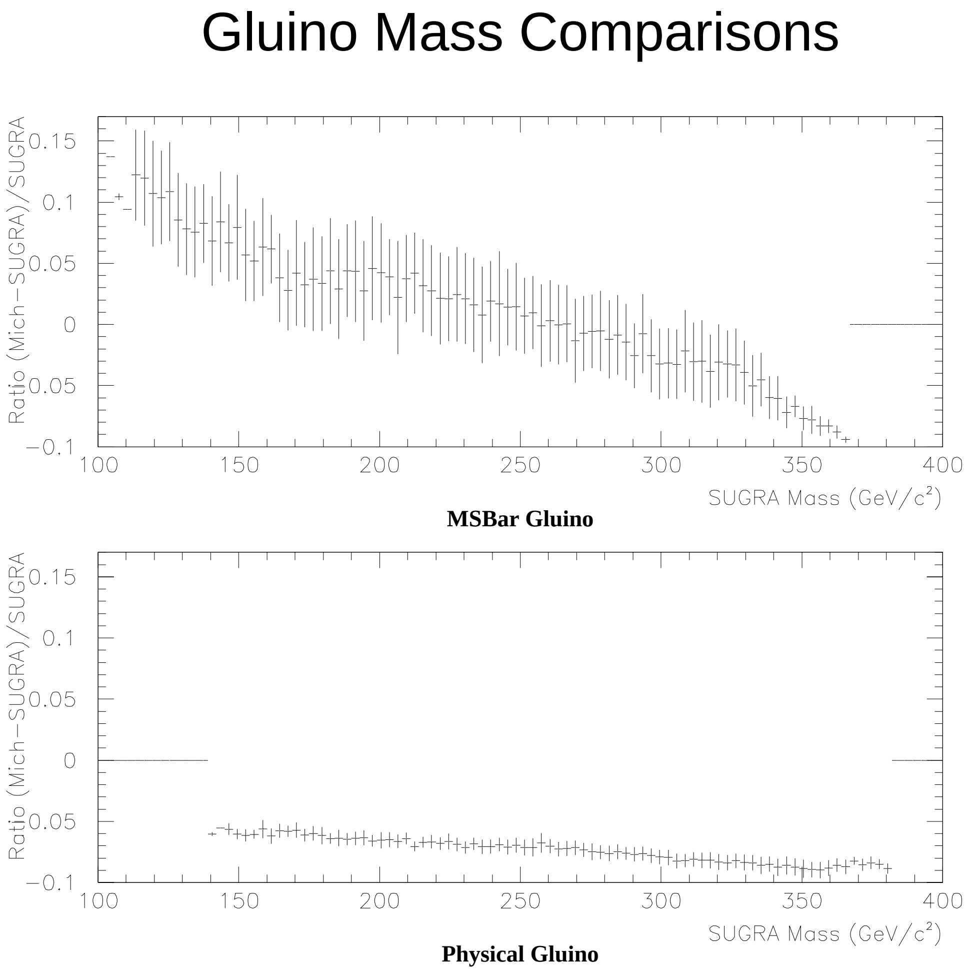
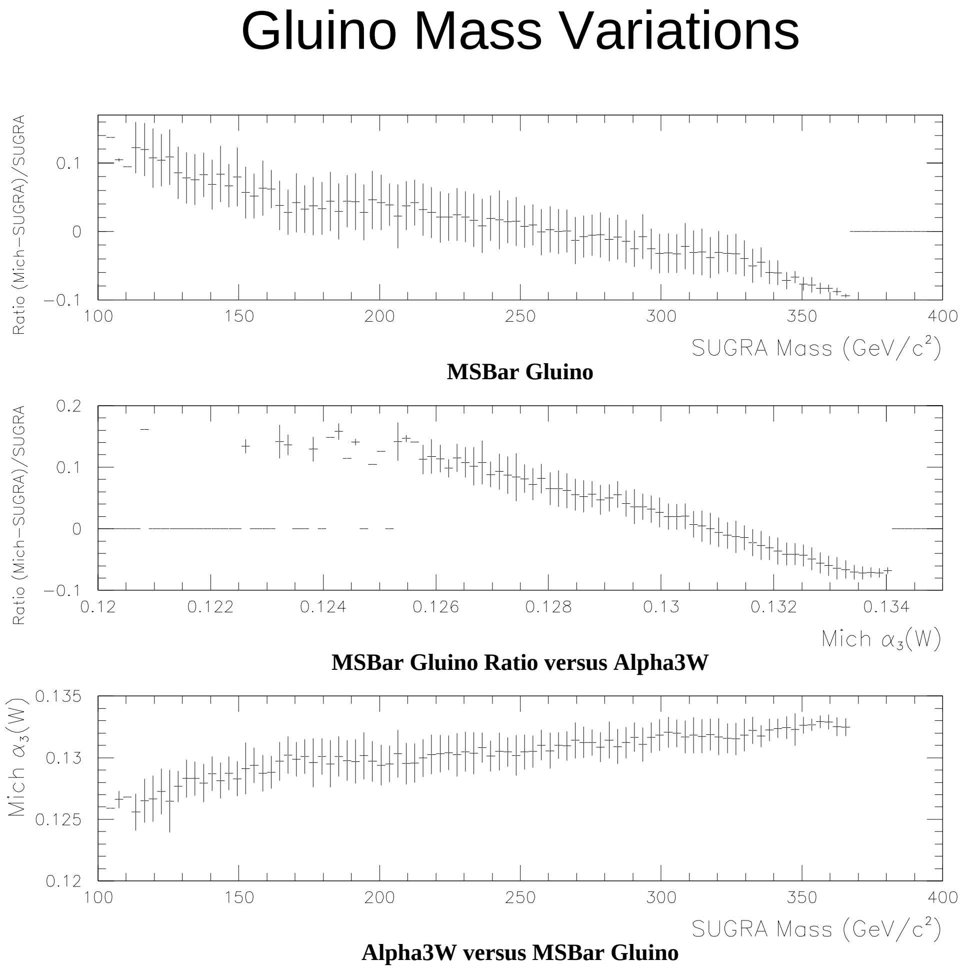


Figure C.6: Comparison of the mass spectra for the gluino. We see agreement to within about 13% for the MSBar gluino mass and a systematic lowering of the Mich-model physical gluino mass by about 10%. This is due to different approximations in the two generators.



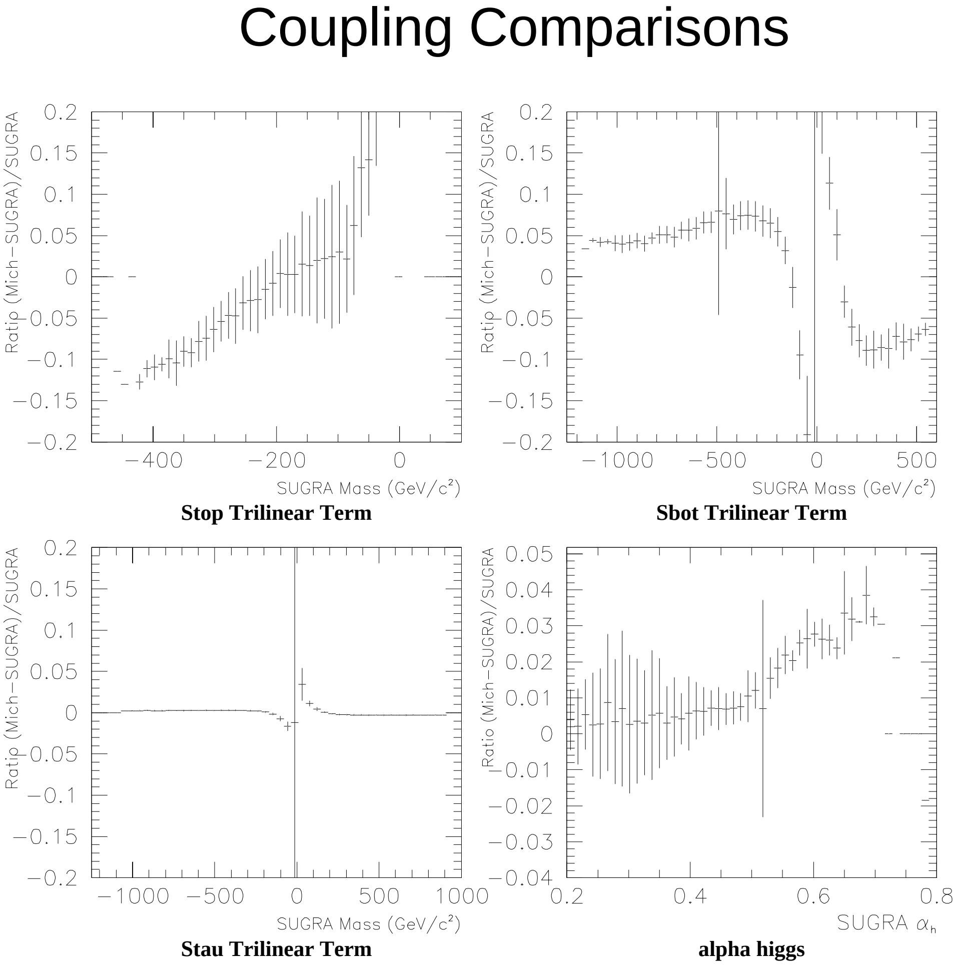
Gluino Mass Variations

MSBar Gluino

MSBar Gluino Ratio versus Alpha3W

Alpha3W versus MSBar Gluino

Figure C.7: Comparison of the mass spectrum for the gluino as a function of $\alpha_3$ calculated at the weak scale. We first repeat the MSbar comparison from Fig. C.6 for reference. The middle plot shows the systematic trend in the gluino mass ratio as a function of $\alpha_3$. The final plot show the trend of $\alpha_3$ versus the MSbar mass from SUGRA. These plots indicate some correlation between the mass ratio and $\alpha_3$, although it could be indirect. Plots of the other quantities used in the translation between the physical and MSbar masses showed no such systematic trends.



**Stop Trilinear Term**

**Sbot Trilinear Term**

**Stau Trilinear Term**

**alpha higgs**

Figure C.8: Comparison of the trilinear terms and the Higgs rotation angle $\alpha_h$. The three trilinear terms show a divergence near zero. The Higgs mixing angles are in agreement to within 4%, once one flips the relative sign of the angle.

clustering around the diagonal. We include the light Higgs comparison plot only because the disagreement looks to be less systematic which may be a clue as to why the light Higgs masses differ so much in the flipped case.

## C.8 Conclusions

We have presented our study of the model mass spectra generated by Michmodel and SUGRA. We see nominal agreement in almost all of the mass parameters, with the exception of the light Higgs. With this comparison done, we move on to an analysis of the differences between ISAJET and SPYTHIA.

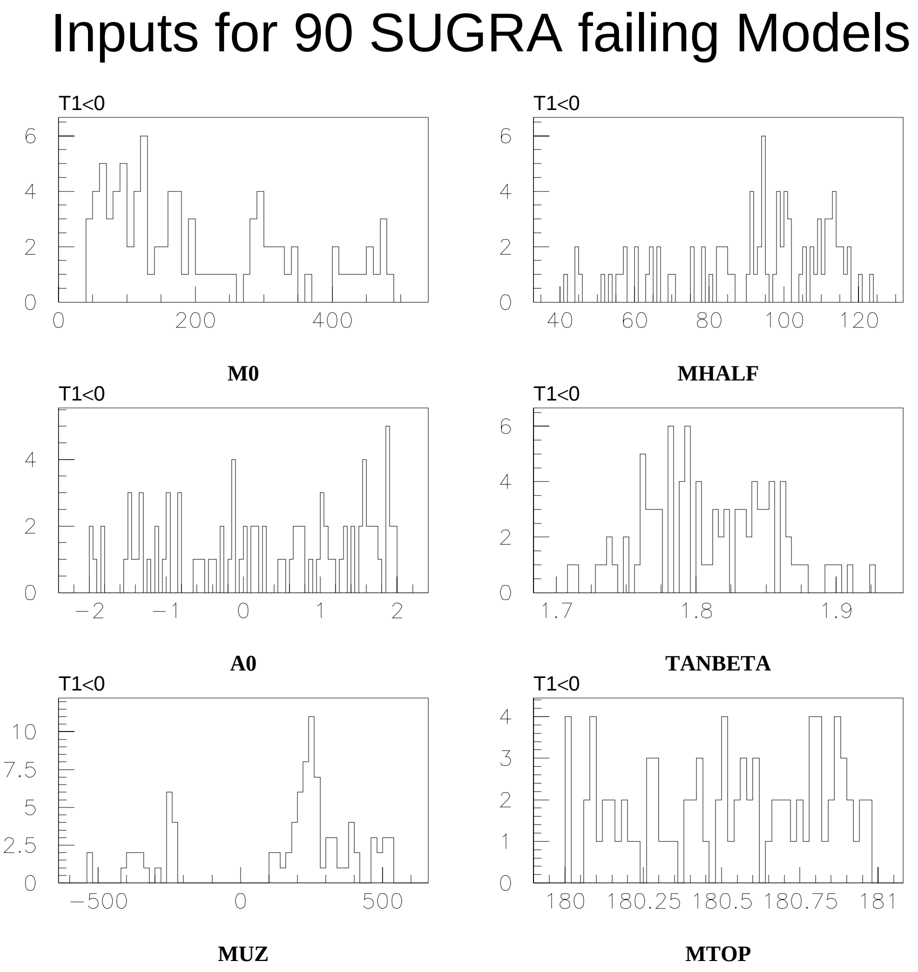
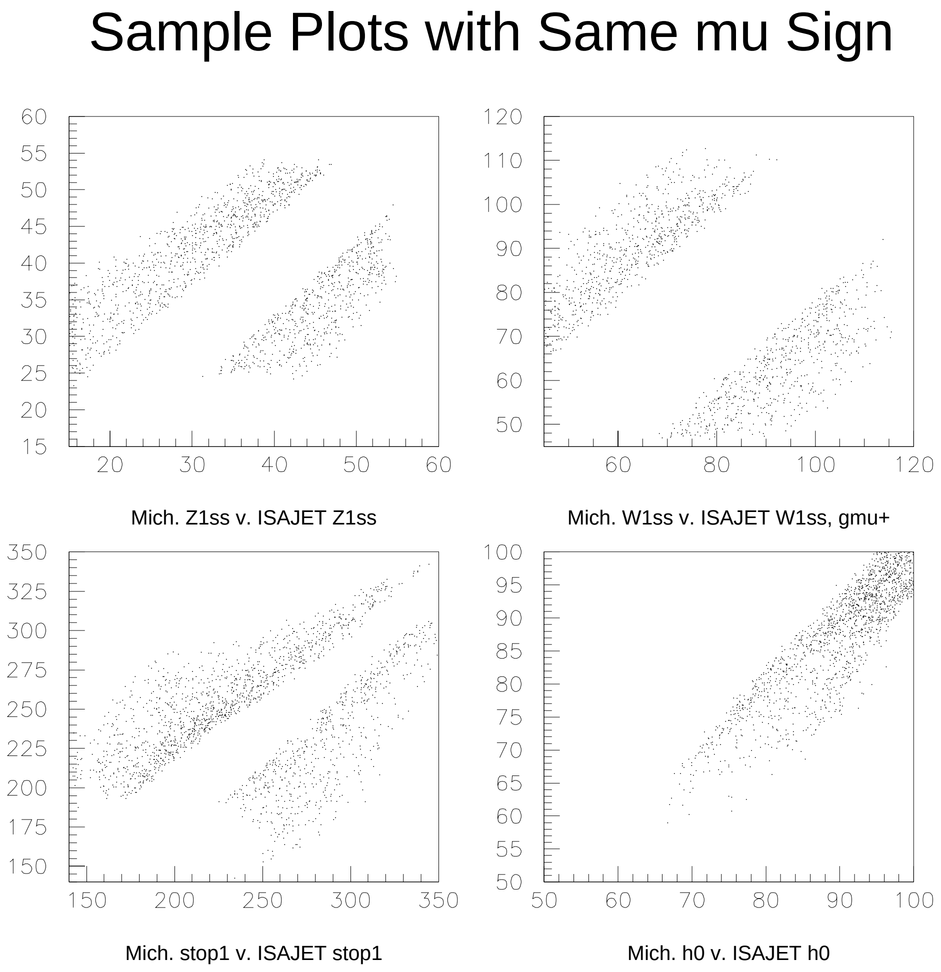


Figure C.9: Spectrum of input parameters for the 90 models which were solved by Michmodel, but caused overflows in SUGRA. We have used the Michmodel convention of $A_0 \equiv A_0 \div m_0$.



Mich. Z1ss v. ISAJET Z1ss

Mich. W1ss v. ISAJET W1ss, gmu+

Mich. stop1 v. ISAJET stop1

Mich. h0 v. ISAJET h0

Figure C.10: Sample of mass spectra for $\mu_G = \mu_H$. The dearth of points along the diagonal indicates that the model generators use a different sign convention.

# Appendix D

# HDIFFB: A Bin by Bin Comparator for HBOOK

## D.1 Introduction

The increasing use of computer controlled data acquisition systems in experimental physics apparatus betide the requirement to analyze and interpret large amounts of data in a short period of time. One of the areas of concern to experimental physicists is comparing many output channels with previous runs of the experiment, calibration curves, or theoretical predictions. HBOOK, a set of subprograms from CERNLIB's PAW physics analysis package, provides a machine independent mechanism for storing data in the form of histograms. HDIFFB, a routine in HBOOK, will test, bin by bin, the contents of two HBOOK histograms, and will return an integer identifying the number of bins which fail the test, along with an array indicating the result of each bin to bin test. HDIFFB assumes that the bins are independent, complimenting the previously available statistical routine HDIFF, which assumes dependence.

Contained herein is a summary of the routine, an extensive user guide, and complete mathematical descriptions of all test options. Additional sections include a description of logic flow, timing considerations, a description of the variable accuracy of statistical comparisons, and description of extensive testing.

## D.2 Routine Summary

| | |
|---|---|
| Routine Name: | HDIFFB |
| Function: | Bin by bin comparator for HBOOK histograms |
| Date of last version: | 27-Oct-1993 |
| Package Size: | Ten subprograms, 2630 lines total |
| Author: | R. J. Genik II |
| Technical Consultants: | J. T. Linnemann, J. McCampbell,<br>J. T. McKinley, D. Gilliland |
| CERNLIB Version: | Versions 93b and after |
| Language: | standard FORTRAN 77 |
| Input files needed: | Two HBOOK histograms |
| Comments: | Routine now maintained by CERN<br>Applications Software Group |

### D.2.1 Preliminaries

Automated data acquisition systems have greatly increased the amount of information obtainable from an experimental apparatus, along with the frequency at which this information is produced. This increase in available data from each run of an experiment has made it more difficult to track small or gradual changes in the output characteristics of an apparatus, especially when a small number of channels partially fail in a complex system; "eyeing-up" results and concluding that "things look O.K." is becoming more and more dangerous as volumes of output skyrocket. Such a situation has befell High Energy Physics and is looming in on physicists in other disciplines with reckless abandon. Faced with this problem, scientists at CERN many years ago began to formally develop data storage and analysis software that would precipitate useful results from oceans of collected information. One of the products of their effort has been PAW, a collection of data analysis software packages specifically designed for physics[40]. HBOOK is one of these packages; it allows experimenters sort, analyze, interpret, and present data in the form of histograms. (More advanced and powerful structures are available in this package; the interested reader should consult CERNLIB Long Write-Up Y-250 for detailed information on HBOOK.) HDIFFB is a subroutine inside HBOOK. HDIFFB will compare two histograms, bin by bin, and return both the number of bins which fail the specified statistical test, with respect to a user input tolerance (`TOL`), and the results of each test.

### D.2.2 Motivation

Previous versions of HBOOK contained only one comparison routine: HDIFF performs the Kolmogorov test on two histograms. HDIFF will compare the two histograms based on their overall shape and return return a probability that the two come from the same parent distribution. HDIFF assumes that the histogram bins are correlated to the extent that a deficit in one bin should be compensated for by excess in neighboring bins; this has the effect of skewing results when one channel has given spurious data, since the Kolmogorov statistic is usually[1] derived from the bin

[1] The Kolmogorov statistic is derived from the largest deviation between the two sample's integrated distribution functions; hence, a spurious bin will usually dominate.

with the largest disagreement. Also, when HDIFF identifies a difference in shape, it is the responsibility of the user to determine, if applicable, where the disagreement is the worst[42, 43, 41]. HDIFFB addresses the above, provides a data to data comparison option that is more powerful when the histogram bins are uncorrelated [45], and will return considerably more information about the two histograms than previously available.

## D.3 User Guide

HDIFFB can be called from a Fortran program, or from PAW (2.00 or later version). To be called from a program, the program must be linked with CERNLIB's GENLIB, KERNLIB, MATHLIB, and HBOOK (accompanying PAW 2.00 or later version). The syntax is:

```
CALL HDIFFB( ID1, ID2, TOL, NBINS, CHOPT, NBAD*, DIFFS*)
```

### D.3.1 Input Parameters

(Unless otherwise noted, the variable data types follow the Fortran-77 standard.)

`ID1` the first histogram to be compared. The *reference* histogram in the C- and A-options.

`ID2` the second histogram to be compared. The *data* histogram in the C- and A-options.

`TOL` is the tolerance for passing the test. Under S- and C-options, `TOL` is a number between 0 and 1 which represents the smallest probability considered as an acceptable match. TOL = 0.05 will cause HDIFFB to reject the bin as bad if there is less than a 5% probability the two bins come from the same distribution.

Under the A-option, TOL is the degree of precision of match required for the test to be considered as passed. TOL=2.0 means that a data bin differing from the reference mean by less than 2.0 times the reference error is compatible.

