INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

University of Alberta

Precision Observables for Particle Physics Experiments

by

Ernest Jankowski 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Physics

Edmonton, Alberta
Fall 2005

Abstract

The objective of this thesis is to develop tools for interpretation of the upcoming particle physics experiments.

We implement and test Optimal Jet Finder (OJF), a jet finding algorithm that is based on the global energy flow in the event. OJF is infrared and collinear safe and resolves overlapping jets dynamically. The shapes of jets are determined dynamically and are not geometrical cones. However, they are more regular than those resulting from $k_\perp$, which should facilitate detector calibration of OJF. We compare the statistical uncertainties of the W-boson mass when using three different jet finding algorithms: $k_\perp$, JADE, and OJF. We find that OJF gives the same accuracy as $k_\perp$ but is faster than $k_\perp$ if a large number of calorimeter cells is analyzed. We present the details of FORTRAN 77 and object-oriented C++ implementations of OJF.

We calculate the rate of the lepton flavour violating $\mu \to e + \gamma$ decay in a particular Grand Unification SO(10) model by Albright and Barr. We assume the Constrained Minimal Supersymmetric Standard Model framework. We interpret the results in view of the recent cosmological observations from Wilkinson Microwave Anisotropy Probe. We find that the SO(10) model is consistent with the experimental limits on the $\mu \to e + \gamma$ branching ratio over a large volume of the supersymmetric parameter space. However, if the branching ratio is further constrained by the MEG experiment, carried out in the Paul Scherrer Institute, below 10^{-13}, the available volume of the parameter space will be significantly reduced.

We calculate the QED suppression of the rate of the lepton flavour violating $\mu \to e + \gamma$ decay. The result does not depend on the details of the mechanism that is responsible for the lepton flavour violation, except for the mass scale that enters the final expression. If this mass scale is between 100 and 1000 GeV, the numerical value of the decrease in the decay rate is between 12% and 17%. If the rare muon decay is observed in the MEG experiment, our result will enhance the precision with which the parameters of the new physics models responsible for this decay can be extracted.

Acknowledgements

I thank my supervisor, Andrzej Czarnecki, for his support throughout my entire degree. I am very grateful to Fyodor Tkachov for his invaluable contributions to a considerable part of my research. I thank for his support and guidance. I am very grateful to Bruce Campbell for the inspiration, encouragement, advice, and support. I am very thankful for his expertise and outstanding teaching. I am obliged to Manuella Vincter, my former committee member, for her helpful assistance and support. Special thanks go to my colleague David Maybury for the opportunity to collaborate with him on the $\mu \to e\gamma$ SO(10) project. I thank James Pinfold for several helpful discussions, his interest in jet finding algorithms, as well as his assistance during my stay at CERN in the summer of 2001. I am grateful to my committee, Subir Bhattacharjee, Andrzej Czarnecki, Douglas Gingrich, Frank Marsiglio, and Krzysztof Jan Sliwa, for the useful comments on my thesis. I am very thankful to the graduate student advisors, Lynn Chandler and Sarah Derr, for their enormous help during the five years of my degree.

I specially thank my wife, Magdalena, for her encouragement and support. I am grateful for the happiness she has brought into my life. I am immensely grateful to my parents, Zofia Jankowska and Jerzy Jankowski, my grandparents, Leokadia Hardel, Tadeusz Hardel, Kazimiera Jankowska, and Edward Jankowski, for all their care and the inspiration I have drawn from them.

I acknowledge with gratitude the financial support from Alberta Ingenuity (the Alberta Heritage Foundation for Science and Engineering Research).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis deals with two separate topics in particle physics phenomenology: reconstruction of hadronic jets (chapters 2-4 and appendices A, B) and the lepton flavour violating $\mu \to e\gamma$ decay (chapters 5, 6, and appendix C).

In this chapter, we sketch the relevant background and outline the organization of this thesis.

1.1 Standard Model

The Standard Model, formulated in the 1970s and established by more than three decades of experimental research, encapsulates our current knowledge about elementary particles and fundamental interactions. It successfully explains an enormous variety of data coming from particle physics experiments that reach energies up to 1 TeV, or equivalently, lengths down to 10^{-18} m.

The Standard Model comprises twelve types (flavours) of elementary fermions with spin 1/2. They fall into two groups: quarks and leptons. The quarks are named: up (u), down (d), charmed (c), strange (s), top (t), and bottom (b); and the leptons are electron (e), electron neutrino (ν_e), muon (μ), muon neutrino (ν_μ), tauon (τ) and tauon neutrino (ν_τ). Table 1.1 organizes the elementary fermions in three rows and four columns. The particles in each column have the same electric charge. Each row is called a generation. The first generation is the most familiar as the particles u, d, and e constitute the matter we are made of (and the neutrino ν_e appears, for example, in the β-decays of unstable elements). The two other generations are heavier versions of the first. Apart from neutrinos, the particles in the second and third generation are unstable as they can decay into lighter particles.

There are four types of interactions between the elementary particles: strong, weak, electromagnetic, and gravitational. The interactions are mediated by gauge bosons, elementary particles with integer spin. Table 1.2 summarizes the interactions and associated gauge bosons. Gravity is not included in the Standard Model. At the energy scales probed by experiments (current and in any conceivable future), gravity is significantly weaker than the other interactions, and quantum effects are negligible.

Table 1.1: Elementary fermions (spin = 1/2). Electric charge Q for each column is displayed in units of the absolute value of the charge of the electron; and the approximate mass in units of the electron's mass is given in brackets.

$Q = +\frac{2}{3}$	$Q = -\frac{1}{3}$	$Q = 0$	$Q = -1$
u (~ 5)	d (~ 12)	ν_e (?)	e (1.0)
c ($\sim 2.4 \times 10^3$)	s ($\sim 2.1 \times 10^2$)	ν_μ (?)	μ (2.1×10^2)
t (3.5×10^5)	b ($\sim 8.3 \times 10^3$)	ν_τ (?)	τ (3.5×10^3)

Table 1.2: Elementary interactions and the associated gauge bosons. Mass of the gauge bosons is given in brackets in the units of the mass of the electron.

INTERACTION	GAUGE BOSONS	SPIN
strong	8 gluons (0)	1
weak	$W^\pm$ (1.6×10^5), Z (1.8×10^5)	1
electromagnetic	photon (0)	1
gravitation	graviton (0)	2

The Standard Model implies the existence of a particle with spin zero, named the Higgs boson. (One such particle is sufficient in the Standard Model, but more are possible or even necessary if extensions of the Standard Model such as supersymmetry are considered.) At higher energies weak and electromagnetic interactions can be described as a more fundamental electroweak interaction mediated by massless gauge bosons W^1, W^2, W^3, and B^0. The Higgs boson is responsible for breaking the electroweak interaction into the weak and electromagnetic sectors. It provides masses for $W^\pm$ and Z bosons, quarks, and the charged leptons. The confirmation of the existence of the Higgs boson or bosons is one of the outstanding challenges of current experimental research.

1.2 Strong interaction, jets, and jet finding algorithms

Quarks and gluons are somewhat peculiar: they do not appear as free particles as, for example, electrons or muons do. Quarks and gluons are the only elementary particles of the Standard Model that experience the strong interaction, which is responsible for confinement of quarks and gluons inside hadrons.

The theory of strong interaction, called Quantum Chromodynamics (QCD), is in many respects similar to Quantum Electrodynamics (QED), which describes interactions between electrically charged particles; but there are important differences. Each quark carries a strong charge called color. There are six types of this charge named red, green, blue, anti-red, anti-green, and anti-blue. The color is analogous to electric charge, except for that there are only two types of

the latter: positive and negative. The gluons that mediate the strong interaction carry color charges themselves and interact between one another. In QED in contrast, photons are neutral. This difference between QED and QCD accounts for the peculiar properties of quarks and gluons. The QCD potential energy between two quarks can be approximated by the expression [1]

$$V(r) = -\frac{4}{3}\frac{\alpha_s}{r} + kr \tag{1.1}$$

where α_s is the strong coupling constant analogous to the fine coupling constant of QED, r is the separation between the quarks, and k is a positive constant of order 1 GeV/fm. Apart from the numerical factor in front, the first term in (1.1) is the same as the QED potential energy between two elementary charges. It is the second term that makes the difference. It is there because gluons, unlike photons, interact between each other. When r is sufficiently small (large momentum transfer between interacting particles), the second term in (1.1) can be neglected. However, for large r, the second term becomes dominant. The limit of $V(r)$ is infinite when r takes arbitrary large values. This means we cannot separate the quarks.

Quarks and gluons are confined inside mesons and hadrons, the composite particles that carry zero net color charge and can avoid the strong interaction at large distances. (A meson is composed of a pair of quark and anti-quark, which carry opposite colors, red and anti-red, green and anti-green, or blue and anti-blue. A baryon is composed of three quarks, each of them carrying a different color; the combination red-green-blue is neutral.) If we try to separate quarks, for example, by giving them initial momenta in the center of mass reference frame, the initial kinetic energy is converted to potential energy of the field according to (1.1). When this energy becomes larger than the rest energy of a quark anti-quark pair, such a pair emerges out of vacuum. The pairs continue to appear until the system rearranges into colorless combinations of hadrons and mesons.

The QCD coupling constant, α_S, determining the strength of the interaction, decreases with the magnitude of the momentum transfer between the interacting particles. (In contrast, the fine coupling constant of QED increases with momentum transfer.) For high momentum transfer (hard scattering processes), we can speak of almost free quarks and gluons. This property of the strong interaction is called asymptotic freedom. It allows us to apply the standard perturbation theory methods to calculate probabilities, cross-sections, and decay rates for processes involving quarks or gluons. The confinement, however, makes experimental studies of such processes more challenging than studying similar processes involving only leptons. In experiments, we observe sprays of hadrons, called hadronic jets, in place of the quarks and gluons. Consider an experiment in which an electron and a positron collide and annihilate. If a muon and anti-muon pair is created, we observe two muons moving in opposite directions (in the center of mass reference frame). If a quark and anti-quark pair is produced, the experimental situation is more involved. In such case, we observe two sprays of hadrons moving in opposite directions.

Some properties of jets, such as direction or energy, can be computed in perturbation theory

using quarks and gluons as final states[1]. Therefore, it is possible to connect, to some degree of accuracy, the properties of quarks and gluons used in computations with the properties of jets observed in experiments. For example, the energy of a jet (understood as the sum of the energies of all particles that belong to that jet) corresponds approximately to the energy of the quark or gluon that originates the jet. Similarly, the direction of the jet (taken as the average direction of the particles that constitute that jet) corresponds to the direction of the quark or gluon state.

Other properties of jets, such as the number of hadrons in a jet, width of a jet, energy distribution in a jet, et cetera, cannot be addressed by the perturbation theory and one has to resort to heuristic models that translate the quarks and gluons into jets. The process in which gluons and quarks evolve into jets is called fragmentation and hadronization. Realistic models of fragmentation and hadronization are quite elaborate and contain many parameters that are tuned according to the data; see [2], for example. Such models are implemented into Monte Carlo generators such as Pythia [3] or Herwig [4].

Processes involving hadronic jets are usually analyzed with jet finding algorithms. A jet finding algorithm determines how to divide the particles in the event (or the particles generated with a Monte Carlo event generator) into a small number of jets (some particles can be left unassigned to any jet). Often, a jet finding algorithm also determines how many jets there are in the event. Reconstructing jets is not always straightforward. There may be many jets in the detector at the same time. In such case, it is very probable that some jets will overlap. It is no longer obvious how to deal with particles in the overlapping region. Even if the jets do not overlap, a detector may contain some extra particles not coming from the process of interest. In such case, reconstructing too wide jets will overestimate the energy of the jets; whereas leaving the jets too narrow will underestimate the jets' energy.

Jet finding algorithms rely on the following property of hadronic jets: the particles that belong to a jet have limited transverse momentum with respect to the axis of the jet or (which is not unrelated) have similar directions. Chapter 2 discusses the commonly used algorithms for jet reconstruction.

Jet finding algorithms are important tools of data analysis in high energy physics experiments. Many processes of interest (and unavoidable background processes) involve production of hadronic jets in the final states. Jet finding algorithms have been often pointed out to contribute significantly to experimental uncertainties in studies involving jets. Many yet-undiscovered particles that experiments are looking for may decay into quarks and gluons. The accuracy of a jet finding algorithm translates into the precision with which the properties of such particles can be studied. A discovery of a new particle may depend on the accuracy of data analysis methods used, including jet finding algorithms.

The first part of this thesis (chapters 2-4 and appendices A and B) gives an account of the studies on Optimal Jet Definition [5], a novel algorithm for identification of hadronic jets.

We introduce hadronic jets and the conventional jet finding algorithms in chapter 2. There,

[1] in terms of probabilities as it is quantum mechanics

we discuss some of the problems of the established jet definitions. Chapter 3 explains Optimal Jet Definition.

One of our research objectives was to construct and run a clear, unambiguous, large scale benchmark test of Optimal Jet Definition. We have compared the algorithm with two other, JADE [6] and Durham [7], in the context of the W-boson mass extraction from fully hadronic decays of pairs of W-bosons at 180 GeV center of mass energy. The details of this test and the results are presented in chapter 4.

Another research objective was to prepare reliable software implementations of Optimal Jet Definition that can be used in high energy physics data analysis. Appendices A and B discuss FORTRAN 77 and object-oriented C++ implementations of the algorithm. Those appendices are intended as a documentation for the users of the programs.

Chapters 2 and 3 serve as an introduction to the subject of jet finding algorithms. Chapter 4 and appendices A and B are based on this thesis research and contain material published in [8, 9, 10].

1.3 Beyond the Standard Model

One evidence of physics beyond the Standard Model comes from neutrino oscillation experiments. Numerous experiments (see [11] for a review) report that neutrinos produced in the Sun, Earth's atmosphere, or nuclear reactors seem to change flavour as they travel to detectors.

This can be explained naturally if neutrinos have masses. Neutrinos are produced and detected as definite flavour states, ν_e, ν_μ, and ν_τ, that is, always in association with a charged lepton, electron, or muon, or tauon (Feynman diagrams in figure 1.1). If neutrinos have masses,

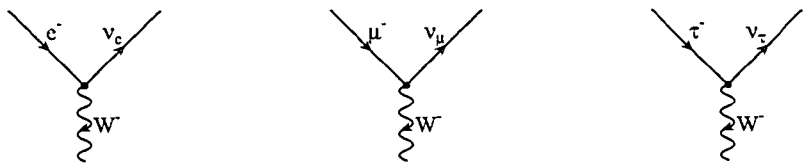

Figure 1.1: Standard Model charged current weak interaction vertices defining neutrino flavour.

it is possible that each flavour eigenstate is a superposition of different mass eigenstates. The wave functions of distinct mass eigenstates have different frequencies, and a superposition that starts as, for example, a definite flavour electron neutrino, after traveling some distance, is no longer equivalent to the electron neutrino but to some mixture of all neutrino flavours. (See chapter 5 for more details.) The important point the oscillation experiments imply is that the neutrino masses are very small in comparison with the masses of all other Standard Model fermions.

The Standard Model assumes massless neutrinos. The term in the Standard Model Lagrangian that describes the mass of the electron, for example, connects the left-handed, e_L, and the right-handed, e_R, helicity states (see the beginning of chapter 5 for more detailed discussion

of the following),

$$y_e e_{\mathrm{L}}^{\dagger} e_{\mathrm{R}} \langle H \rangle + \mathrm{h.c.}, \tag{1.2}$$

where y_e is the (dimensionless) electron's Yukawa coupling, and $\langle H \rangle$ is the Higgs field vacuum expectation value. Similar terms can be written for μ, τ, and all quarks. The neutrinos in the Standard Model have only left-handed helicity states and no such mass terms are possible.

It is possible to extend the Standard Model particle content by adding three extra ("right-handed") neutrino states, one for each Standard Model neutrino flavour. The mass terms analogous to (1.2) can be added to the Lagrangian. The smallness of the neutrino masses can be arranged if the extra neutrino states are neutral with respect to the electroweak and strong interactions (that is, are the Standard Model gauge group singlets). In such case, the extra neutrinos can have Majorana type mass terms in the Lagrangian. For example, for one of the three extra neutrino states, ν_{R}, such a Majorana mass term can be written as

$$\frac{1}{2} M_{\nu} \nu_{\mathrm{R}}^{\mathrm{T}} \sigma^2 \nu_{\mathrm{R}} + \mathrm{h.c.}, \tag{1.3}$$

where M_{ν} is a mass-dimension coefficient possibly of order of 10^{10} to 10^{16} GeV, and σ^2 is the Pauli σ-matrix. (No such mass terms are possible for the Standard Model particles as they would violate the gauge invariance of the theory.) As explained in chapter 5, with the addition of the heavy Majorana mass terms, such as (1.3), the neutrino mass eigenstates split into three light and three heavy states. This way of explaining small neutrino masses by introduction of heavy singlet neutrino states is known as the see-saw mechanism [12]. The heavy singlet neutrinos play an essential role in inducing lepton flavour violation in supersymmetric extensions of the Standard Model, such as the one considered in chapter 5.

Another important indication of physics beyond the Standard Model comes from cosmological observations, which suggest that the total amount of matter in the Universe significantly (by several times) supersedes the amount of the matter that we know from the Standard Model. The most accurate determination of the density of non-baryonic dark matter comes from the measurements of the anisotropies of the cosmic microwave background temperature performed by the COBE[2] [13] and WMAP[3] [14] satellites.

Perhaps the best candidate for the dark matter is supersymmetry. (Other possibilities are discussed, for example, in [15]. Here, we confine ourselves to supersymmetry as the assumption that particles required by supersymmetry constitute the dark matter is central to the research reported in chapter 5.) Supersymmetry is an extension of the Poincare group of spacetime symmetries that links fermions (particles with half-integer spin number) to bosons (particles with integer spin number). Extending the Standard Model to a supersymmetric theory involves introducing a large number of new particles. (The possibility that some particles of the Standard Model are superpartners of others has been excluded.) For example, the electron of the Standard Model is required to have two superpartners with spin 0 named left-selectron, $\tilde{e}_{\mathrm{L}}$, and right-

[2] COsmic Background Explorer
[3] Wilkinson Microwave Anisotropy Probe

Table 1.3: Particles of the Minimal Supersymmetric Standard Model

PARTICLES	SPIN	SUPERPARTNERS	SPIN
e, μ, τ	$s = \frac{1}{2}$	$\tilde{e}_{\rm L}$, $\tilde{e}_{\rm R}$, $\tilde{\mu}_{\rm L}$, $\tilde{\mu}_{\rm R}$, $\tilde{\tau}_{\rm L}$, $\tilde{\tau}_{\rm R}$	$s = 0$
ν_e, ν_μ, ν_τ		$\tilde{\nu}_{e\rm L}$, $\tilde{\nu}_{\mu\rm L}$, $\tilde{\nu}_{\tau\rm L}$	
u, c, t		$\tilde{u}_{\rm L}$, $\tilde{u}_{\rm R}$, $\tilde{c}_{\rm L}$, $\tilde{c}_{\rm R}$, $\tilde{t}_{\rm L}$, $\tilde{t}_{\rm R}$	
d, s, b		$\tilde{d}_{\rm L}$, $\tilde{d}_{\rm R}$, $\tilde{s}_{\rm L}$, $\tilde{s}_{\rm R}$, $\tilde{b}_{\rm L}$, $\tilde{b}_{\rm R}$	
B^0, W^k, $k = 1,2,3$	$s = 1$	$\tilde{B}^0$, $\tilde{W}^k$, $k = 1,2,3$	$s = \frac{1}{2}$
g^a, $a = 1,\ldots,8$		$\tilde{g}^a$, $a = 1,\ldots,8$	
$H_{\rm u}^+$, $H_{\rm u}^0$, $H_{\rm d}^0$, $H_{\rm d}^-$	$s = 0$	$\tilde{H}_{\rm u}^+$, $\tilde{H}_{\rm u}^0$, $\tilde{H}_{\rm d}^0$, $\tilde{H}_{\rm d}^-$	$s = \frac{1}{2}$

selectron $\tilde{e}_{\rm R}$. (We need two scalar fields in order to have the same number of degrees of freedom.) Similar superpartners with spin 0 have to be introduced for other fermions of the Standard Model, see table 1.3. The Standard Model strong and electroweak gauge bosons with spin 1 will have fermionic superpartners, gauginos, with spin 1/2. The Higgs fields with spin 0 will have superpartners with spin 1/2. The Standard Model requires two scalar Higgs fields, H^+ and H^0, aligned into one electroweak doublet (although more are possible). After electroweak symmetry breaking, the doublet gives rise to one physical Higgs particle. A supersymmetric extension of the Standard Model requires at least four scalar Higgs fields (labeled as $H_{\rm u}^+$, $H_{\rm u}^0$, $H_{\rm d}^0$, and $H_{\rm d}^-$) and their fermionic superpartners with spin number 1/2. A single Higgs doublet cannot give masses to both up-type (u, c, t) and down-type quarks (d, s, b) in a theory that is supersymmetric. Another reason for extending the Higgs sector to two electroweak doublets is the cancellation of chiral anomalies. The renormalizability of the Standard Model requires cancellation of certain Feynman diagrams, which takes place because of an interplay of quantum numbers of all fermions existent in the theory. (For example, the net charge of all fermions of the standard model is zero, $3 \times 2/3 + 3 \times (-1/3) + (-1) = 0$, for each generation. Adding a single positively charged fermion would spoil this condition.)

None of the superpartners has been observed yet. Nature cannot be strictly supersymmetric. The superpartners — if they exist — are heavier than the Standard Model particles. To construct a realistic supersymmetric extension of the Standard Model, we have to introduce supersymmetry breaking terms in the Standard Model Lagrangian, which give the extra masses to the superpartners. Such a theory, supersymmetric only in the high energy limit, still possesses many phenomenologically attractive features if the supersymmetry breaking terms are of order of 1 TeV. The supersymmetric extension of the Standard Model with the minimal number of particles added and containing the explicit supersymmetry breaking is called the Minimal Supersymmetric Standard Model (MSSM).

The supersymmetry breaking brings over a 100 new parameters to the theory. None of them has been measured yet. Some are, however, constrained by data. Non-observation of superpartners sets certain limits on the lower value of the extra masses. The mixing of flavours that supersymmetry introduces cannot be too large in order not to violate the current experimental

bounds, for example, on the $\mu \rightarrow e\gamma$ decay rate (to be discussed later). The theory with so many free parameters has very limited predictive power. A way around it (suggested by the strong limits on lepton flavour violation placed by experiments) is the idea that at some high energy scale the supersymmetry breaking parameters are universal. That means that, at the high energy scale the extra "breaking" masses of all scalars are identical and equal to some parameter m_0. Similarly, all the gaugino masses are identical and equal to a parameter $m_{1/2}$. The number of parameters is reduced to a few. Those relevant for the project discussed in chapter 5 are: m_0, a mass parameter that determines the extra masses of all sleptons and squarks; $m_{1/2}$, a mass parameter that determines the extra masses of all gauginos (the superpartners of standard model gauge bosons); $\tan\beta$ the ratio of the vacuum expectation value of H_u^0 to the vacuum expectation value of H_d^0; the fourth discreet parameter is the sign of the μ term in the supersymmetric potential (defined by the equation C.1). The absolute value of the μ parameter is determined (see appendix C), but the sign can be both positive or negative.