NBINS is the number of bins in the comparison. For a 1-dimensional histogram, this is the number of bins plus 0, 1 or 2, depending on whether the overflow and underflow channels are included. For a 2-dimensional histogram, this will have the total number of bins plus room for overflow bins along any of the axes requested. For more detail, see the discussion of DIFFS below.

CHOPT is a character string allowing specification of the several options (see below).

### D.3.2 Output Parameters

NBAD* is the number of bins failing the comparison according to the criteria defined by TOL and CHOPT.

DIFFS* is an array of length the number of bins being compared. It contains the results of the test bin by bin (probabilities for C- and S-options, $z$ values for the A-option). The results are passed back in the form:

**1-dimensional:**

DIFFS(NX) for no overflow/underflow, or for profile histograms.

DIFFS(0:NX) for underflow.

DIFFS(NX+1) for overflow.

DIFFS(0:NX+1) for overflow and underflow.

**2-dimensional:**

DIFFS(NX,NY) or similar to above depending on selected overflow/underflow options

The input array must be dimensioned this way in order for the user to be able to locate the results for each bin.

NOTE: The calculation of DIFFS is dependent upon the choice of TOL, and the contents of each bin, in addition to the type of histograms and the test selected.

### D.3.3 Selectable options

The following options are available to the user through the input parameter CHOPT:

N Use the absolute contents of each histogram, thus including the normalization of the histogram as well as its shape in the comparison. By default, for standard 1- and 2-dimensional histograms, the means are adjusted for the relative numbers of entries (including any overflow or underflow bins requested) in ID1 and ID2. No adjustment is ever made for profile histograms.

O Overflow, requests that overflow bins be taken into account.

U Underflow, requests that underflow bins be taken into account.

R Right overflow bin. For a 2-dimensional histogram, it includes the X-Axis overflow bin in the comparisons. If option O is selected, this is automatic.

L Left underflow bin. Same as above, but the X-Axis underflow is used. Option U uses this automatically.

T Top overflow bin. Same as option R but for the Y-Axis

B Bottom underflow bin. Option L for the Y-Axis

S Statistical comparison. Calculates the probability that both bins were produced from a Poisson distribution with the same mean. This probability is referred to in TOL and DIFFS. For profile histograms, performs Student *t*-test. The S-option should be used when comparing two data histograms.

C Compatibility test. Considers bins of the reference histogram (ID1) as perfectly describing the true distribution. Calculates the probability that the data (from ID2) was produced from that distribution. For 1- or 2-dimensional histograms, the Poisson mean is deduced from ID1. For profile histograms, the test assumes a Gaussian with mean and standard deviation given by the ID1. The C-option should be used when comparing data to a function, a well-known reference, or a calibration distribution.

A Absolute test. Similar to the C-option, except that TOL and DIFFS are in terms of the number of standard deviations, rather than probability. The test is on the number of standard deviations by which the data from ID2 deviates from the mean. Both the mean and the standard deviation are deduced from ID1. In addition, the A-option is intended to allow the user to set the value of the histogram error bars (via HPAK and HPAKE). An application might be to include systematic as well as statistical errors in the calculation. (This is also possible with the C- and S-options, unweighted, but the user should be aware that, in this case, HDIFFB will not check for proper calling of HPAKE.)

Error bars must be on for this option. This forbids overflow bins, underflow bins, and 2-dimensional histograms. The A-option ignores bins with zero contents in reference histogram.

Z Ignores bins with zero contents in the comparison. The S-option will ignore bins with zero contents in either histogram. The C- and A-options will ignore bins with zero contents in the reference histogram. The default action is to consider all bins as significant.

D Debug printout, dumps the critical variables in the comparisons, along with indicators of its weight, etc.

The default (no options selected) is the S-option, ignoring underflow and overflow bins, and automatically correcting for the difference in total entries between ID1 and ID2.

### D.3.4 When to use HDIFFB instead of HDIFF

HDIFFB treats the histogram bins individually, while HDIFF treats the histogram as a whole. In HDIFF, one is comparing the overall shapes of a probability distribution. Typically, an event is entered only in one channel, and the choice of channel depends on a measured value of a continuous coordinate, so that it makes sense for downward fluctuations in one bin to be considered as compensated by upward fluctuations in another bin. In HDIFFB, each bin is considered independently, except, perhaps, for an overall normalization factor which is the sum over all bins.

Thus HDIFFB is appropriate when:

- it makes sense to identify a single channel as ”bad”, for example if the bin contents correspond to hits in a given detector element.

- the data is heterogeneous, for example if the contents are counts vs trigger bit.
- one has already found a discrepancy on a shape with HDIFF and wishes to focus on where disagreement is worst.

A plot of hits vs detector element, where the detector elements cover some angular range, is an example of a histogram which might be considered with either comparison utility. The choice depends on the solution sought.

- If one wants to know if the angular distribution looks the same, use HDIFF.
- If one wants a report on bad detector elements, use HDIFFB.

### D.3.5 Choice of `TOL`

If one chooses .05 for `TOL`, one should expect 5 or so bad bins per trial from a histogram with 100 channels. For monitoring, one must compromise between the number of false messages (based on the total number of channels monitored), and the amount of data needed to claim a channel is bad. In general, a somewhat smaller fraction of channels than `TOL` will be flagged as bad, since for discrete distributions (Poisson statistics), the probability is quantized. For example, the probability might be .053 for 4 entries, and .021 for 3. If `TOL` $= 0.05$, only bins with 3 or fewer entries would be flagged as bad.

### D.3.6 When to use the C-option

The C-option assumes that the reference histogram contains the theoretically expected values with no (or negligible) errors. Examples might be a flat distribution hand-inserted as the expectation for a phi distribution, or a long data run to be compared with shorter data runs.

### D.3.7 When to use the A-option

The A-option can be used as an equivalent to the C-option by choosing `TOL` in terms of standard deviations instead of probability. HDIFFB will return $z$-values in `DIFFS` for each bin.

The A-option is also intended for setting by hand absolute minima and maxima. To restrict an efficiency between 80 and 100%, load the reference histogram with a mean of .9 (via HPAK) and the error bar of .1 (via HPAKE), and use HDIFFB with `TOL` $= 1.0$ and the A-option. Option `N` should also be selected for this application.

### D.3.8 When to use the S-option

The S-option should be used when both histograms are filled with statistical data, for example a momentum distribution from two successive data runs. Using the S-option when comparing data to a function or known reference yields poor results because it attributes errors to both histograms. In this case, the C-option should be selected.

### D.3.9 Comparison of Weighted vs Unweighted events

This is in general undesirable, as one is forced into the less accurate Gaussian approximation. Thus it is preferable, for example, to have unweighted Monte Carlo events to use HDIFFB to compare with data. The only useful case is if the weighted histogram is the reference histogram in the C-option, which only makes sense if one has much better accuracy than the data.

### D.3.10 Using Profile histograms

Option N is irrelevant for profile histograms. The overflow/underflow options are not allowed for profile histograms because insufficient information is stored in HBOOK to calculate the error bars. None of the test options (S, C, or A) check on the number of entries in a profile histogram bin (to do that, a separate 1-dimensional histogram must be made). This has an unexpected effect when the number of entries are small. Bins with no entries always pass the S- and C-options (no data is compatible with any distribution), so in such cases more bins pass for a given TOL than one might expect.

### D.3.11 Returned Values of DIFFS

The value returned in DIFFS may depend somewhat on the value of TOL chosen, as the approximation chosen to calculate DIFFS depends on both the number of entries and on the size of TOL (how accurately DIFFS must be calcuated).

The S-option sometimes returns a confidence level of 1.0 in the small statistics calculation, i.e. there is no probability that the two numbers come from different distributions. This is due to finite precision. Values slightly higher than 1.0 will be returned when the two content values are identical, since no statistical test could claim they come from different distributions.

### D.3.12 Scaling factor and negative bin contents

The normalization scaling (used unless option `N` is selected) is based on channel contents for all channels requested (including overflow/underflow bins when selected).

Negative bin contents are flagged as bad bins in S-, C-options.

### D.3.13 Statistical methods and numerical notes

(For simplicity, this is written as if the histograms contain the same number of total entries.)

The methods used for the S- and C-options are correct for unweighted events and Poisson statistics for 1- or 2- dimensional histograms. Errors may result in either the S- and C-options for small tolerances if bin contents are greater than the largest allowed integer.

For weighted events, the S- and C-options use a Gaussian approximation. This results in `DIFFS` values which are too low. HDIFFB rejects too many bins for weighted events, particularly for small numbers of equivalent events.

For the profile histogram S-option, HDIFFB calculates the $t$-test probability that both bin means were produced from a population with the same mean. The C-option calculates the probability of finding the `ID2` contents given a Gaussian with $\mu$ and $\sigma$ given by the `ID1` contents. Small numbers of entries (less than 5) for either test give `DIFFS` values which are too large, and HDIFFB will reject too few bins in such cases.

Finally, common to all statistical analysis is the limitations on the applicability of tests. Physicists can generally ignore formalisms when using HDIFFB to compare different results from the same apparatus; however, the user should be aware that

- HDIFFB assumes that the data was chosen at random from a Poisson distributed range of values.

- The Guassian approximation, used for weighted events and large statistics, is very poor at the tails of the distribution (orders of magnitude when farther than $3\sigma$ away from $\mu$).

- HDIFFB makes no discrimination between systematic and statistical errors. This should especially be noted for comparing results from different apparatus.

- HDIFFB is testing a sample size of 2.

### D.3.14 Errors reported by HDIFFB:

`Warning: Zero tolerance.`

The passed value `TOL` is less than or equal to 0. `TOL` = 0 can be used to force highest accuracy in the S-option.

`Warning: Only one comparison at a time, please.`

More than one type of comparison was selected. Only one of options S, C, and A may be used. As default, the S-option will be used.

`Warning: Different binning.`

The X-Min values for a 1-dimensional histogram or the X-Min and/or Y-Min values on a 2-dimensional histogram are different. This may give inaccurate results.

`Warning: Weighted or saturated events in 2-dimensions.`

HBOOK does not compute error bars for two dimensional histograms, thus weighted events are not allowed, and HDIFFB can not compute the correct statistics. An answer is still given, but it is probably not right. The only reliable case is a weighted 2-dimenension histogram as the reference histogram for the C-option.

`Sum of histogram contents is zero!` The sum of the content bins is zero.

`Both histograms must be the same dimension.`

A 1-dimensional and a 2-dimensional histogram have been specified. In order for the routine to work, both must be the same dimensionality.

`Both histograms must be the standard or profile type.`

Two different types of histograms have been specified. Both must be profile or non-profile.

`Not enough bins DIFFS to hold result.`

The parameter `NBINS` is less that the number of bins in the histograms.

`Number of channels is different.`

The number of channels in the two histograms to compare are different. They must be the same before the routine will process the data.

`U/O/L/R/T/B Option with weighted events.`

HBOOK does not compute an error bar for over-/underflow bins, thus it may not be used with weighted events.

`U/O/L/R/T/B Option with profile histograms.`

HBOOK does not compute an error bar for over-/under-flow bins, thus it may not be used with profile histograms.

`Weighted options and no HBARX.`

The user had not told HBOOK to figure the error bars for the histograms. Therefore, the operations will not be valid.

`A-option with no error bars on reference histogram.`

The user has not told HBOOK to compute error bars for the reference histogram. This error is also returned when the user attempts to select A-option to compare 2-dimensional histograms.

## D.4 Statistical Tests

The following is a mathematical description of the tests used by HDIFFB: C-option, compatibility; A-option, compatibility; and S-option, comparison[2]. The convention is used that $r$ is the bin contents of the first, or *reference* histogram, and $d$ is the bin contents of the second, or *data* histogram. Also, $\lambda$ is the overall scaling correction; the ratio of total entries in the data histogram divided by the total entries in the reference histogram

$$\lambda = \frac{\text{no. entries in data histogram}}{\text{no. entries in reference histogram}}$$

[2] The letters C, A, and S were selected as acronyms. The C-option and A-option, compatibility tests, are different in that the C-option results are expressed in terms of probabilities, whereas the A-option, an *Absolute* comparison, expresses results in terms of the number of standard deviations. The S-option is a statistical comparison between two runs of the experiment

For clarity in the text, it is assumed that the histograms have the same number of total entries ($\lambda = 1$); however, for completeness, $\lambda$ is shown for all relevant equations. Furthermore, while 1-dimensional histograms are described, the extension to two-dimensions is straightforward. *Data bin* and *reference bin* refer to corresponding bins, *i.e.*, the first bin of the reference histogram corresponds to the first bin of the data histogram, the second to the second, and so on. *diffs* is the returned `DIFFS` array element, the calculated probability. The term *small statistics* is here meant to mean low statistics *or* small values of `TOL`.

## D.4.1 C-option, compatibility

### Description

The C-option is a statistical compatibility test. Here it is assumed that the reference bin contains the expected entries for the data bin. The returned value is the probability that $d$ came from a Poisson distribution with mean $r$. The test is then performed:

$$\text{BIN IS PASSED IF} : \textit{diffs} \geq \texttt{TOL}$$

$$\text{BIN IS FAILED IF} : \textit{diffs} < \texttt{TOL}$$

*diffs* is calculated below.

### Unweighted Histograms

For unweighted histograms, a two-tailed[3] test is used

$$\mathit{diffs} = 2 \times \mathrm{MIN}(L, U)$$

where $L$ (or $U$) is the probability of randomly picking $d$ or less (or greater) from a Poisson distribution with $\mu = \lambda r = \langle d \rangle$. Specifically,

$$L = \sum_{k=0}^{d} \frac{e^{-\langle d \rangle} \langle d \rangle^k}{k!}$$

$$U = \sum_{k=d}^{\infty} \frac{e^{-\langle d \rangle} \langle d \rangle^k}{k!}$$

Using the following identity[46, Eq. 26.4.21]:

$$\mathrm{Q}(\chi^2|\nu) = \sum_{j=0}^{\frac{\nu}{2}-1} \frac{e^{-\frac{\chi^2}{2}} \left(\frac{\chi^2}{2}\right)^j}{j!}$$

where $\mathrm{Q}(\chi^2|\nu)$ is the upper tail cumulate of the $\chi^2$ distribution with $\nu$ degrees of freedom, $L$ and $U$ obtain the form

$$L = \mathrm{Q}(2\langle d \rangle | 2d)$$

[3] It is two-tailed because of the definitions of TOL and *diffs* above. For example, passing TOL $=$ 0.05 requests HDIFFB to "throw away" 2.5% at each tail of the probability distribution. This is the source of the factor of 2 which will appear in subsequent calculations of *diffs*. One could just as easily let $\mathit{diffs} = \mathrm{MIN}(L, U)$ (or equivalent for the other options), and compare that with $\frac{\mathtt{TOL}}{2}$; however, in this case, the user would have been responsible for determining whether he was interested in one- or two-tailed tests. It was decided that for almost all physics applications, the two-tailed test is appropriate, since one is usually concerned with being alerted to large statistical fluctuations in general, rather than a specific directional fluctuation. The returned values from the A-option are left signed for users interested in directional fluctuations since it is more natural to ask, for instance, for a $2\sigma$ high fluctuation, rather than ask for the 2.5% upper tail. Note that for this application, the user is responsible for determining how many bins failed in the direction he is concerned with, since the A-option uses $|\mathit{diffs}|$ to compare with TOL.

$$U = 1 - \mathrm{Q}(2\langle d\rangle|2d+2)$$

This form allows the use of several CERNLIB functions to directly calculate $L$ and $U$ from two of the four functions[4] `PROB` (G100)[5], `GAMDIS` (G106), `DGAPNC` (C334), and `DGAGNC` (C334). `PROB` is the upper tail cumulate of the $\chi^2$-distibution

$$\texttt{PROB}(X,N) = \frac{1}{\sqrt{2^N}\,\Gamma\left(\frac{N}{2}\right)}\int_X^\infty t^{\frac{N}{2}-1}e^{\frac{t}{2}}\,dt = \mathrm{Q}(\chi^2|\nu)_{\chi^2=X,\nu=N} = L$$

while `GAMDIS` is the lower tail of the Gamma distribution

$$\texttt{GAMDIS}(x,a) = \frac{1}{\Gamma(a)}\int_0^x e^{-t}t^{a-1}\,dt = 1 - \mathrm{Q}(\chi^2|\nu)_{\chi^2=x,\nu=a} = U$$

with $X$ , $N$ , $x$ and $a$ passed appropriately. If higher accuracy is needed than guaranteed from `PROB` and `GAMDIS`, the double precision Incomplete Gamma Functions

$$\texttt{DGAGNC}(X,A) = \frac{1}{\Gamma(a)}\int_x^\infty e^{-t}t^{a-1}\,dt = L$$

$$\texttt{DGAPNC}(X,A) = \frac{1}{\Gamma(a)}\int_0^x e^{-t}t^{a-1}\,dt = U$$

may be used. It is important to note that even though different functions are used, the only material difference is the numerical technique used to evaluate the integral. The demarcations are set such that no abrupt transitions are present; moreover, they depend on `TOL` and the usable range of the function. To determine if high accuracy

[4] See [46] equation 26.4.19 and G106, Note 2, for relations between these distributions.

[5] The alphanumeric value in parenthesis refers to the CERNLIB short write-up which describes the function and the numerical approximation used to calculate its value.[44]

is needed, the $z$-value is calculated

$$z = \frac{d - \langle d \rangle}{\sqrt{\langle d \rangle}}$$

When $z < -3$ and `TOL` $< 10^{-4}$, `DGAGNC` is used for $L$; otherwise `PROB` is used. If $z > 3$, `TOL` $< 10^{-4}$, and $\langle d \rangle < 10^4$, `DGAPNC` is used for $U$; else, `GAMDIS` is used for $z > 2$ and $\langle d \rangle < 5000$ [6]; otherwise, the following is used[7]:

$$U = 1 - \mathtt{PROB}(2\langle d \rangle, 2d)$$

### Weighted Events and Large Statistics in the C-option

When the data histogram contains weighted events, or $\langle d \rangle$ is larger than 1 million[8], the $\chi^2$ test is performed

$$\mathit{diffs} = \mathrm{P}(\chi^2 | \nu = 1)$$

where

$$\chi^2 = \frac{(d - \langle d \rangle)^2}{\sigma_d^2}$$

with

$$\langle d \rangle = \sigma_d^2 = \lambda \times \langle r \rangle \quad , \quad \langle r \rangle = \begin{cases} r & \text{(unweighted)} \\ \\ \sum w_r & \text{(weighted)} \end{cases}$$

[6] The value 5000 avoids convergence problems in the technique used to calculate `GAMDIS`

[7] It is worth noting that, since the Poisson distribution is quantized, $L + U > 1$.

[8] If `TOL` is less than $10^{-2}$ and the histogram is unweighted, then the previously described method is used. Also, as previously noted, the value of 1 million was selected because the difference between the approximations used tested to be negligible at this level of statistics.

For this calculation, the following identity is used:

$$\mathrm{P}(\chi^2|\nu = 1) = 2 - 2 \times \texttt{FREQ}(|\chi|)$$

where FREQ is the CERNLIB function evaluating the lower tail of the Normal distribution

$$\texttt{FREQ}(X) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{1}{2}t^2}\, dt$$

### Accuracy

There are two ways to interpret accuracy for the C-option: the significant digits in the returned *diffs* value, or whether HDIFFB incremented NBAD[9] correctly, *i.e.*, whether the first digit of *diffs* is correct. The former being important for normal applications, and the latter coming into significance for extreme values of $\langle d \rangle$ and TOL. Both cases are discussed below, and the methods used to determine the quoted values are described in Section D.8.

The first three digits of the returned *diffs* value can be considered correct for unweighted events, $\langle d \rangle < 10^4$, and $\texttt{TOL} > 10^{-6}$. For weighted events, accuracy depends upon the significance of the weight values; widely varying weights will decrease accuracy[10].

Considering unweighted events, there is an implied range of integers which should pass the test: for a given $\langle d \rangle$, there exists a range of passing integers $(d^0_{min}, d^0_{max})$ for a fixed TOL, and $d$ within this range should pass. HDIFFB will correctly pass this range for $\langle d \rangle < 10^5$ and $\texttt{TOL} > 10^{-15}$. For $\langle d \rangle < 10^6$, $\texttt{TOL} > 10^{-6}$, $d^{passed}_{min}$ and $d^{passed}_{max}$

[9] NBAD is the returned integer number of bins which fail the specified test.

[10] Quantifying the decrease for all situations is not possible. The user is hence cautioned to stay within "theoretical applicability" of the test.

vary from $d^0_{min}$ and $d^0_{max}$ by less than 2. Finally, for large $\langle d \rangle$, there is a maximum $d$ beyond which *diffs* always returns zero; however, for $\langle d \rangle < 10^7$, `TOL` $> 10^{-9}$, this is irrelevant.

## D.4.2 A-option, compatibility

The A-option is similiar to the C-option large statistics. However, *diffs* is returned in terms of standard deviations instead of probibility; in addition, $\sigma_r$ is read from the histogram instead of calculated. This allows the user to set the values of the error bars used in the calculation

$$diffs = \frac{(d - \langle d \rangle)}{\sigma_d}$$

where

$$\langle d \rangle = \lambda \times r, \quad \sigma_d = \lambda \times \sigma_r, \quad \sigma_r = \begin{cases} \sqrt{r} & \text{(unweighted)} \\ \\ \sqrt{\sum w^2} & \text{(weighted)} \\ \sigma & \text{(user set)} \end{cases}$$

$$\text{BIN IS PASSED IF} : |diffs| \leq \texttt{TOL}$$

$$\text{BIN IS FAILED IF} : |diffs| > \texttt{TOL}$$

### Accuracy

The A-option calls no external functions; hence, full machine accuracy should be expected. However, it is the opinion of the author that it is unwise to take more than 2 digits as significant; the calculations used in the subroutine are more precise than the *a priora* assumptions required to conduct the tests.

### D.4.3 S-option, comparison

The S-option is a statistical *comparison*, rather than a *compatibility* test. Here, neither bin is given preference, and the returned *diffs* value is the probability that the two numbers come from a Poisson distribution with the same, undetermined mean.

#### Unweighted Histograms

For unweighted histograms, no assumption is made concerning the mean of the distribution from which $r$ and $d$ were chosen: all possible means are considered. This allows the use of an Uniformly Most Powerful unbiased test. Complete theoretical development is beyond the current subject matter; however, it should be noted that the sum of two such numbers constitutes a Binomial distribution[45]. Again, a two-tailed test is used

$$\textit{diffs} = 2 \times \mathrm{MIN}(L, U)$$

but now $L$ and $U$ are determined from the binomial distribution

$$L = \sum_{k=0}^{r} \begin{pmatrix} r+d \\ k \end{pmatrix} \times p^k \times q^{r+d-k}$$

$$U = \sum_{k=r}^{r+d} \begin{pmatrix} r+d \\ k \end{pmatrix} \times p^k \times q^{r+d-k}$$

with

$$p = \frac{1}{1+\lambda}$$

and

$$q = 1 - p$$

For timing reasons, this test is only used when $r$ or $d$ is less than 25, or when the user requests a TOL of less than 0.001.

### Weighted Events and Large Statistics in the S-option

When the histograms contain weighted events, or the criteria for performing the small statistics test is not satisfied, the assumption is made that the $\mu$ of the distribution from which $r$ and $d$ were chosen is approximated by their arithmetic mean

$$\langle r \rangle \;=\; \tfrac{r+d}{1+\lambda} \quad , \quad \langle d \rangle \;=\; \tfrac{r+d}{1+\frac{1}{\lambda}}$$

The $\chi^2$ test is then performed

$$\mathit{diffs} = \mathrm{P}(\chi^2 | \nu = 1)$$

where

$$\chi^2 = \frac{(r - \langle r \rangle)^2}{e_r^2} + \frac{(d - \langle d \rangle)^2}{e_r^2}$$

with

$$e_r^2 = \tfrac{\sigma_r^2 + \sigma_d^2}{1+\lambda^2} \quad , \quad e_d^2 = \tfrac{\sigma_r^2 + \sigma_d^2}{1+\frac{1}{\lambda^2}} \quad , \quad \sigma_{r,d}^2 = \begin{cases} r, d & \text{(unweighted)} \\ \\ \sum w_{r,d}^2 & \text{(weighted)} \end{cases}$$

FREQ is used to evaluate this, as in the C-option and

$$\text{BIN IS PASSED IF : } \mathit{diffs} \geq \mathtt{TOL}$$

$$\text{BIN IS FAILED IF : } \mathit{diffs} < \mathtt{TOL}$$

### Accuracy

For the small statistics calculation, when *diffs* is greater than `TOL`, the first two digits are correct. When *diffs* is less than `TOL`, the first two digits to the right of the first non-zero digit of `TOL` are significant; for example, if $\texttt{TOL} = 0.0001, 0.000xxx$ are significant. One can force higher accuracy by setting `TOL` smaller than required, but calculation time will increase. Highest accuracy can be obtained by setting $\texttt{TOL} = 0$ (in this case, HDIFFB will issue a warning). The technique used to obtain this accuracy is described in the Section D.7.

For large statistics ($r, d > 25$, and $\texttt{TOL} > 0.001$) a Guassian approximation is used. Three digits are guaranteed correct. For weighted events, the author advises against taking more than two digits as significant. As in the C-option, accuracy will decrease if weights vary dramatically.

## D.4.4 Profile Histograms

HDIFFB will also accept Profile histograms[11] as input. Modifications to the above tests are described below.

### C-option

C-option again performs the $\chi^2$ test with the following modifications:

$$\chi^2 = \left(\frac{d - r}{s_d}\right)^2$$

[11] Profile histograms are described in the HBOOK manual[41].

where

$$s_d = s_{\bar{r}}\sqrt{\frac{N_r}{N_d}} = \frac{s_r}{\sqrt{N_d}}$$

and $s_r$ is the error bar from the refernce histogram.

### A-option

The A-option returns the number of standard deviations $d$ varies from $r$

$$\mathit{diffs} = \frac{d-r}{\sigma_r}$$

where $\sigma_r$ is the reference histogram error bar.

### S-option

The S-option performs the Student t-test on the expectation values of $r$ and $d$, which are calculated as before. CERNLIB's `STUDIS` (G104) is referenced for this calculation

$$\texttt{STUDIS}(t,n) = \mathrm{F}(t,n) = \frac{,\,(\frac{1}{2}(n+1))}{\sqrt{\pi n},\,(\frac{1}{2}n)} \int_{-\infty}^{t} (1+\frac{x^2}{n})^{-\frac{n+1}{2}}\,dx$$

with

$$t = \frac{|\langle r\rangle - \langle d\rangle|}{s_\triangle}$$

where

$$s_\triangle = \frac{N_r+N_d}{N_rN_d}\left(\frac{(N_r-1)s_r^2+(N_d-1)s_d^2}{N_r+N_d-2}\right)$$

and $s_{r,d}^2 = [e_{r,d}^2]$ are the error bars on the histograms. The two-tailed test is then performed

$$\mathit{diffs} = 2\times(1-\texttt{STUDIS}(t,N_r+N_d-2))$$

## D.5 Logic Flow of Subroutine

HDIFFB is a package of several related subroutines and functions. The package includes ten subprograms:

HDIFFB The top routine. It calls HDBINI, checks a returned error flag, then calls the appropriate calculation routine, and finally RETURNs to the calling program.

HDBINI The initialization routine. It initializes the variables in the HCDIFB common block, checks for errors in the input, decodes the option string and sets appropriate flags, and determines what type of histograms were sent: 1- or 2-dimensional, standard or profile, and weighted or unweighted. It references HGCONT to retrieve information about the histograms. In addition, it also references the following HBOOK and ZEBRA subprograms: HGIVE, HBUG, HUOPTC, HFIND, HDCOFL, JBIT, and HNOENT. Information about these routines may be obtained from CERNLIB documentation or, more likely, from the well commented source code of HBOOK. Syntax:

```
CALL HDBINI( ID1, ID2, TOL, NBINS, CHOPT, ERRORS)
```

HDBCOP The C-option routine. Calculates NBAD and the DIFFS array as described in the text. HDBCOP references the following external subprograms: HGCONT, FREQ(C301), GAMDIS(G106), DGAPNC(C334), DGAGNC(C334), and PROB(G100). More information about these functions is available in CERNLIB Short Write-Ups. Syntax:

```
CALL HDBCOP( TOL, NBINS, NBAD, DIFFS)
```

HDBAOP The A-option routine. Calculates NBAD and the DIFFS array as described in the text. References: HGCONT. Syntax:

```
CALL HDBAOP( TOL, NBINS, NBAD, DIFFS)
```

HDBSOP The S-option routine. Calculates NBAD and the DIFFS array as described in the text. References: HGCONT, FREQ(C301), and HBNSUM. Syntax:

```
CALL HDBSOP( TOL, NBINS, NBAD, DIFFS)
```

HDBPRF The routine which compares profile histograms. Calculates NBAD and the DIFFS array as described in the text. References: HGCONT, FREQ(C301), and STUDIS(G104). Syntax:

```
CALL HDBPRF( TOL, NBINS, NBAD, DIFFS)
```

HGCONT Real function returning various contents on histogram Zebra banks. Syntax:

```
X = HGCONT( ID, IX, IY, FUNCT)
```

where

ID Is the integer histogram reference number from HBOOK.

IX, IY Are the X and Y bin numbers ( Y is zero for 1-dimensional histograms).

FUNCT An integer controling the function to perform: '1' will return the contents of bin X,Y; '2' will return the error bar for the bin X (default to $\sqrt{\text{bin contents}}$ if no error bars exist); '3' will return the number of Y-channel entries, for bin X, from a profile histogram.

HBNSUM Double precision function returning a partial sum of terms in a Binomial distribution (used by HDBSOP). Syntax:

```
DX = HBNSUM( NSTRT, NEND, NMAX, P, QQQ, ACCUR)
```

where we calculate

$$\mathtt{HBNSUM} = \sum_{k=\mathtt{NSTRT}}^{\mathtt{NEND}} \begin{pmatrix} \mathtt{NMAX} \\ k \end{pmatrix} \times \mathtt{P}^{k} \times \mathtt{QQQ}^{\mathtt{NMAX}-k}$$

until the remaining terms become less than `ACCUR` (see Section D.7). NOTE: `QQQ` was used to prevent interference with Zebra.

HLBINO Double precision function returning the logarithm of Binomial coefficients (used by HBNSUM). When possible, `DBINOM` (B100) is referenced; otherwise, HLNFCT is used. Syntax:

```
DX = HLBINO( N, IR)
```

Returned value

$$\mathtt{HLBINO(N,IR)} = \log \begin{pmatrix} \mathtt{N} \\ \mathtt{IR} \end{pmatrix}$$

HLNFCT Double precision function returning the logarithm of the factorial of an integer using the next-to-leading order Stirling approximation. Syntax:

```
DX = HLNFCT(IX)
```

Returned value:

$$\mathtt{HLNFCT(X)} = (\mathtt{X}+0.5)\log(\mathtt{X}+1) - (\mathtt{X}+1) + \log\sqrt{2\pi} + \log\left(1 + \frac{1}{12(\mathtt{X}+1)}\right)$$

## D.6 Timing Considerations

HDIFFB was studied for CPU time used, on a VAXstation 3100 M76, for the C-, A-, and S-options. For large statistics cases, all used less than 200 $\mu$s per bin for the comparison. The small statistics calculations in the C-option run on the order of milliseconds per bin, with larger $\langle d \rangle$ taking slightly longer. The small statistics calculations for the S-option generally run on the order of milliseconds per bin; however, CPU time is highly dependent upon TOL and $\langle d \rangle$. The slowest calculation done ran with $\mathtt{TOL} = 0$ and $\langle d \rangle = 10^6$: approximately 300 milliseconds were required per bin for this calculation. The CERNLIB subroutine TIMED (Z007) was used for this analysis. The above numbers are intended as a guide since HDIFFB is available for the widest range of hardware environments. When timing is of great concern, the user should perform these simple studies on the intended system. (TIMED is located in KERNLIB.)

## D.7 S-option Variable Accuracy

The S-option was given variable accuracy because timing studies indicated possible gains of an order of magnitude for large calculations. This gain was achieved by terminating the sum in HBNSUM according to the passed value ACCUR. ACCUR is the largest allowed error; for example, sending HBNSUM an ACCUR value of 0.01 will ensure that the one-hundredths place will be returned with the correct significant digit.

Specifically, HDBSOP sends HBNSUM

$$\mathtt{ACCUR} = \frac{\mathtt{TOL}}{2 \times 10^{\mathtt{ACDIGT}+1}}$$

where ACDIGT is set equal to 2 in HDBINI. HBNSUM then calculates SMLNUM.

$$\mathtt{SMLNUM} = \log\left(\frac{\mathtt{ACCUR}}{10 \times |\mathtt{NSTRT} - \mathtt{NEND}|}\right)$$

where |NSTRT − NEND| represents the total number of terms in the sum. When the log of a term in the sum is less than SMLNUM, HBNSUM checks to insure that the terms are decreasing. When this condition is satisfied, HBNSUM returns the current sum.

Technical notes:

- The sum is determined to be decreasing if the current term is less than the last term calculated.
- The sum proceeds from the most to least significant for a $p$ value of 0.5.
- SMLNUM defaults to -LNBIGP, minus the logarithm of the largest allowed floating point number. LNBIGP is calculated in HDBINI from BIGP, the machine dependent largest floating point number from the HCPRIN common block of HBOOK.

## D.8 Description of Testing

Testing of the routines in HDIFFB was conducted using calls from programs written in Vax Fortran. Each option was tested individually. Descriptions of the testing of C-, A-, and S-option appear below.

### D.8.1 C-option testing

For testing C-option unweighted, two types of tests were performed. The first comparing a flat distribution with uniform random numbers, and the second comparing a flat distribution with a linearly increasing distribution.

The first series of tests set the mean of the flat distribution, generated Poisson contents around that mean for the second distribution, and recorded the number of bad bins as returned in NBAD with TOL set to 0.05. This was done several thousand times at each mean, for means from 4 to $10^6$. The results were then averaged for each test and compared with the expected result of 5% failure.

The second series of tests set the mean of the flat distribution, generated a linearly increasing distribution such that the minimum and maximum value had a Poisson probability of less than $10^{-40}$ (with the exception that 0 was the lowest data bin content), and cycled through the returned values in the DIFFS array. TOL was set to $10^{-5}$ for the call, and a table was made, for for each mean, with minimum and maximum passing bins calculated by comparing the returned value with desired TOL level. The table extended from TOL $= 10^{-2}$ to TOL $= 10^{-15}$. These values were then compared with the upper and lower tails of the quantized Poisson distribution, evaluated at each mean, as calculated directly from summing the Poisson probabilities. The later program made use of the extended precision available in Vax Fortran. Means ranged from 4 to $10^9$. Histograms were used containing up to 500,000 bins. Integer steps between bins were used when possible.

The results of both tests confirmed that the statistical methods used by HDIFFB are correct; furthermore, the results of the second series were used in determining accuracy at small values of TOL. The accuracy of larger TOL values was evaluated from guaranteed limits on called functions, and checked against hand calculation when possible.

Weighted events were tested by filling each bin in the data histogram $\mu$ times with a uniform random weight between 0 and 2. The test was run several thousand times for means up to $10^4$, and several hundred times for means up to $10^6$. The results here agreed with expectations.

### D.8.2 A-option testing

The A-option, unweighted, was tested with similar Monte Carlo techniques as described above. Also, a linearly increasing histogram was compared with a flat reference and the resulting `DIFFS` array was compared with hand calculation. Weighted events were tested as with the C-option. The results of the A-option tests were satisfactory.

### D.8.3 S-option testing

The S-option, small statistics, unweighted, was tested in several ways, in several steps, due to the fact that it references few tested external functions.

Each numerical subroutine written was tested within the range of allowed integer values on a VAX 3100, and the results were compared with hand calculation, and extended range versions of the same routines. HLNFCT, HLBINO, and HBNSUM all tested accurate up to or above single machine precision.

The correct method for testing the S-option is with Monte Carlo simulations, and comparing the results with (tedious) hand calculation when possible. The simulation produced two histograms with Poisson bin contents distributed around a common $\mu$, called HDIFFB, and recorded `NBAD`. Each simulation was run several thousand times and the results were averaged. Means up to 25 were tested in this manner. For means up to $10^9$, a specialized program was developed to evaluate the expected results directly from the Poisson distribution using extended precision. The later program was able to reproduce hand calculations. S-option gave satisfactory results in all cases evaluated.

Weighted events were tested with a similar uniform random weight routine as in

C-option. The results agreed with expectations.

## D.9 Summary

In conclusion, HDIFFB provides HBOOK with an important new tool for quickly identifying statistical fluctuations in histograms. The bins are considered as independent, complimenting the previously available routines. In addition, the assumption of independence allows users to pack histograms with uncorrelated channels, providing a new approach to analyzing large data sets, and real-time monitoring of complex experimental apparatus. The generality and completeness of the statistical tests available makes HDIFFB a powerful new tool for all scientists. Finally, the amount of information returned allows for automatic location of fluctuating channels, and relieves the experimenter of the burden of hand scanning reams of information to determine if a complete system is operating within user set tolerances.

# Appendix E

# Autocompare: A Monitor for Level 2 Filters

## E.1 Introduction

This document is intended as the complete reference to the Autocompare_Histogram package. The experienced reader may skip to the second section to find which files are required and where they are, alter the appropriate variables in the RCP's, and go from there. The second section contains lengthier descriptions of the parameters if the comments in the RCP file itself seem ambiguous.

### E.1.1 Purpose

Autocompare_Histogram (Autocompare) is a statistical comparator for HBOOK histograms. It is designed primarily for the histograms written out by the Global Monitor and Global Examine2 during each Run. In short, Autocompare is intended as an aid to the tedious process of hand-scanning histograms and comparing them to a reference distribution. Autocompare will notify the user of 'poor' fits to the reference and attempt to identify the exact bins within the histogram that are suspect. It is also useful to tune cut values for individual histograms for use in the Mega_Autocompare package.

### E.1.2 Description

#### Basic Action

Given 2 Run numbers, beginning and ending, Autocompare will sequentially cycle through the existing Runs between those limits (inclusive) and print out the histogram bins which have a low probability of being produced by random variation from the reference.

#### Requirements

The Autocompare package consists of 2 stand alone executables, 3 RCP files, 2 command files, one bin code description file, and one description file. The user should maintain a copy of the .RCP's, .COM's, the description file, and know the location of the executables on their cluster. If not available on their cluster, the .exe's combined are less than 3000 blocks and can copied. The primary copy is maintained on FNALDØ currently.

#### RCP files, General: (Those with experience using RCP's may move on.)

RCP stands for Run Control Parameters. This type of file allows the user to change various parameters used by the executables without the need to re-compile and re-link. For example, to change the cut value in an analysis, such as Autocompare, the user need only change the value in the RCP next to the variable name, instead of obtaining the source code, editing it, compiling it, and building the executable by linking with several libraries. The executable, of course, has to know which variables to initialize from the RCP; hence, the user cannot add to, and should never subtract from, the variables listed. Removing variables from the RCP will cause Autocompare to crash and dump to the screen the last variable it attempted to read. Adding variables is moot because the executables just ignore them and their value. Once the user has their own copy of the RCP's, it is a good idea to add their own comments for future reference: this practice is encouraged by the author, but **only** for such stand alone utilities as Autocompare, where each user customizes their own RCP. Comments are added by preceding them with an exclamation point. Note that everything from `!` to the end of a line will be ignored so don't put them in front of variables to be read.

**Sample lines from an RCP**

```
Name        Value             Comments
--------------------------------------------------------------
SIG_TOL     0.10            ! Number of sigmas for test,real.
                            ! %stat prob for S,C options
PCT_TOL     .20             ! Percent difference cut.
```

**Editing RCP files** To edit any RCP file, the user should open the file in EDFOR, the default EVE editor at D-Zero, change the values desired, and hit enter-R to write the RCP file in the appropriate format. (Users without access to EDFOR should consult their local expert on how to run RCPSIZE on a modified file.) RCPSIZE is also run automatically when the user exits EDFOR via ctrl-Z.

## E.2 Using Autocompare

The most difficult part of any statistical comparison routine is developing an appropriate reference. The Autocompare package is divided into 2 executables: Autocompare_Histogram (Autocompare) and Make_Histogram_Ref (Make_Ref). Autocompare will be discussed first (so as not to discourage the reader immediately:-), then Make_Ref will be addressed along with how to chose test options and cut values. Before beginning, obtain a copy of the following files:

```
Autocompare_Histogram.rcp, Make_Histogram.rcp,
Autocompare_Histogram.com, Make_Histogram.com,
Autocompare_Histogram.descr, if_codes.txt,
Run_Sum_info.rcp
```

At this release, these files are maintained in

```
FNALD0::tmp$root215:[data.genik.autocompare.production]
```

If the executables are not available on your node, copy them also. The parameters below are extensive; some are present with future versions and options in mind.

### E.2.1 Setting up the Parameters

Autocompare is intended to be low maintenance, once the initial set-up is complete. After initial set-up, the only changes which should be required are adjusting the minimum and maximum Run numbers, and perhaps some tweaking of the cut values.