With this assumption of universal supersymmetry breaking, the results of calculations involving supersymmetry can be parametrized by the three[4] numbers m_0, $m_{1/2}$, $\tan\beta$, and the sign of μ. Such a theory has been named the Constrained Minimal Supersymmetric Standard Model (CMSSM).

If we assume that the superpartners constitute the dark matter, from the requirement that the proper amount of this dark matter was produced in the early Universe, further constraints can be placed [16] on the values of the parameters m_0, $m_{1/2}$, $\tan\beta$ (the allowed regions are indicated in figure 5.3).

From a theoretical standpoint, there are many questions that the Standard Model does not address. Why are there three different interactions and three separate coupling constants? Why is the charge of the up-type quarks $+2/3\,|e|$, and the charge of down-type quarks $-1/3\,|e|$? (Another rephrasing of this question is: why does the proton have exactly opposite charge to the electron?) Why do the chiral anomalies in the Standard Model cancel? (For example, why is the net charge of each generation of fermions zero?) How are the masses of different particles related?

Some of these questions are, at least partially, addressed by Grand Unification Theories that arise from the desire to unify the strong and electroweak interactions. They are based on the idea that there is a larger gauge symmetry group, for example, SU(5) or SO(10), that contains the Standard Model SU(3)×SU(2)×U(1) as a subgroup. At some very high energy (roughly 10^{16} GeV) the larger gauge symmetry of the theory breaks spontaneously leaving the Standard Model gauge group (similarly to how the electroweak symmetry of the Standard Model breaks to the weak interaction and electromagnetism). The SO(10) grand unification gauge group is especially promising; a 16-dimensional spinor representation of SO(10) has exactly the right number of degrees of freedom to accommodate all the fermions of a single generation of the Standard Model particles plus a heavy singlet neutrino necessary for the see-saw mechanism. The fact that each

[4] We have skipped the discussion of the so called trilinear couplings, which would require another universal parameter.

Table 1.4: The lepton flavour numbers. Antiparticles have opposite lepton flavour numbers. All other Standard Model particles have all lepton flavour numbers equal to 0.

	l_e	l_μ	l_τ
e, ν_e	1	0	0
μ, ν_μ	0	1	0
τ, ν_τ	0	0	1

generation is a single representation of SO(10) explains why the electric charge of the up-type quarks is $+2/3\,|e|$, and the electric charge of the down-type quarks is $-1/3\,|e|$ (or why the electric charge of the proton is opposite to that of the electron). Because the SO(10) gauge group does not contain any Abelian U(1) component, the charges are quantized. (In the Standard Model, the charges for U(1) group are introduced by hand according to data.) The anomaly cancellation is automatic, and certain relations between the masses of the particles exist.

1.4 Decay $\mu \rightarrow e\gamma$

The muon, μ, is an unstable particle and decays according to

$$\mu \rightarrow e\,\bar{\nu}_e\,\nu_\mu \tag{1.4}$$

(Feynman diagram in figure 1.2). Apart from radiative modifications (for example, an extra pho-

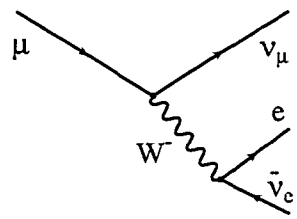

Figure 1.2: $\mu \rightarrow e\,\bar{\nu}_e\,\nu_\mu$ in the Standard Model.

ton or electron-and-anti-electron pair), this is the only decay channel observed in experiments and the only one allowed by the Standard Model. All Standard Model processes conserve separately the electron flavour number l_e, the muon flavour number l_μ, and the tauon flavour number l_τ. Table 1.4 shows the assignment of the lepton flavour numbers to the Standard Model leptons. For antiparticles, the lepton flavour numbers are opposite. All other Standard Model particles carry the lepton flavour numbers equal to zero. As can be easily seen, the process 1.4 conserves all three lepton flavour numbers.

If the neutrinos have masses, and the mass eigenstates are not identical with the weak interaction eigenstates, lepton flavour violating processes, such as $\mu \rightarrow e\gamma$, are possible. Figure

1.3 shows an example Feynman diagram contributing to the $\mu \rightarrow e\gamma$ decay rate with massive neutrinos. However, because of the smallness of the neutrino masses (by comparison to the mass of the W-boson) the decay rate is far beyond the reach of experiments.

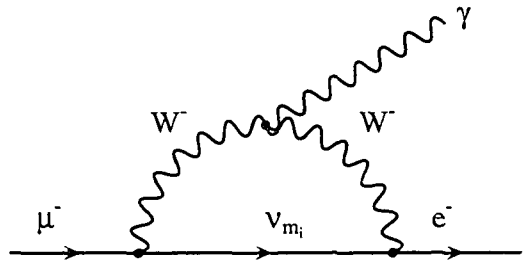

Figure 1.3: Feynman diagram contributing to $\mu \rightarrow e\gamma$ with massive neutrinos, ν_{m_i}.

Supersymmetric extensions of the Standard Model give, in general, non-zero rates for lepton flavour violating processes, in particular, $\mu \rightarrow e\gamma$. In fact, the historical reason for introducing the universality of supersymmetry breaking was to avoid the lepton flavour violation that significantly exceeds the experimental bounds. With the heavy singlet neutrinos, even with the universality of supersymmetry breaking masses at the Grand Unification Theory scale, lepton flavour violation is still induced through radiative corrections. The exact value of the decay rate is model specific.

In chapter 5, we present the results of the branching ratio of the $\mu \rightarrow e + \gamma$ decay in a framework of the supersymmetric Grand Unification Theory proposed by Albright and Barr [17]. It has been suggested [18] that this Grand Unification model, very successful otherwise, predicts the rate of $\mu \rightarrow e + \gamma$ that exceeds the current experimental limits, and therefore, cannot be valid. We have evaluated the rate of $\mu \rightarrow e + \gamma$ decay, interpreted the results in view of recent cosmological observations from Wilkinson Microwave Anisotropy Probe [19, 20], and concluded that the model is consistent with the current experimental limits on the branching ratio of $\mu \rightarrow e + \gamma$. Chapter 5 constitutes a longer version of the published paper [21].

In chapter 6, we present the calculation of a quantum electrodynamics radiative correction to the branching ratio of $\mu \rightarrow e + \gamma$. We report that the branching ratio $\mu \rightarrow e + \gamma$ is suppressed by several percent. (The exact value depends on the mass scale of the physics involved in the decay, see chapter 6 for details.) If the $\mu \rightarrow e + \gamma$ decay is observed in the experiment run at Paul Scherrer Institute [22], our results will improve the accuracy with which the experimental data can be interpreted. Chapter 6 constitutes the publication [23].

Credits

Optimal Jet Definition and its first Oberon Pascal implementation were invented by F. Tkachov [5]. The FORTRAN 77 version of Optimal Jet Finder [9] was prepared in collaboration with D. Grigoriev and F. Tkachov. The C++ version of Optimal Jet Finder [10] was developed in collaboration with S. Chumakov and F. Tkachov. The W-boson mass benchmark test of Optimal Jet Definition [8] was done together with D. Grigoriev and F. Tkachov. The Albright-Barr SO(10) $\mu \to e + \gamma$ project [21] was completed together with my colleague graduate student David Maybury. The author is very grateful to Bruce Campbell for many helpful discussions. The QED suppression to the $\mu \to e + \gamma$ branching ratio [23] was computed together with A. Czarnecki.

Bibliography

[1] D. H. Perkins. Introduction to high-energy physics. Cambridge University Press (2000).

[2] Roger J. Barlow. Jets in high-energy interactions. *Rept. Prog. Phys.*, 56:1067–1144, 1993.

[3] Torbjorn Sjostrand et al. High-energy-physics event generation with PYTHIA 6.1. *Comput. Phys. Commun.*, 135:238–259, 2001.

[4] G. Corcella et al. Herwig 6.5 release note. 2002.

[5] Fyodor V. Tkachov. The definition of jets. *Int. J. Mod. Phys.*, A17:2783–2884, 2002.

[6] W. Bartel et al. Experimental studies on multi-jet production in e+ e- annihilation at PETRA energies. *Z. Phys.*, C33:23, 1986.

[7] S. Catani, Yuri L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber. New clustering algorithm for multi-jet cross-sections in e+ e- annihilation. *Phys. Lett.*, B269:432–438, 1991.

[8] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Towards a standard jet definition. *Phys. Rev. Lett.*, 91:061801, 2003.

[9] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder. *Comput. Phys. Commun.*, 155:42–64, 2003.

[10] S. Chumakov, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder (C++ v1.0). Submitted for publication in Comput. Phys. Commun.

[11] M. C. Gonzalez-Garcia and Yosef Nir. Developments in neutrino physics. *Rev. Mod. Phys.*, 75:345–402, 2003.

[12] Pierre Ramond. *Journeys Beyond the Standard Model.* Reading, Mass., Perseus Books, 1999.

[13] http://lambda.gsfc.nasa.gov/product/cobe/.

[14] http://map.gsfc.nasa.gov/.

[15] S. Eidelman et al. Review of particle physics. *Phys. Lett.*, B592:1, 2004.

[16] John R. Ellis, Keith A. Olive, Yudi Santoso, and Vassilis C. Spanos. Supersymmetric dark matter in light of WMAP. *Phys. Lett.*, B565:176–182, 2003.

[17] Carl H. Albright and Stephen M. Barr. Construction of a minimal higgs SO(10) SUSY GUT model. *Phys. Rev.*, D62:093008, 2000.

[18] Xiao-Jun Bi and Yuan-Ben Dai. Lepton flavor violation and its constraints on the neutrino mass models. *Phys. Rev.*, D66:076006, 2002.

[19] C. L. Bennett et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Preliminary maps and basic results. *Astrophys. J. Suppl.*, 148:1, 2003.

[20] D. N. Spergel et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Determination of cosmological parameters. *Astrophys. J. Suppl.*, 148:175, 2003.

[21] Ernest Jankowski and David W. Maybury. Lepton flavour violation in a class of lopsided SO(10) models. *Phys. Rev.*, D70:035004, 2004.

[22] T. Mori. Status and future of mu $\rightarrow$ e gamma: The PSI experiment. *Nucl. Phys. Proc. Suppl.*, 111:194–199, 2002.

[23] Andrzej Czarnecki and Ernest Jankowski. Electromagnetic suppression of the decay mu $\rightarrow$ e gamma. *Phys. Rev.*, D65:113004, 2002.

Chapter 2

Jet Finding Algorithms

Jet finding algorithms are essential tools in high energy physics data analysis. Many standard model or new physics processes studied or expected to be studied with particle colliders involve production of hadronic jets in the final state. The accuracy of jet identification has a strong impact on the precision with which the standard model or new physics parameters can be determined.

In this chapter, we briefly review conventional jet finding algorithms and discuss their shortcomings.

2.1 Hadronic Jets

Jets of hadrons that appear in the final states of scattering experiments in high energy physics correspond, to the first approximation, to quarks and gluons produced in the collisions. Quarks and gluons, interacting strongly, are not observed as free particles. Only colorless combinations of quarks and gluons, hadrons, can escape the strong interaction at large distances and only those combinations appear in experiments. If the energy of the colliding particles is sufficiently high, the quarks and gluons produced in the collision manifest themselves as sprays of hadrons, called *hadronic jets*. The particles within each spray move roughly in the same direction, which is interpreted as the direction of the underlying quark or gluon.

Hadronic jets were first observed by the MARK I Collaboration [1] in 1975. The physics of the strong interaction and hadronic jets is described in many places; a broad review can be found, for instance, in [2] and [3].

Let us consider an example high energy physics event, to which we will refer in chapter 4. Many such events were observed at LEP [4]. An electron and positron collide with the center of mass energy equal to 180 GeV. The electron and positron annihilate and a pair of W-bosons is produced. Each of the W-bosons decays into two quarks. When the quarks move away from each other, potential energy of the strong interaction between them increases quickly with the

distance[1], and new pairs of quarks and anti-quarks emerge. The many quarks and anti-quarks combine into colorless hadrons, which form 4 or more[2] jets. We are interested, for instance, in extracting the W-boson mass from a collection of such events. It would be much easier if we were able to observe directly the quarks coming from the decaying W-bosons. But we observe jets of hadrons instead, and when we make the analysis, we have to deal with jets. The task of analyzing hadronic events is performed by means of jet finding algorithms.[3]

2.2 Jet algorithms

The analysis of events with many hadrons is often performed with the use of jet finding algorithms. A *jet finding algorithm* is a procedure to reconstruct jets and their properties from the observed hadrons. A jet algorithm is often referred to as a *jet definition* because it defines what is precisely meant by a 'jet'.

There are two logically distinct steps in any analysis involving jets: (1) dividing the observed particles into jets (along with deciding how many jets there are if appropriate) and (2) a prescription how to derive the properties of the jets, such as their four-momenta, from the properties of the particles. The second step is called a *recombination scheme.*

A simple and logical recombination scheme, but not necessarily the only possible (see section 2.2.2 and [15] for more detailed discussion), is that the four-momentum q_{jet} of the jet, is the sum of the four-momenta p_a of all particles that belong to that jet:

$$q_{\text{jet}} = \sum_{a \in \text{ jet}} p_a. \tag{2.1}$$

There have been many jet definitions developed by various collaborations over the years; see [2, 3, 15, 16] for a review. Conventional jet definitions fall into two main categories: cone algorithms [17] and successive recombination algorithms [18]. Cone algorithms define a jet as all particles with trajectories within a circle of a certain radius R in $\eta \times \phi$ (pseudorapidity and azimuthal angle) space. Individual variants differ by how the center of the circle is found and by the recombination scheme. Successive recombination algorithms merge two (or more) particles at each step. Variants differ by the order of merging, by the condition on which the merging is terminated, and the recombination scheme. Below, we review the most frequently used jet definitions.

[1] $\sim$1 GeV per fm.

[2] More jets can result from gluon bremsstrahlung.

[3] Jet variables, such as sphericity [5], thrust [6, 7], the Fox-Wolfram moments [8], spherocity [9, 10, 11], tripodity [12], acoplanarity [13], et cetera, can be applied directly to the particles in the event, but they would be of no use in the mentioned example of extracting the W-boson mass. The Jet Energy Flow Project [14] suggests a more accurate way of analyzing hadronic events without using jet finding algorithms, but this approach is very computationally intensive and has no practical realizations at present.

2.2.1 Preliminary definitions

The input of a jet finding algorithm is a list of n_{parts} particles[4], each characterized by a four-momentum

$$p_a = \left(E_a, p_a^{(\text{x})}, p_a^{(\text{y})}, p_a^{(\text{z})}\right), \qquad a = 1, 2, \ldots, n_{\text{parts}}. \tag{2.2}$$

The index a labels the particles. The output of a jet finding algorithm is a collection of n_{jets} jets with four-momenta

$$q_j = \left(E_j, q_j^{(\text{x})}, q_j^{(\text{y})}, q_j^{(\text{z})}\right), \qquad j = 1, 2, \ldots, n_{\text{jets}}. \tag{2.3}$$

The index j labels the jets. We will also use generic label 'jet' in place of j whenever this is more convenient. Usually, the particles are described by the energy and direction

$$E_a,\ \theta_a,\ \phi_a, \tag{2.4}$$

where θ_a is the standard polar angle from the z-axis, and ϕ_a is the standard azimuthal angle in the xy-plane. The z-axis is chosen along the beam direction. The massless[5] four-momentum $\left(p_a^2 = 0\right)$ can be computed from the energy and direction according to

$$p_a = E_a \left(1, \sin\theta_a \cos\phi_a, \sin\theta_a \sin\phi_a, \cos\theta_a\right), \tag{2.5}$$

In hadron collisions, we do not know the center of mass of colliding partons, and we should use variables that are insensitive to the lack of this information.
Rapidity y is defined as (index a can apply in the equations below if needed)

$$y = \frac{1}{2} \ln \frac{E + p^{(\text{z})}}{E - p^{(\text{z})}}. \tag{2.6}$$

Under a boost with velocity u in the z-direction, rapidity changes as

$$y' = y - \operatorname{arctanh} u, \tag{2.7}$$

and the difference in rapidities is invariant, $\Delta y = \Delta y'$. Usually, we do not have sufficient information from measurements to calculate rapidity, but we use *pseudorapidity* instead

$$\eta = -\ln \tan\left(\frac{\theta}{2}\right). \tag{2.8}$$

For massless and approximately for relativistic particles,

$$y = \eta. \tag{2.9}$$

[4]For brevity, we use the term 'particles' for the input of a jet finding algorithm, however, depending on circumstances, 'particles' can mean calorimeter cells, towers, preclusters, particle tracks from a tracking detector, particles from a Monte Carlo generator, partons from theoretical computations, et cetera.

[5]Particles are relativistic, and measurements are not complete in general to construct the full four-momentum.

The *transverse momentum* $p^{\perp}$ is defined as

$$p^{\perp} = \sqrt{\left(p^{(x)}\right)^2 + \left(p^{(y)}\right)^2}, \tag{2.10}$$

and the *transverse energy* $E^{\perp}$ as

$$E^{\perp} = E \sin\theta \tag{2.11}$$

(for massless and approximately for relativistic particles, $E^{\perp} = p^{\perp}$).

2.2.2 Cone algorithms

Cone algorithms [17] define a jet as all particles with trajectories within a circle of a certain radius R in $\eta \times \phi$ space. The center of the circle is usually found as a result of an iterative procedure that requires the geometrical center of the circle and the center of the jet (defined in a moment) to match. Distance in $\eta \times \phi$ space between the particle η_a, ϕ_a and the center η_{center}, ϕ_{center} of the circle is defined as

$$d_a = \sqrt{\left(\eta_a - \eta_{\text{center}}\right)^2 + \left(\phi_a - \phi_{\text{center}}\right)^2}. \tag{2.12}$$

The particle a belongs to a jet if it is contained within the circle;

$$a \in \text{jet} \quad \Leftrightarrow \quad d_a \leq R; \tag{2.13}$$

the radius R is a chosen parameter (the standard value of R is 0.7 or 0.4).

The center η_{center}, ϕ_{center} is found iteratively: (1) Some initial position $\eta_{\text{center(0)}}$, $\phi_{\text{center(0)}}$ for the center of the circle is chosen; (2) From all particles that belong to the circle (according to condition 2.13), the following jet direction is computed using transverse-energy weighting,

$$\eta_{\text{jet}} = \frac{\sum_{a \in \text{jet}} E_a^{\perp} \eta_a}{\sum_{a \in \text{jet}} E_a^{\perp}}, \tag{2.14}$$

$$\phi_{\text{jet}} = \frac{\sum_{a \in \text{jet}} E_a^{\perp} \phi_a}{\sum_{a \in \text{jet}} E_a^{\perp}}; \tag{2.15}$$

(3) The center of the circle is moved to the jet direction η_{jet}, ϕ_{jet}. As the center of the circle changes its position, in general, the content of the jet changes, according to (2.13). The new jet direction is computed, and the center of the circle is moved there, et cetera. That is, the points (2) and (3) are repeated until a stable circle is found: the geometrical center of the circle and

the center of the jet coincide,

$$\eta_{\text{jet}} = \eta_{\text{center}}, \tag{2.16}$$

$$\phi_{\text{jet}} = \phi_{\text{center}}. \tag{2.17}$$

Other conditions to define stable circles have been used, such as the requirement that the direction of the net 3-momentum of all particles within the circle coincide with the geometrical center of the circle, or that the energy inside the circle is maximized. In the simplest versions of the cone algorithm, such as the one used in ATLFAST package [19], no iterative procedure is performed, and the direction of jet is given by (2.14) and (2.15) for the initial position of the center of the circle.

The iterative procedure described above requires an initial position, $\eta_{\text{center(0)}}$, $\phi_{\text{center(0)}}$. Conventionally, directions of the most energetic particles are chosen for the starting positions. Those energetic particles are called *seeds*. Particles are considered to be seeds when their energy is above some threshold (set up individually for each process). As we discuss in section 2.3.1, the use of seeds brings many problems, both from the theoretical and experimental side.

A variety of recombination schemes has been applied by various experiment collaborations. In recent years, the simplest choice (2.1) has been recommended by the CDF and D0 collaborations [15]. We list several frequently used schemes (q_{jet} in the formulas below is given by equation 2.1).

- Original Snowmass recombination scheme [20]:

$$E_{\text{jet}}^{\perp} = \sum_{a \in \text{jet}} E_a^{\perp}, \tag{2.18}$$

$$\eta_{\text{jet}} = \frac{\sum_{a \in \text{jet}} E_a^{\perp} \eta_a}{\sum_{a \in \text{jet}} E_a^{\perp}}, \tag{2.19}$$

$$\phi_{\text{jet}} = \frac{\sum_{a \in \text{jet}} E_a^{\perp} \phi_a}{\sum_{a \in \text{jet}} E_a^{\perp}}. \tag{2.20}$$

- Modified Run I[6] D0 recombination scheme [15]:

$$\theta_{\text{jet}} = \arctan \frac{\sqrt{\left(q_{\text{jet}}^{(x)}\right)^2 + \left(q_{\text{jet}}^{(y)}\right)^2}}{q_{\text{jet}}^{(z)}}, \tag{2.21}$$

[6] "Run I" and "Run II" refer to the running periods of the Tevatron collider at Fermilab, respectively, before and after the upgrade that took place between 1995 and 2001.

$$\eta_{\rm jet} = -\ln\tan\frac{\theta_{\rm jet}}{2}, \tag{2.22}$$

$$\phi_{\rm jet} = \arctan\frac{q^{(\rm y)}_{\rm jet}}{q^{(\rm x)}_{\rm jet}}, \tag{2.23}$$

$$E^{\perp}_{\rm jet} = \sum_{a\in{\rm jet}} E^{\perp}_{a}, \tag{2.24}$$

- Modified Run I CDF recombination scheme [15]:

$$\theta_{\rm jet} = \arctan\frac{\sqrt{\left(q^{(\rm x)}_{\rm jet}\right)^2+\left(q^{(\rm y)}_{\rm jet}\right)^2}}{q^{(\rm z)}_{\rm jet}}, \tag{2.25}$$

$$\eta_{\rm jet} = \frac{1}{E^{\perp}_{\rm jet}}\sum_{a\in{\rm jet}} E^{\perp}_{a}\eta_a, \tag{2.26}$$

$$\phi_{\rm jet} = \frac{1}{E^{\perp}_{\rm jet}}\sum_{a\in{\rm jet}} E^{\perp}_{a}\phi_a. \tag{2.27}$$

- Four-vector scheme [15]:

$$q^{\perp}_{\rm jet} = \sqrt{\left(q^{(\rm x)}_{\rm jet}\right)^2+\left(q^{(\rm y)}_{\rm jet}\right)^2}, \tag{2.28}$$

$$y_{\rm jet} = \frac{1}{2}\ln\frac{E_{\rm jet}+q^{(\rm z)}_{\rm jet}}{E_{\rm jet}-q^{(\rm z)}_{\rm jet}}, \tag{2.29}$$

$$\phi_{\rm jet} = \arctan\frac{q^{(\rm y)}_{\rm jet}}{q^{(\rm x)}_{\rm jet}}. \tag{2.30}$$

2.2.3 Successive recombination algorithms in e^+e^- collisions

Successive recombination algorithms [18] used in e^+e^- collisions, in the simplest variant, work as follows. The algorithm starts with a list of initial particles; the particles are labeled, as before, with the index $a = 1, 2, \ldots, n_{\rm parts}$. (1) The distance d_{ab} between any two particles is computed according to some definition. The examples of the distance definitions are given in table 2.1. The various successive recombination algorithms differ by the choice of the distance definition. (2) The pair with the smallest distance d_{ab} is found and merged into one *pseudo*-particle with the four-momentum given (for example) by $p_{ab} = p_a + p_b$. In this way, the number of (pseudo-)particles is reduced by one. The procedure is repeated until (i) the required number of pseudo-particles is left (if we know in advance how many jets we desire) or (ii) until $d_{ab} > d_{\rm cut}$ for all a, b, where

Table 2.1: Distance measure for different binary joining algorithms.