Below is the complete list of parameters (in no particular order), parsed from the RCP file as set up for the L2EM 'ifailed' histogram, followed by a description. The logicals `act$his`, `act$root`, `act$local`, `act$ref`, and `act$run_sum` are defined via the .com file as the directories containing the histograms, the executable (Autocompare), the `if_code` file and description file, the reference file,and the run summaries, respectively; however, there is nothing special about these logicals (*i.e.* they are not system table logicals).

```
RUN_NUM_MIN      60400
RUN_NUM_MAX      60500
BASE_FILE_NAME   'act$his:RUN'
BASE_FILE_EXTENTION  '.HST'
```

This tells Autocompare where to find the histogram data files. Autocompare will search for existing files named :

```
BASE_FILE_NAMExxxxxBASE_FILE_EXTENTION
```

from `xxxxx` equal `RUN_NUM_MIN`, to `RUN_NUM_MAX`. It is suggested to use the complete Run number since this integer is used as an output parameter. For example, one could include the '60' in the `BASE_FILE_NAME` instead of in the Run limits; however, the output would identify Run 60400 as `Run:   400`, etc. Note that the sample definitions for these files and logicals were set for Run 1a file locations and standard file naming conventions. These names and locations are set by persons other than the author; therefore, consult your local expert as to where the files you wish to use are located.

```
OUTFILE_LOC      'SYS$OUTPUT'
```

This tells Autocompare where to write all the useful information it figures out. Note: `SYS$OUTPUT` is the name of the default VMS output device, usually the screen.

```
IF_CODE_FILE 'act$local:IF_CODES.TXT' ! this file must exist, but
                                      ! can be a null file.
```

This file is read in as the bin codes for the histogram it is analyzing. The bin codes are labels to the specific channels of the histogram, if appropriate. These are written to the output file next to the failing channel. A sample output line from the ifailed histogram is:

```
Run 60479 sig =  0.05 diff =  -57.8% fails for  16,  EM3 HIGH
```

Here, histogram channel 17, bin code 16, failed the tests performed. This histogram channel corresponded to the EM3 HIGH cut from L2EM. Note the name *bin codes*, they do **not** equal the bin channel numbers of the histogram, but are displaced by -1 for historical reasons. (This decision precedes the tenure of the current author.) The first bin has a bin code of 0, the second 1, etc. up to the `NX` bin, the number of bins booked in the histogram, having a code of `NX-1`. The underflow bin has a code of -1 and the overflow bin has a code on `NX`. (underflow and overflow bins are incremented during the filling of the histogram by attempting to fill at an $x$ value less than minimum, or more than the maximum, respectively, of the booked allowed values. For instance, if a histogram was booked with an `XMAX`, as defined in HBOOK, value of 101., and then data point `X=103.` was entered into the histogram, the overflow bin would be incremented.) The required format for this file is as follows:

```
CIIIXA32
```

where `C` is either a space, telling Autocompare to read in this line, or an exclamation point, `!`, telling Autocompare to ignore this line (it is a comment line). In-line comments are not allowed. `III` is an integer indicating the bin code as described above. Leading zeroes are not required. `X` is a required space between the integer and the beginning of the label. `A32` the next 32 characters on the line: the name, or *tag*, associated with that histogram channel. [ For Fortran jocks: the `READ` is from `FORMAT(A1,I3,A32)`, and the A1 is identified before the returned `IOSTAT` value is evaluated for errors. ] The bin codes need not be in any specific order, or even all there; the default label is `Unlabeled bin code`. Note that this file **must** exist. Autocompare will assume something is wrong if it can't find it; however, it can be a null file.

```
IGTUNIT_USERID  666
```

This tells Autocompare the `GTUNIT` user id for the process at hand. If the reader knows what this is, substitute your id. If the reader has no clue, don't worry unless Autocompare crashes and outputs an error message saying `Cannot get unit number`. Just change the ’666’ to another number between 101 and 999 and try again. Repeat until this message goes away.

```
SUMMARIZE_ALL   .true.
```

This flag tells Autocompare whether to print the Summary line for each run, or print just a message saying `Run XXXXX OK`. `.TRUE.` will print the Summary, `.FALSE.` won't. Two examples are

```
Summary: Run 60417
     6 Sig fails,  2 Pct fails,  0 Ok zeroes,  0 Combined failures
```

and

```
 Run 60417 OK
```

The former is useful in tuning cuts on a small number of Runs (described later), the latter is useful for scanning output from several Runs. There is no functional difference between the two options, just what information is output. The summary line is printed when there are combined failures (*i.e.* the purpose of the program).

```
    DESCR_FILE      'act$local:autocompare_histogram.descr'
```

This tells Autocompare where to get the description file. (Don’t bother looking for the name ’description file’ in any manual, Jim Linnemann made it up and the author likes the idea :-) A description file is a text file which is printed out verbatim at the end of initialization and before computations begin. It can be described as the ‘header’ or ‘title’ of the output. The description file for the L2EM analysis prints this:

```
            Running Autocompare for L2EM ifailed histogram
```

The user can customize this to provide any information they consider useful. The only limit on the file is that each line has a maximum length of 80 characters. Lines in the file which begin with an exclamation point, !, are skipped. The ! must be the first character for comment lines.

```
    REFFILE     'act$ref:HISTOGRAM_REF.HST' ! named in make_ref RCP
    REF_ID      500                ! From Make_ref rcp file, hist
                                   ! id for reference.
```

This tells Autocompare the name of the reference file created via Make_Ref, and the histogram id given to it. The file for the reference is named in Make_Ref's RCP file, as is the ID written to it.

```
TEST_SELECTION    'B'        ! Option of whether to call HDIFFB
                             ! or HDIFF. See HDIFFB.DOC for info
                             ! on which is appropriate.
TEST_OPT_STAT_HDB    'CZ'    ! 10 chr max. sent to hdiffb, 1st
                             ! char must be 'A', 'C', or 'S'
TEST_OPT_STAT_HD     'D'     ! 10 chr max. sent to hdiff, see
                             ! HBOOK manual for possible options
TEST_OPT_PCT     'AZU'       ! sent to Hdiffb to compute %diff 'A'
                             ! option, the user can add U,O,Z,N
```

These tell Autocompare about the various statistical options. Choosing which options are appropriate is discussed below. In short: TEST_SELECTION should be B to call HDIFFB, any other character to call HDIFF instead (Maximum of one character). TEST_OPT_STAT_HDB is the option string sent to HDIFFB, while the option string sent to HDIFFis TEST_OPT_STAT_HD. O and U are accepted by both routines to indicate the inclusion of over/underflow bins in the comparison. The first character of TEST_OPT_STAT_HDB **must** be the HDIFFB test selection S, C, or A. (This requirement is imposed by Autocompare, HDIFFB doesn't care which order the options are in.) TEST_OPT_PCT is the option string sent to HDIFFB for the calculation of percent differences. The first character **must** be an A and the string **cannot** contain an S or C. Other options are discussed below. Note on compatibility: if U or O is selected, they should be in both TEST_OPT_STAT and TEST_OPT_PCT, **and** the appropriate UNDERFLOW and/or OVERFLOW flag (below) must agree with this. See below for details on which is given priority if conflicts are present.

```
SIG_TOL     0.10            ! Number of sigmas for test, real.
                            ! %stat prob for S,C options, and HDIFF
PCT_TOL     .20             ! Percent difference cut.
```

These are the tolerances for passing the tests. SIG_TOL: for all but the HDIFFB A option, this is the statistical probability that the two bins are from the same distribution. A value of 0.05 means that there is only a 5% chance that the bins are identical. PCT_TOL: this is the 'percent error bar' on the reference bin. For instance, to allow for a +/- 15% variation in values without being notified that something is wrong, set PCT_TOL to 0.15. When a give channel fails both of these cuts, it is termed a *combined failure*, and the calculated values, along with the *bin code* are dumped to the screen. An example of the output is

```
    Run 60417 sig =  0.02 diff =  -29.7% fails for  15, EM3 LOW
```

Note that the PCT_TOL value is positive, but it is applied symmetrically around the reference bin contents.

```
    UNDERFLOW   .true.          ! Copies underflows into all internal
                                !  histos
    OVERFLOW   .false.          ! Copies overflows into all internal
                                !  histos
```

These flags tell Autocompare whether to consider over/underflow bins in the comparison. (Remember to include the over/under options when building the reference. This is set in the Make_Ref RCP file.) Compatibility (Version 1.0): If either of these flags is set .true., and the test option strings do not contain the appropriate U or O, they are added to the strings before sending them to the comparison routines. If these flags are set .false. and the option strings contain O's or U's, Autocompare will become very confused, and output unpredictable results. Note that 'confusion' is orders of magnitude worse than crashing: the user is mislead instead of notified something is wrong.

```
HIST_ID     6             ! Integer ID of histogram in data file
PATHNAME    '/L2EM'       ! Dir path to histogram, '/' must be
                          ! first character if hist is in
                          !  sub-dir, or '' if in top dir.
```

This tells Autocompare which histogram it is supposed to compare with the reference. Autocompare will take HIST_ID from the PATHNAME directory of the data files as the data histogram to send to the comparison routines. If the histogram is in a sub-directory of the HBOOK file, PATHNAME must begin with a /; else, if in the top directory, enter ”. Note that the HIST_ID used has a tendency to change when major versions of the Global Monitor are released. This should result in an error being generated from the comparison routines. To find this information, if not known, copy a histogram file to your local directory and enter PAW. Open the file in PAW and list the directories. Change to the appropriate directory and list the available histograms. The command sequence in below (note the .hst file must be in you current directory. The current directory is the last directory you were in before you entered PAW.) is an example for the /L2EM directory.

```
PAW>HIST/FILE 1  RUN60447.HST
PAW>HCDIR /L2EM
```

```
     PAW>HIST/LIST
```

```
   AUTOBOOK    .true.    ! Will read the booking paramters of
                         ! the reference histogram and set the
                         ! below  to their returned values.
                         !  Note: the paramters are still read in.
   hist_nbins  100       ! Integer number of bins, 200 current
                         !   max allowed.
   hist_xmin   0.        ! Real lower edge of channel 1
   hist_xmax   100.      ! Real upper edge of last channel
```

These tell Autocompare the booking parameters of the histogram to be analyzed. If your not sure what the booking parameters are, set AUTOBOOK .true. and they will be read off the reference histogram. The parameters are still read in, and must have values associated with them, but they are replaced during execution if `AUTOBOOK` is on. The current maximum bins allowed is 200 (Version 1.0).

```
   RS_CHECK    .false.
   LINE_MIN    2
   LINE_MAX    6
```

This tells Autocompare whether to dump `LINE_MIN` to `LINE_MAX` of the indicated Run Summary file. The type of Run Summary is set in RunSum_Info.RCP; e.g., the Run Summary files you get when you use the RunSum utility, or the Global Monitor run summary. Lines 2-6 of the 'usual' Run Summary are the header, an example is

```
Attempting to access Run Summary...
Run # 60416 Date: 10-FEB-1993 03:37:22.13 Duration: 0 03:52:14.79
```

```
GL Log 4 page 255  Keyword: Beam-Beam
Configuration: V71.GLB
Begin comment: lum = 1.7 e30           trig= v71-01e30
Evaluation Good Run     End comment: 4 HOUR BREAK
```

Below is the parsed RUNSUM_INFO.RCP file with additional clarifications.

```
\ARRAY BASE_RS_NAME
'act$run_sum:run_summary_00'
'D0$RS_loc:run_0'
\END
\ARRAY EXT_RS_NAME ! CHARACTER EXTENTION AFTER RUN NUM
'.dat'
'.GM_SUM'
\END
```

These two arrays set which files to look for. We search for filenames

```
BASE\_RS\_NAMExxxxxEXT\_RS\_NAME
```

where `xxxxx` is the `I5` integer sent to runsum_info.for. Trailing blanks are ignored for `BASE_RS_NAME`. An array was used to allow the user to search several file names and locations. Note that if different run summary types are listed, the min and max lines the user wants to see may change. The size of this array is set in this file with the parameter `NPLACES`. Remember to change this parameter if you change the size of the arrays. If the number of elements differ between the two arrays, unknown things will happen. RunSum_Info also has it's own `GTUNIT` user id; see above for what to do with this.

Make_Histogram_Ref (Make_Ref) will make reference histograms for use with the Autocompare package. In short, Make_Ref will add, bin by bin, the contents of an .rcp set list of histograms. One to 25 histograms may be added together. There are several checks within Make_Ref which ensure you are adding together compatible things.

Make_Ref consists of one executable file, one .rcp file, and one .com file to define various logicals. You should have your own copies of the .rcp and .com files in order to make your own references.

Most parameters are the same as for Autocompare, with some slight changes in variable names intended to ease modification, such as REF_ID from above is set here as HIST_ID_OUT. There are a few parameters; however, which need further explanation.

```
NO_RUNS 1              ! size of CHFILE_NAME array
                       ! MAX SET TO 25 IN CODE
\ARRAY CHFILE_NAME
    'MRF$HIS:run60414.HST'
\END
```

NO_RUNS is the integer size of the array CHFILE_NAME, the array of filenames from which to get references. A maximum of 25 histograms may be added together.

# Appendix F

# Level 1.5 CalTrig Readout Control P2 Paddle Board Description (ROC-P2PB)

This is a brief technical description of the hardware and logic involved in the ROC-P2PB. Additional information is available in other L1.5 documentation.

The Readout Control P2 Paddle Board (ROC-P2PB) performs four functions: TAS Protocol decoding, TAS Number latching, Front End Busy transmission, and signal level conversion. It plugs into the VSB (also called VMX) P2 connector of an Ironics IV-1623 VMEbus Parallel I/O Board through the Backplane of the Level 1.5 CalTrig crate. There are three connectors on the board:

**P1** A 40-pin Universal Header connector with latches which carries the differential ECL Trigger Acquisition and Synchronization (TAS) signals.

**P2** A 96-pin Eurocard connector for the VSB bus.

**P3** A 34-pin Universal Header connector with latches that acts as communication port between the P2 paddle Boards of the crate (called the Board to Board connector).

**TAS Protocol (TASP) decoding** ROC-P2PB monitors the TASP from the TAS cable via a dual rising-edge triggered D-Latch. SD and HT are D1 and D2, while each inverted signal acts as a clock for the other. The first to fall will latch the other high. TASP will either set Readout Required high or Dump Event high as outputs to an Ironics port. These outputs are cleared when Readout Complete is asserted through an input from another Ironics port.

**TAS Number Latching** Readout Required also clocks two octal D-Latches, grabbing the TAS number from the TAS cable. The outputs from these chips are connected to two ports on the Ironics.

**Front End Busy transmission** ROC-P2PB ORs Readout Control Front End Busy from an Ironics port with two inputs from the ROC-P2PB P3 connector (the Board to Board connector); the result is put onto the TAS Cable.

**Signal level conversion** ROC-P2PB translates all input signals from the L1.5 Framework from differential ECL to TTL. The FEB output to the L1.5 framework is translated from TTL to differential ECL. Six 10H125 quad translators, and one 10H124 quad translator perform this function. (ECL → TTL, and TTL → ECL, respectively.)

# Appendix G

# Level 1.5 Physics Commissioning

We describe the goals, design, implementation, and performance of the DØ Level 1.5 Electromagnetic Trigger. This system has been added to the DØ hardware trigger and is based on Digital Signal Processors (DSP). The Level 1.5 Calorimeter Trigger is a combined hardware and software system which can identify objects, such as electromagnetic (EM) showers, by the patterns of energy deposition in the calorimeter. The system uses the Level 1 Calorimeter Trigger Tower data and applies filtering algorithms based on neighbor sums and ratios of the EM and total (EM + hadronic) transverse energies ($E_T$). Eleven "local" DSP's, operating in parallel, process data from overlapping sections of the calorimeter and transfer their results to a "global" DSP where the Level 1.5 Calorimeter Trigger decision is made. This decision is then sent to the Level 1.5 Trigger Framework as one of the inputs to the final trigger decision. The online performance is defined and results of the commissioning for two EM algorithms in use at DØ are presented.[1]

[1] This paper was presented at the 1995 IEEE Conference on Real Time Computer Applications in Nuclear, Particle, and Plasma Physics (East Lansing, MI, May 1995).

## G.1 The DØ Experiment

The DØ Experiment [49, 7] is a high-energy physics experiment at Fermi National Accelerator Laboratory (Fermilab), in Batavia, IL. Fermilab is home to the world's most powerful particle collider, the Tevatron, which produces, every 3.5 $\mu$s, proton–antiproton collisions ("beam crossings") with a 1.8 TeV center–of–mass energy. The DØ Detector is located in one of the Tevatron's six interaction regions. The primary components of the detector are an eight meter long cylindrical uranium–liquid argon calorimeter surrounding central tracking chambers, and enclosed by a magnetized iron muon detector. DØ was designed to stress measurement of high transverse momentum (high $p_T$) parton jets using the finely segmented, nearly hermetic, calorimeter.

High–energy physics experiments at hadron colliders, such as DØ, must sort through the hundreds of thousands of collisions per second to determine if the collision produced an "interesting event", or one which can provide insight into the fundamental particles and forces of nature. For the physics under study at DØ, about 1 in 100,000 beam crossings produces an interesting event to be saved for offline analysis, the rest are ignored. The DØ̃Trigger [50] is responsible for making this selection, the *trigger decision*, based on the patterns of energy deposition in the various detector components. This trigger decision is made in two stages, the hardware stage (Level 1) [51] and the software stage (Level 2) [52]. The goal of the hardware trigger is to process a quick overview of every beam crossing and initiate a high–precision data readout for the most promising events. The software trigger analyzes the full data set and makes the final decision to save the event on tape for offline study. Level 1.5 is a hybrid of hardware and software, and operates between Level 1 and Level 2.

The Level 1 Trigger consists of Level 1 Trigger Subsystems and the Level 1 Trigger Framework. Each Level 1 Trigger Subsystem processes detector–specific information and produces, for every beam crossing, an indication of the activity seen by its associated Detector component. These indications are transmitted to the Level 1 Trigger Framework, which determines whether the event is to be rejected, or digitized for further analysis. This decision is made between beam crossings, before the information is overwritten in the Detector Front-Ends. The Level 1 Trigger Subsystems perform, in hardware, relatively simple fixed algorithms with programmable parameters. The Level 1 Trigger accepts about 1 event out of every 300-600 beam crossings.

The Level 1.5 Trigger produces additional rejection by allowing more complex data processing, at the expense of some experiment deadtime. This not only purifies the event sample sent to the software trigger, but also allows for a more efficient usage of the limited bandwidth available in the triggering and data acquisition system. The bandwidth to Level 2 is limited to about 130 MBytes/second. The Level 1.5 Trigger consists of Level 1.5 Trigger Subsystems (again associated with Detector components) and a Level 1.5 Trigger Framework. A maximum of 250 $\mu$s is available for the Level 1.5 Trigger Subsystems to produce their triggering information.[2] The Level 1.5 Trigger is used to further qualify a subset of the events which have been accepted by Level 1, and accepts about 1 out of every 3 submitted events.

The Level 1 Trigger Framework, Level 1 Calorimeter Trigger, Level 1.5 Trigger Framework, and Level 1.5 Calorimeter Trigger [52, 53] were designed and built at Michigan State University.

Level 2 consists of 48 MicroVAX processors operating in parallel. Each processor

[2] This is the maximum time the analog front end buffers of the high–precision data readout can hold their signal accurately.

receives a single event passed by the Level 1 Framework and analyzes it. Based on this analysis, the final trigger decision is made and the event is either logged to tape or dumped. The Level 2 input rate is about 150 Hz, and its output rate is approximately 3 Hz for full detector readout, limited by the 2.1 Mbytes/sec sustainable rate to tape.

## G.2 Motivation for the Level 1.5 Calorimeter Trigger

The DØ Calorimeter is divided into two sections, the inner electromagnetic layers (EM), and the outer hadronic layers. Electrons and photons, whose scattering and subsequent showering is governed by bremsstrahlung in the uranium plates, will deposit almost all of their energy in the inner EM layers. Jets of hadrons, whose showering is governed by strong interactions with uranium nuclei, deposit their energy much deeper in the calorimeter,[3] in the outer hadronic layers.

The Level 1 Calorimeter Trigger receives signals from 1280 projective Trigger Towers, 40 in the polar direction ($\eta$) by 32 in the azimuthal direction ($\phi$). We use the physics relevant pseudo–rapidity $\eta = -\log(\tan\theta/2)$ for addressing the Trigger Towers. [50] Each Trigger Tower has 2 sections identified as EM and Hadronic. The data available at Level 1 to make a trigger decision are the EM and EM + Hadronic Trigger Tower transverse energies. The transverse energy, $E|\sin\theta|$ ($\theta = 0$ is along to the beam direction), is the quantity of interest in $\overline{p}p$ collisions because large

[3] Roughly, this is because the range of the electromagnetic interaction is much greater than that of the strong interaction. We also note here that electrons and photons shower identically in the calorimeter; we discriminate between them by signals in other detectors, such as the Central Drift Chambers, where electrons will leave a charged particle track, while photons will traverse without leaving a signal. For the rest of this paper we will only refer to electrons, although the discussion is valid for photons also.

components of momentum perpendicular to the original direction of motion indicate either violent collisions, or decays products of very massive (thus interesting) particles. For the rest of this paper energy will always refer to the transverse energy, $E_T$.

Although electron showers are typically smaller than Trigger Towers, the energy deposited is frequently shared between 2 (or more) neighboring Trigger Towers whose signals must be taken into account to determine the true energy deposit. The Level 1 Calorimeter Trigger does not have this capability. The thresholds that can be safely applied at Level 1, while preserving good efficiency, must thus be set to a fraction of the desired electron energy. For jets, which span many Trigger Towers, this problem is more severe. This lowering of the thresholds degrades background rejection and leads to high trigger rates. Furthermore, the ability to trigger on multiple electrons or jets is limited.

To address these limitations, the Level 1.5 Calorimeter Trigger was designed to allow signals from adjacent Trigger Towers to be combined, providing a more accurate determination of a particles true energy. In addition, the system can calculate ratios of these combinations, for example the EM energy in a 2x1 cluster divided by the total of EM and hadronic energy. Preliminary studies showed that clustering up to 5x5 Trigger Towers centered on the Level 1 candidate, or *seed* Tower, would be sufficient to produce useful algorithms.

## G.3 The Level 1.5 Calorimeter Trigger

Purely hardware designs were seriously considered, but would have fixed the algorithm by choice of the architecture. For this reason, the DØ Collaboration chose to support the processor–based design presented here.

Early in the design studies, it became clear that the major technical challenge of the Level 1.5 Calorimeter Trigger was *not* in the implementation of a particular algorithm, but rather in making the Trigger Tower data available to the processing elements. We chose DSPs over conventional CPUs because they are better optimized for this type of data manipulation.

We selected the Texas Instruments TMS320C40, a commercial 32–bit floating–point DSP, which provides the necessary high I/O bandwidth via its 6 high–speed Communication Ports and associated DMA Channels.

To implement the system we selected commercial VME products. The Hydra-II card, from Ariel Corp., offers 4 'C40 processors, and provides external access to 4 Comm Ports for each 'C40, while using the other 2 Comm Ports to ring-connect the processors. Additionally, each Hydra-II provides both VME slave and VSB master interfaces. [58]

## G.4 Level 1.5 Calorimeter Trigger Software

The Level 1.5 Calorimeter Trigger processing was decomposed into the following 4 basic steps:

1. Collect all Trigger Tower energy data
2. Scan all 1280 Trigger Towers looking for seeds[4]
3. Use an algorithm to confirm or reject these candidates

[4] Although the list of seed Towers is available at Level 1, it was much more straightforward to rebuild the list in the Local DSP's. In addition, the time and cost to build a system to do this were prohibitive given the marginal increase in performance which may have been achieved.

Table G.1: Level 1.5 Calorimeter Trigger Timing

| Level 1.5 Calorimeter<br>Trigger Activity | Step<br>Time ($\mu$s) | Elapsed Time<br>(1 Tool Call, $\mu$s) |
|---|---|---|
| Processing request for event n | 0 | 0 |
| All energy data arrives at Local DSP's | 18.8 | 18.8 |
| Local DSP candidate scan (with no Tool calls) | 30.7 | 49.5 |
| Each Local Tool call | 6.6 | 56.1 |
| Local DSP's transfer confirmed objects to Global DSP | 56 | 112.1 |
| Global DSP confirmed object scan | 5.5 | 117.6 |
| Each Global Tool call | 0.5 | 118.1 |
| Answers presented to Level 1.5 Trigger Framework | 9.4 | 127.5 |
| Experiment deadtime ends | 8.5 | 136 |
| Data Block for event n built | 252 | 388 |

4. Combine the results and form a decision based on a global analysis

Fundamental to the architecture of the Level 1.5 Calorimeter Trigger is the realization that the Trigger Tower data need only be examined in small portions. The neighbor energy summing process is a *local* process, requiring data from a contiguous set of Trigger Towers. The calorimeter is therefore partitioned geographically and serviced in parallel by the Local DSP's. The Local DSP's are responsible for the collection, scanning, and candidate confirmation phases of the Level 1.5 Calorimeter Trigger processing. They pass the confirmed objects to a Global DSP, which is responsible for the global analysis. A Service CPU oversees the operations of the DSP's, communicates with the rest of the trigger system, and provides error recovery.

There are 11 Local DSP's, each of which receives data from 256 Trigger Towers in a patch of 32 $\phi$ by 8 $\eta$ Towers. Each Local DSP is responsible for finding candidates from the central region of 4 $\eta$ Towers. The 2 $\eta$–Tower–wide patches flanking the central region are provided to meet the neighbor range goal described in Section II.

The Trigger Tower data arrive via Comm Ports on each Local DSP. Forty Comm

Ports are used in parallel for Trigger Tower data input, providing an overall input data bandwidth of 400 MBytes/sec. The data come from a "hook" that was designed into the Level 1 Calorimeter Trigger, and are captured and transported via custom FPGA cards [60].

Each Local DSP passes its confirmed objects to the Global DSP via a Comm Port. The Global DSP performs a global analysis of the confirmed objects (*e.g.* determines if one or more Level 1.5 electrons[5] have been found), and returns its results to the Level 1.5 Trigger Framework via the Service CPU. If the event is passed, the Level 1.5 Calorimeter Trigger records its results in a Data Block, which is provided to the Level 2 Trigger, and is kept as part of the event's data record.

The tasks of the Level 1.5 Calorimeter Trigger are divided into two classes. Roughly these are: application of an algorithm, the Tool code, and movement of data, the Frame code. The Frame code is responsible for receiving Trigger Tower energy data, scanning for candidates, calling the Tool for each candidate, building the Data Block, handshaking with the Service CPU, and performing initialization tasks including receiving and checking operating parameters.

By keeping the Tool distinct from the Frame, the algorithm can change without interfering with the Frame code. The Frame is by far the largest component of the DSP software. All of the DSP code was written in assembly language.

Although our use does not constitute a traditional DSP application, the powerful data access and manipulation features of the 'C40 are used to efficiently perform the required tasks. As currently operating with the algorithm described below, the Level 1.5 Calorimeter Trigger introduces an average deadtime of 136 $\mu$s/event, and an

[5] We mention here that these are electron *candidates*. The degree of confidence is described by the analysis level the candidate has be subjected to. A Level 2 electron is a better candidate than a Level 1.5, and so on, up to an offline electron which passes tight quality cuts.

overall experimental deadtime of about 2%. A detailed breakdown of the processing time is given in Table G.1.

An offline simulator emulating the hardware and software aspects of the Level 1.5 Calorimeter Trigger was developed in parallel, and was used during the algorithm selection phase and the commissioning phase of the project.

## G.5 Performance of the System

We may precipitate the motivating factors for Level 1.5 as follows: the *luminosity*, or rate of interesting events per beam crossing, increased significantly in the current collider run; the input rate to Level 2 is limited; and, the physics channels under study would be impaired if this increased luminosity forced a raising of Level 1 energy thresholds or front end *prescaling* of triggers. Prescaling is programming Level 1 to only accept a trigger 1 of $n$ times, thus making moot the increased luminosity.

The quality of an algorithm is measured by its:

- Efficiency at confirming "good" events (*i.e.* events which contain an electron)
- Rejection factor for "bad" events (*i.e.* no electron)

We consider the performance of the system to be the additional rejection of background obtained by use of the Level 1.5 system. We quantify this performance by producing algorithm dependent acceptance curves, described in Figure G.1. We use such curves to determine where to set the online cut values to stay within allowed trigger rates.

## G.6 The Electromagnetic Algorithm

To trigger on electrons, we exploit two aspects of the EM showers previously mentioned: we make clusters of 2x1 EM $E_T$ Towers to better measure the shower energy, and we calculate fraction of energy in these clustered Trigger Towers deposited in the EM layers. These two quantities are denoted Em2x1 and EM Fraction. In the Local Tool, this algorithm:

- Calculates the Em2x1 from the sum of a candidate Tower plus the most energetic of its 4 neighbors in the $\eta$ or $\phi$ direction (the catercorner Tower is not considered)
- Calculates the 2x1 Total $E_T$ of the two Towers
- Confirms a candidate as an electron if both
  1. Em2x1 $\geq$ Em2x1 Energy Threshold[6]
  2. Em2x1 / 2x1 Total $E_T \geq$ EM Fraction Threshold

In the Global Tool, this algorithm:

- Confirms the event if the number of confirmed electrons is $\geq$ Electron Count Threshold

The Em2x1 Threshold, the EM Fraction Threshold, and the Electron Count Threshold are all programmable. These 3 thresholds, taken together, are called a Parameter Set, denoted, for example, EX(1,12,0.85) for requiring 1 Level 1.5 electron at or above 12 GeV Em2x1 with an EM Fraction at or above 0.85.

[6] When we apply a threshold value such as this, we call it "making a *cut*" on the sample. Candidates which are above the threshold pass the cut, while candidates below it fail the cut. The applied threshold itself is called the cut value.

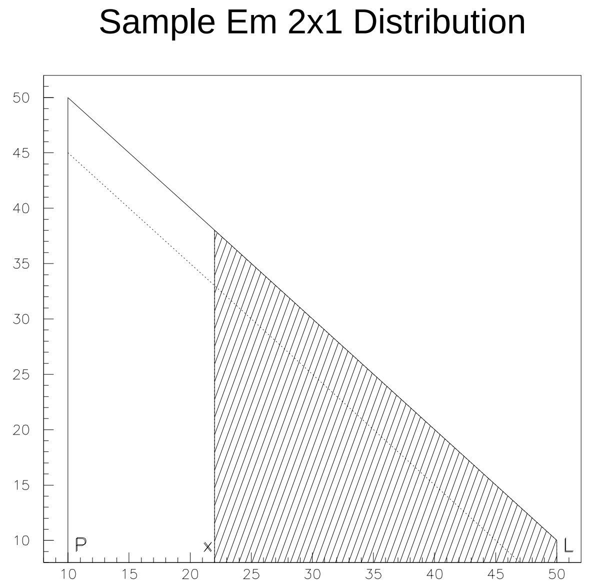


Figure G.1: Example of a distribution showing how the acceptance curves were obtained. In order to determine the acceptance curve, we histogram the value to be cut on, for example, the above Em2x1 distribution. The percent accepted for a given cut value $x$ is then the integral from $x$ to $L$ divided by the integral from $P$ to $L$. We then plot this result as a function of the cut value $x$. If multiple cuts are made, for example a Level 1 tower above 10 GeV then an Em 2x1 cut, the percent accepted from the first cut is the area under the dashed line, divided by the total area. The acceptance for the combined cut is then as above but the intergration of $x$ to $L$ now under the dashed line.

The efficiency is determined by analyzing events with electrons. For this algorithm, two samples were studied. One sample contained single high $E_T$ electrons per event. The other contained two high $E_T$ electrons per event. The events in these samples, obtained without Level 1.5 triggering, were considered "golden", meaning that any new triggering system or algorithm which decided not to accept nearly 100% of these events would be rejected from consideration for inclusion in the DØ Trigger. It was shown that the requirement EX(1,17,0.92) was 100% efficient for the single electron sample, while EX(1,14,0.92) was 100% efficient for the di–electron sample. Once this maximum cut is set, one needs to determine what value will give enough rejection to stay within an allocated bandwidth, with the caveat that the cut values should be far enough away from the 100% value to ignore the biases in the samples used for its determination.

Background samples for determining the rejection were obtained from two special configurations of the DØ Trigger; Run 82034 passed events based only on Level 1 EM Tower energies with thresholds set at 3, 7, and 12 GeV [EM(1,3), EM(1,7), EM(1,12)], and Run 86863 passed events containing two EM Towers above 7 GeV [EM(2,7)]. In both cases, Level 2 was in *mark and pass* mode—no filtering was done at this level, the Level 1 candidates were passed and written to tape. Since nearly all of the events in the mark and pass runs are background,[7] one processes the raw events through simulated Level 1.5 with nominal threshold values and makes the acceptance curve using the method described in Figure G.1. We use the acceptance rather than the traditional rejection power (1/acceptance) because it is a clearer representation of how much reduction in rate one can expect online for rejection powers between 1 and 3. To determine the acceptance curve for the single electron

[7] In normal running, on the order of 1 in 1000 Level 1 electrons produce a high quality offline electron.

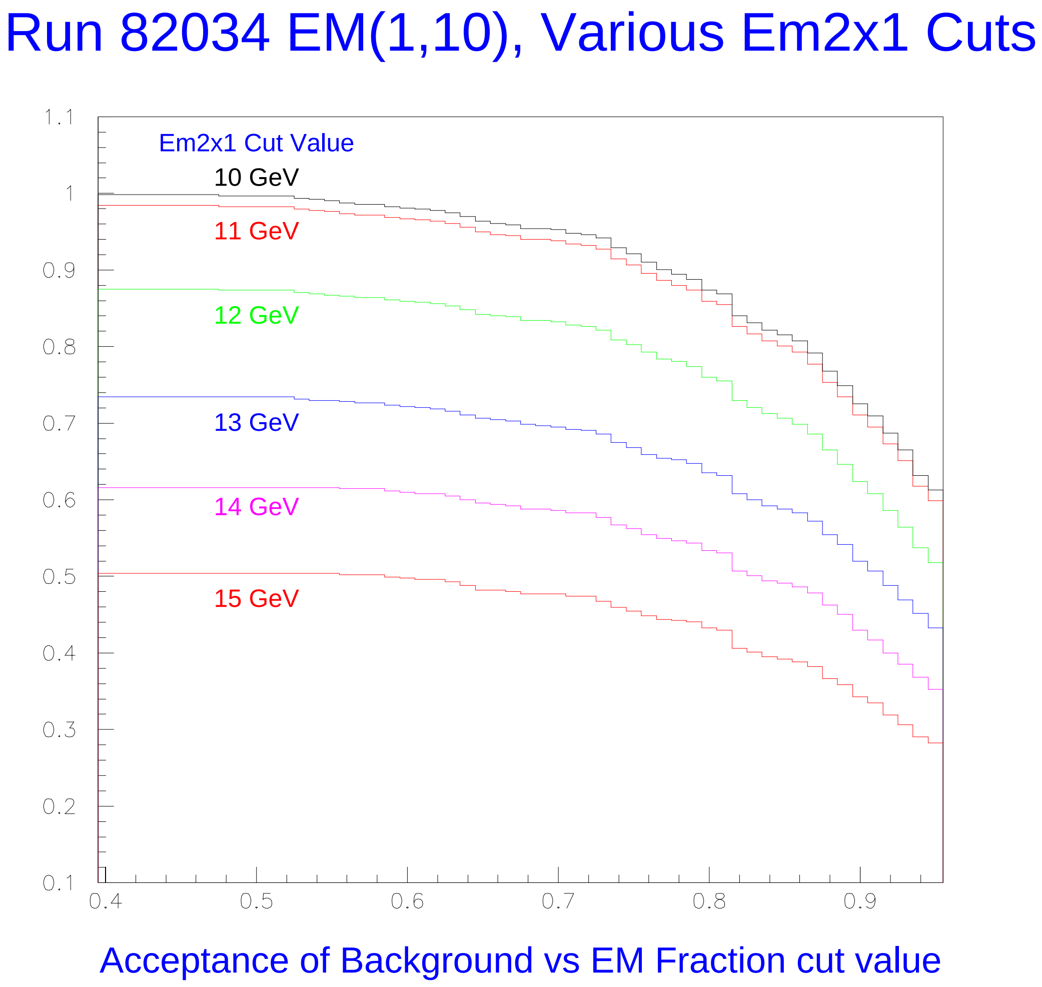


Figure G.2: The family of acceptance curves for single electron events. A cut below 10 GeV on Em2x1 produced the same results as the 10 GeV cut and thus shows the EM fraction acceptance without an Em2x1 cut.

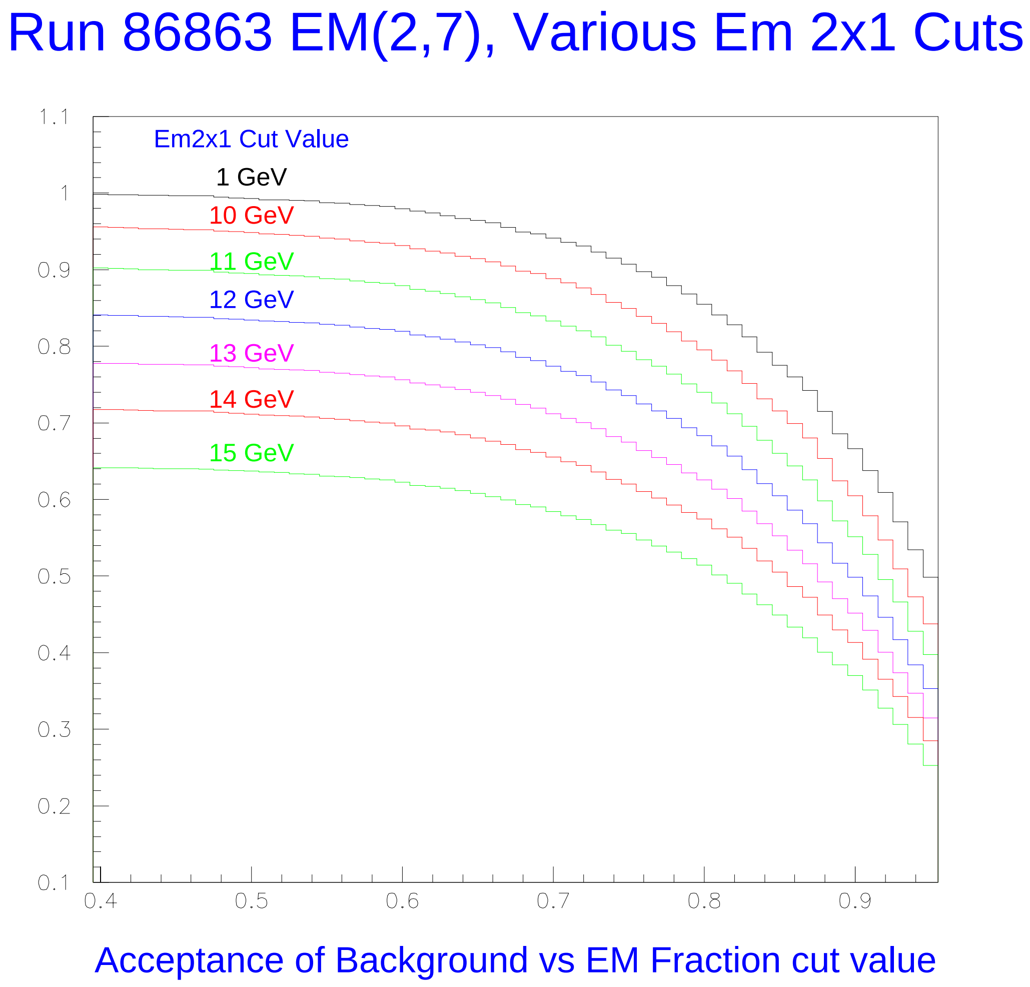


Figure G.3: The family of acceptance curves for di–electron events. A cut of 1 GeV shows the EM fraction acceptance without an Em2x1 cut.

trigger, portions of Run 82034 were filtered through the Level 1 simulation requiring EM(1,10).

Plots of the acceptance curves were generated from the special background runs using the method described in Figure G.1. With each sample, the highest EM fraction object within the event which passed an Em2x1 cut was histogramed thus providing event acceptance.[8] FigureG.2 shows the acceptance curves for the EM(1,10) sample. FigureG.3 shows the acceptance curves for Run 86863 with the same selection as above.

## G.7 Use of Level 1.5 for Triggering on $W$ and $Z$ Boson Decays

The $W$ and $Z$ boson Level 1 triggers, EM(1,10) and EM(2,7), together have an allocated bandwidth into Level 2 of 70 Hz. The first use of the Level 1.5 system was to apply a Level 1.5 Em2x1 threshold of 15 GeV to EM(1,10). Without the Level 1.5 rejection, EM(1,10) would have been raised to EM(1,12) at Tevatron luminosity of $15 \times 10^{30}$ cm$^{-2}$s$^{-1}$ in order to stay within allocated bandwidth. The sum of EM(1,10),EX(1,15) and EM(2,7),EX(1,10) fired at about 98 Hz at $20 \times 10^{30}$ cm$^{-2}$s$^{-1}$, and roughly scales linearly, thus requiring the additional rejection of the EM fraction cut. The current configuration in use at DØ, where luminosities are regularly above $20 \times 10^{30}$ cm$^{-2}$s$^{-1}$, is to apply EX(1,15,0.85) to EM(1,10), and EX(1,12,0.85) to EM(2,7). Without the additional rejection provided by Level 1.5, both triggers would have required front end prescales at these luminosities.

[8] Note that the $y$-intercept of the curve is the acceptance of the Em2x1 cut value without an EM fraction cut.

## G.8 Conclusions

We have shown the acceptance curves for the two Level 1.5 algorithms in use at the DØ detector. The estimated values have been verified online. The use of the Level 1.5 system has allowed DØ to adapt to the increased luminosity at the Tevatron without altering the scope of the physics program pertaining to electromagnetic final states. Specifically, it has allowed DØ to keep $W$ and $Z$ boson electromagnetic decay triggers unprescaled, and at their respective 1992-93 Tevatron Run Level 1 thresholds, by introducing only a few percent deadtime. The flexibility of the system allows DØ to make maximum use of the delivered luminosity through the current Tevatron Run.

# Appendix H

# A Study of Shower Depth Bias in Electron Cluster $z$ Position Determination for the Central Calorimeter

The calorimeter cluster z position determination is studied. A correction to the standard parameterization based on shower depth is presented. An alternate parameterization to the standard clustering and a new clustering algorithm is discussed. The plan for applying these latter corrections to data is shown. Finally, plots are presented for comparison between the new corrections and the current standard.

## H.1 Introduction

The center of gravity (cog) of an electron cluster in the DØ calorimeter originally was only determined using a energy log-weighting of non-zero suppressed cells in the EM3 layer through the routine CM3POS. The algorithm is described in section 3.13.5 of *DØ Software Documentation*. Also mentioned there is a theta dependent correction to the cog $z$ position done in the routine CM3POS_PV (PV for primary vertex, needed to compute theta) which arises from the sampling and the finite cell size. This latter $z - bias$ correction is known to be weakly energy dependent for high $E_T$ (above 25 GeV) clusters. The parameterization of the bias was done with 50 GeV Monte Carlo single track electrons and the energy dependence contributes to the calorimeter $z$ position resolution (the *standard parameterization*). The standard parameterization doesn't work well for lower energy electrons. As an example, Figure H.1 shows a comparison of the error in cluster determined $z$ for 50 GeV energy and 2 GeV $E_T$ electrons (plate MC without noise).

The Monte Carlo $z$ is determined by projecting the ISAJET track to a radius determined from the $x, y$ calorimeter cluster positions. The $x, y$ cluster resolution is negligible for the purpose at hand. Nearly the same comparison is obtained using a nominal EM3 radius of 91.68 cm. Plots for 3-25 GeV $E_T$ are similar, although the bias decreases with increasing energy. One approach to correct for the energy dependence is to parameterize at several energies and apply an appropriate correction to data. A better approach, the subject of this note, is to understand the energy dependence, remove it from the correction, and apply a second energy dependent correction if needed.