ALGORITHM	DISTANCE DEFINITION (up to a constant factor)
JADE [21]	$d_{ab} = E_a E_b (1 - \cos\theta_{ab})$
Durham ($k_\perp$) [22]	$d_{ab} = \min\left(E_a^2, E_b^2\right)(1 - \cos\theta_{ab})$
Luclus [18]	$d_{ab} = \frac{\lvert\mathbf{p}_a\rvert^2 \lvert\mathbf{p}_b\rvert^2 (1-\cos\theta_{ab})}{(\mathbf{p}_a+\mathbf{p}_b)^2}$
Geneva [23]	$d_{ab} = \frac{E_a E_b (1-\cos\theta_{ab})}{(E_a+E_b)^2}$
Cambrigde [24]	same as Durham

d_{cut} is a chosen parameter. The remaining pseudo-particles are the final jets. The described scheme corresponds to so called binary algorithms, which merge only two particles at a time ($2 \to 1$). Other variants correspond to $3 \to 2$ (for example, Arclus [25]) or more generally to $m \to n$ merging scheme (for example, [26]).

2.2.4 $k_\perp$ algorithms for hadron-hadron collisions

The successive recombination algorithms for hadron-hadron collisions are more complicated than those used in e^+e^- collisions. We review two variants, one suggested by Catani, Dokshitzer, Seymour and Webber [27], and second proposed in [28] by Ellis and Soper and adopted, after changes, by D0 and CDF collaborations as a standard Run II $k_\perp$ algorithm [15].

Run II $k_\perp$ algorithm

The algorithm starts with a list of particles labeled with index $a = 1, 2, \ldots, n_{\text{parts}}$. Below, we use the term particles also for the objects generated from merging other particles.[7] The algorithm works as follows (we present the Run II version)

1. For all particles calculate d_a, and for all pairs of particles calculate d_{ab} according to

$$d_a = \left(p_a^\perp\right)^2, \tag{2.31}$$

$$d_{ab} = \min\left(\left(p_a^\perp\right)^2, \left(p_b^\perp\right)^2\right) \frac{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}{R^2}. \tag{2.32}$$

(2.33)

[7] Those objects are sometimes called pseudo-particles or proto-jets, but we prefer not to clutter the terminology.

2. Find the minimum $d_{\min}$ of all d_a and d_{ab}.

3. (a) If $d_{\min}$ is one of d_{ab}, remove the corresponding particles with indices a and b, and add a new particle with four-momentum p_{ab}

$$p_{ab} = p_a + p_b. \tag{2.34}$$

   (b) If $d_{\min}$ is one of d_a, remove the corresponding particle with index a, and put it on the list of *final jets.*

4. If any particle remains go to step 1.

Variant of $k_\perp$ proposed by Catani, Dokshitzer, Seymour, Webber

1. For all particles calculate d_a, and for all pairs of particles calculate d_{ab} according to

$$d_a = \left(p_a^\perp\right)^2, \tag{2.35}$$

$$d_{ab} = \min\left(\left(p_a^\perp\right)^2, \left(p_b^\perp\right)^2\right)\left((\eta_a - \eta_b)^2 + (\phi_a - \phi_b)^2\right). \tag{2.36}$$

$$\tag{2.37}$$

2. Find the minimum $d_{\min}$ of all d_a and d_{ab}.

3. (a) If $d_{\min}$ is one of d_{ab}, remove the corresponding particles with the indices a and b, and add a new particle with four-momentum p_{ab} ([22] considers also other recombination schemes)

$$p_{ab} = p_a + p_b \tag{2.38}$$

   (b) If $d_{\min}$ is one of d_a, remove the corresponding particle with index a, and put it on the list of *beam jets.*

4. Repeat (1) - (3) for all particles that are not the beam jets until

$$d_a \geq d_{\text{cut}}, \quad \text{and} \quad d_{ab} \geq d_{\text{cut}} \tag{2.39}$$

   or a specified number of particles remains.

5. The remaining particles are the *hard final state jets.*

Unlike the Run II $k_\perp$ algorithm, the algorithm described in this paragraph returns two separate sets of jets: *beam jets* and *hard final state jets.*

2.3 Problems with conventional algorithms

The conventional jet finding algorithms, reviewed in this chapter, have numerous shortcomings that are avoided in the jet definition presented in the next chapter.

2.3.1 Seeds

The use of seeds in cone algorithms poses many important problems and renders the comparison of experimental data and theoretical results more difficult and less accurate. The final configuration of jets often depends on whether a jet algorithm is applied to calorimeter cells, particles form Monte Carlo simulations, or partons from perturbative calculations. When we apply a cone algorithm involving seeds to partons beyond the leading order in perturbative QCD calculations, soft radiation or collinear splitting of partons may change the jet configuration significantly, whereas the corresponding experimental jet configuration remains unchanged. The final jet configuration is also unstable with respect to small fluctuations in detectors, such as splitting of energy between several calorimeter cells. Below, we provide three examples of seed related problems that could render completely different experimental and theoretical jet configurations (figures 2.1 - 2.3).

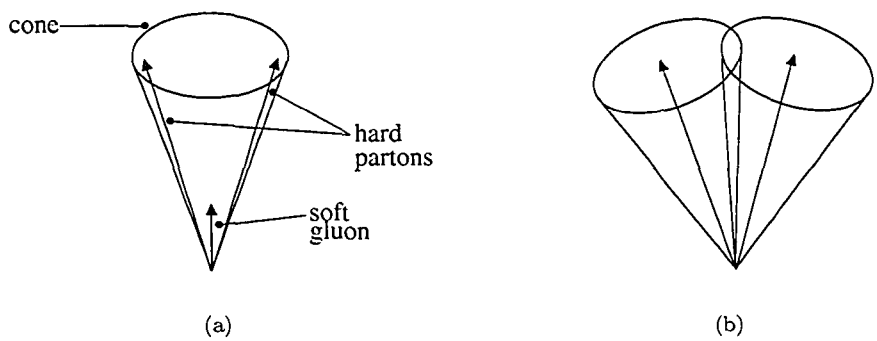

Figure 2.1: Soft gluon radiation at the next to leading order in perturbative QCD. (a) A soft gluon serves as a seed, and the two hard partons are reconstructed as one jet. (b) Without the soft gluon, the two partons are reconstructed to two separate jets if they are separated by a distance between R and $2R$.

- At the next to leading order in perturbative QCD calculations, a soft gluon can serve as a seed, and the two hard partons in figure 2.1(a) are reconstructed as one jet. Without the soft gluon, the two partons are reconstructed to two separate jets if they are separated by a distance between R and $2R$; figure 2.1(b). (Of course, both parton configurations should correspond to the same experimental situation.) This is an example how a small perturbation can result in a large change in the final jet configuration. (For example, if the number of jets is used to classify events in figure 2.1, (a) and (b) would belong to separate

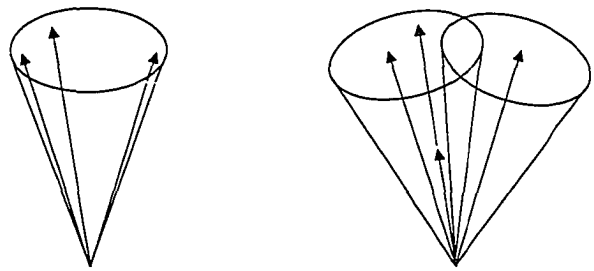

Figure 2.2: Collinear splitting at the next to next to leading order in perturbative QCD. (a) The event is reconstructed to one jet using the most energetic, middle parton as a seed. (b) The middle parton is split into two collinear partons; the parton on the left is now the most energetic, and the first jet is reconstructed there, forcing the right parton to be in a separate jet.

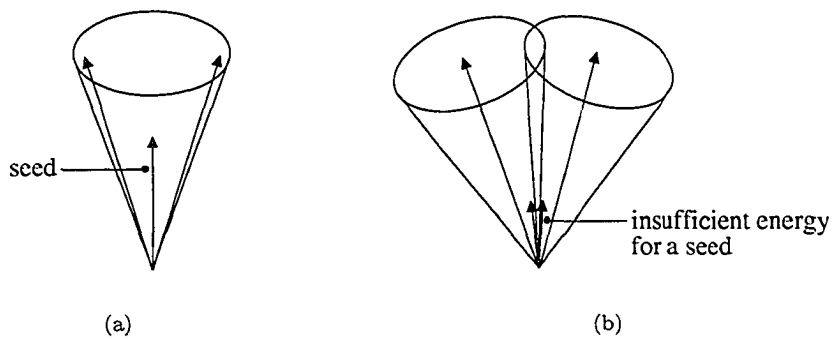

Figure 2.3: Segmentation of energy in a calorimeter. (a) A high energy particle hits a single calorimeter cell; the cell collects a sufficient amount of energy to be considered as a seed, and a single jet is reconstructed. (b) The high energy particle hits the boundary of calorimeter cells and deposits energy in two cells. None of the cells has separately enough energy to be considered as a seed, and two jets are reconstructed.

classes, which is not a small difference any more.) We would like to have a jet finding algorithm in which a small perturbation corresponds to similarly small change in the final jet configuration. This is indeed the case with the Optimal Jet Finder, described in the next chapter.

- In figure 2.2(a), at the next to next to leading order perturbative QCD calculations, the event is reconstructed to one jet using the most energetic, middle parton as a seed. If the middle parton splits into two collinear parts, figure 2.2(b) (the situation which is experimentally indistinguishable from the previous), the parton on the left is now the most

energetic, and the first jet is reconstructed there, forcing the right parton to be in a separate jet.

- In figure 2.3(a), a high energy particle hits a single calorimeter cell. The cell collects a sufficient amount of energy to be considered as a seed, and a single jet is reconstructed. In figure 2.3(b), a high energy particle hits the boundary of calorimeter cells and deposits energy in several cells (two in the figure). None of the cells has separately enough energy to be considered as a seed. Two jets are reconstructed.

Since the described problems are not inherent to the cone algorithms directly but to the use of seeds, two solutions were proposed to overcome the problems: Seedless cone algorithm [15] and Improved Legacy Cone Algorithm [29]. A seedless cone algorithm looks for all stable circles, that is, it starts "everywhere" (for example at every calorimeter cell). This approach is very computationally demanding, which is its major disadvantage. However, it is not impossible with the use of special optimization tricks [15]. The Improved Legacy Cone Algorithm uses seeds, but in addition to conventional seeds, it also takes pairs, triplets, and n-tuples of the conventional seeds and considers them as new seeds as well. That is, if p_a, p_b, p_c, ... are the conventional seeds, also

$$p_{ab} = p_a + p_b \qquad a \neq b, \tag{2.40}$$

$$p_{abc} = p_a + p_b + p_c \qquad a \neq b, a \neq c, b \neq c, \tag{2.41}$$

$$p_{abcd} = p_a + p_b + p_c + p_d \qquad \text{all } a, b, c, d \text{ are distinct}, \tag{2.42}$$

$$\text{et cetera}$$

are also considered as seeds. (Indices a, b, c, d run over all conventional seeds.) The use of extra seeds mostly eliminates the problems discussed in this section.

2.3.2 Non-uniqueness of final jet configuration

The final position of the center of the circle, obtained as the result of the iterative procedure described in section 2.2.2, depends on the initial starting point. This is a disturbing fact, as the cone algorithm does not have any inherent way of saying which of the few final configurations is best.

2.3.3 Overlapping jets

Another problem with cone algorithms is the fact that some of the identified jets may overlap. In this case, an artificial and arbitrary prescription has to be used to decide whether to split or merge the two jets. One example of such prescription is: If the fraction f of overlapping energy (with respect to the smaller energy jet) exceeds some threshold (for example, $f = 0.5$ or $f = 0.75$), the two jets should be merged; otherwise, the two jets should be split, and the

particles in the overlapping region assigned to the closest jet. If there are more than two jets overlapping, the final result may depend on the order of merging. Therefore, some standard (and again arbitrary) convention for the order has to be specified. The important problem with merging and splitting procedure is that it is very difficult, if not impossible, to account for it in theoretical parton level calculations.

2.3.4 Calibration

A very strong advantage of the cone algorithms is the relative simplicity of the jet energy calibration. Because the cone jets are reconstructed from a fixed geometrical shape (circle), it is relatively easy to account for the extra energy from underlying event and detector noise. (This is the main reason why the cone algorithm is standard in hadron collisions, whereas binary algorithms are standard in e^+e^- collisions where there is no underlying event.)

The $k_\perp$ algorithms are free of the seed related problems and overlapping circles discussed above, but significantly more difficult to calibrate in hadron collisions. The $k_\perp$ algorithms compare the distance between only two particles and merge only two particles at a time.[8] Instead of the global energy flow in the event, only limited information is taken into account at each single step of that procedure. Jets resulting from such a procedure have often very irregular shapes, which are not motivated by the underlying physics. From the experimental point of view, jets with very irregular shapes are difficult to calibrate.

2.3.5 Speed

For the existing implementations of the $k_\perp$ algorithm [30, 31], the running time scales as n_{parts}^3, where n_{parts} is the number of particles in the input. It has been pointed out in [15] that because of this n_{parts}^3 dependence, it is impossible to apply the algorithm directly to the analysis of all detector cells (of order of a few thousands for D0 and CDF, and ~100 000 for ATLAS [32]). A step called preclustering has to be introduced to reduce the number of input data to ~100 pieces before the $k_\perp$ can be applied. An important question emerges here: how the preclustering step affects measurements. It is impossible, or very difficult at least, to model the preclustering in theoretical calculations. Because Optimal Jet Finder depends only linearly on the number n_{parts} of particles, it becomes an ideal algorithm for analysis of enormous sets of data, such as analyzing all cells of the ATLAS detector.

It is important to mention here that, even though the running time of the existing implementations [30, 31] scales as n_{parts}^3, it is possible to implement $k_\perp$ algorithm so that its running time dependence is $\sim n_{\text{parts}}^2 \log n_{\text{parts}}$. The $\sim n_{\text{parts}}^2 \log n_{\text{parts}}$ implementation [33] becomes important when extrapolations to large n_{parts} values are discussed.

[8] Diclus (Arclus) algorithm [25] for e^+e^- collisions is based on $3 \to 2$ merging scheme, but that does not change the point we try to make here.

Bibliography

[1] G. Hanson et al. Evidence for jet structure in hadron production by e+ e- annihilation. *Phys. Rev. Lett.*, 35:1609–1612, 1975.

[2] Roger J. Barlow. Jets in high-energy interactions. *Rept. Prog. Phys.*, 56:1067–1144, 1993.

[3] R. K. Ellis, W. James Stirling, and B. R. Webber. *QCD and Collider Physics.* Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol. Vol. 8, 1996.

[4] Salvatore Mele. Physics of W bosons at LEP. *Phys. Rept.*, 403-404:255–270, 2004.

[5] J. D. Bjorken and Stanley J. Brodsky. Statistical model for electron - positron annihilation into hadrons. *Phys. Rev.*, D1:1416–1420, 1970.

[6] S. Brandt, C. Peyrou, R. Sosnowski, and A. Wroblewski. The principal axis of jets. An attempt to analyze high- energy collisions as two-body processes. *Phys. Lett.*, 12:57–61, 1964.

[7] Edward Farhi. A QCD test for jets. *Phys. Rev. Lett.*, 39:1587–1588, 1977.

[8] Geoffrey C. Fox and Stephen Wolfram. Event shapes in e+ e- annihilation. *Nucl. Phys.*, B149:413, 1979.

[9] Howard Georgi and Marie Machacek. A simple QCD prediction of jet structure in e+ e- annihilation. *Phys. Rev. Lett.*, 39:1237, 1977.

[10] G. Parisi. Super inclusive cross-sections. *Phys. Lett.*, B74:65, 1978.

[11] John F. Donoghue, F. E. Low, and So-Young Pi. Tensor analysis of hadronic jets in quantum chromodynamics. *Phys. Rev.*, D20:2759, 1979.

[12] O. Nachtmann and A. Reiter. A new quantity for finding four - jet events in electron - positron annihilation into hadrons. *Zeit. Phys.*, C14:47, 1982.

[13] A. De Rujula, John R. Ellis, E. G. Floratos, and M. K. Gaillard. QCD predictions for hadronic final states in e+ e- annihilation. *Nucl. Phys.*, B138:387, 1978.

[14] C. F. Berger et al. Snowmass 2001: Jet energy flow project. *eConf*, C010630:P512, 2001.

[15] Gerald C. Blazey et al. Run II jet physics. 2000.

[16] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Towards a standard jet definition. *Phys. Rev. Lett.*, 91:061801, 2003.

[17] George Sterman and Steven Weinberg. Jets from quantum chromodynamics. *Phys. Rev. Lett.*, 39:1436, 1977.

[18] Torbjorn Sjostrand. The Lund monte carlo for e+ e- jet physics. *Comput. Phys. Commun.*, 28:229, 1983.

[19] Elzbieta Richter-Was, Daniel Froidevaux, and Poggioli Luc. Atlfast 2.0 a fast simulation package for ATLAS. ATLAS Internal Note ATL-PHYS-98-131.

[20] John E. Huth et al. Toward a standardization of jet definitions. Presented at Summer Study on High Energy Physics, Reaearch Directions for the Decade, Snowmass, CO, Jun 25 - Jul 13, 1990.

[21] W. Bartel et al. Experimental studies on multi-jet production in e+ e- annihilation at PETRA energies. *Z. Phys.*, C33:23, 1986.

[22] S. Catani, Yuri L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber. New clustering algorithm for multi-jet cross-sections in e+ e- annihilation. *Phys. Lett.*, B269:432–438, 1991.

[23] S. Bethke, Z. Kunszt, D. E. Soper, and W. James Stirling. New jet cluster algorithms: Next-to-leading order QCD and hadronization corrections. *Nucl. Phys.*, B370:310–334, 1992.

[24] Yuri L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber. Better jet clustering algorithms. *JHEP*, 08:001, 1997.

[25] L. Lonnblad. Arclus: A new jet clustering algorithm inspired by the color dipole model. *Z. Phys.*, C58:471–478, 1993.

[26] S. Youssef. Clustering with local equivalence relations. *Comput. Phys. Commun.*, 45:423–426, 1987.

[27] S. Catani, Yuri L. Dokshitzer, M. H. Seymour, and B. R. Webber. Longitudinally invariant k(t) clustering algorithms for hadron hadron collisions. *Nucl. Phys.*, B406:187–224, 1993.

[28] Stephen D. Ellis and Davison E. Soper. Successive combination jet algorithm for hadron collisions. *Phys. Rev.*, D48:3160–3166, 1993.

[29] S. Catani et al. The QCD and standard model working group: Summary report. 2000.

[30] http://hepwww.rl.ac.uk/theory/seymour/ktclus/.

[31] J. M. Butterworth, J. P. Couchman, B. E. Cox, and B. M. Waugh. KtJet: A C++ implementation of the k(t) clustering algorithm. *Comput. Phys. Commun.*, 153:85–96, 2003.

[32] W. W. Armstrong et al. ATLAS: Technical proposal for a general-purpose p p experiment at the Large Hadron Collider at CERN. CERN-LHCC-94-43.

[33] Ernest Jankowski. $O\left(n^2\right)$ implemetation of kt algorithm. In preparation.

Chapter 3

Optimal Jet Definition

In this chapter, we describe Optimal Jet Definition [1], a jet finding algorithm that avoids many of the shortcomings of the conventional schemes reviewed in the previous chapter. In Optimal Jet Definition, the shapes of jets are determined dynamically from the global structure of the energy flow of the event. This is in contrast with cone algorithms, where jets are restricted to fixed cones. In Optimal Jet Definition, jet overlaps are handled automatically without the necessity of any arbitrary prescriptions. Final jet configuration is independent of whether input particles are split into collinear groups (collinear safety). Optimal Jet Definition is also infrared safe: Any soft particle radiation results in a small change in the structure of jets. Thus, Optimal Jet Definition avoids the serious problems of cone algorithms employing seeds. Optimal Jet Definition, as opposed to successive recombination algorithms, takes into account the global structure of the energy flow in the event rather then merging a single pair of particles at a time. The resultant jets have more regular shapes than $k_\perp$ jets. This makes the Optimal Jet Definition better suited for calibration for hadron-hadron collisions.

With one exception, we present Optimal Jet Definition in an implementation independent manner, leaving the practical software details for appendices A and B. The only implementation detail that we introduce in this chapter is parameter n_{tries}.

First, we summarize the motivations behind the Optimal Jet Definition. Then, we describe Optimal Jet Definition for e^+e^- collisions. In the end, we explain the differences for hadron-hadron collisions. We continue to use the notation introduced in the previous chapter.

3.1 Motivations

The theory of Optimal Jet Definition was developed in [1, 2, 3]. Below, we summarize the principal points of the theory.[1]

(1) Calorimetric measurements with hadronic final states $\mathbb{P}$ must rely on observables $f(\mathbb{P})$ that

[1] A version of this section has been included in [4]. © 2003 Elsevier.

possess a special "calorimetric", or C-continuity [3]. An observable f is C-continuous only if (for each event) it is continuous in the parameters (energies and angles) of the particles constituting the event and is insensitive to splitting of the particles into collinear fragments. Fyodor Tkachov pointed out C-continuous analogues for a variety of observables usually studied via intermediacy of jet algorithms [3]. The fundamental role of such observables is highlighted by two facts: (i) An observable inspired by [3] played an important role in the selection of top quark events in the fully hadronic channel at D0 [5, 6]. (ii) The Jet Energy Flow project [7] provides numerical evidence that C-continuous observables may indeed help to go beyond the intrinsic limitations of conventional procedure based on jet algorithms in the quest for the 1% precision level in the physics of jets.

(2) The proposition that the observed event $\mathbb{P}$ inherits information (as measured by calorimetric detectors) from the underlying quark-and-gluon event $\mathfrak{q}$ is expressed as

$$f(\mathfrak{q}) \approx f(\mathbb{P}) \quad \text{for any } C\text{–continuous } f. \tag{3.1}$$

(3) For each parameter M on which the probability distribution $\pi_M(\mathbb{P})$ of the observed events $\mathbb{P}$ may depend, there exists an optimal observable $f_{\text{opt}}(\mathbb{P}) = \partial_M \ln \pi_M(\mathbb{P})$ for the best possible measurement of M [8]. This is a reinterpretation of the Rao-Cramer inequality and the maximal likelihood method of mathematical statistics in terms of the method of moments. In the context of multi-hadron final states as "seen" by calorimetric detectors, such an observable is automatically C-continuous.

(4) If the dynamics of hadronization is such that (3.1) holds, then good approximations for f_{opt} could exist among functions that depend only on $\mathbb{Q}$ which is a parameterization of $\mathbb{P}$ in terms of a few pseudo-particles (jets), found from a condition modeled after (3.1):

$$f(\mathbb{Q}) \approx f(\mathbb{P}) \quad \text{for any } C\text{–continuous } f. \tag{3.2}$$

This simply translates the meaning of jet finding as an inversion of hadronization into the language of C-continuous observables.