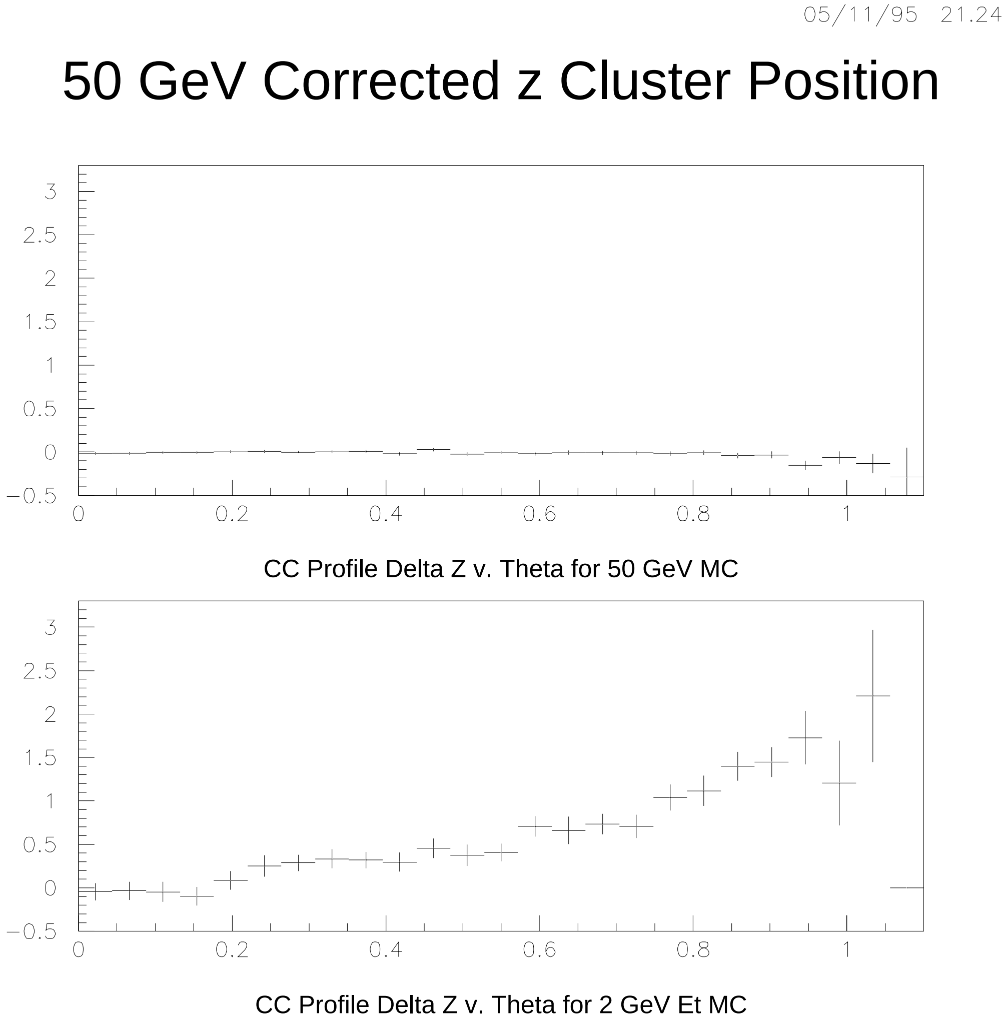


Figure H.1: Profile histograms for two samples of CC Monte Carlo electrons. The x-axis is the angle relative to normal incidence into EM3 in radians. The y-axis is the cluster determined $z$ position in EM3 minus the Monte Carlo $z$ position.

## H.2 Energy Dependence in the Correction

The primary contribution to the energy dependence is introduced in the standard parameterization itself (Figure H.2). The DØ calorimeter was designed such that 25 GeV electron showers at normal incidence have a cog radius equal to the nominal radius of the center of EM3. Depending upon shower energy and incident angle, the radius of the shower cog and the nominal center of EM3 may differ, producing a systematic error in the determination of the projected ISAJET track. To calculate the EM3 cog along the electron track, I model EM3 as a 6.77 cm deep layer containing 6.8 radiation lengths, preceeded by 4.1 radiation lengths of pre-EM3 material. Ignoring the contribution of other materials (LAr, G10, for example) to the shower profile function

$$X_{cog} = \frac{\int_{X_1}^{X_2} f(x)x\,dx}{\int_{X_1}^{X_2} f(x)\,dx}$$

where $X_1$ and $X_2$ are the starting and ending radiation lengths of EM3; for example, normal incidence would set $X_1 = 4.1$ and $X_2 = 10.9$, while tracks incident at angle $\theta = \pi/4$ have $X_1 = 4.1/\cos(\theta) = 5.8$ and $X_2 = 15.4$. The shower profile function, $f(x)$, is taken from PDG [6] for pure uranium. To illustrate, Figure H.3 shows shower cog for normal incidence for energies between 1 and 75 GeV, while Figure H.4 shows cog for various energy showers as a function of incident angle.

The energy dependence was introduced when the parameterization was done to correct the 50 GeV sample to the projected EM3 z position at nominal radius instead of the actual cog radius. This is described in Figure H.2. If the finite cell size and sampling are ignored, the log weighted average z position should return z at cog. To

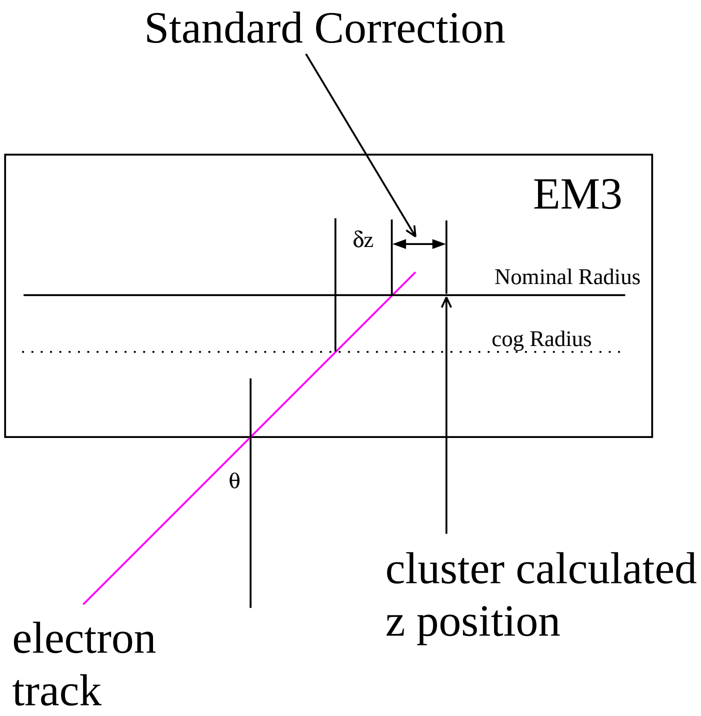


Figure H.2: Shown is an electron track an incident angle $\theta$. The standard correction compensates for the theta bias by parameterizing a sample of constant energy, 50 GeV. This correction moves the cluster calculated cog to the Monte Carlo z position at nominal EM3 radius. The energy and angle determine a unique radius for the actual cog, shown, where one should correct to in order to avoid the bias shown in Figure H.1. The shower depth bias for this track is shown as $\delta z$.

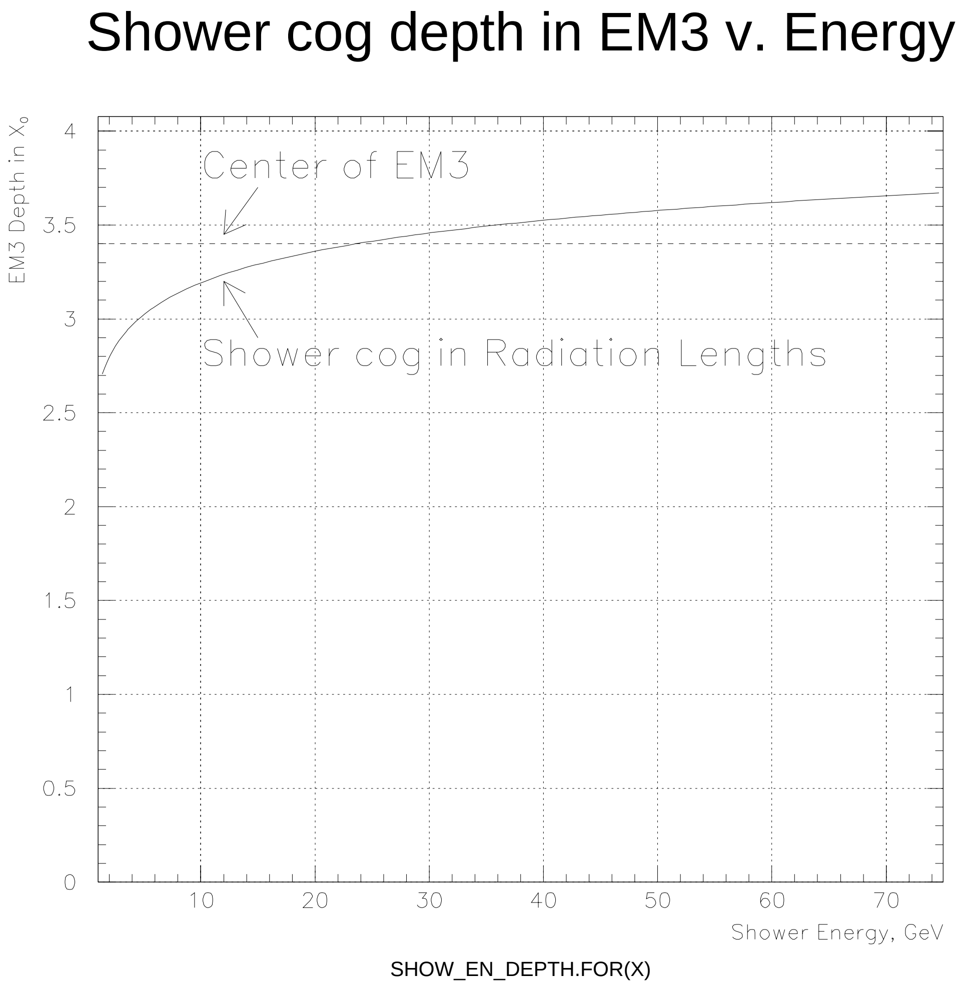


Figure H.3: Perpendicular depth, in radiation lengths, of shower center of gravity in EM3 as a function of shower energy. Note that the fourth of seven uranium plates cover approximately 3-4 radiation lengths; thus, for energies above 4 GeV, normal incidence leads to cog in the same plate.

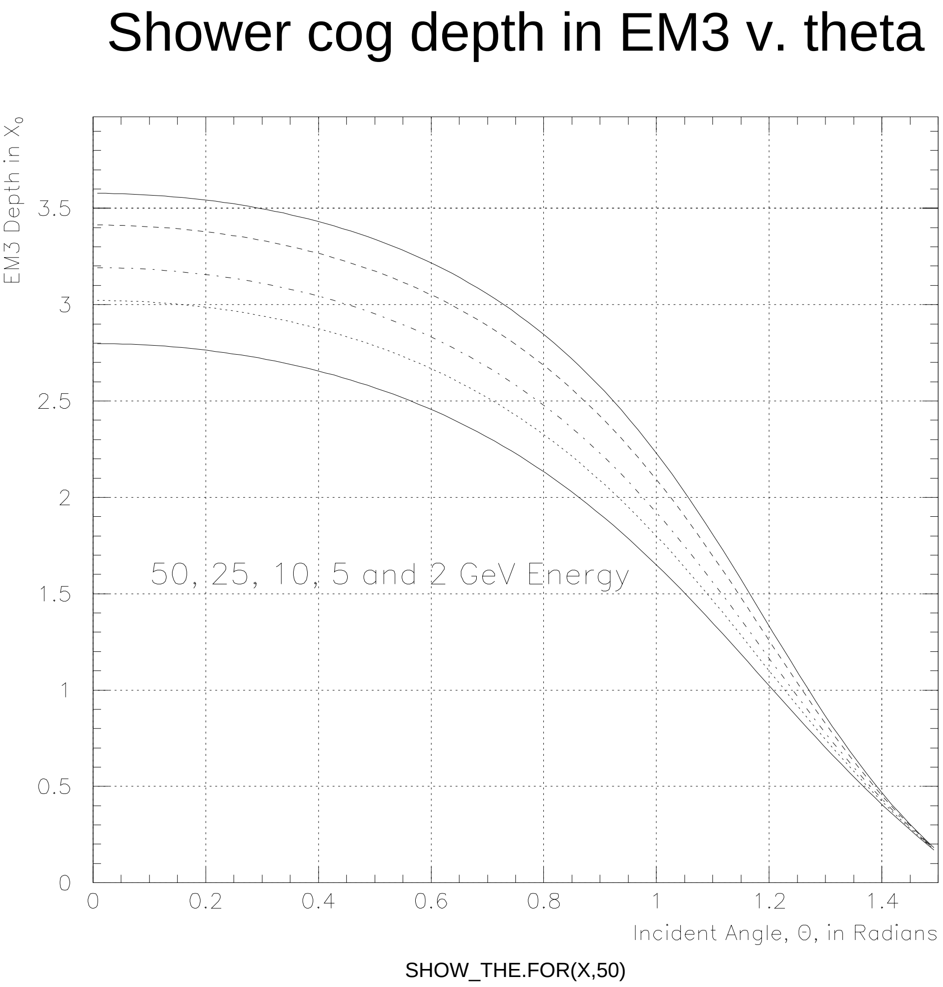
approximate this position in the EM3 layer, I take the ratio

$$\frac{X_\perp}{X_{EM3}} = \frac{D_\perp}{D_{EM3}}$$

where the two numerators are radiation and physical cog depth into EM3, and the two denominators are total EM3 depths (both measured perpendicular to the beam axis). The depth, or perpendicular distance into EM3, is calculated from $X_{cog}$ and incident angle, and the cog radius is determined from $D_\perp$ and the inner radius of the EM3 layer. The difference between the EM3 radius calculated this way, and the nominal position is shown in Figure H.5 for several energies. The effect on the z position of changing the radius necessarily includes a dependence on incident angle. Geometrically, $\Delta z = \Delta R \tan\theta$, and this z-bias is shown in Figure H.6 for the energies from Figure H.5, and in Figure H.7 for a few values of $E_T$.

## H.3 The Standard Correction and Clustering

The sampling and finite cell size introduce a bias primarily as a function of incident angle. The standard parameterization in effect corrects these biases to the nominal EM3 radius. Since the parameterization was done at a single energy, the effect shown here can be included by normalizing to the shower depth z-bias of the 50 GeV reference curve. To this end, the reference sample is used to correct for biases arising from sampling and finite cell size, and the difference in the shower depth z-bias between the electron and the reference is applied to correct for most of the energy dependence. In effect, one parameterizes the bias at 50 GeV, and includes the difference between the 50 GeV curve in FigureH.6 and the appropriate curve based on the energy of the electron. While this doesn't remove the entire energy


Shower cog depth in EM3 v. theta

SHOW_THE.FOR(X,50)

Figure H.4: Perpendicular depth, in radiation lengths, of shower center of gravity in EM3 as a function of incident angle for energies 50 (top curve), 25, 10, and 5 GeV. Note that the shower cog for all energies plotted can occur in plates 2, 3, or 4.

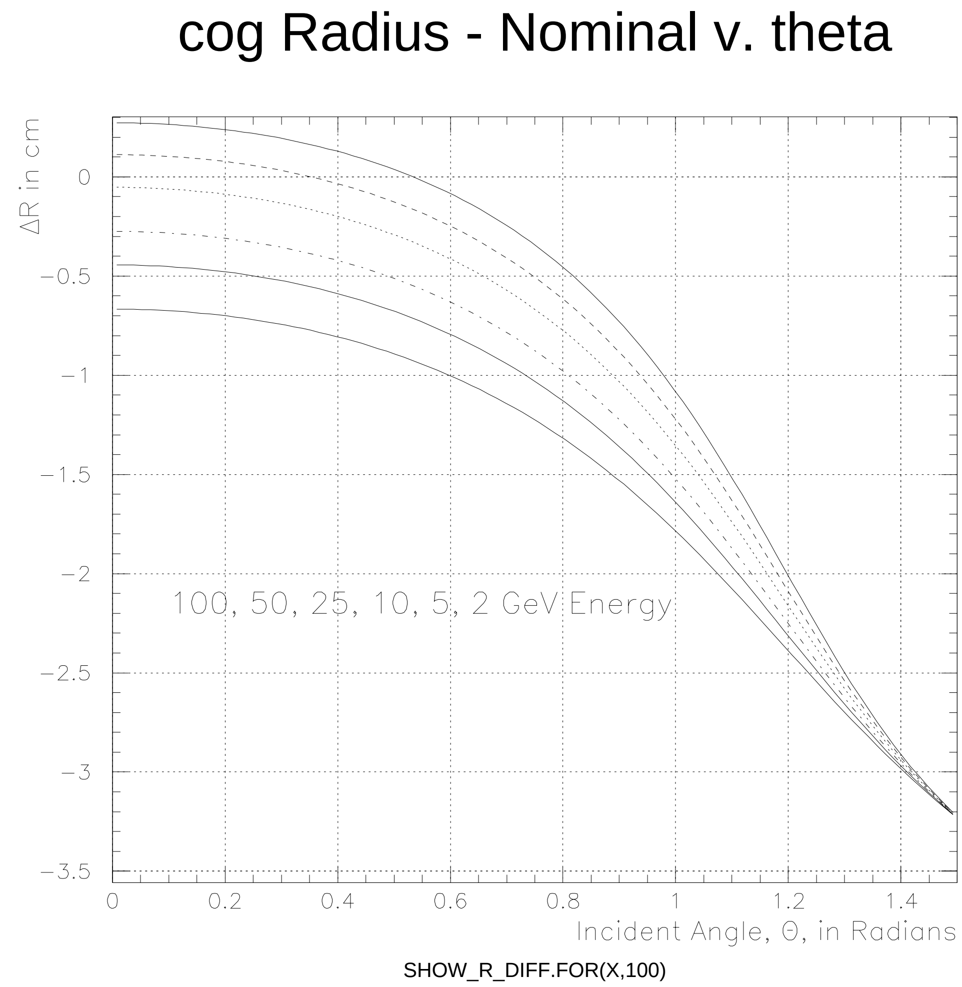


Figure H.5: Difference of shower center of gravity radius and the nominal radius of EM3, in cm, as a function of incident angle for shower energies 50 (top curve), 25, 10, 5, and 2 GeV.

dependence, it reduces it noticeably. Figure H.8 shows this correction to z as a function of incident angle for various energies. The effect of applying this energy and theta correction is shown in Figure H.9 for the 2 GeV $E_T$ sample presented in the Introduction. Applying this correction to 3-25 GeV $E_T$ samples doesn't give as good of results for high $\theta$ as presented in the Figure. This is due to the IETA dependent correction discussed below (it is implied that the really nice agreement for 2 GeV $E_T$ might have been coincidental). Applying this correction to the standard parameterization does give an improved $z$ position. To optimize, one should include the new radius in the parameterization, which we proceed to next.

## H.4 Reclustering

For low energy electrons, the standard clustering algorithm doesn't do as well as a fixed window size algorithm. This is due to the fact that noise cells, which are picked up in low and high energy clusters, affect the cluster centroid position determination to a greater degree. Using a 1.5x1.5 tower cluster around the hottest EM3 cell reduces the effect of noise on low energy electrons. This size is a compromise between concentrating on the core of the cluster, and keeping enough information to make a reasonable centroid determination. There are up to nine EM3 cells in this clustering. For a change in the clustering, one needs to redo the reference parameterization. Figure H.10 show fit parameterizations for the standard cluster and the 3x3 EM3 clusters using the new radius and a flat weight cut. The flat weight cut refers to an IETA dependant cut on weights in the cluster centroid determining algorithm. The standard parameterization includes an IETA dependance in the weight

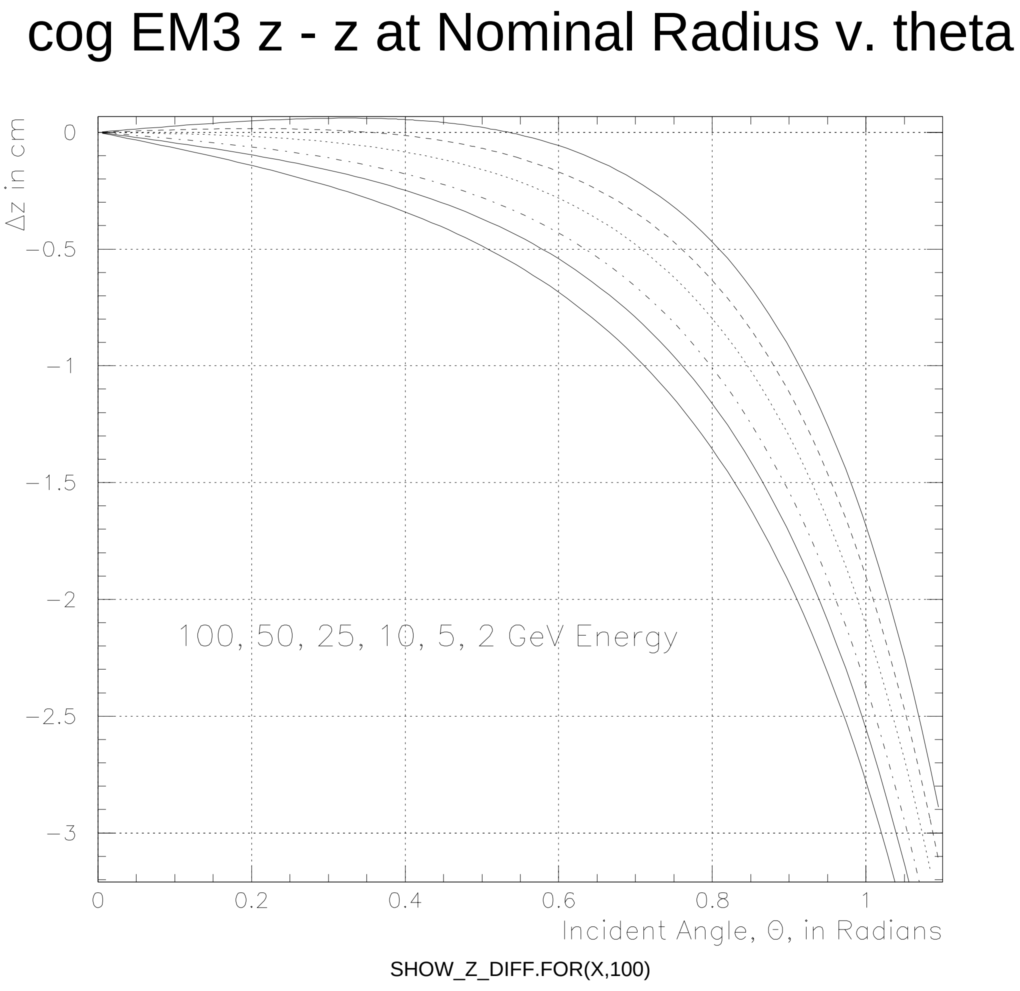


Figure H.6: Difference between EM3 z position at nominal and calculated radius, in cm, as a function of incident angle for energies 100 (top curve), 50, 25, 10, 5, and 2 GeV. Note the difference between the 50 GeV curve, where the parameterization was done, and the 25-100 GeV range.