(5) C-continuous observables can be approximated by sums of products of simplest such observables that are linear in particles' energies E_a:

$$f(\mathbb{P}) = \textstyle\sum_a E_a f(\hat{\mathbf{p}}_a), \tag{3.3}$$

where a runs over all particles in the event and $\hat{\mathbf{p}}_a$ (a unit vector) denotes the direction of the 3-momentum of the a-th particle; f is any continuous function of a direction only. (The relevant theorems can be found in [3] and [1].)

(6) Thus, it is sufficient to explore criterion 3.2 with only f's of the form in (3.3). Then one can perform a Taylor expansion in angular variables and obtain a factorized bound of the form

$$|f(\mathbb{P}) - f(\mathbb{Q})| \leq C_{f,R} \times \Omega_R[\mathbb{P}, \mathbb{Q}], \tag{3.4}$$

where all the dependence on f is localized within $C_{f,R}$ (so the bound remains valid for any C-continuous f) and $\Omega_R[\mathbb{P},\mathbb{Q}]$ is a function of the jet configuration $\mathbb{Q}$ (and the event $\mathbb{P}$) only.

(7) Since the collection of values of all f on a given event $\mathbb{P}$ is naturally interpreted as the event's physical information content, bound 3.4 means that the distortion of such content in the transition from $\mathbb{P}$ to $\mathbb{Q}$ can be controlled by a single function; so the loss of physical information in the transition is minimized if $\mathbb{Q}$ corresponds to the global minimum of $\Omega_R[\mathbb{P},\mathbb{Q}]$. (The minimization of the information loss motivates the name "Optimal".) The Optimal Jet Definition amounts to finding the jet configuration $\mathbb{Q}$ which minimizes $\Omega_R[\mathbb{P},\mathbb{Q}]$, depending on specific application, either with a given number of jets or with a minimum number of jets while satisfying the restriction $\Omega_R[\mathbb{P},\mathbb{Q}] < \omega_{\text{cut}}$ with some parameter $\omega_{\text{cut}} > 0$ which is similar to the jet resolution d_{cut} of recombination algorithms. The parameter ω_{cut} can be interpreted as an upper bound on the information loss.

3.2 Optimal Jet Definition for e^+e^- collisions

Optimal Jet Definition works as follows. It starts with a list of particles (hadrons, calorimeter cells, partons) and ends with a list of jets. To find the final jet configuration, we define Ω, some function of a jet configuration. The momenta of the input particles enter Ω as parameters. The final, optimal jet configuration is found as the configuration that minimizes Ω.

A jet configuration is described by the so-called recombination matrix $\{z_{aj}\}$, where a=1, 2, ..., n_{part} indexes the input particles with four-momenta p_a, and j=1, 2, ..., n_{jets} indexes the jets. z_{aj} is interpreted as the fraction of the a-th particle that belongs to the j-th jet. The conventional schemes correspond to restricting z_{aj} to either one or zero depending on whether or not the a-th particle belongs to the j-th jet. Here, we require only that

$$0 \leq z_{aj} \leq 1, \tag{3.5}$$

and

$$\sum_j z_{aj} \leq 1. \tag{3.6}$$

Any value $\{z_{aj}\}$ of the recombination matrix satisfying conditions 3.5 and 3.6 describes a jet configuration.

The four-momentum of the j-th jet is given by the recombination scheme:

$$q_j = \sum_a z_{aj} p_a. \tag{3.7}$$

The light-like $(\tilde{q}_j^2 = 0)$ four-direction of the j-th jet is defined as

$$\tilde{q}_j = (1, \hat{\mathbf{q}}_j), \tag{3.8}$$

where $\hat{\mathbf{q}}_j = \mathbf{q}_j / |\mathbf{q}_j|$ is a unit direction vector obtained from $q_j = (E_j, \mathbf{q}_j)$. The explicit form of Ω is

$$\Omega(\{z_{aj}\}) = \frac{2}{R^2} \sum_j q_j \tilde{q}_j + \sum_a \left(1 - \sum_j z_{aj}\right) E_a, \tag{3.9}$$

where R is a positive parameter.

If the number of jets is already known, we find $\{z_{aj}\}$ that minimizes $\Omega(\{z_{aj}\})$ given in the above equation. This value of $\{z_{aj}\}$ describes the final, desired configuration of jets.

The minimization problem is non-trivial because of the large dimension of the domain in which to search the global minimum, $n_{\text{part}} \times n_{\text{jets}} = O(100{-}1000)$ of continuous variables z_{aj}. However, it is possible to solve it due to the known analytical structure of Ω and the regular structure of the domain of z_{aj}. An implementation, called Optimal Jet Finder [4], is described in appendices A and B. We need to mention here that the program starts with some initial value of $\{z_{aj}\}$, which in the simplest case can be entirely random, and descends iteratively into a local minimum of Ω. In order to find the global minimum, random initial values of z_{aj} are generated several times, parameter n_{tries}, and the deepest minimum is chosen out of the local minima obtained at each attempt.

If the number of jets should be determined in the process of jet finding, we repeat the procedure described above for the number of jets equal to 1,2,3,... The final jet configuration is the one with the smallest number of jets for which the minimum of Ω is sufficiently small, that is,

$$\Omega < \omega_{\text{cut}}, \tag{3.10}$$

where ω_{cut} is a positive parameter chosen by the user. ω_{cut} has a similar meaning to the d_{cut} parameter in successive recombination algorithms.

3.3 Interpretation of Ω

The first term in (3.9), called *fuzziness*, 'measures' the dynamical width of the jets. This is more easily seen when $q_j \tilde{q}_j$ is rewritten as

$$q_j \tilde{q}_j = 2 \sum_{a=1}^{n_{\text{parts}}} z_{aj} E_a \sin^2 \frac{\theta_{aj}}{2}, \tag{3.11}$$

or for small θ_{aj},

$$q_j \tilde{q}_j \simeq \frac{1}{2} \sum_{a=1}^{n_{\text{parts}}} z_{aj} E_a \theta_{aj}^2, \tag{3.12}$$

where θ_{aj} is the angle between the a-th particle and j-th jet, that is, the angle between the vectors $\mathbf{p}_a$ and $\hat{\mathbf{q}}_j$.

The second term in (3.9), called *soft energy*, is the fraction of the energy of the event that does not take part in any jet formation.

The positive parameter R has a similar meaning as the radius parameter in cone algorithms in the sense that a smaller value of R results in narrower jets and more energy left outside jets. Large values of R ($\gtrsim 2$) force the energy left outside jets to zero, and, effectively, only the first term in (3.9) is minimized.

3.4 Hadron-hadron collisions

The mechanism of Optimal Jet Definition for hadron-hadron collisions is the same as described above for e^+e^- processes, but the definition of the Ω function is different. The light-like ($\tilde{q}_j^2 = 0$) four-direction of the j-th jet is defined as

$$\tilde{q}_j = (\cosh\eta_j, \cos\varphi_j, \sin\varphi_j, \sinh\eta_j)\,, \tag{3.13}$$

where

$$\eta_j = \frac{\sum_{a=1}^{n_{\text{parts}}} z_{aj} E_a^{\perp} \eta_a}{\sum_{a=1}^{n_{\text{parts}}} z_{aj} E_a^{\perp}}, \tag{3.14}$$

$$\cos\phi_j = \frac{q_j^{(\mathrm{x})}}{\sqrt{\left(q_j^{(\mathrm{x})}\right)^2 + \left(q_j^{(\mathrm{y})}\right)^2}}, \tag{3.15}$$

$$\sin\phi_j = \frac{q_j^{(\mathrm{y})}}{\sqrt{\left(q_j^{(\mathrm{x})}\right)^2 + \left(q_j^{(\mathrm{y})}\right)^2}}. \tag{3.16}$$

The Ω is defined as

$$\Omega\left(\{z_{aj}\}\right) = \frac{2}{R^2}\sum_j q_j \tilde{q}_j + \sum_a \left(1 - \sum_j z_{aj}\right) E_a^{\perp}. \tag{3.17}$$

The second term in equation 3.17, soft energy, is now the *transverse* energy that does not belong to any jet. The first term has similar interpretation as for e^+e^- collisions; it is the dynamical width of jets.

Bibliography

[1] Fyodor V. Tkachov. The definition of jets. *Int. J. Mod. Phys.*, A17:2783–2884, 2002.

[2] F. V. Tkachov. Measuring the number of hadronic jets. *Phys. Rev. Lett.*, 73:2405–2408, 1994.

[3] Fyodor V. Tkachov. Measuring multijet structure of hadronic energy flow or what is a jet? *Int. J. Mod. Phys.*, A12:5411–5529, 1997.

[4] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder. *Comput. Phys. Commun.*, 155:42–64, 2003.

[5] N. Amos et al. The random grid search: A simple way to find optimal cuts. Prepared for International Conference on Computing in High-Energy Physics (CHEP 95), Rio de Janeiro, Brazil, 18-22 Sep 1995.

[6] Pushpalatha C. Bhat, Harrison Prosper, and Scott S. Snyder. Top quark physics at the Tevatron. *Int. J. Mod. Phys.*, A13:5113–5218, 1998.

[7] C. F. Berger et al. Snowmass 2001: Jet energy flow project. *eConf*, C010630:P512, 2001.

[8] Fyodor V. Tkachov. Approaching the parameter estimation quality of maximum likelihood via generalized moments. *Part. Nucl. Lett.*, 111:28–35, 2002.

Chapter 4

W-boson Mass Benchmark Test

A version of this chapter has been published [1].

In this chapter, we discuss a benchmark Monte Carlo test of Optimal Jet Definition. We compare three jet finding algorithms: JADE [2], Durham [3], and Optimal Jet Definition [4] in the context of the W-boson mass extraction from fully hadronic decay channel at 180 GeV. We determine how good a jet definition is based on how small the statistical uncertainty in the extracted W-boson mass is (assuming a fixed number of experimental events). The W-boson mass extraction procedure was inspired by the analysis performed by the OPAL collaboration from LEP II data [5].

First, we explain the details of the test and present the results. Next, we talk about running-time efficiency of the algorithms and discuss a possible optimization. At the end, we provide more details on how the presented results were obtained.

4.1 Details of the test

We simulated the process $e^+e^- \to W^+W^- \to$ hadrons at the center of mass energy of 180 GeV using PYTHIA 6.2 [6]. We reconstructed each event to four jets using Optimal Jet Definition [4] and two binary jet algorithms: $k_\perp$ (Durham) [3] and JADE [2] for comparison. We employed the Optimal Jet Finder[1] software [7] (described in appendix A), an implementation of Optimal Jet Definition. For the $k_\perp$ and JADE algorithms, we used KTCLUS implementation [9]. The JADE algorithm was obtained by modifying the measure in KTCLUS. For Optimal Jet Definition, we chose $R=2$ and employed the most primitive variant of Optimal Jet Finder based algorithm with a fixed $n_{\text{tries}}=10$ for all events.

[1] For clarity, "Optimal Jet Definition" [4] is a name of the jetfinding algorithm regardless of the implementation. "Optimal Jet Finder" is a name of the specific FORTRAN and C++ implementations [7, 8] described in appendices A and B.

Table 4.1: Results of the benchmark test: statistical uncertainty of W-boson mass corresponding to a 1000 experimental events.

ALGORITHM	$\delta M_{\text{exp}} \pm 3$ MeV
Optimal Jet Definition [4]	106
$k_\perp$ [3]	105
JADE [2]	118

The four jets can be combined into two pairs in three different ways (each pair supposedly resulting from the decay of a single W-boson). We choose the combination with the smallest difference in invariant masses between the two pairs and calculated the average m of the two masses. We generated the probability distribution $\pi_M(m)$ with the W-boson mass M as a parameter. The minimal statistical error of parameter estimation corresponding to the number N_{exp} of experimental events, as given by Rao-Frechet-Cramer theorem (see, for instance, [10, 11, 4]), is

$$\delta M_{\text{exp}} = \frac{1}{\sqrt{N_{\text{exp}}}} \cdot \frac{1}{\sqrt{I_{\text{opt}}}}. \tag{4.1}$$

I_{opt} is Fisher's information, that is, the informativeness $I[f_{\text{opt}}(m)]$ of the optimal observable $f_{\text{opt}}(m)$, given by

$$I_{\text{opt}} = I[f_{\text{opt}}] = \int \pi_M(m) f_{\text{opt}}^2(m)\, dm, \tag{4.2}$$

and

$$f_{\text{opt}}(m) = \frac{\partial \ln \pi_M(m)}{\partial M}. \tag{4.3}$$

The first factor in (4.1) depends only on the number of available experimental events and, as far as data analysis is concerned, is fixed. The second factor in (4.1) depends on the probability distribution $\pi_M(m)$ (and how it changes with M). Since different algorithms give (slightly) different $\pi_M(m)$, we can use δM_{exp} to compare jet finding algorithms.

4.2 Results

We computed numerically the statistical error δM_{exp} of the W-boson mass for the three jet finding algorithms using $\sim 10^7$ simulated events. δM_{exp} corresponding to a 1000 experimental events is displayed in table 4.1. (The error of 3 MeV in our results is dominated by the uncertainties in the numerical differentiation with respect to M. The value of the error is taken to be equal to $I_{\text{opt}}(\Delta M = 0) - I_{\text{opt}}(\Delta M = 1 \text{ GeV})$; see table 4.3.) Within the obtained precision Durham and Optimal Jet Finder are equivalent with respect to the accuracy; JADE appears to be worse.

4.3 Speed of the algorithms

An important aspect is the speed of the algorithms. The average processing time per event depends on the number of particles or detector cells in the input n_{parts}. We observed the following empirical relations[2] (time in seconds):

$$\begin{array}{ll} 1.2 \times 10^{-8} \times n_{\text{parts}}^3 & \text{for } \mathrm{k}_\perp, \\ 1.0 \times 10^{-4} \times n_{\text{parts}} \times n_{\text{tries}} & \text{for OJF.} \end{array} \tag{4.4}$$

n_{parts} varied from 50 to 170 in our sample, with the mean value of 83. However, the behavior was verified for n_{parts} up to 1700 by splitting each particle into 10 collinear fragments (similarly to how a particle may hit several detector cells).

We observed that Optimal Jet Finder is slower for small number of particles or detector cells, whereas, for a large number of particles, it appears to be relatively much faster. In the process we studied, Optimal Jet Finder starts to be more efficient for $n_{\text{parts}} \approx 90\sqrt{n_{\text{tries}}}$.

The dramatically better behavior of Optimal Jet Finder at large n_{parts} makes it a candidate for work at the level of detector cells, perhaps even on-line as all n_{tries} minimization attempts can be done in parallel.

4.4 Running time optimization

We tested a simple optimization, implemented using only the routines from the Optimal Jet Finder library (described in appendix A). It relies on the well-known fact that the jet structure is often determined by the most energetic particles. We select the most energetic particles (a skeleton event), and precluster them by running the minimization routine. Then, we add the remaining particles with random values of z_{aj} and run the minimization again. With a threshold of 2 GeV to select the energetic particles, $n_{\text{tries}} = 5$ at the preclustering phase and $n_{\text{tries}} = 1$ at the final stage, only a 1% change was observed for δM_{exp} (curiously, an improvement) whereas the speed significantly increased, with the dependence of the time per event on n_{parts} now given roughly by

$$2.5 \times 10^{-2} \times n_{\text{parts}} \tag{4.5}$$

with a hint at a slower growth at large n_{parts}. This is faster than KTCLUS starting from $n_{\text{parts}} \approx 140$, and the speed advantage increases sharply for higher n_{parts}: for $n_{\text{parts}} \approx 200$ this is twice as fast as KTCLUS, and an extrapolation to $n_{\text{parts}} \approx 1000$ yields the factor of 50.

[2] As already discussed in section 2.3.5, the running time of the public implementations [9, 12] of the $\mathrm{k}_\perp$ algorithm scales as n_{parts}^3, but a more efficient implementation for large n_{parts} is possible [13]. This should be kept in mind when discussing the speed issue.

Table 4.2: Number of simulated events in millions.

simulated mass [GeV]	Durham	OJF	JADE
79.450	9	1	1
79.700	9	-	-
79.950	9	9	9
80.200	9	-	-
80.350	9	-	-
80.450	9	1	1
80.550	9	-	-
80.700	9	-	-
80.950	9	9	9
81.200	9	-	-
81.450	9	1	1

4.5 Technical details of computing I_{opt}

In this section, we explain how we obtained the value of I_{opt}. We state the details of the computation and discuss the stability of the final results with respect to variation of the parameters used in the computation.

4.5.1 Procedure

We describe the procedure that we followed to compute I_{opt}.

Each event generates one m value, as described above. Table 4.2 shows the numbers of simulated events for each value of M. We bin m values with the number of bins equal to n_{points}. The corresponding width of the bins is $\Delta m = 100\ \text{GeV}/n_{\text{points}}$. We count the number N_i of events that have m in the range associated with that bin, that is, $m_i - \frac{\Delta m}{2} < m < m_i - \frac{\Delta m}{2}$. We calculate

$$\pi_i = \frac{N_i}{\Delta m N_{\text{events}}}, \tag{4.6}$$

where N_{events} is the total number of events. In the limit of infinite number of events

$$\pi_i = \frac{1}{\Delta m} \int_{m_i - \frac{\Delta m}{2}}^{m_i + \frac{\Delta m}{2}} \pi(m)\, dm \approx \pi(m_i). \tag{4.7}$$

Next, for each i, we compute

$$f_{\text{opt}}(m_i) \approx \frac{\pi_{M+\frac{\Delta M}{2}}(m_i) - \pi_{M-\frac{\Delta M}{2}}(m_i)}{\Delta M} \tag{4.8}$$

and integrate

$$I_{\text{opt}} \approx \Delta m \sum_{i=0}^{n_{\text{points}}} \pi_{Mi} f_{\text{opt}}^2(m_i), \tag{4.9}$$

where we take π_{Mi} directly from data or as

$$\pi_{Mi} \approx \frac{\pi_{(M-\frac{\Delta M}{2})i} + \pi_{(M-\frac{\Delta M}{2})i}}{2} \tag{4.10}$$

if π_{Mi} is not available.

4.5.2 Smoothing

We apply smoothing to $\pi(m_i)$ obtained in the way described in section 4.5.1. On small ranges of m values, $\pi(m)$ can be approximated by some polynomial. The smoothing procedure is as follows. For each point i, we fit a polynomial to all points (j, π_j) such that $|j - i| \leq n_{\text{sm}}$, where n_{sm} is some parameter that defines the range over which the polynomial is fitted. To determine the optimal range, we plot the distance $\langle d \rangle$,

$$\langle d \rangle = \Delta m \sum_i \pi_i \left| \pi_i - \pi_i^{\text{sm}} \right| \approx \int dm\, \pi(m) \left| \pi(m) - \pi^{\text{sm}}(m) \right|, \tag{4.11}$$

where π_i and π_i^{sm} are respectively probability distributions before and after smoothing, versus the number of points taken to fit the polynomial, n_{sm}. The results are presented in figure 4.1 for a 6-th order polynomial; it is clear that n_{sm} should be around 200 for $n_{\text{points}} = 10\,000$. Each of the tests presented below were done twice, with and without smoothing.

4.5.3 Number of events in the sample

We check how the calculated I_{opt} depends on the number of events in the sample. We take the whole available sample: 9×10^6 events and divide it into:

- 18 independent samples of 5×10^5 events each,
- 9 independent samples of 10^6 events each,
- 4 independent samples of 2×10^6 events each,
- 3 independent samples of 3×10^6 events each,
- 2 independent samples of 4×10^6 events each.

We compute I_{opt} for each sample. The results are presented in the figures 4.2 and 4.3. In the first case, no smoothing is applied; and in the latter, 6-th order polynomial smoothing is applied over the range of 2 GeV each side. (The choice of the smoothing range was discussed above.) As expected, fluctuations decrease when the number of events in the sample grows in both cases; but when no smoothing is applied, I_{opt} changes *systematically* as well.

4.5.4 Variable number of bins

We repeat the procedure of finding I_{opt} using various number of bins (n_{points}). We plot the results in figure 4.4. It can be seen that when smoothing is applied, I_{opt} is quite stable with respect to n_{points}.

4.5.5 Lower and upper integration limit

We estimate the contribution to I_{opt} coming from different values of m (see equation 4.9). Because of the low statistics, it is difficult to evaluate f_{opt} far from the real W-boson mass, but we can see here that it hardly matters. Figures 4.5 and 4.6 illustrate how the calculated I_{opt} is affected by the change of a lower and upper integration limit in (4.2).

4.5.6 Different values of M and ΔM

Different values of M and ΔM are used to evaluate I_{opt}. The results for available data are presented in figures 4.7 (without smoothing) and 4.8 (with smoothing). The circles represent $\Delta M = 0.25$ GeV, the boxes $\Delta M = 0.5$ GeV, and the triangles $\Delta M = 1$ GeV. It can be seen that the calculated I_{opt} depends systematically on ΔM. This is pictured in figure 4.9. All points in the figure are obtained for central value $M = 80.450$ GeV but with different ΔM. The solid circles represent the results without smoothing and the empty boxes with smoothing. It is apparent from that picture that we should consider $\Delta M \leq 0.2$ GeV. However, reducing the value of ΔM increases the minimal required number of events as discussed in the next section.

4.5.7 Relation between ΔM and the minimal number of events

In the previous section, we mention that the computed value of I_{opt} depends on the value of ΔM used to calculate derivatives. The true value of I_{opt} is approached when $\Delta M \to 0$. However, there are practical limits for taking ΔM smaller and smaller. Below, we estimate how statistical error of I_{opt} is related to ΔM. Suppose, we calculate the informativeness of some known observable f

$$I[f] = \left(\frac{\partial \langle f \rangle}{\partial M}\right)^2 \frac{1}{\text{Var}f}, \tag{4.12}$$

where

$$\text{Var}f = \int_0^\infty dm \pi(m) \left(f(m) - \langle f \rangle\right)^2. \tag{4.13}$$

We compute this taking finite ΔM

$$I[f] = \left(\frac{\langle f \rangle_{M+\Delta M} - \langle f \rangle_M}{\Delta M}\right)^2 \frac{1}{\text{Var}f}. \tag{4.14}$$

The statistical errors for the averages are

$$\delta\langle f\rangle_{M+\Delta M} = \delta\langle f\rangle_M = \sqrt{\frac{\text{Var}f}{N_{\text{events}}}}, \tag{4.15}$$

where N_{events} is the number of events. The resulting statistical error for $I[f]$ is

$$\delta I[f] = \frac{2\sqrt{2}\left|\langle f\rangle_{M+\Delta M} - \langle f\rangle_M\right|}{(\Delta M)^2\,\text{Var}f}\delta\langle f\rangle, \tag{4.16}$$

where we neglect the error of $\text{Var}f$. Using (4.15) and (4.14), we obtain

$$\frac{\delta I[f]}{I[f]} = \frac{2\sqrt{2}}{\sqrt{I[f]}}\cdot\frac{1}{\Delta M\sqrt{N}}. \tag{4.17}$$

Demanding that (our assumed accuracy)

$$\frac{\delta I[f]}{I[f]} \sim 0.01, \tag{4.18}$$

we get an estimation for the required number of events

$$N \sim \frac{10^6}{(\Delta M)^2}, \tag{4.19}$$

where we have used $I[f] \approx 0.087$ (as in our case), and ΔM is in GeV. This is only the minimal estimation for the error or the number of necessary events assuming that f is known. When calculating $I_{\text{opt}} = I[f_{\text{opt}}]$ the precision with which f_{opt} is known depends on N_{events} as well.

4.5.8 Extrapolation of I_{opt} to $\Delta M = 0$

The calculated values of I_{opt} depend on ΔM quite visibly, as shown in figure 4.9. Because it is relatively easy to calculate I_{opt} for larger ΔM, like 1 GeV, we estimate how the calculated value of I_{opt} depends on ΔM with the intention to extrapolate to $\Delta M = 0$. We checked this dependence experimentally, figure 4.9, but now we want to study it in a more analytical manner.

We define $I_{\text{opt}}(\Delta M)$ as follows

$$I_{\text{opt}}(\Delta M) = \int \frac{\pi(m; M+\Delta M) + \pi(m; M)}{2} \times \left[\frac{\ln\pi(m; M+\Delta M) - \ln\pi(m; M)}{\Delta M}\right]^2 dm. \tag{4.20}$$

We expand $\pi(M+\Delta M)$ and $\ln\pi(M+\Delta M)$ in series around M and substitute into the previous

equation

$$I_{\text{opt}}(\Delta M) = \int dm \left[\pi + \frac{1}{2}\sum_{i=1}\frac{1}{i!}\frac{\partial^i \pi}{\partial M^i}(\Delta M)^i\right]\left[\sum_{j=1}\frac{1}{j!}\frac{\partial^j \ln \pi}{\partial M^j}(\Delta M)^{j-1}\right]^2, \tag{4.21}$$

where π and all derivatives are taken at M. After multiplying the brackets, we get what can be described symbolically as

$$I_{\text{opt}}(\Delta M) = I_0 + I_1 \cdot \Delta M + I_2 \cdot (\Delta M)^2 + I_3 \cdot (\Delta M)^3 + \ldots, \tag{4.22}$$

where I_0, I_1, I_2, I_3, ... do not depend on ΔM and can be explicitly calculated from (4.21). I_0 is the desired I_{opt}. Of course, general numerical computation of I_1, I_2, I_3, ... is even more difficult than I_0 itself (higher order derivatives). On the other hand, they are only small corrections so they do not need to be known with so high accuracy as I_0. In our specific case (not in general) π is roughly a Breight-Wigner shape plus some background $b(m)$ which does not depend on M

$$\pi(m; M) = \frac{h}{(m - m_0)^2 + \left(\frac{\Gamma}{2}\right)^2} + b(m), \tag{4.23}$$

where $m_0 = M + \text{const}$ and all other constants do not depend implicitly on M. They can be estimated from simulated π. Then all the derivatives in (4.21) can be calculated analytically, that is, expressed in terms of the constants h, m_0, Γ, and the background $b(m)$, which is given numerically.

In figure 4.10, dots represent the direct numerical values for $I_{\text{opt}}(\Delta M)$ (as in figure 4.9). The smooth curve comes from calculation described above (for Durham), that is

$$\begin{aligned} I_{\text{opt}}(\Delta M) &= I_{\text{opt}} \\ &+ 0.000222\,(\Delta M) - 0.00338\,(\Delta M)^2 \\ &+ 0.0000148\,(\Delta M)^3 + 0.000156\,(\Delta M)^4 + \ldots \end{aligned} \tag{4.24}$$

Resulting corrections, that is $I_{\text{opt}}(\Delta M = 0) - I_{\text{opt}}(\Delta M = 1\text{ GeV})$ are given in table 4.3, and, as

Table 4.3: Corrections $I_{\text{opt}}(\Delta M = 0) - I_{\text{opt}}(\Delta M = 1\text{ GeV})$

ALGORITHM	$I_{\text{opt}}(\Delta M = 0) - I_{\text{opt}}(\Delta M = 1\text{ GeV})$
Optimal Jet Definition	0.0023
$k^{\perp}$	0.0025
JADE	0.0020

expected, are similar for all algorithms. (The corrections should be especially similar for Durham and Optimal Jet Finder.) The results extrapolated to $\Delta M = 0$ are given in table 4.1.

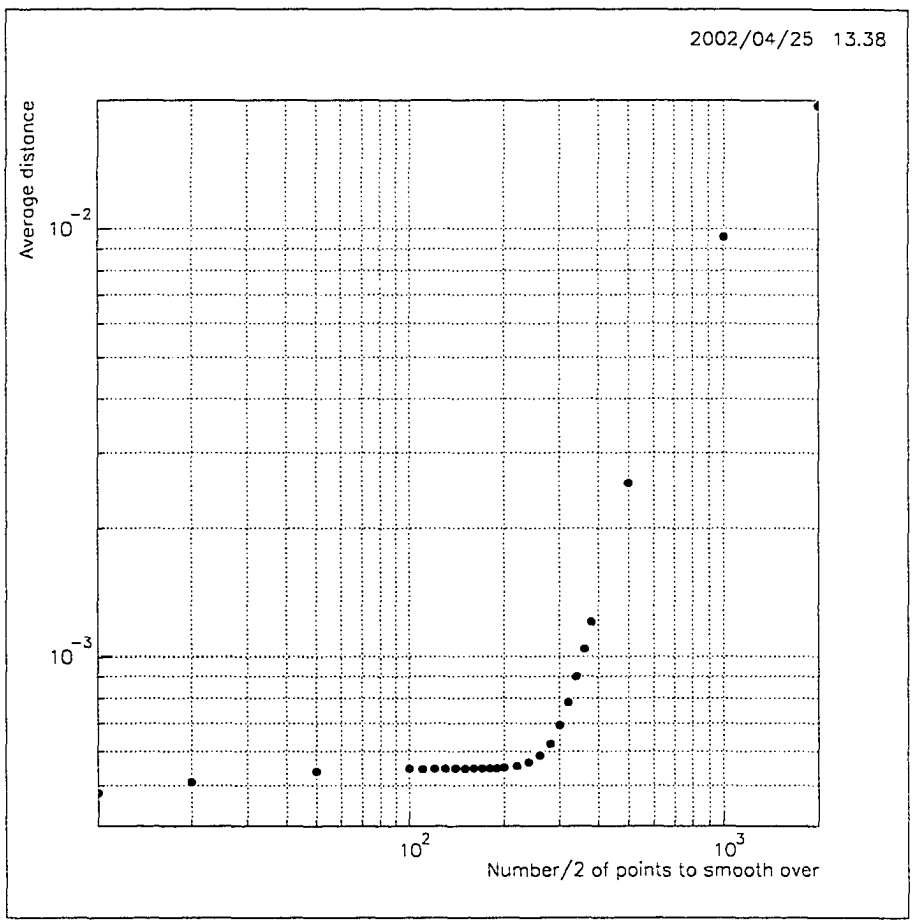

Figure 4.1: Average distance $\langle d \rangle$ versus the number of points used to fit the polynomial, n_{sm}. The entire number of points (bins) is 10 000.

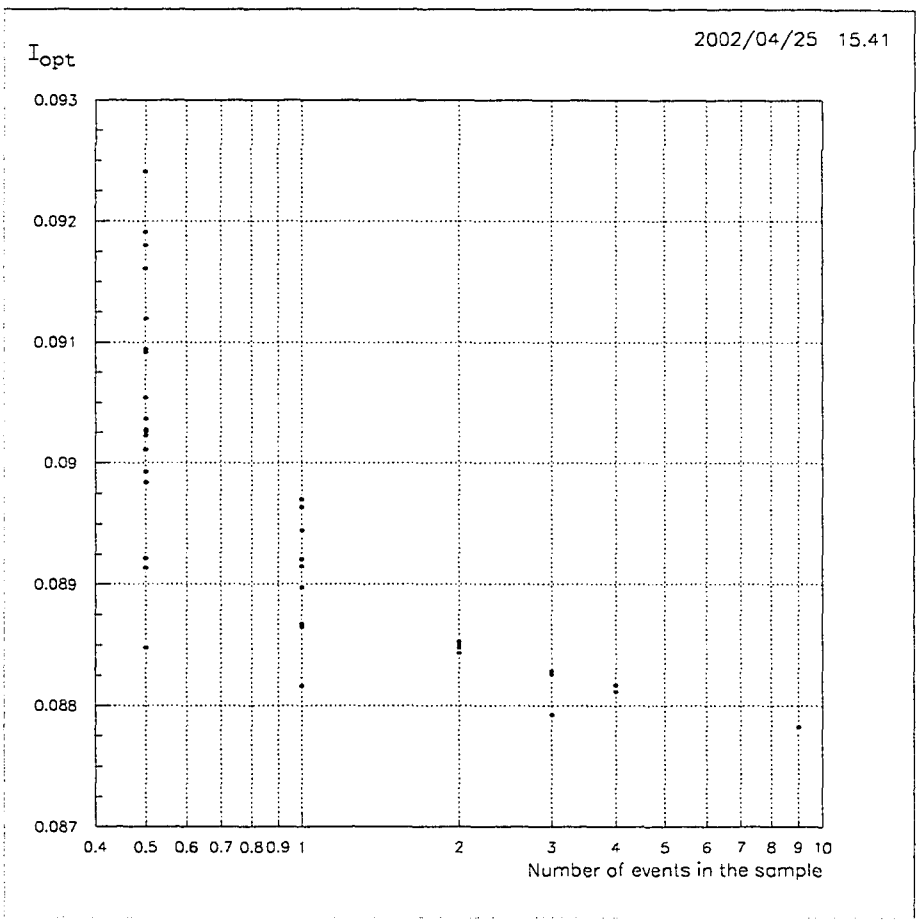

Figure 4.2: I_{opt} computed from different number of events in the sample. No smoothing.

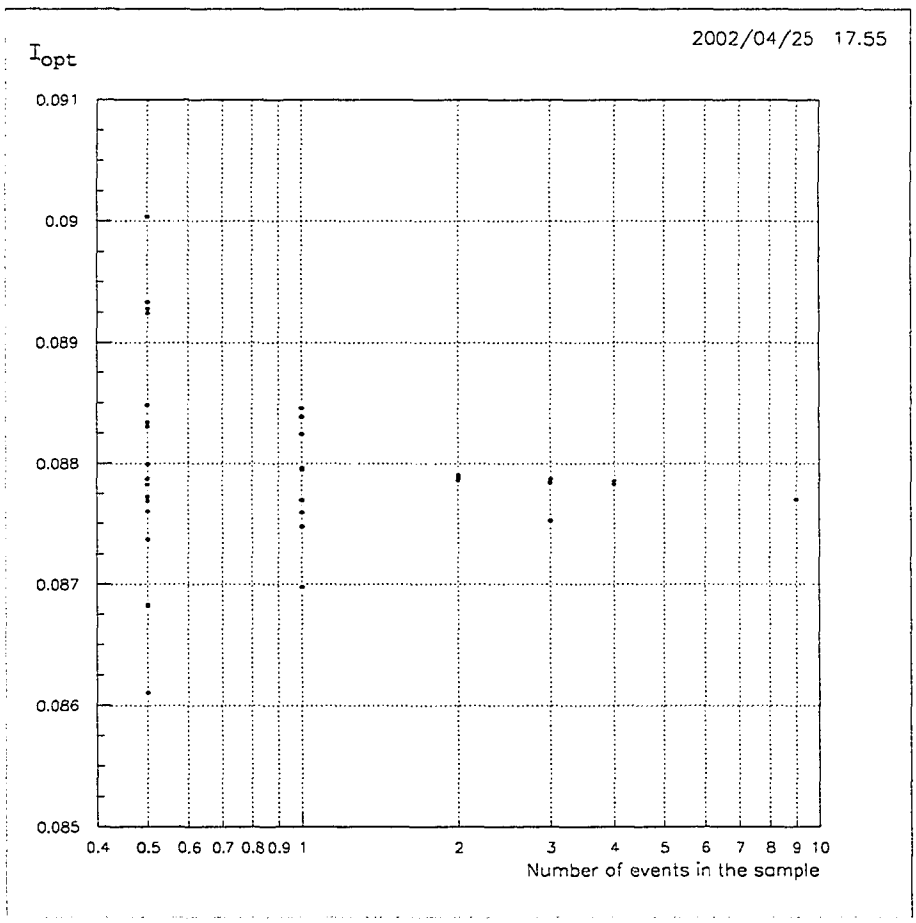

Figure 4.3: I_{opt} computed from different number of events in the sample. Smoothing applied.

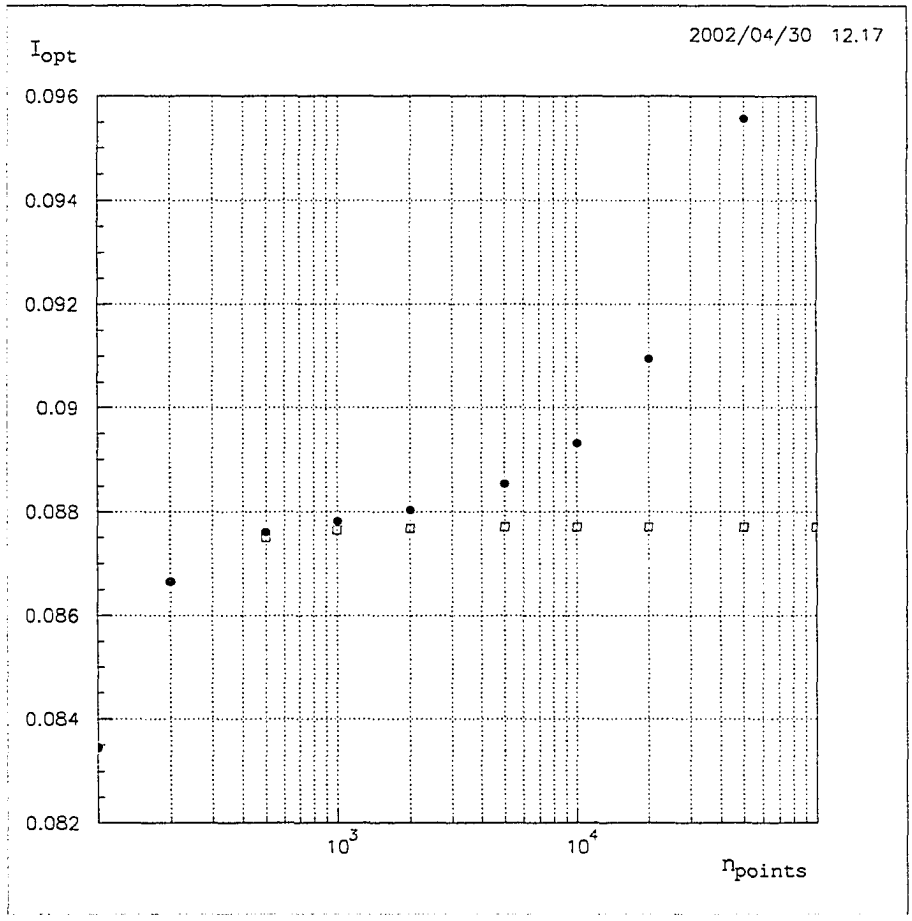

Figure 4.4: I_{opt} versus the number of bins, n_{points}. No smoothing: solid circles; smoothing applied: empty squares.

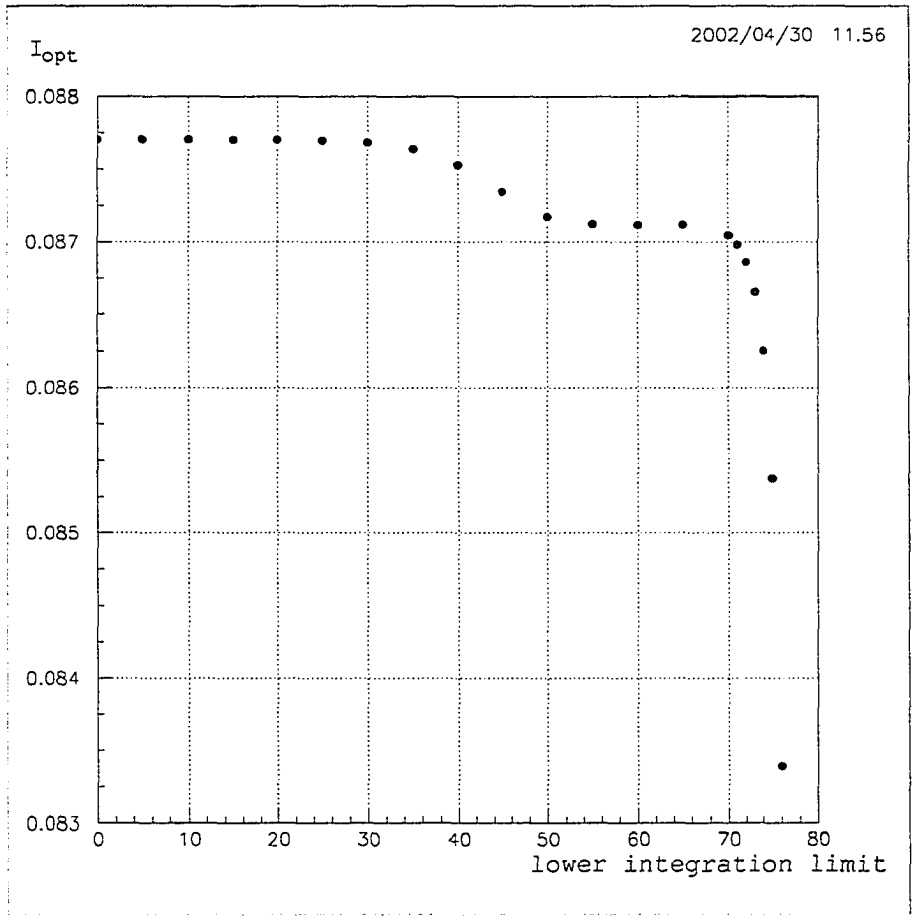

Figure 4.5: I_{opt} versus the lower integration limit. Smoothing makes no relative difference.

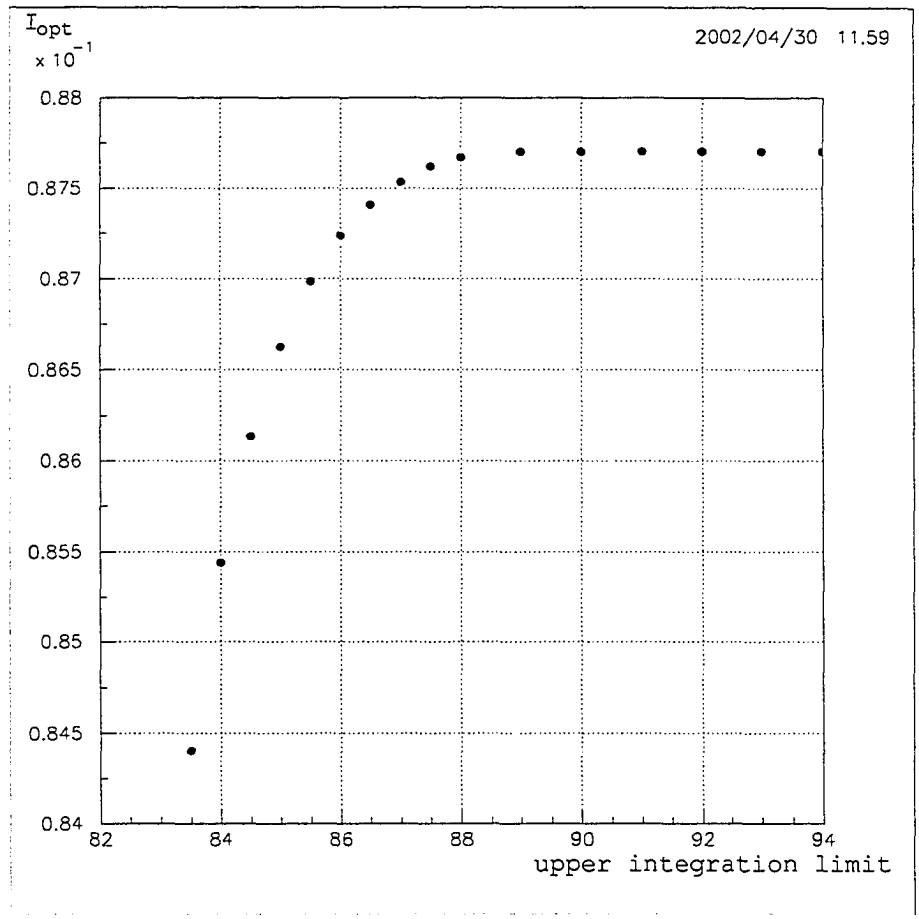

Figure 4.6: I_{opt} versus the upper integration limit. Smoothing makes no relative difference.

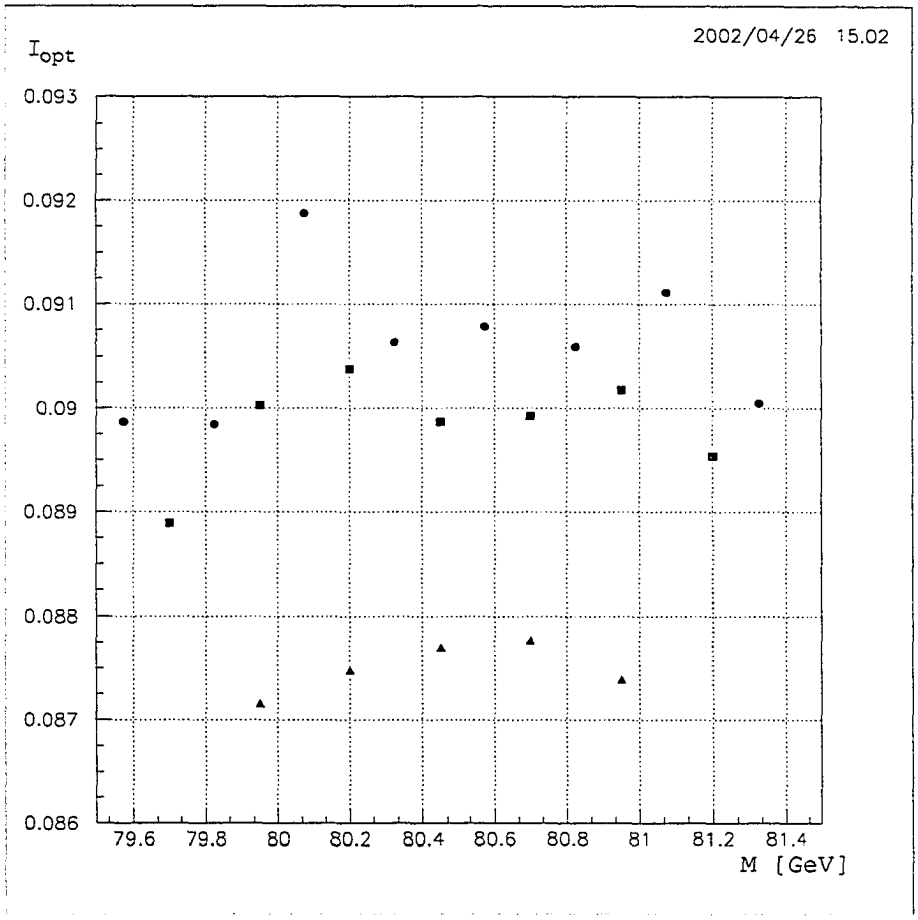

Figure 4.7: I_{opt} versus M. No smoothing. The circles represent $\Delta M = 0.25$ GeV, the boxes $\Delta M = 0.5$ GeV, and the triangles $\Delta M = 1$ GeV.