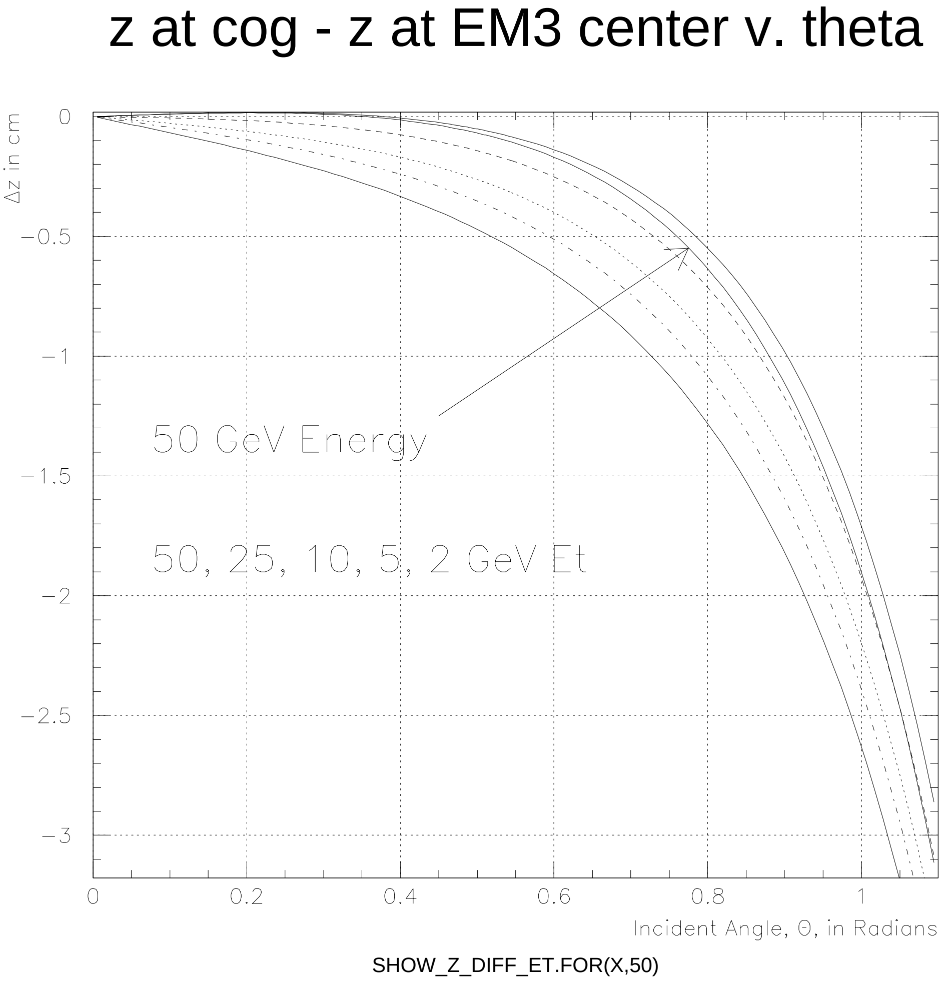


Figure H.7: Difference between EM3 z position at nominal and calculated radius, in cm, as a function of incident angle for transverse energies 50 (top curve), 25, 10, 5, and 2 GeV. Note the difference, about a mm, from the 50 GeV energy curve, and the two high $E_T$ curves as compared with the Figure H.6

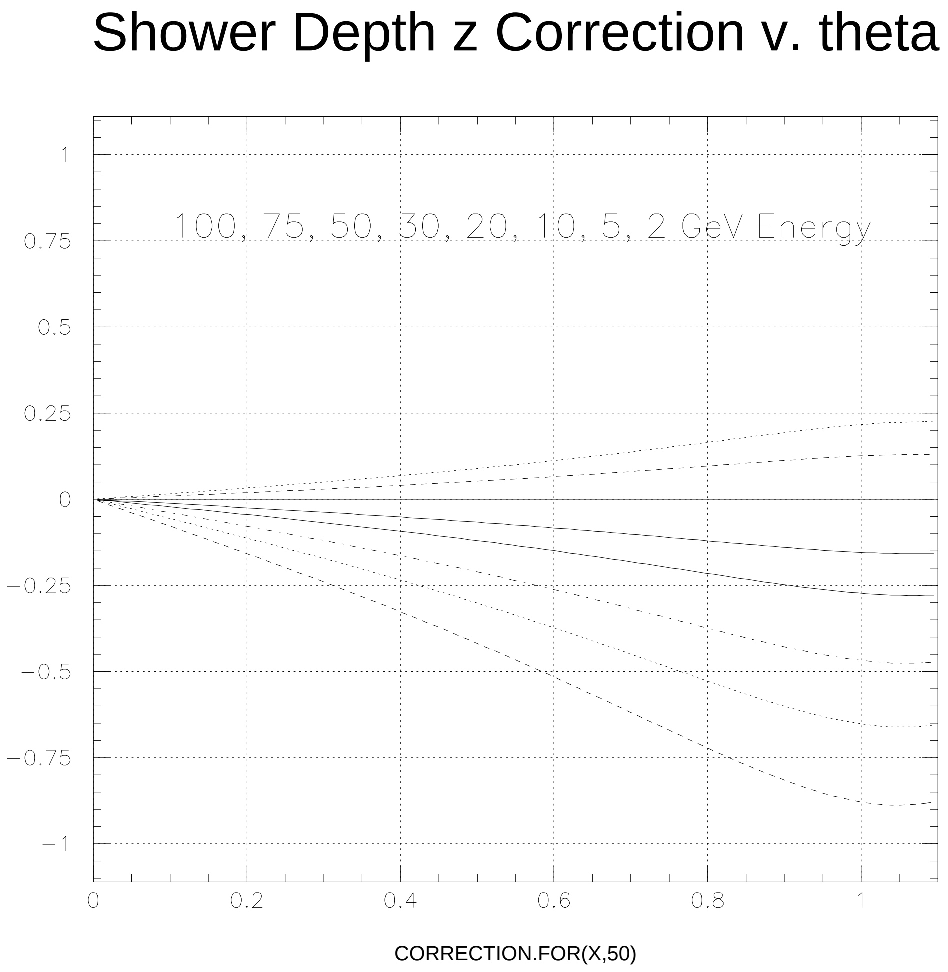
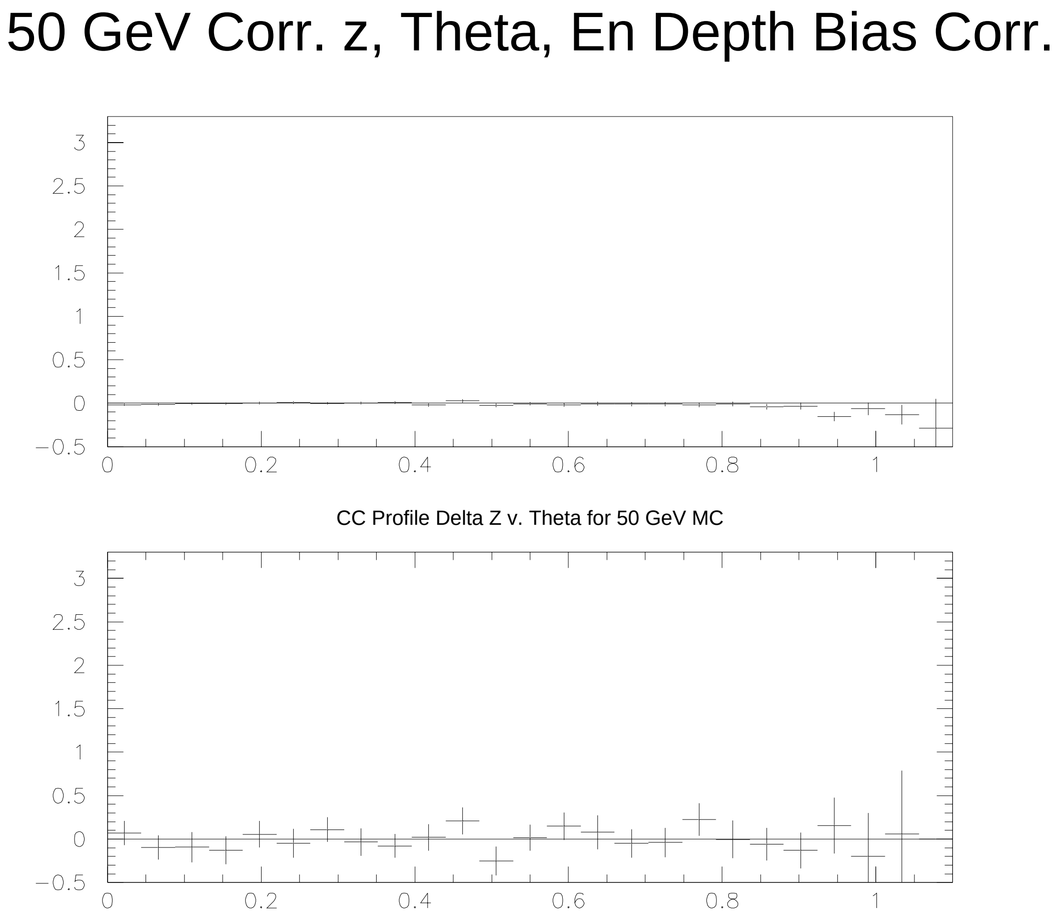


Figure H.8: Difference between the z bias of various energy electrons and the reference curve. Basically, the difference between the 50 GeV curve and others in Figure H.6.

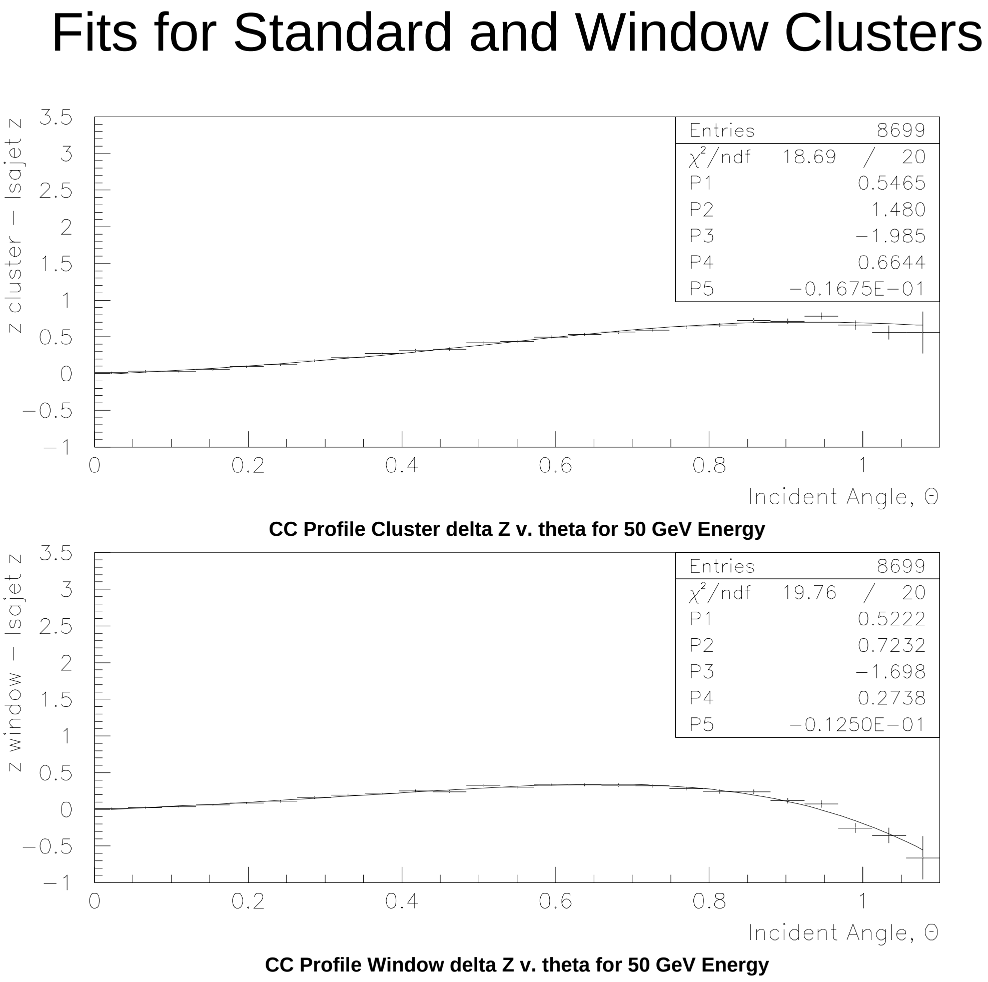


50 GeV Corr. z, Theta, En Depth Bias Corr.

CC Profile Delta Z v. Theta for 50 GeV MC

CC Profile Delta Z v. Theta for 2 GeV Et MC

Figure H.9: The same plots for the same samples as Figure H.1 but the theta and energy shower depth z bias correction has been applied to the 2 GeV $E_T$ sample.

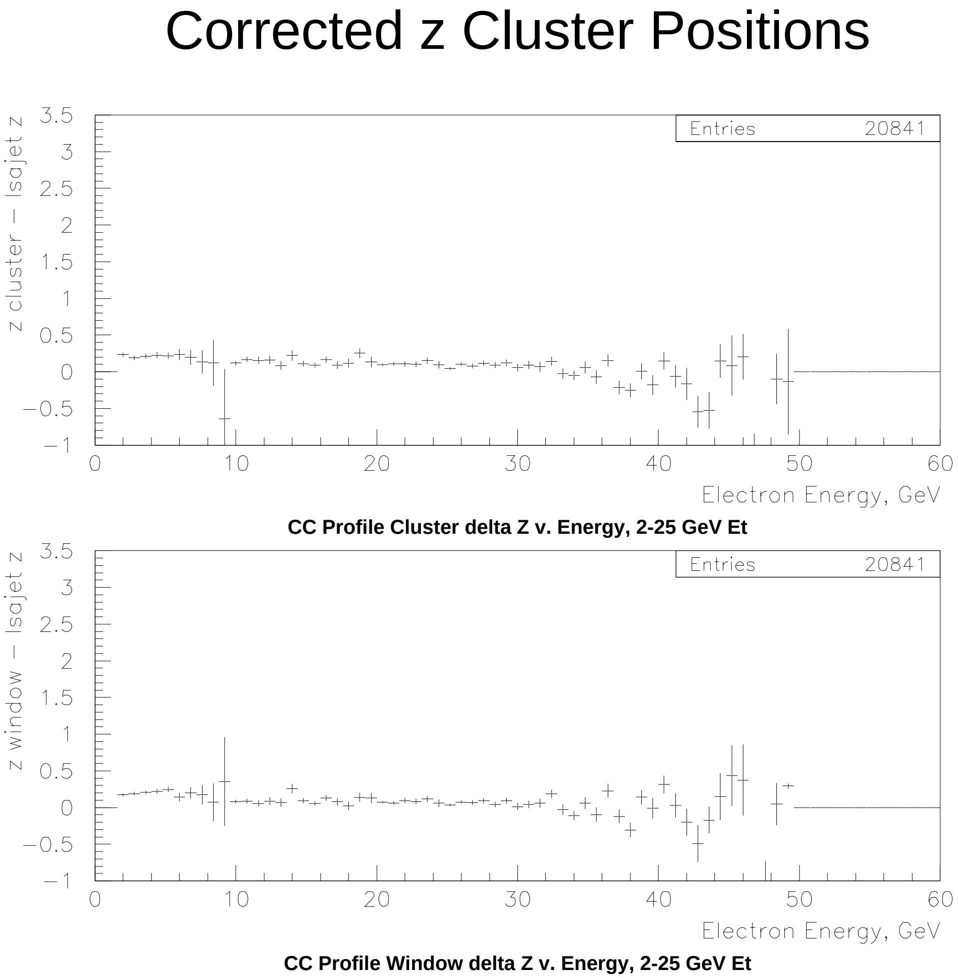


Fits for Standard and Window Clusters

CC Profile Cluster delta Z v. theta for 50 GeV Energy

CC Profile Window delta Z v. theta for 50 GeV Energy

Figure H.10: Z bias fits for standard and 1.5x1.5 tower clusters. The function used is $\mathrm{P1}x + \mathrm{P2}x^3 + \mathrm{P3}x^5 + \mathrm{P4}x^7 + \mathrm{P5}$. In both cases, the cog radius was used along with flat weight cuts.



**CC Profile Cluster delta Z v. Energy, 2-25 GeV Et**

**CC Profile Window delta Z v. Energy, 2-25 GeV Et**

Figure H.11: Corrected z positions for 2 to 25 GeV $E_T$ Monte Carlo, shown as a function of energy. Above is corrected for standard clusters, while below is for 1.5x1.5 tower clusters.

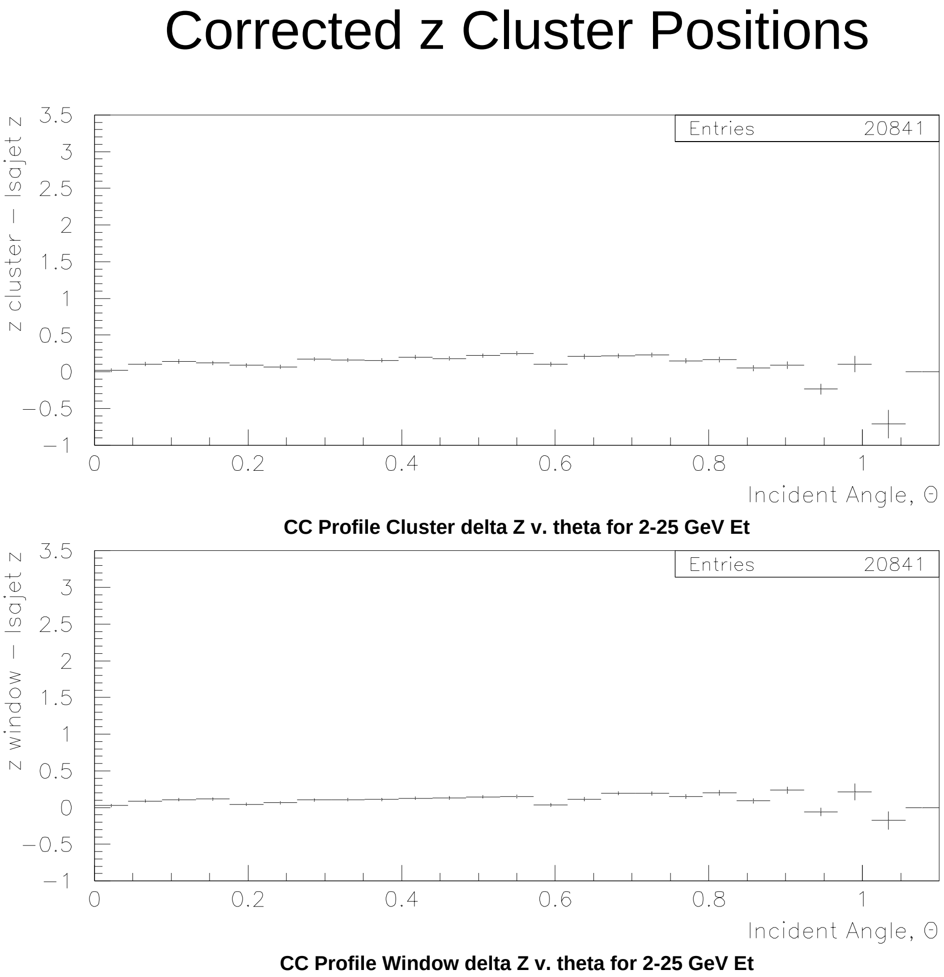
cut determined by optimizing resolution as a function of IETA.[1] This optimization interferes with the energy correction and is therefore removed. Doing the energy correction instead of the IETA parameterization doesn't have any obvious negative effect on resolution. Figure H.11 and Figure H.12 show the results of applying these corrections to a combined sample of single track electrons generated with $E_T$ of 2, 3, 4, 5, 10, 15, 20, and 25 GeV. The former shows the residual bias as a function of energy, while the latter plots the bias as a function of incident angle. The term *window* refers to the 3x3 EM3 clustering, also called the 1.5x1.5 tower clustering.

## H.5 Applying corrections to Data

Applying the shower depth correction to data is straightforward; the necessary elements are the electron energy, and its incident angle. Both of these quantities have associated errors; however, since the correction is well behaved, the effect on the size of the correction is less than the correction. The difficulty in using the Monte Carlo results directly to data is the deformity in the calorimeter. Figure H.13 shows the calculated cluster radius for a sample of 24,000 electrons from $W$ boson decays. The sample is divided into four Z bins between ± 100 cm in EM3. This shows the deformation of the calorimeter, and its skewed angle with respect to the beam axis and thus the DØ coordinate system. The position determination is dominated by the survey constants and is not determined as a function of energy.[2] Given this situation, one uses the $x, y$ position of the EM cluster to determine the local EM3

[1] This effectively parameterizes the needed correction for electrons according to the distribution of energies from $Z \rightarrow ee$ events.