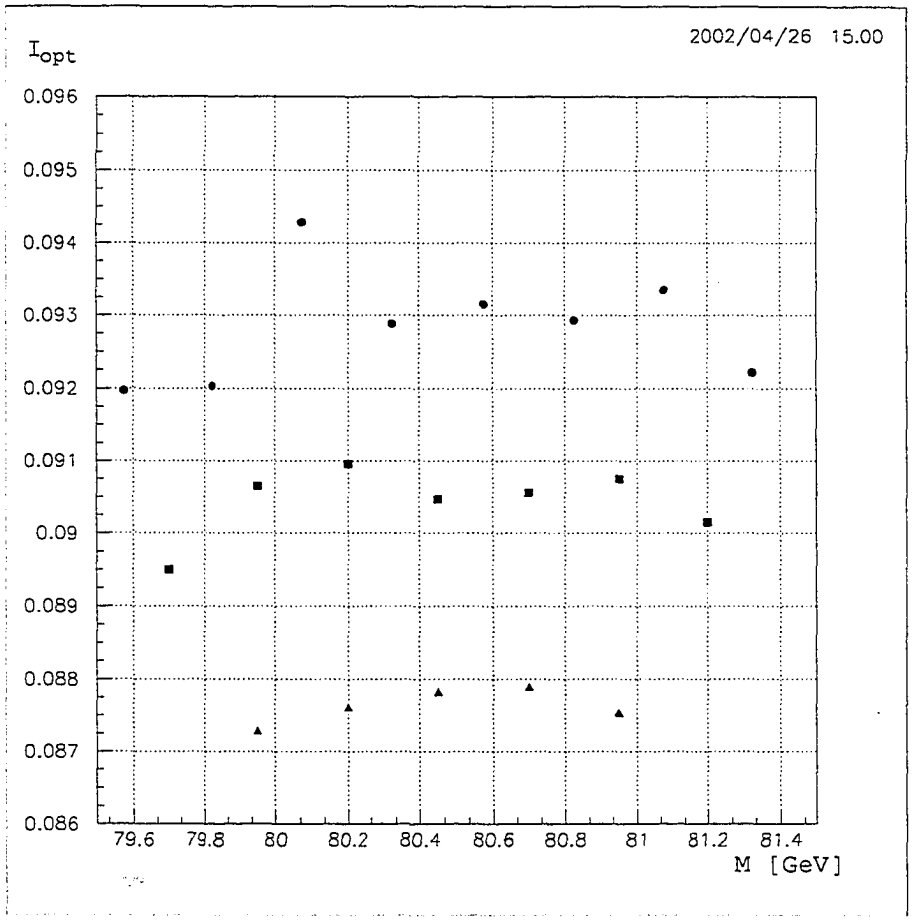

Figure 4.8: I_{opt} versus M. Smoothing applied. The circles represent $\Delta M = 0.25$ GeV, the boxes $\Delta M = 0.5$ GeV, and the triangles $\Delta M = 1$ GeV.

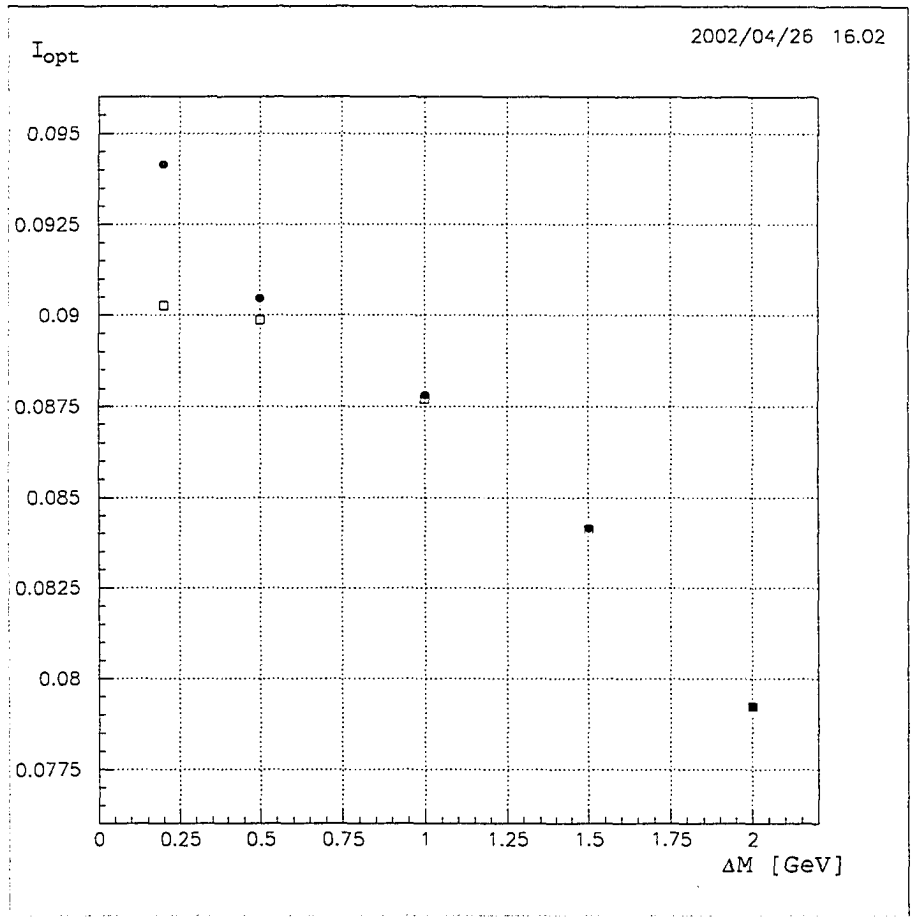

Figure 4.9: I_{opt} versus ΔM. The solid circles represent the results without smoothing; the empty boxes with smoothing.

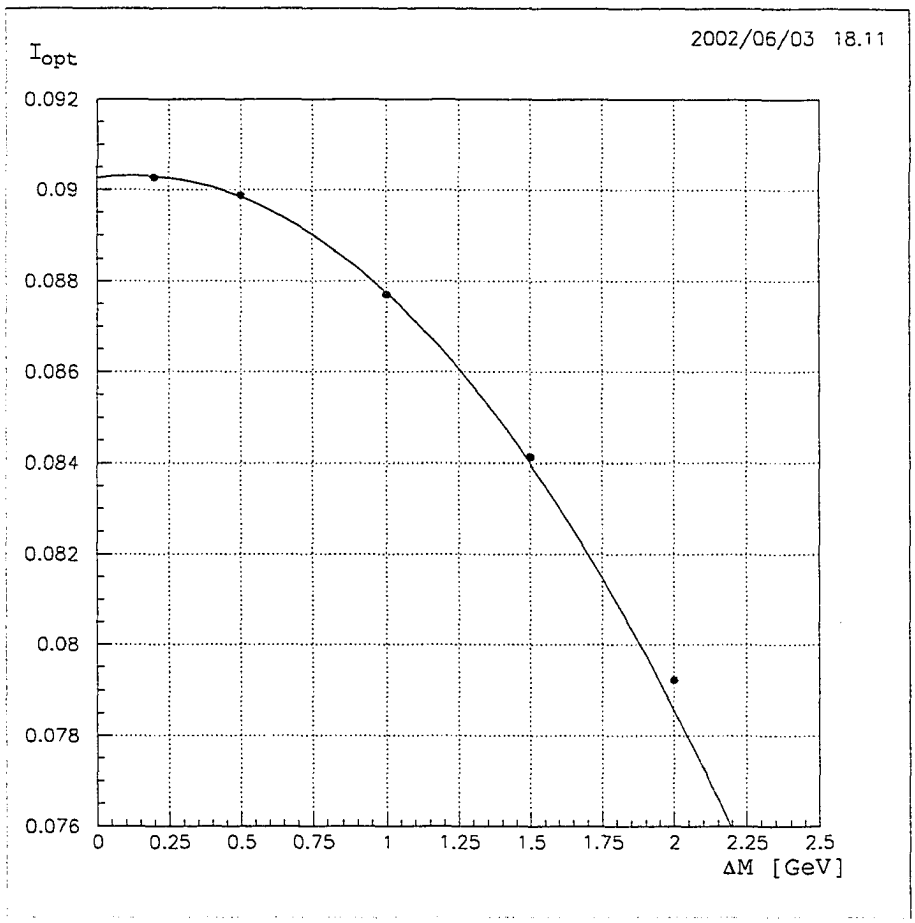

Figure 4.10: I_{opt} versus ΔM (horizontal axis) for Durham. The solid circles represent the results (with smoothing) computed for several values of ΔM. The continuous curve represents expression 4.24.

Bibliography

[1] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Towards a standard jet definition. *Phys. Rev. Lett.*, 91:061801, 2003.

[2] W. Bartel et al. Experimental studies on multi-jet production in e+ e- annihilation at PETRA energies. *Z. Phys.*, C33:23, 1986.

[3] S. Catani, Yuri L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber. New clustering algorithm for multi-jet cross-sections in e+ e- annihilation. *Phys. Lett.*, B269:432–438, 1991.

[4] Fyodor V. Tkachov. The definition of jets. *Int. J. Mod. Phys.*, A17:2783–2884, 2002.

[5] G. Abbiendi et al. Measurement of the mass and width of the W boson in e+ e- collisions at 189-GeV. *Phys. Lett.*, B507:29–46, 2001.

[6] Torbjorn Sjostrand et al. High-energy-physics event generation with PYTHIA 6.1. *Comput. Phys. Commun.*, 135:238–259, 2001.

[7] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder. *Comput. Phys. Commun.*, 155:42–64, 2003.

[8] S. Chumakov, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder (C++ v1.0). Submitted for publication in Comput. Phys. Commun.

[9] http://hepwww.rl.ac.uk/theory/seymour/ktclus/.

[10] W. T. Eadie, D. Drijard, F. E. James, M. Roos, and Sadoulet. *Statistical Methods in Experimenta Physics.* North-Holland, Amsterdam (1971).

[11] G. Cowan. *Statistical data analysis.* Oxford, UK: Clarendon (1998) 197 p.

[12] J. M. Butterworth, J. P. Couchman, B. E. Cox, and B. M. Waugh. KtJet: A C++ implementation of the k(t) clustering algorithm. *Comput. Phys. Commun.*, 153:85–96, 2003.

[13] Ernest Jankowski. $O\left(n^2\right)$ implemetation of kt algorithm. In preparation.

Chapter 5

See-saw Induced $\mu \longrightarrow e + \gamma$ Branching Ratio from Albright-Barr SO(10) Grand Unified Theory

A version of this chapter has been published [1].

Neutrinos have been observed to oscillate between flavour states [2, 3, 4]; a detailed review of the neutrino oscillation experiments can be found in [5]. The standard model charged current weak interactions produce the ν_e, ν_μ, and ν_τ neutrinos in association with charged leptons, electron, muon, and tau. That is, the neutrinos originate as definite flavour eigenstates. The corresponding interaction vertices are shown in figure 5.1. If the neutrino flavour and mass

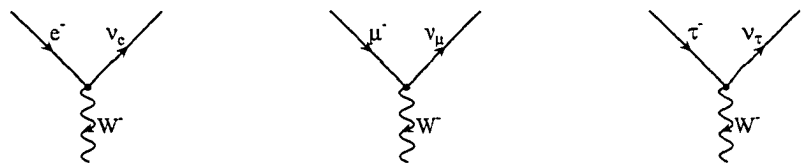

Figure 5.1: Standard model charged current weak interaction vertices involving neutrinos.

eigenstates are different — similarly to how the mass eigenstates of the down, strange, and bottom quarks are different from the corresponding weak interaction eigenstates — a neutrino that is produced as a definite flavour is a mixture of different mass eigenstates. Since the distinct mass eigenstates propagate differently depending on their mass, the combination of mass eigenstates

at the detection point is different from the original combination at the production point and, therefore, is no longer a definite flavour eigenstate.

To be more specific, we illustrate the neutrino mixing using the plane wave approximation (we assume that all the neutrino states are eigenstates of the momentum operator). Let us denote the flavour eigenstates as $|\nu_\alpha\rangle$ where $\alpha = e,\ \mu,\ \tau$ for electron, muon, or tau neutrino, respectively. We denote the mass eigenstates as $|\nu_{m_i}\rangle$ where $i = 1, 2, 3$. We assume only three light neutrino mass eigenstates, which seems proper in the context of the SO(10) model studied in this chapter and is also supported by data from Sudbury Neutrino Observatory (SNO), which has measured the total flux of the ν_e, ν_μ, and ν_τ neutrinos [6]. The relation between the mass and the flavour eigenstates can be written as

$$|\nu_\alpha\rangle = \sum_{i=1}^{3} U^*_{\alpha i} |\nu_{m_i}\rangle , \tag{5.1}$$

where $U_{\alpha i}$ are elements of the Maki-Nakagawa-Sakata (MNS) unitary matrix [7], which is the leptonic counterpart of the Cabbibo-Kobayashi-Maskawa matrix [8] that relates the mass and weak interaction states in the quark sector of the standard model. The mass eigenstates evolve with time and position as

$$|\nu_{m_i}(x)\rangle = e^{-ipx} |\nu_{m_i}(0)\rangle , \tag{5.2}$$

where $p = (E, \mathbf{p})$ is the four-momentum of the neutrino, $x = (t, \mathbf{x})$ is the position four-vector, and we take $x = 0$ at the point where the neutrinos are produced. (The factors of $\hbar$ and c have been omitted; $\hbar = c = 1$.) Since the neutrinos are highly relativistic, we have

$$px = Et - \mathbf{p}\cdot\mathbf{x} \simeq |\mathbf{p}| \left(1 + \frac{m_i^2}{2|\mathbf{p}|^2}\right) t - \mathbf{p}\cdot\mathbf{x} \simeq \frac{m_i^2 L}{2E} \tag{5.3}$$

where $L \simeq t$ is the distance between the production and detection point. (We assume that $\mathbf{p}$ points along the direction between the production and detection point, and the neutrinos are produced with the same momentum, $\mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_3 = \mathbf{p}$.) The neutrino that was originally produced with flavour α after traveling a distance L evolves to

$$|\nu_\alpha(L)\rangle = \sum_{\beta=e,\mu,\tau} \left(\sum_{i=1}^{3} e^{-i\frac{m_i^2 L}{2E}} U^*_{\alpha i} U_{\beta i} \right) |\nu_\beta\rangle , \tag{5.4}$$

which results from combining (5.1)-(5.3). The last equation implies that the probability $|\langle\nu_\beta|\nu_\alpha(L)\rangle|^2$ of detecting a neutrino with flavour β different from the original flavour α is non-zero only if at least two of m_1, m_2, m_3 are different, and $U_{\alpha i}$ is not diagonal.

Equation 5.4 explains how neutrino mixing implies that the neutrino have non-zero masses. In fact, if $m_1 = m_2 = m_3 = m$ (which includes the case of zero masses), from the unitarity of

the $U_{\alpha i}$ matrix,

$$\sum_{i=1}^{3} e^{-i\frac{m_i^2 L}{2E}} U_{\alpha i}^* U_{\beta i} = e^{-i\frac{m_i^2 L}{2E}} \delta_{\alpha\beta}, \tag{5.5}$$

and (5.4) reduces to

$$|\nu_\alpha(L)\rangle = e^{-i\frac{m_i^2 L}{2E}} |\nu_\alpha\rangle . \tag{5.6}$$

As can be inferred from (5.4), neutrino oscillation experiments are sensitive to Δm_{ij}^2. The data indicate that Δm_{ij}^2 are very small in comparison with the electroweak mass scale. More specifically, an example global analysis [5] of neutrino oscillation data gives

$$2.4 \times 10^{-5}\ \text{eV}^2 \ < \ \Delta m_{21}^2 < 2.4 \times 10^{-4}\ \text{eV}^2, \tag{5.7}$$

$$1.4 \times 10^{-3}\ \text{eV}^2 \ < \ \Delta m_{32}^2 < 6.0 \times 10^{-3}\ \text{eV}^2. \tag{5.8}$$

($\Delta m_{31}^2 = \Delta m_{32}^2 + \Delta m_{21}^2$, and is not a free parameter.) The smallness of Δm_{ij}^2, in turn, suggests that the neutrino masses m_1, m_2, m_3 are small. Unfortunately, the existing direct upper limits on the neutrino masses [9, 10, 11] are far above the mass scale suggested by the oscillation experiments;

$$m_{\nu_e}^{(\text{eff})} \ < \ 2.8\ \text{eV}, \tag{5.9}$$

$$m_{\nu_\mu}^{(\text{eff})} \ < \ 170\ \text{keV}, \tag{5.10}$$

$$m_{\nu_\tau}^{(\text{eff})} \ < \ 18.2\ \text{MeV}, \tag{5.11}$$

where

$$m_{\nu_\alpha}^{(\text{eff})} = \sqrt{\sum_{i=1}^{3} |U_{\alpha i}|^2 m_i^2}. \tag{5.12}$$

It is puzzling why the neutrino masses are so much smaller than the masses of quarks and leptons. This relative difference of mass scales suggests that the mechanism responsible for neutrino masses is different from purely electroweak symmetry breaking that generates masses for other standard model particles. A natural mechanism for generating small neutrino masses is the see-saw mechanism [12] described below.

Another intriguing feature revealed by the neutrino oscillation experiments is that the structure of the MNS matrix is significantly different from the CKM matrix. While the CKM matrix is almost diagonal, the mixing of neutrinos is substantial. A global fit [5] to neutrino oscillation data gives the following ranges of the moduli of the MNS matrix elements

$$\begin{pmatrix} 0.73 \text{ to } 0.89 & 0.45 \text{ to } 0.66 & 0.00 \text{ to } 0.24 \\ 0.23 \text{ to } 0.66 & 0.24 \text{ to } 0.75 & 0.52 \text{ to } 0.87 \\ 0.06 \text{ to } 0.57 & 0.40 \text{ to } 0.82 & 0.48 \text{ to } 0.85 \end{pmatrix}. \tag{5.13}$$

For comparison, the ranges of the moduli of the CKM matrix elements are [13]

$$\begin{pmatrix} 0.9739 \text{ to } 0.9751 & 0.221 \text{ to } 0.227 & 0.0029 \text{ to } 0.0045 \\ 0.221 \text{ to } 0.227 & 0.9730 \text{ to } 0.9744 & 0.039 \text{ to } 0.044 \\ 0.0048 \text{ to } 0.014 & 0.037 \text{ to } 0.043 & 0.9990 \text{ to } 0.9992 \end{pmatrix}. \tag{5.14}$$

(The CKM matrix is defined by the equation

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}, \tag{5.15}$$

where d, s, b are the mass eigenstates for down, strange, and bottom quarks, whereas d', s', b' are the charged current weak interaction eigenstates.)

To explain the see-saw mechanism [12], we will use a simplified "standard model" with only one generation of particles, d, u, e, ν. The part of the Lagrangian that contains the mass terms for those particles reads

$$\mathcal{L}_{\text{mass}} = -m_d \bar{\Psi}_d \Psi_d - m_u \bar{\Psi}_u \Psi_u - m_e \bar{\Psi}_e \Psi_e \tag{5.16}$$

where Ψ_d, Ψ_u, Ψ_e are the Dirac four-component spinors corresponding to the down and up quarks, and electron with masses m_d, m_u, and m_e, respectively. It is convenient to rewrite (5.16) using two-component spinors d_{L}, d_{R}, u_{L}, u_{R}, e_{L}, e_{R} corresponding to left- and right-handed portions of the down and up quarks, and electron. In the Weyl basis,

$$\Psi_d = \begin{pmatrix} d_{\text{L}} \\ d_{\text{R}} \end{pmatrix}, \quad \frac{1-\gamma_5}{2}\Psi_d = \begin{pmatrix} d_{\text{L}} \\ 0 \end{pmatrix}, \quad \frac{1+\gamma_5}{2}\Psi_d = \begin{pmatrix} 0 \\ d_{\text{R}} \end{pmatrix}, \tag{5.17}$$

and similarly for Ψ_u, Ψ_e. Using the two-component spinors,

$$\mathcal{L}_{\text{mass}} = -m_u u_{\text{L}}^\dagger u_{\text{R}} - m_d d_{\text{L}}^\dagger d_{\text{R}} - m_e e_{\text{L}}^\dagger e_{\text{R}} \quad + \quad \text{h. c.} \tag{5.18}$$

In the standard model, only the left-handed neutrino ν_{L} is present; the lack of the right-handed counterpart part ν_{R} prevents from writing the corresponding mass term in (5.18) and the neutrino remains massless.

If only Lorentz invariance and renormalizability of $\mathcal{L}_{\text{mass}}$ are demanded, other mass terms are possible in our example, namely,

$$M_{d_{\text{L}}} d_{\text{L}}^{\text{T}} \sigma^2 d_{\text{L}}, \quad M_{u_{\text{L}}} u_{\text{L}}^{\text{T}} \sigma^2 u_{\text{L}}, \quad M_{e_{\text{L}}} e_{\text{L}}^{\text{T}} \sigma^2 e_{\text{L}}, \quad M_{\nu_{\text{L}}} \nu_{\text{L}}^{\text{T}} \sigma^2 \nu_{\text{L}}, \tag{5.19}$$

$$M_{d_{\text{R}}} d_{\text{R}}^{\text{T}} \sigma^2 d_{\text{R}}, \quad M_{u_{\text{R}}} u_{\text{R}}^{\text{T}} \sigma^2 u_{\text{R}}, \quad M_{e_{\text{R}}} e_{\text{R}}^{\text{T}} \sigma^2 e_{\text{R}}, \tag{5.20}$$

where σ^2 is the Pauli sigma matrix, $M_{\ldots}$ are parameters with the mass dimension (and T denotes

transpose). No such terms are present in the standard model Lagrangian because they violate the SU(3) × SU(2) × U(1) gauge group invariance. However, we can imagine adding to the list of particles an extra right-handed neutrino field ν_{R} that does not change under the SU(3) × SU(2) × U(1) gauge group transformations (a singlet representation of the gauge group). With this additional neutrino field, two extra mass terms are possible in the Lagrangian,

$$\mathcal{L}_{\mathrm{mass}} = -m_u u_{\mathrm{L}}^{\dagger} u_{\mathrm{R}} - m_d d_{\mathrm{L}}^{\dagger} d_{\mathrm{R}} - m_e e_{\mathrm{L}}^{\dagger} e_{\mathrm{R}} - m_\nu \nu_{\mathrm{L}}^{\dagger} \nu_{\mathrm{R}} - \frac{1}{2} M_\nu \nu_{\mathrm{R}}^{\mathrm{T}} \sigma^2 \nu_{\mathrm{R}} \quad + \quad \text{h. c.}, \tag{5.21}$$

where m_ν and M_ν are parameters with the mass dimension. In the following, we focus only on the neutrino masses and omit the remaining part of the mass Lagrangian (5.21); that is, we examine

$$\mathcal{L}_{\mathrm{mass},\,\nu} = -m_\nu \nu_{\mathrm{L}}^{\dagger} \nu_{\mathrm{R}} - \frac{1}{2} M_\nu \nu_{\mathrm{R}}^{\mathrm{T}} \sigma^2 \nu_{\mathrm{R}} \quad + \quad \text{h. c.} \tag{5.22}$$

The last equation can be conveniently rewritten in the matrix notation,

$$\mathcal{L}_{\mathrm{mass},\,\nu} = -\frac{1}{2} \begin{pmatrix} \nu_{\mathrm{L}}^{\mathrm{T}} \sigma^2 & \nu_{\mathrm{R}}^{\dagger} \end{pmatrix} \begin{pmatrix} 0 & m_\nu \\ m_\nu & M_\nu \end{pmatrix} \begin{pmatrix} \nu_{\mathrm{L}} \\ \sigma^2 \nu_{\mathrm{R}}^{*} \end{pmatrix} \quad + \quad \text{h. c.} \tag{5.23}$$

The eigenvalues of the mass matrix $\begin{pmatrix} 0 & m_\nu \\ m_\nu & M_\nu \end{pmatrix}$ are $\dfrac{M_\nu \mp \sqrt{M_\nu^2 + 4m_\nu^2}}{2}$, which in the limit of $m_\nu \ll M_\nu$ reduces to

$$-\frac{m_\nu^2}{M_\nu}, \quad M_\nu. \tag{5.24}$$

We can expect that m_ν, similarly to m_e, m_d, m_u, is generated by the electroweak symmetry breaking, and, therefore, is of the order of 1 MeV to 100 GeV. If, for example, the light neutrino mass $\sim 10^{-2}$ eV (in agreement with the current observations), the scale for the heavy neutrino is 10^5 to 10^{15} GeV. The heavy neutrino is not accessible to experiments, operating at energies $\ll M_\nu$, however, according to (5.24), its presence reduces the mass of the light neutrino in comparison with the electroweak scale by the factor of m_ν/M_ν, which can be many orders of magnitude if M_ν arises from physics at the Grand Unification Theory (GUT) scale. This mechanism of generating light neutrino masses is called the see-saw mechanisms [12].

Generalization to the case with three generations of fermions is straightforward. We have to supplement the standard model particles with three heavy, SU(3) × SU(2) × U(1) gauge group singlet neutrinos. The mass matrix in an expression analogous to (5.23) is six dimensional and has three small eigenvalues corresponding to masses of the three light standard model neutrinos and three large eigenvalues corresponding to the three heavy neutrinos, which, because of their mass, are beyond direct observations.

While the see-saw mechanism is an economical and natural way to understand the smallness of the inferred neutrino masses, there are many possible methods of implementing it, and therefore detailed neutrino observations can be used to constrain GUT models. Perhaps the most elegant GUT uses the grand unifying group SO(10) in four spacetime dimensions. The spinor

representation of SO(10) is 16 dimensional, which accommodates all the helicity states of one fermion family plus an extra singlet degree of freedom for a Majorana neutrino. The generations are simply three copies of the spinor representation. Since GUTs relate quark and lepton masses and mixings, it is perplexing from a model building perspective as to why lepton mixing is so different from that in the quark sector. More specifically, it is of interest to understand why $|U_{\mu 3}|$ of the MNS matrix, (5.1) and (5.13), is so much larger than $|V_{cb}|$ of the CKM matrix, (5.14). Over the last few years a number of models have been developed to address this difference [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Recently, a particularly interesting and highly successful class of supersymmetric SO(10) GUTs has emerged that makes use of asymmetric mass matrices known as lopsided textures [14, 15, 16]. In these models, the charged lepton sector is responsible for the large atmospheric mixing angle while the Majorana singlet neutrino matrix has a simple form that results in the large solar mixing angle. Throughout this chapter we will refer to these models as the AB model class [14].

After GUT breaking, these models reduce to the R-parity conserving minimal supersymmetric standard model (MSSM) with specific model dependent relationships amongst the Yukawa couplings. In addition to the constraints already provided by the neutrino physics (and the demand that these models reproduce all the low energy physics of the standard model), the Wilkinson Microwave Anisotropy Probe (WMAP) satellite observations of the cosmic microwave background temperature fluctuations [25, 26] provide strong constraints on the available supersymmetric parameter space if the lightest supersymmetic particle (LSP) is assumed to compose the dark matter [27, 28, 29, 30, 31, 32, 33, 34]. With the WMAP data constraints, the definite flavour structure of the AB models will result in specific soft supersymmetry breaking parameters. Therefore, the AB model class gives well defined predictions for lepton flavour violation and in particular $\mu \to e\gamma$. It is of considerable interest to determine how the lepton flavour changing neutral current bounds restrict the Constrained Minimal Supersymmetric Standard Model (CMSSM) parameters for the AB model class in light of the WMAP data.

We consider $\mu \to e\gamma$ since at the present time, with the current bound [35] of $\mathrm{BR}(\mu \to e\gamma) < 1.2 \times 10^{-11}$, this process gives the strongest constraints on lepton flavour violation in the class of models that we discuss. Furthermore, the MEG experiment at Paul Scherrer Institute [36] expects to improve on this bound with the expected sensitivity of $\mathrm{BR}(\mu \to e\gamma) \lesssim 5 \times 10^{-14}$. This experiment will provide stringent limits on models with charged lepton flavour violation.

We organize this chapter as follows. In section 5.1, we outline the essential details of the AB models. In section 5.2, we discuss the supersymmetric parameter space, and display our numerical results with the combined constraints from $\mu \to e\gamma$ and the WMAP satellite observations. In section 5.3, we present our conclusions. Appendix C provides further calculational details.

5.1 The AB Model Definition

The AB model class is based on an SO(10) GUT with a U(1)×Z_2×Z_2 flavour symmetry and uses a minimum set of Higgs fields to solve the doublet-triplet splitting problem [14, 15, 16]. The interesting feature of these models is the use of an asymmetric ("lopsided") texture. The approximate form of the charged lepton and the down quark mass matrix in these models is given by

$$\mathbf{L} \sim \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \epsilon \\ 0 & \sigma & 1 \end{pmatrix}, \quad \mathbf{D} \sim \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \sigma \\ 0 & \epsilon & 1 \end{pmatrix} \tag{5.25}$$

where $\sigma \sim 1$ and $\epsilon \ll 1$. As pointed out by the authors of [14], this asymmetric structure naturally occurs within a minimal SU(5) GUT where the Yukawa interaction for the down quarks and leptons is of the form $\lambda_{ij}\bar{\mathbf{5}}_i\mathbf{10}_j\mathbf{5_H}$ ($\mathbf{5_H}$ denotes the Higgs scalars). In an SU(5) GUT, the left-handed leptons and the charge conjugate right-handed down quarks belong to the $\bar{\mathbf{5}}$ while the **10** contains the charge conjugate right-handed leptons and the left-handed down quarks. Therefore the lepton and down quark mass matrices are related to each other by a left-right transpose. Since SU(5) is a subgroup of SO(10), this feature is retained in an SO(10) GUT. This lopsided texture has the ability to explain why $|U_{\mu 3}| >> |V_{cb}|$. Making use of this observation, the AB models contain the Dirac mass matrices $\mathbf{U}, \mathbf{N}, \mathbf{D}, \mathbf{L}$ for the up-like quarks, neutrinos, down-like quarks, and the charged leptons respectively [16],

$$\mathbf{U} = \begin{pmatrix} \eta & 0 & 0 \\ 0 & 0 & \epsilon/3 \\ 0 & -\epsilon/3 & 1 \end{pmatrix} M_U, \quad \mathbf{N} = \begin{pmatrix} \eta & 0 & 0 \\ 0 & 0 & -\epsilon \\ 0 & \epsilon & 1 \end{pmatrix} M_U, \tag{5.26}$$

$$\mathbf{D} = \begin{pmatrix} 0 & \delta & \delta' e^{i\phi} \\ \delta & 0 & \sigma + \epsilon/3 \\ \delta' e^{i\phi} & -\epsilon/3 & 1 \end{pmatrix} M_D, \quad \mathbf{L} = \begin{pmatrix} 0 & \delta & \delta' e^{i\phi} \\ \delta & 0 & -\epsilon \\ \delta' e^{i\phi} & \sigma + \epsilon & 1 \end{pmatrix} M_D, \tag{5.27}$$

where

$$\begin{aligned} M_U &\approx 113\ \text{GeV}, & M_D &\approx 1\ \text{GeV}, \\ \sigma &= 1.78, & \epsilon &= 0.145, \\ \delta &= 8.6 \times 10^{-3}, & \delta' &= 7.9 \times 10^{-3}, \\ \phi &= 126°, & \eta &= 8 \times 10^{-6}. \end{aligned} \tag{5.28}$$

Dimensionless Yukawa couplings $\mathbf{Y_U}, \mathbf{Y_N}, \mathbf{Y_D}$, and $\mathbf{Y_E}$ can be extracted from the Dirac matrices. The given values of M_D and M_U best fit the low energy data with $\tan\beta \approx 5$. It should be noted that larger values of $\tan\beta$ are easily accommodated by altering the values of M_U and M_D while retaining accurate fits to the low energy data after renormalization group running. The lopsided texture of the AB model class nicely fits the large atmospheric mixing angle; however, in order to obtain the large solar mixing angle a specific hierarchical form of the heavy Majorana singlet

neutrino matrix needs to be chosen [15, 16], namely,

$$\mathrm{M_N} = \begin{pmatrix} b^2\eta^2 & -b\epsilon\eta & a\eta \\ -b\epsilon\eta & \epsilon^2 & -\epsilon \\ a\eta & -\epsilon & 1 \end{pmatrix} \Lambda_\mathrm{N}. \tag{5.29}$$

where the parameters ϵ and η are as defined in (5.28). The parameters a and b are of order 1 and $\Lambda_\mathrm{N} \sim 2 \times 10^{14}$ GeV. Since the Majorana singlet neutrino matrix is not related to the Dirac Yukawa structure, it is not surprising that this matrix should take on a form independent from the rest of the model. Once these choices have been made, the AB model class is highly predictive and accurately fits all the low energy standard model physics and the neutrino mixing observations.

It should be emphasized that all these relations are defined at the GUT scale and are therefore subject to renormalization group running [12, 37]. If we conservatively assume that the GUT symmetry breaks directly to the standard model gauge symmetries, SU(3)×SU(2)×U(1), and that supersymmetry is broken super-gravitationally through a hidden sector in a flavour independent manner, the AB model class will give well defined predictions for charged lepton flavour violation. There may also be significant contributions to the off-diagonal elements from renormalization group running between the GUT and gravity scales [38, 39]. Since the particulars of GUT and supersymmetry breaking – as well as the possibility of new physics above the GUT scale – can have model dependent effects on the branching ratio for $\mu \to e\gamma$, we do not consider an interval of running between the GUT and gravity scales.

The specific model predictions for the Dirac Yukawa couplings and the form of the Majorana singlet neutrino matrix will feed into the soft supersymmetry breaking slepton mass terms through renormalization group running, generating off diagonal elements that will contribute to flavour changing neutral currents [40]. The amount of flavour violation contained in the AB model class can be examined through the branching ratio of the process $\mu \to e\gamma$.

5.2 Numerical Results for $\mu \to e\gamma$

After GUT and supersymmetry breaking, we have the constrained minimal supersymmetric standard model with heavy gauge singlet neutrinos to make use of the see-saw mechanism. The leptonic part of the superpotential is

$$W = \epsilon_{\alpha\beta} H_\mathrm{d}^\alpha \mathbf{E} \mathbf{Y}_\mathrm{E} \mathbf{L}^\beta + \epsilon_{\alpha\beta} H_\mathrm{u}^\alpha \mathbf{N} \mathbf{Y}_\mathrm{N} \mathbf{L}^\beta + \frac{1}{2} \mathbf{N} \mathbf{M}_\mathrm{N} \mathbf{N} \tag{5.30}$$

where $\mathbf{Y}_\mathrm{E}$, $\mathbf{Y}_\mathrm{N}$ are Yukawa matrices, and $\mathrm{M_N}$ is the singlet Majorana neutrino mass matrix. The totally antisymmetric symbol is defined $\epsilon_{12} = +1$. We explain our notation in detail in appendix

C. On integrating out the heavy singlet neutrinos, (5.30) reduces to

$$W = \epsilon_{\alpha\beta} H_{\rm d}^{\alpha} \mathbf{E} \mathbf{Y}_{\rm E} \mathbf{L}^{\beta} - \frac{1}{2} \nu^{\rm T} \mathbf{m}_{\nu} \nu \tag{5.31}$$

where

$$\mathbf{m}_{\nu} = \frac{v^2}{2} \mathbf{Y}_{\rm N}^{\rm T} \mathbf{M}_{\rm N}^{-1} \mathbf{Y}_{\rm N} \sin^2 \beta \tag{5.32}$$

is the see-saw induced light neutrino mass matrix. The coefficients β and v are defined in terms of Higgs fields expectation values by

$$\frac{v^2}{2} = \langle H_{\rm d}^0 \rangle^2 + \langle H_{\rm u}^0 \rangle^2 = (174\,{\rm GeV})^2, \qquad \tan\beta = \frac{\langle H_{\rm u}^0 \rangle}{\langle H_{\rm d}^0 \rangle}. \tag{5.33}$$

The neutrino mass matrix, equation 5.32, is in general not diagonal and this is the source of lepton flavour violating interactions.

We assume that supersymmetry is broken softly in that breaking occurs through operators of mass dimension 2 and 3. The soft supersymmetry breaking Lagrangian relevant to lepton flavour violation studies is

$$\begin{aligned}
\mathcal{L}_{\rm breaking} \;=\; & -\delta_{\alpha\beta} \tilde{\mathbf{L}}^{\alpha\dagger} \mathbf{m}_{\tilde{\rm L}}^2 \tilde{\mathbf{L}}^{\beta} - \tilde{\mathbf{E}} \mathbf{m}_{\tilde{\rm E}}^2 \tilde{\mathbf{E}}^{\dagger} - \tilde{\mathbf{N}} \mathbf{m}_{\tilde{\rm N}}^2 \tilde{\mathbf{N}}^{\dagger} \\
& - m_{\rm H_d}^2 \delta_{\alpha\beta} H_{\rm d}^{\alpha *} H_{\rm d}^{\beta} - m_{\rm H_u}^2 \delta_{\alpha\beta} H_{\rm u}^{\alpha *} H_{\rm u}^{\beta} \\
& + \left(-B \epsilon_{\alpha\beta} H_{\rm d}^{\alpha} H_{\rm u}^{\beta} - \frac{1}{2} \tilde{\mathbf{N}} \mathbf{B}_{\tilde{\rm N}} \tilde{\mathbf{N}} \;+\; {\rm c.\,c.} \right) \\
& + \left(-\epsilon_{\alpha\beta} H_{\rm d}^{\alpha} \tilde{\mathbf{E}} \mathbf{A}_{\rm E} \tilde{\mathbf{L}}^{\beta} - \epsilon_{\alpha\beta} H_{\rm u}^{\alpha} \tilde{\mathbf{N}} \mathbf{A}_{\rm N} \tilde{\mathbf{L}}^{\beta} \;+\; {\rm c.\,c.} \right) \\
& + \left(-\frac{1}{2} M_1 \tilde{B} \tilde{B} - \frac{1}{2} M_2 \tilde{W}^a \tilde{W}^a \;+\; {\rm c.\,c.} \right)
\end{aligned} \tag{5.34}$$

(see appendix C for the notational details). The CMSSM assumes universal soft supersymmetry breaking parameters at the supersymmetry breaking scale, which we take to be of order the GUT scale, leading to the following GUT relations:

$$\mathbf{m}_{\tilde{\rm L}}^2 = \mathbf{m}_{\tilde{\rm E}}^2 = \mathbf{m}_{\tilde{\rm N}}^2 = m_0^2 \cdot \mathbf{I}, \tag{5.35}$$

$$m_{\rm H_d}^2 = m_{\rm H_u}^2 = m_0^2, \tag{5.36}$$

$$\mathbf{A}_{\rm E} = \mathbf{A}_{\rm N} = 0, \tag{5.37}$$

$$M_1 = M_2 = m_{1/2} \tag{5.38}$$

where m_0 and $m_{1/2}$ denote the universal scalar mass and the universal gaugino mass respectively ($\mathbf{I}$ is the 3×3 unit matrix). We conservatively assume that the trilinear terms $\mathbf{A}_{\rm E}$ and $\mathbf{A}_{\rm N}$ vanish at the supersymmetry breaking scale.

We run the parameters of the CMSSM using the renormalization group equations (see appendix C) working in a basis where the Majorana neutrino singlet matrix is diagonal, integrating

out each heavy neutrino singlet at its associated scale. After integrating down to the electroweak scale, we rotate the Yukawa couplings to the mass eigenbasis. In order to understand the origin of flavour violation in this model class, we first give a qualitative estimate. The leading log approximation of the off-diagonal slepton mass term is given by

$$\left(\Delta \mathbf{m}_{\tilde{\mathrm{L}}}^2\right)_{ij} \approx -\frac{3}{8\pi^2} m_0^2 (\mathbf{Y}_\nu{}^\dagger \mathbf{Y}_\nu) \ln\left(\frac{M_{\mathrm{GUT}}}{\Lambda_{\mathrm{N}}}\right), \tag{5.39}$$

(assuming that the trilinears vanish at the GUT scale), and using this approximation together with mass insertion techniques [39, 41], the branching ratio for $\mu \to e\gamma$ is

$$\begin{aligned} \mathrm{BR}(\mu \to e\gamma) &\sim \frac{\alpha^3}{G_{\mathrm{F}}^2} \frac{\left(\left(\mathbf{m}_{\tilde{\mathrm{L}}}^2\right)_{12}\right)^2}{m_{\mathrm{s}}^8} \tan^2\beta \\ &\approx \frac{\alpha^3}{G_{\mathrm{F}}^2 m_{\mathrm{s}}^8} \left|\frac{3}{8\pi^2} m_0^2 \ln\frac{M_{\mathrm{GUT}}}{\Lambda_{\mathrm{N}}}\right|^2 \left|\left(\mathbf{Y}_\nu{}^\dagger \mathbf{Y}_\nu\right)_{12}\right|^2 \tan^2\beta \end{aligned} \tag{5.40}$$

where m_s is a typical sparticle mass. We see that since the flavour structure of the AB model class is specified so precisely, the branching ratio for $\mu \to e\gamma$ is well determined. In our calculation of the decay rate, we use the full one-loop expressions derived from the diagrams in figure 5.2 (see appendix C for more details).

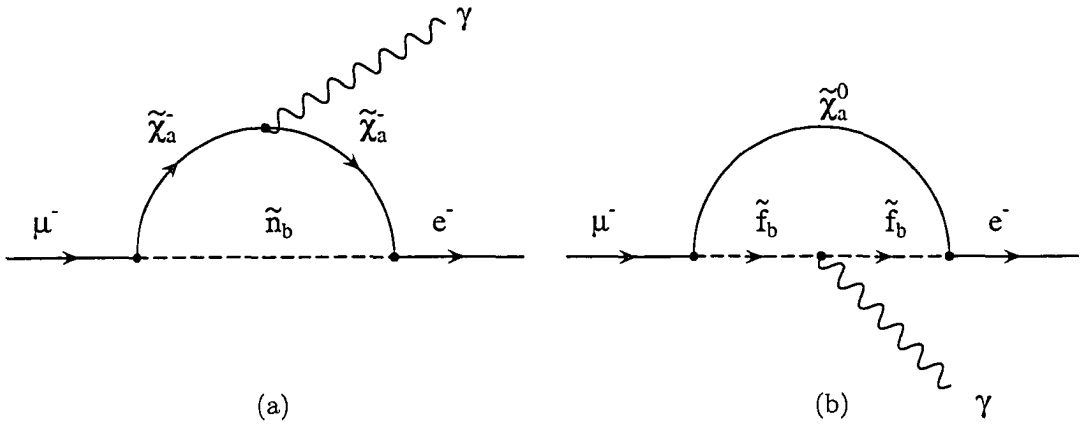

Figure 5.2: Feynman diagrams contributing to $\mu \to e\gamma$.

The WMAP satellite observations [25, 26] combined with constraints from $b \to s\gamma$ and LEP direct searches [42] strongly limit the available CMSSM parameter space if the LSP composes the dark matter [27, 28, 29, 30, 31, 32, 33, 34]. In addition to these constraints, realistic supersymmetric GUT models must also survive lepton flavour violation bounds, such as the limit on $\mu \to e\gamma$. In particular, using all of the available bounds, both cosmological and laboratory, we can further restrict the AB model class.

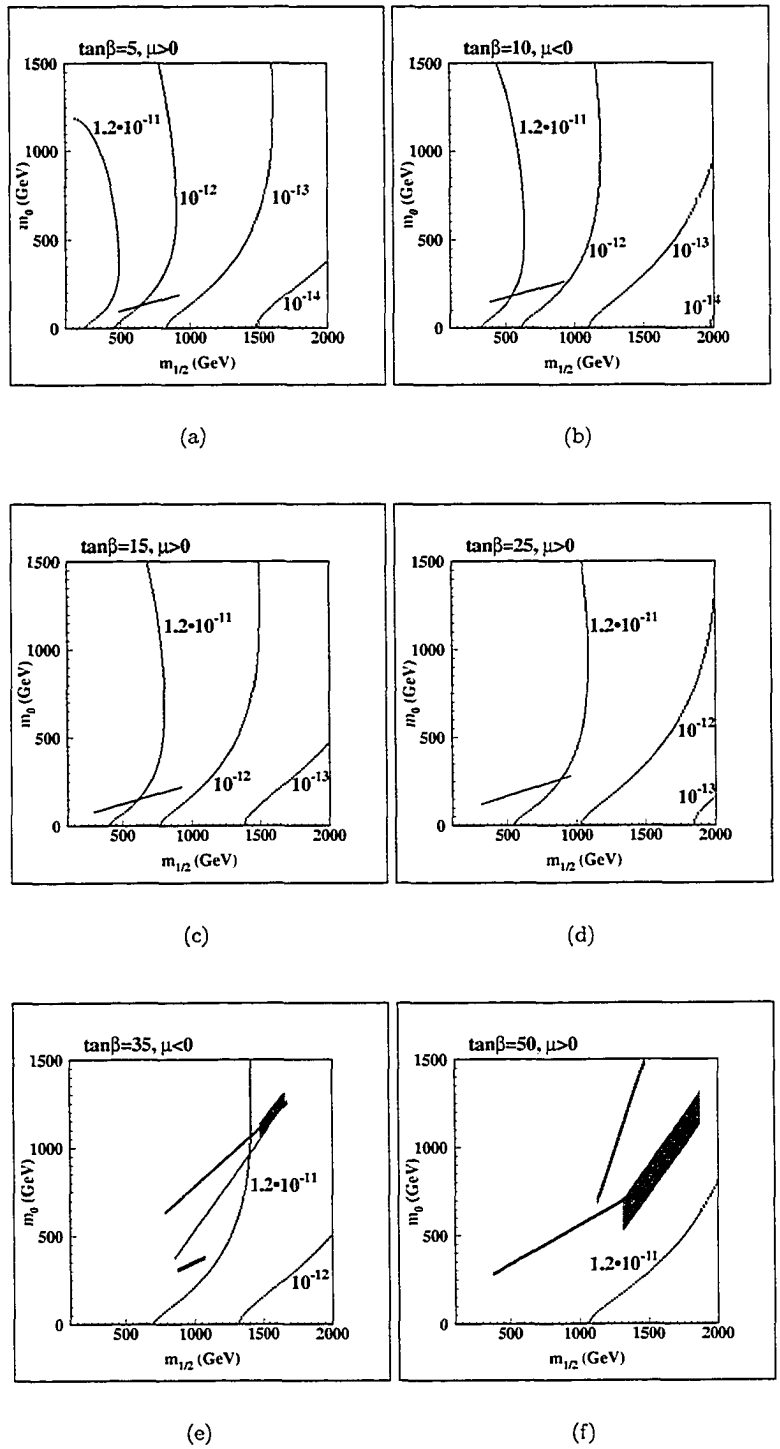

Figure 5.3: Contour plots of BR($\mu \to e\gamma$) in the $m_0 - m_{1/2}$ plane: Panels (a),(c),(d), and (f) show the contours of the branching ratio for $\tan\beta = 5, 15, 25, 50$ respectively with $\mu > 0$. Panels (b) and (e) show the contours with $\tan\beta = 10, 35$ respectively with $\mu < 0$. In all cases, the shaded region corresponds to the approximate combined WMAP and laboratory constraints.