[2] The log weighting of cells in the cluster centroid determination uses the nominal center of the cell for each point in the algorithm. These positions define a radius determined from surveys of uranium plate support structure. The energy deposited in the cell has no effect on the $x, y, z$ coordinate used for the cell.


Corrected z Cluster Positions

CC Profile Cluster delta Z v. theta for 2-25 GeV Et

CC Profile Window delta Z v. theta for 2-25 GeV Et

Figure H.12: Corrected z positions for 2 to 25 GeV $E_T$ Monte Carlo, shown as a function of incident angle. Above is corrected for standard clusters, while below is for 1.5x1.5 tower clusters.

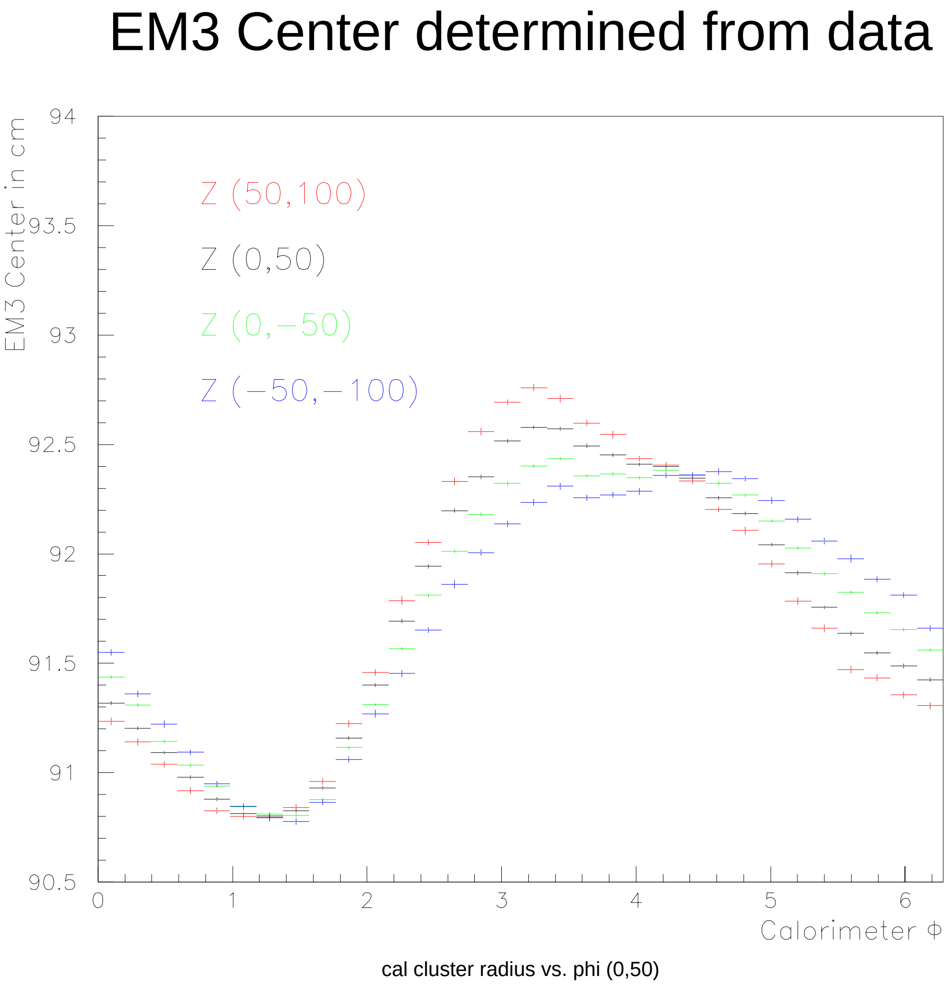
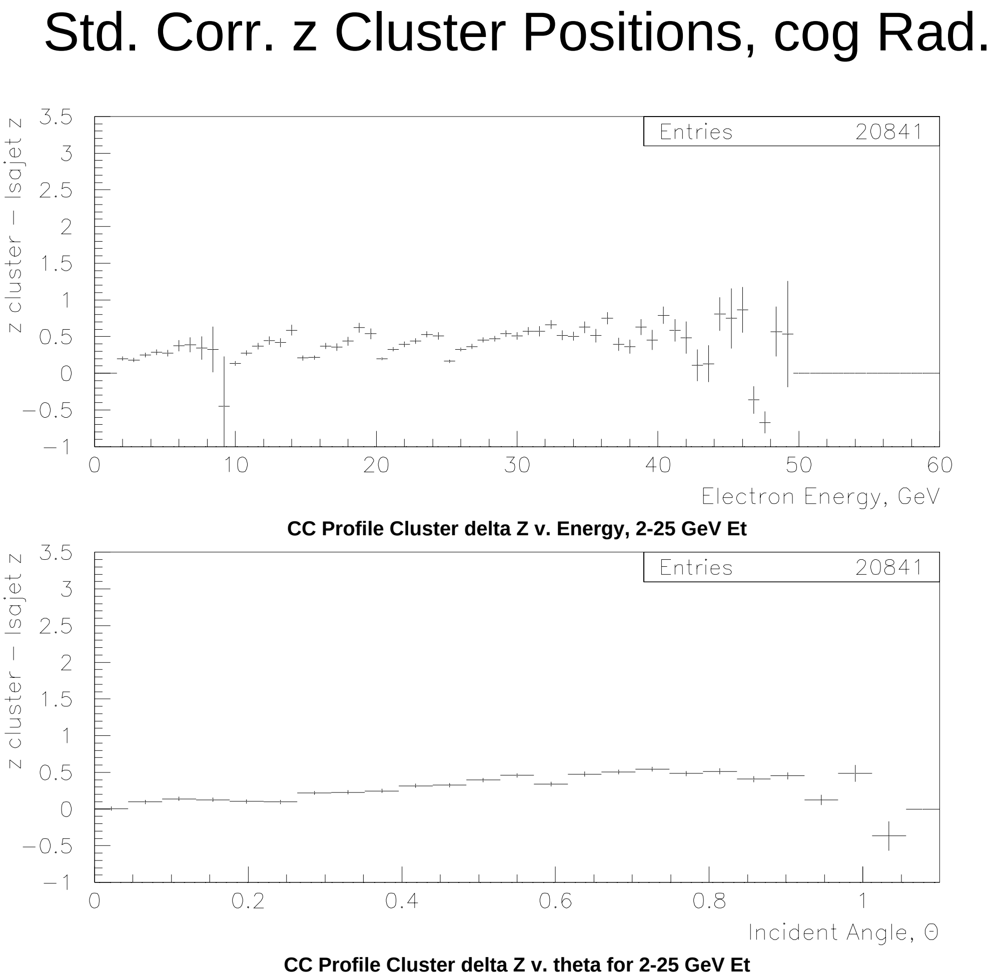


Figure H.13: EM3 radii from electromagnetic clusters in the $W$ boson sample.



Std. Corr. z Cluster Positions, cog Rad.

CC Profile Cluster delta Z v. Energy, 2-25 GeV Et

CC Profile Cluster delta Z v. theta for 2-25 GeV Et

Figure H.14: Cluster z bias calculated with respect to the cog radius. The standard parameterization has been applied, and no additional corrections.

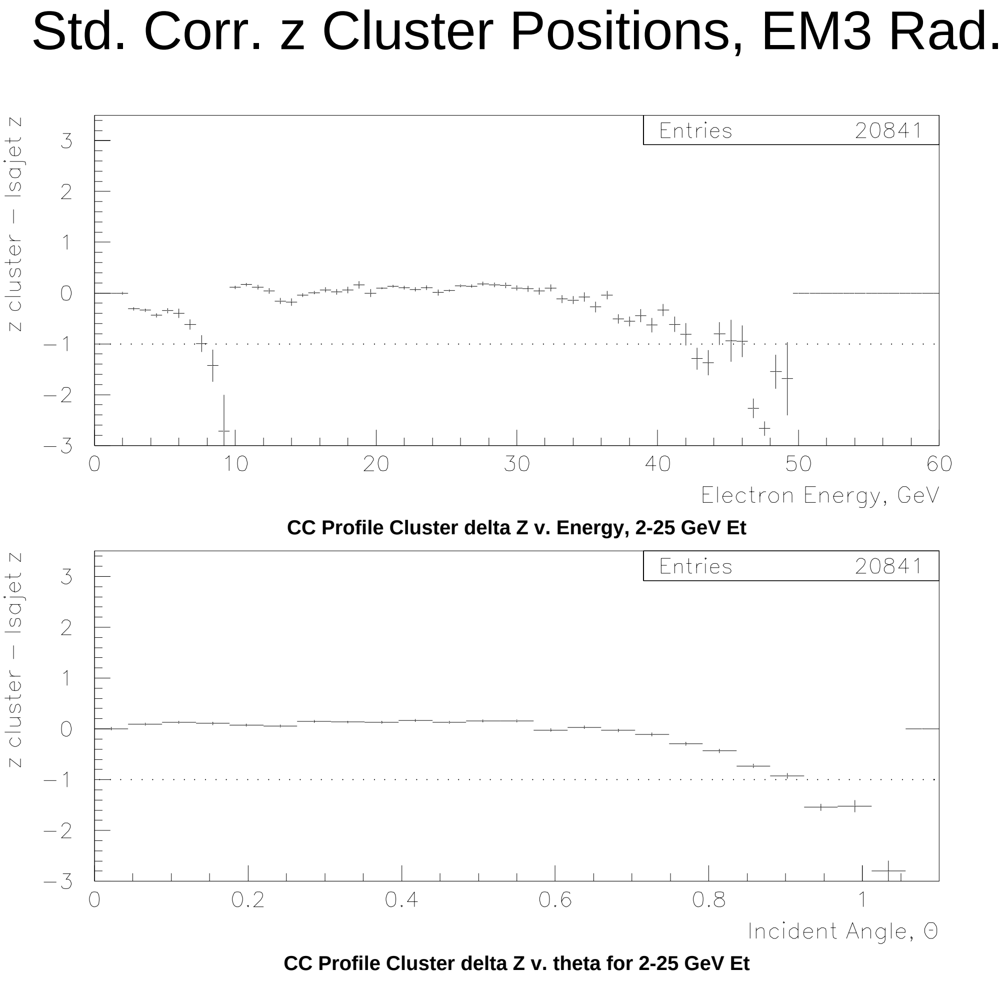
center, and subtracts half of the EM3 design thickness to obtain the inner radius of the EM3 layer. Then the cog correction proceeds as in Monte Carlo with the inner radius now a function of $z$ and $\phi$. [3]

The corrections described here will be written into a new version of CM3POS_PV, to be called CM3POS_PVG. One should remember that the returned $x, y, z$ will be at shower cog and not at nominal EM3 center.

## H.6 Comparisons with the Standard Parameterization

For the sake of comparison with the current correction, two additional plots are included that are generated from the above mentioned sample of 2-25 GeV $E_T$. Figure H.14 shows the current cluster bias using the ISAJET track projected up the calculated cog radius. Figure H.15 shows the current bias where the ISAJET track has been projected up to the local EM3 radius. The radius in Figure H.15 is the value usually used and is an indication of the dependence in the current electron identification standard.

[3] This procedure necessarily assumes that the internal module structure remains intact and the deformation is in the superstructure aligning the modules. As understood by the author, the *dogbones* and inter-plate spacers preserve module integrity, while slight deformation of the supporting ring has produced the observed radii. The skewed nature as a function of z is just due to slight misalignment with the beam axis. The magnitude of these effects are impressively small given the size and weight of the calorimeter.



Std. Corr. z Cluster Positions, EM3 Rad.

CC Profile Cluster delta Z v. Energy, 2-25 GeV Et

CC Profile Cluster delta Z v. theta for 2-25 GeV Et

Figure H.15: Cluster z bias calculated with respect to the EM3 radius. The standard parameterization has been applied, and no additional corrections. Since the some entries extends below the scale of the previous plots, a dotted line has been included to indicate the former minimum.

## H.7 Conclusions

An energy dependent correction to the shower center of gravity z position has been presented. The new corrections exhibit less energy dependence, especially at low $E_T$ when compared with the standard corrections. It is hoped that this study, along with other studies such as TRD information, will significantly improve the identification efficiency of low transverse energy electrons.

# Bibliography

[1] M. Kaku, *Quantum Field Theory*, Oxford University Press, 1993.

[2] B.Lee, C. Quigg, and H. Thacker, *Phys. Rev.* **D16** (1977).

[3] H. Baer, et al., "Low Energy Supersymmetry Phenomenology", hep-ph/9503479.

[4] V. Barger, R. Phillips, *Collider Physics* (Adddison-Wesley, Menlo Park, CA), 1987.

[5] M. Carena, et al., "The Search for Supersymmetry at the Tevatron Collider", hep-ex/9712022.

[6] Particle Data Group, *Phys. Rev.* **D54**, part 1 (1996).

[7] S. Abachi, et al. (DØ Collaboration), *Nucl. Instr. and Meth.* **A338** 185 (1994).

[8] J. Krane, "The Ratio of Inclusive Jet Cross Sections at $\sqrt{s} = 630$ GeV and $\sqrt{s} = 1800$ GeV", Ph.D Thesis, University of Nebraska, October 1998 (unpublished), Chapter 2.

[9] J. McKinley, "A Measurement of the Inclusive Drell-Yan $e^+e^-$ Cross Section in the Invariant Mass Range of 30-60 GeV/c$^2$ from $p\overline{p}$ Collisions as $\sqrt{s} = 1800$ GeV", Ph.D Thesis, Michigan State University, 1996 (unpublished), Chapter 3.

[10] A.R. Clark et al., *Nucl. Instr. and Meth.* **A315** 193 (1992).

[11] J. Kadyk, *Nucl. Instr. and Meth.* **A300** 436 (1991).

[12] J.D. Jackson, *Classical Electrodynamics*, John Wiley and Sons, 1975 (Second Edition).

[13] V.L. Ginzberg and I.M. Frank, *JETP* **16** 15 (1946).

[14] F. Feinstein, Ph.D. Thesis, Northwestern University (unpublished) (1992).

[15] A.R. Clark et al., *Nucl. Instr. and Meth.* **A279** 243 (1987).

[16] R. Avery et al., *IEEE Trans. Nucl. Sci.* **NS-40** 573 (1993).

[17] T. Ferbel, "Experimental Techniques in High-Energy, Nuclear, and Particle Physics," World Scientific, 1991, pg 6.

[18] W. Guryn, *Proc. Second Int. Conf. on Calorimetry in High Energy Physics*, Capri, Italy (1991).

[19] H. Aihara et al., *Nucl. Inst. and Meth.* **A325** 393 (1993).

[20] Terry Geld, *The DØ Intercryostat Detector: Design Considerations and Test Beam Studies and Initial Performance*, Ph.D. Thesis, University of Michigan (unpublished) (1993).

[21] R. Angstadt, et. al., Proc. IEEE Meeting, Nov. 1991 (Santa Fe, NM).

[22] M. Goossens, CERN Program Library Long Writeup **Q100** (unpublished) (1991).

[23] B. Hirosky, "Jet Efficiencies and Seed Tower Distributions in the DØ Cone Algorithm," DØ Internal Note #2405, January 1995.

[24] L. Babukhadia, et. al., "Standard Jet Cuts and their Efficiencies for Run 1b DØ Collider Data," DØ Internal Note #3407, March 1998.

[25] G. Manning (Editor), "DØ Software Documentation," DØ Internal Publication (1995). The jet preclustering algorithm description was titrated from the RECO source code, subroutine CLUPRE.

[26] I. Adam, Ph.D. Thesis, Columbia University (Unpublished) (1997). E. Flattum, Ph.D. Thesis, Michigan State University (Unpublished) (1996).

[27] R. Brun and C. Carminati, *CERN Program Library Writeup* **W5013** (unpublished) (1993) release v3.14.

[28] M. Narian and U. Heintz, "Electron ID with Transition Radiation Detector and Central Drift Chamber", DØ Internal Note #2355 (1994), "A Likelihood Test for Electron ID", DØ Internal Note #2386 (1994).

[29] J. M. Butler, *et. al.*, "Measurement of the Top quark production cross section using lepton + jets events," DØ Internal Note # 2978 (Dec. 1995).

[30] I. Bertram, et al., "A Recipe for the Construction of Confidence Limits", DØ Internal Note #2775 (1995).

[31] J. Linnemann, "How Hard Should You Cut When the Data Sample is Limited", at Computing in High Energy Physics, Rio De Janiero, September 1995. World Scientific (1996), p. 205-209.

[32] B Knuteson, "Algorithms for Computing Significance", DØ Internal Note #3345 (1997).

[33] R. Partridge, "Optimizing Statistical Significance for Small Signals", DØ Internal Note #3383 (1998).

[34] S. Mrenna, "SPYTHIA, A Supersymmetric Extension to PYTHIA 5.7", ANL-HEP-PR-96-63 (unpublished).

[35] R. J. Genik II, "Backgrounds", DØNote in preparation. See Chapter 5 for this information.

[36] S. Eno, L. Lueking, and R. Genik, "Verification of the Fast MC QSIM", DØNote in preparation. S. Eno and J. Hobbs, "A Manual and Description of the QSIM Smearing", DØNote in preparation. J. Hobbs, "TSIM and RSIM Manual and Description", DØNote in preparation.

[37] T. Sjöstrand, "PYTHIA 5.7 and JETSET 7.4, Physics and Manual", CERN-TH.7112/93, Dec. 1993 revision.

[38] F. James, "RANLUX: A Fortran implementation of the high-quality pseudorandom number generator of Lüscher", Computer Phys. Commun. 79 (1994) 111-114. Also, see CERNLIB Short Write-up V115.

[39] G. L. Kane, Chris Kolda, Leszek Roszkowski, and James D. Wells, "Study of Constrained Minimal Supersymmetry ", UM-TH-93-24, Oct 1993. 77pp. Published in Phys.Rev.D49:6173-6210,1994.

[40] CERN Program Library Long Write-Up Q121, *PAW*, (CERN, Geneva, Switzerland, 1989).

[41] CERN Program Library Long Write-Up Y-250, *HBOOK*, (CERN, Geneva, Switzerland, 1992).

[42] F. J. Wall, *Statistical Data Analysis Handbook*, (McGraw-Hill, New York, 1986), chapters 15, 17.

[43] W. T. Eadie, D. Drijard, F. E. James, M. Ross, B. Sadoulet, *Statistical Methods in Experimental Physics* , (North-Holland Publishing Company, Amsterdam, 1971).

[44] CERN Program Library Short Write-Ups, (CERN, Geneva, Switzerland, 1989).

[45] E. L. Lehmann, *Testing Statistical Hypotheses*, (Wiley, New York, 1959), pp. 140-3.

[46] M. Abramowitz and I. A. Stegun (eds.), *Handbook of Mathematical Functions*, (National Bureau of Standards, Washington, D. C., 1964).

[47] J. McKinley, J. Linnemann, "Description of the Level 2 Electromagnetic Filter Algorithm and Study of the Effects of Shower Generation Method, Zero Suppression, and Event Generation Method", DØ Internal Note #1354 (1992).

[48] R. J. Genik II, "Calorimeter Level 1.5 Electromagnetic Physics Commissioning", DØ Internal Note #2411 (1993).

[49] M. Abolins, et. al, "Design Report: The DØ Experiment at the Fermilab Antiproton–Proton Collider", Fermilab Publication, 1984.

[50] M. Abolins, D. Edmunds, P. Laurens, J. Linnemann, and B. Pi, "A High Luminosity Trigger Design for the Tevatron Collider Experiment in DØ," IEEE Trans. on Nucl. Sci.,**36**, pp. 384-389.

[51] M. Abolins, D. Edmunds, P. Laurens, and B. Pi, "The Fast Trigger for the DØ Experiment," Nucl. Instrum. and Meth.,**A289**, pp. 543-560, 1990.

[52] J. Linnemann, "The DØ Software Trigger," in *Proc. of the Int. Conf. on Computing in High Energy Physics*, Annency, France, Sept. 1992, pp. 199-201.

[53] J. Linnemann, et. al, "The DØ Level 1.5 Calorimeter Trigger," in *Proc. of the Int. Conf. on Computing in High Energy Physics*, San Francisco, CA, May 1994, pp. 88-90.

[54] D. Edmunds, S. Gross, and P. Laurens, for the DØ Collaboration, "A DSP Based Calorimeter Trigger for the DØ Experiment", *Nuclear Science Symposium 1994 IEEE Conference Record NSS26-01, Vol 2*, pp. 819-823

[55] S. Abachi, et. al., "Search for the Top Quark in $\overline{p}p$ collisions at $\sqrt{s} = 1.8$ TeV. " Phys.Rev.Lett., **72**, pp. 2138-2142, 1994.

[56] Texas Instruments Inc., Dallas TX.

[57] Texas Instruments Inc., *TMS320C4x User's Guide*, Texas Instruments Inc., 1993.

[58] Ariel Corp., Highland Park, NJ.

[59] Ariel Corp., *Preliminary User's Manual for the Hydra-II*, Ariel Corp., 1994.

[60] J. Drinkard, "The D0 Level 1.5 Calorimeter Trigger", FERMILAB-CONF-94-320-E, Presented at 1994 Meeting of the American Physical Society, Division of Particles and Fields (DPF 94), Albuquerque, NM, 2-6 Aug 1994.

[61] F. Paige and S. Protopopescu, *BNL Report,* **BNL38034** (unpublished) (1986) release v7.13.