In figure 5.3, we show contours of the branching ratio $\mu \to e\gamma$ in the $m_{1/2}$-m_0 plane for a variety of $\tan\beta$ with the μ parameter both positive and negative. The parameters of the AB model class have been chosen such that all the low energy predictions fit the standard model data, and we have chosen $a = 1$ and $b = 2$ for the Majorana singlet neutrino mass matrix given in (5.29). As indicated in [16], there are a number of possible model choices for the Majorana singlet parameters a and b that are consistent with the LMA solution. However, we find that the rate for $\mu \to e\gamma$ is largely unaffected by the allowed range [16] for these parameters. Panel (a) demonstrates the lepton flavour bounds for $\tan\beta = 5$ with $\mu > 0$. The small line-like shaded area in the lower part of the panel is the allowed region from the combined WMAP and laboratory limits. The remaining panels show that the contours of constant branching ratio migrate to the right of the plots (that is, to high values of $m_{1/2}$ and m_0) as $\tan\beta$ is increased. In each case, we overlay the approximate WMAP and laboratory constraint bounds represented by a shaded region [27]. The choice for the sign of μ is indicated in each panel. As $\tan\beta$ is pushed up, larger portions of the parameter space become excluded. This is an expected feature since the branching ratio is proportional to $\tan^2\beta$. Notice that by $\tan\beta \sim 25$, $\mu > 0$, the branching ratio allowed contours no longer have a significant overlap with the WMAP region. As a result, we find that the AB model class is consistent with the current experimental bound on $\mu \to e\gamma$ for low $\tan\beta$ (that is, $\tan\beta \lesssim 20$) for $\mu > 0$. For completeness, in panels (b) and (e), we show two cases where $\mu < 0$. The branching ratio of $\mu \to e\gamma$ is largely insensitive to the sign of μ, however the WMAP region is moderately affected [28]. A small part of the allowed WMAP region is currently permitted for larger $\tan\beta$ (that is, ~ 35) as indicated in panel (e). The upcoming limits [36] that MEG will establish, $\mathrm{BR}(\mu \to e\gamma) \lesssim 5 \times 10^{-14}$, will effectively rule out this model class if lepton flavour violation is not seen. Interestingly, if lepton flavour violation is seen at MEG, this model will suggest that $\tan\beta$ is low based on flavour bounds alone.

5.3 Conclusions

The AB model class [14, 15, 16], based on a $\mathrm{U}(1)\times\mathrm{Z}_2\times\mathrm{Z}_2$ flavour symmetry, is a highly successful and predictive GUT scenario. This model class has the ability to accommodate all the observed neutrino phenomena and reproduce the low energy physics of the standard model. If it is assumed that supersymmetry is broken via mSUGRA and that the GUT breaks directly to the CMSSM, the AB model class is highly restrictive and hence allows for a precise determination for the rate of charged lepton flavour violation. In particular, we examined the process $\mu \to e\gamma$, since at the present time this flavour violating muon decay channel gives the strongest constraints on flavour changing neutral currents in the lepton sector.

As the WMAP satellite data [25, 26] and laboratory direct searches [42] have already severely restricted the available CMSSM parameter space, the $\mu \to e\gamma$ flavour bounds allow a strong test of the AB model class. We find that given the current bounds [35] on $\mu \to e\gamma$, $\mathrm{BR}(\mu \to e\gamma) < 1.2 \times 10^{-11}$, the AB model class favours low $\tan\beta$ (that is, $\lesssim 20$) with $\mu > 0$, however, there is a

small region that is not excluded for $\tan\beta \lesssim 35$ with the sign of μ negative. If MEG [36] does not detect a positive lepton flavour violation signal, $\mathrm{BR}(\mu \to e\gamma) \lesssim 5\times10^{-14}$, the AB model class will be effectively ruled out, given our conservative assumptions concerning GUT and supersymmetry breaking. It remains an open question as to whether or not other supersymmetry and/or GUT breaking schemes within the AB model class will be able to avoid these flavour violating bounds.

Bibliography

[1] Ernest Jankowski and David W. Maybury. Lepton flavour violation in a class of lopsided SO(10) models. *Phys. Rev.*, D70:035004, 2004.

[2] R. Davis. Nobel lecture: A half-century with solar neutrinos. *Rev. Mod. Phys.*, 75:985–994, 2003.

[3] Dzh. N. Abdurashitov et al. Results from SAGE. *Phys. Lett.*, B328:234–248, 1994.

[4] P. Anselmann et al. Solar neutrinos observed by GALLEX at Gran Sasso. *Phys. Lett.*, B285:376–389, 1992.

[5] M. C. Gonzalez-Garcia and Yosef Nir. Developments in neutrino physics. *Rev. Mod. Phys.*, 75:345–402, 2003.

[6] S. N. Ahmed et al. Measurement of the total active B-8 solar neutrino flux at the Sudbury Neutrino Observatory with enhanced neutral current sensitivity. *Phys. Rev. Lett.*, 92:181301, 2004.

[7] Z. Maki, M. Nakagawa, and S. Sakata. Remarks on the unified model of elementary particles. *Prog. Theor. Phys.*, 28:870, 1962.

[8] M. Kobayashi and T. Maskawa. Cp violation in the renormalizable theory of weak interaction. *Prog. Theor. Phys.*, 49:652–657, 1973.

[9] C. Weinheimer et al. High precision measurement of the tritium beta spectrum near its endpoint and upper limit on the neutrino mass. *Phys. Lett.*, B460:219–226, 1999.

[10] K. Assamagan et al. Upper limit of the muon-neutrino mass and charged pion mass from momentum analysis of a surface muon beam. *Phys. Rev.*, D53:6065–6077, 1996.

[11] R. Barate et al. An upper limit on the tau neutrino mass from three- and five-prong tau decays. *Eur. Phys. J.*, C2:395–406, 1998.

[12] Pierre Ramond. *Journeys Beyond the Standard Model.* Reading, Mass., Perseus Books, 1999.

[13] S. Eidelman et al. Review of particle physics. *Phys. Lett.*, B592:1, 2004.

[14] Carl H. Albright and Stephen M. Barr. Construction of a minimal higgs SO(10) SUSY GUT model. *Phys. Rev.*, D62:093008, 2000.

[15] Carl H. Albright and Stephen M. Barr. Realization of the large mixing angle solar neutrino solution in an SO(10) supersymmetric grand unified model. *Phys. Rev.*, D64:073010, 2001.

[16] Carl H. Albright and S. Geer. Neutrino superbeam and factory tests of grand unified model predictions for the large mixing angle and LOW solar neutrino solutions. *Phys. Lett.*, B532:311–317, 2002.

[17] K. S. Babu, Jogesh C. Pati, and Frank Wilczek. Fermion masses, neutrino oscillations, and proton decay in the light of SuperKamiokande. *Nucl. Phys.*, B566:33–91, 2000.

[18] Zurab Berezhiani and Anna Rossi. Predictive grand unified textures for quark and neutrino masses and mixings. *Nucl. Phys.*, B594:113–168, 2001.

[19] T. Blazek, S. Raby, and K. Tobe. Neutrino oscillations in an SO(10) SUSY GUT with U(2) x U(1)**n family symmetry. *Phys. Rev.*, D62:055001, 2000.

[20] W. Buchmuller and D. Wyler. CP violation, neutrino mixing and the baryon asymmetry. *Phys. Lett.*, B521:291–298, 2001.

[21] Mu-Chun Chen and K. T. Mahanthappa. CP violation in a supersymmetric SO(10) x U(2)F model. *Phys. Rev.*, D65:053010, 2002.

[22] Ryuichiro Kitano and Yukihiro Mimura. Large angle MSW solution in grand unified theories with SU(3) x U(1) horizontal symmetry. *Phys. Rev.*, D63:016008, 2001.

[23] Nobuhiro Maekawa. Neutrino masses, anomalous U(1) gauge symmetry and doublet- triplet splitting. *Prog. Theor. Phys.*, 106:401–418, 2001.

[24] Graham G. Ross and L. Velasco-Sevilla. Symmetries and fermion masses. *Nucl. Phys.*, B653:3–26, 2003.

[25] C. L. Bennett et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Preliminary maps and basic results. *Astrophys. J. Suppl.*, 148:1, 2003.

[26] D. N. Spergel et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Determination of cosmological parameters. *Astrophys. J. Suppl.*, 148:175, 2003.

[27] John R. Ellis, Keith A. Olive, Yudi Santoso, and Vassilis C. Spanos. Supersymmetric dark matter in light of WMAP. *Phys. Lett.*, B565:176–182, 2003.

[28] M. Battaglia et al. Updated post-WMAP benchmarks for supersymmetry. *Eur. Phys. J.*, C33:273–296, 2004.

[29] John R. Ellis, Keith A. Olive, Yudi Santoso, and Vassilis C. Spanos. Likelihood analysis of the CMSSM parameter space. *Phys. Rev.*, D69:095004, 2004.

[30] Howard Baer and Csaba Balazs. Chi**2 analysis of the minimal supergravity model including WMAP, g(mu)-2 and b → s gamma constraints. *JCAP*, 0305:006, 2003.

[31] Howard Baer, Csaba Balazs, Alexander Belyaev, Tadas Krupovnickas, and Xerxes Tata. Updated reach of the CERN LHC and constraints from relic density, b → s gamma and a(mu) in the mSUGRA model. *JHEP*, 06:054, 2003.

[32] A. B. Lahanas and D. V. Nanopoulos. WMAPing out supersymmetric dark matter and phenomenology. *Phys. Lett.*, B568:55–62, 2003.

[33] Utpal Chattopadhyay, Achille Corsetti, and Pran Nath. WMAP constraints, SUSY dark matter and implications for the direct detection of SUSY. *Phys. Rev.*, D68:035005, 2003.

[34] R. Arnowitt, Bhaskar Dutta, and B. Hu. Dark matter, muon g-2 and other SUSY constraints. 2003.

[35] M. L. Brooks et al. New limit for the family-number non-conserving decay mu+ - ¿ e+ gamma. *Phys. Rev. Lett.*, 83:1521–1524, 1999.

[36] T. Mori. Status and future of mu → e gamma: The PSI experiment. *Nucl. Phys. Proc. Suppl.*, 111:194–199, 2002.

[37] M. Fukugita and T. Yanagida. *Physics of Neutrinos and Applications to Astrophysics.* Springer, Berlin, Germany, 2003.

[38] Riccardo Barbieri, Lawrence J. Hall, and Alessandro Strumia. Violations of lepton flavor and CP in supersymmetric unified theories. *Nucl. Phys.*, B445:219–251, 1995.

[39] J. Hisano and Daisuke Nomura. Solar and atmospheric neutrino oscillations and lepton flavor violation in supersymmetric models with the right- handed neutrinos. *Phys. Rev.*, D59:116005, 1999.

[40] Francesca Borzumati and Antonio Masiero. Large muon and electron number violations in supergravity theories. *Phys. Rev. Lett.*, 57:961, 1986.

[41] J. Hisano, T. Moroi, K. Tobe, and Masahiro Yamaguchi. Lepton-flavor violation via right-handed neutrino Yukawa couplings in supersymmetric standard model. *Phys. Rev.*, D53:2442–2459, 1996.

[42] K. Hagiwara et al. Review of particle physics. *Phys. Rev.*, D66:010001, 2002.

Chapter 6

QED Radiative Suppression of $\mu \longrightarrow e + \gamma$ Branching Ratio

The only observed decay channel of the muon is $\mu^- \rightarrow e^- \bar{\nu}_e \nu_\mu$ (with possible photon or electron-positron pair emission). However, since the discovery of the muon more than half a century ago, searches have been undertaken for the decay $\mu \rightarrow e\gamma$. Initially, when the muon was thought to be an excited state of the electron, this was expected to be its dominant decay channel. It was soon realized that it is very strongly suppressed (the early experiments are summarized in [2]). When an intermediate boson was proposed to explain the mechanism of weak interactions [3], the absence of $\mu \rightarrow e\gamma$ led to the hypothesis that the two neutrinos in the muon decay (figure 6.1(a)) have different flavours so that the interaction shown in figure 6.1(b) cannot occur [4, 5]. The existence of the muon neutrino, distinct from the electron one, was demonstrated in the

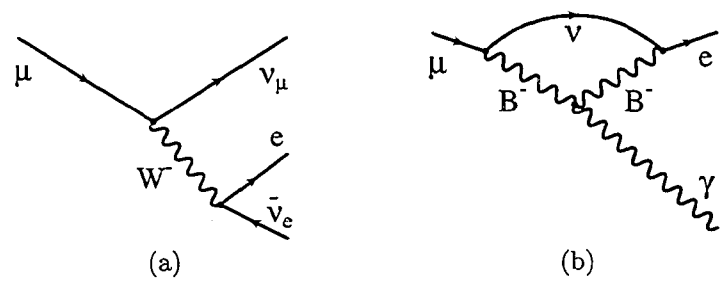

Figure 6.1: (a) Ordinary muon decay; (b) The puzzle of $\mu \rightarrow e\gamma$ absence in the early models with an intermediate vector boson.

classic 1962 experiment in Brookhaven [6]. In this way, the limits placed on the branching ratio

for $\mu \to e\gamma$ helped establish the concept of families or generations of fermions, which became one of the cornerstones of the standard model.

In fact, the standard model with massless neutrinos strictly forbids the lepton-flavour non-conserving transitions like $\mu \to e\gamma$. Even if the neutrinos have a small mass, the rate is still very small, $\mathcal{O}\left((m_\nu/m_W)^4\right)$ [7, 8, 9, 10]. However, most extensions of the standard model, containing some new physics at the hitherto unexplored mass scales, predict a higher rate of $\mu \to e\gamma$. For example, in supersymmetry (SUSY) neutrinos have heavy "partners", scalar sneutrinos, whose mixing could generate $\mu \to e\gamma$ transitions through the interaction with charginos $\tilde{\chi}^{\pm}$, as shown in figure 5.2(a). Scalar partners of the charged leptons, interacting with neutralinos $\tilde{\chi}^0$, could also contribute to this decay (figure 5.2(b)).

Explicit supersymmetric grand unified models [11, 12, 13, 14, 15] predict a $\mu \to e\gamma$ rate just below the present 90% CL upper bound from the MEGA experiment, [16],

$$\frac{\Gamma(\mu \to e\gamma)}{\Gamma(\mu \to e\bar{\nu}_e\nu_\mu)} < 1.2 \times 10^{-11}. \tag{6.1}$$

In the near future, a new search for $\mu \to e\gamma$ will be undertaken at the Paul Scherrer Institute (PSI) [17], with a single event sensitivity corresponding to the branching ratio of 2×10^{-14}. In view of the SUSY GUT predictions, it is not inconceivable that this experiment will find of order of 100 $\mu \to e\gamma$ decay events. At such rate, precision studies of lepton-number violating interactions will become possible. It is therefore interesting to theoretically evaluate model-independent electromagnetic effects which turn out to decrease the rate of $\mu \to e\gamma$ by several percent.

6.1 QED suppression of the dipole operators

The effective Lagrangian which gives rise to $\mu \to e\gamma$ has the form

$$\overline{e}\,\sigma^{\mu\nu}(f_M + f_E\gamma_5)\,\mu F_{\mu\nu}, \tag{6.2}$$

where f_i $(i = M, E)$ are form-factors, calculable in explicit models of physics beyond the standard model and dependent on the parameters of those models (see (C.45) for an example); e, μ are the Dirac fields for electron and muon respectively; $\sigma^{\mu\nu} = i\left(\gamma^\mu\gamma^\nu - \gamma^\nu\gamma^\mu\right)/2$; and $F_{\mu\nu}$ is the electromagnetic field tensor. In terms of f_i, the tree-level decay rate $\Gamma^{(0)}$ $(\mu \to e\gamma)$ is

$$\Gamma^{(0)}(\mu \to e\gamma) = \frac{m_\mu^3}{8\pi}\left(|f_M|^2 + |f_E|^2\right). \tag{6.3}$$

It is well known that the chirality-flipping electric and magnetic dipole operators in (6.2) have (the same) large QED anomalous dimension. It was first computed in the context of hadron decays in QCD [18, 19, 20, 21], and plays an important role in various electromagnetic processes like the radiative decay $b \to s\gamma$ [22] or the muon anomalous magnetic moment [23, 24, 25] (see

also [26]).

We denote the coefficient of the dipole-transition operators in (6.2), computed in a full theory violating lepton flavour, by $f_i(\Lambda)$, where Λ is a characteristic mass scale of the relevant new physics. For example, in SUSY, $f_i(\Lambda)$ would result from the one-loop diagrams in figure 5.2, and Λ would be the characteristic mass of the superpartners. If we now consider an effective theory at an energy of the order of the muon mass, the heavy exotic fields are not dynamical degrees of freedom and we can consider the effects of figure 5.2 as point-like interactions given by the Lagrangian (6.2), figure 6.2(a).

However, when we consider higher-order electromagnetic corrections to this interaction, such as the one shown in figure 6.2(b), we find that they are logarithmically divergent in the ultraviolet (UV). This is not surprising, since the dimension of the operators in (6.2) is 5, which signals non-

Figure 6.2: (a) The effective interaction that gives rise to $\mu \to e\gamma$. (b) An example of an electromagnetic correction which contributes to the suppression of the $\mu \to e\gamma$ decay rate.

renormalizability. An explicit calculation shows that the effect of those corrections amounts to

$$f_i(\Lambda) \to f_i(\Lambda)\left(1 - \frac{4\alpha}{\pi}\ln\frac{\Lambda}{m_\mu} + \mathcal{O}(\alpha)\right), \tag{6.4}$$

where we have taken the UV cut-off to be equal to Λ, since around that magnitude of the loop momentum, it is no longer justified to treat the flavour-changing vertex as point-like. The interaction is weakened; we can denote its effective strength at the muon mass scale by $f_i(m_\mu)$, which includes the leading logarithmic effect,

$$f_i(m_\mu) = f_i(\Lambda)\left(1 - \frac{4\alpha}{\pi}\ln\frac{\Lambda}{m_\mu}\right). \tag{6.5}$$

This effect can be quite large, since the rate (6.3) of the decay is proportional to the sum of squares of f_i,

$$\Gamma(\mu \to e\gamma) \simeq \left(1 - \frac{8\alpha}{\pi}\ln\frac{\Lambda}{m_\mu}\right)\Gamma^{(0)}(\mu \to e\gamma). \tag{6.6}$$

If Λ is of order 250 GeV, which is a typical SUSY mass scale in the models considered in [11], this corresponds to about a 14% decrease of the rate.

It is possible to sum up the leading-logarithmic effects to all orders in $\alpha^n \ln^n \Lambda/m_\mu$ (see, for example, [27, 28]). In the absence of mixing with other lepton-flavour non-conserving operators, the scale dependence of the coefficients f_i can be expressed in an iterative form,

$$f_i(m_<) = f_i(m_>) \cdot \left(\frac{\alpha(m_<)}{\alpha(m_>)}\right)^{\gamma/b}, \tag{6.7}$$

where in our case the anomalous dimension is $\gamma = -8$ and b is determined using the charges Q_j of all particles contributing to the running of the fine structure constant between the scales $m_<$ and $m_>$:

$$b = -\frac{4}{3}\sum_j Q_j^2. \tag{6.8}$$

The explicit result for $f_i(m_\mu)$ depends on the mass spectrum of a concrete new physics scenario. However, higher order leading-logarithmic effects are not expected to significantly change the magnitude of the $\mu \to e\gamma$ rate decrease given in (6.6), because of cancellation between the running of the fine structure constant and the effects of higher orders in the anomalous dimension. Similar cancellation was observed in the muon $g-2$ calculation [25].

Typical lepton-flavour violating amplitudes, like the ones in figure 5.2, contain two new physics masses, which in general may be quite different. One can ask the question, what should be taken as the argument Λ of the logarithm in (6.6). As long as the ratio of the two large scales is small compared to their size relative to the muon mass, this is an issue of non-leading corrections, which we have been neglecting. In the case of $\mu \to e\gamma$ induced by the small neutrino masses (where the rate is extremely small, as discussed above), the scale $\Lambda = m_W$ in (6.6) is the larger of the two masses in the loop. The inverse of m_W determines the size of the effective interaction range.

6.2 Four-fermion operators

New physics effects can also induce lepton-flavour violating four-fermion operators such as $(\bar{e}\Gamma\mu)(\bar{f}\Gamma f)$ (figure 6.3(a)). They contribute to $\mu \to e\gamma$ through loop effects (figure 6.3(b,c)) in the same order in $\frac{\alpha}{\pi}\ln\frac{\Lambda}{m_\mu}$ as the suppression effect in (6.6).

In theories such as R-parity conserving SUSY, four-fermion contributions are suppressed relative to the dipole operators (figure 5.2) by two powers of a coupling constant and are not expected to contribute significantly to $\mu \to e\gamma$. It is, however, interesting to see to what extent we can estimate such contributions in a model-independent way.

Virtual fermions f other than muon or electron contribute only through "closed" loops, as shown in figure 6.3(c). Large logarithms arising from such diagrams cancel at least partially in

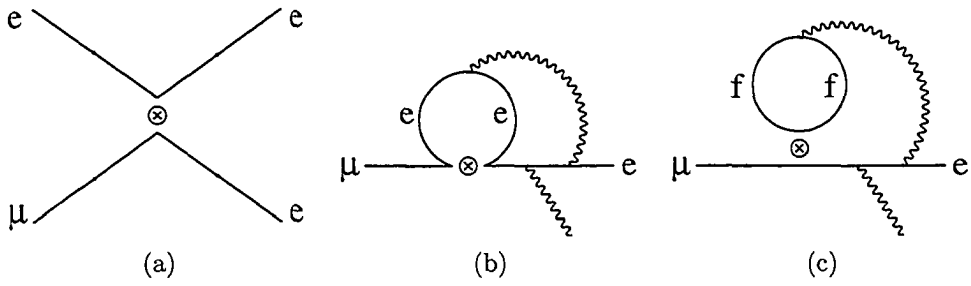

Figure 6.3: (a) Lepton flavour-violating four-fermion operator; (b) Example of a contribution to $\mu \to e\gamma$ for $f = e$ or μ; (c) Example of other fermions' contribution.

anomaly-free theories, and we will neglect the contributions shown in figure 6.3(c).

Here we will consider a specific example of the operator

$$\mathcal{O}_x = G_x(\bar{e}\gamma^\nu L\mu)(\bar{e}\gamma_\nu Le), \qquad L \equiv \frac{1-\gamma_5}{2}, \tag{6.9}$$

whose anomalous dimension and mixings with other flavour-violating operators can be found using well-known results found in studies of the radiative quark decay $b \to s\gamma$. We will demonstrate that the bound on G_x obtained from searches for $\mu \to eee$ renders the contribution of this operator to $\mu \to e\gamma$ negligible. We may expect that contributions of other Dirac structures and of operators $(\bar{e}\Gamma\mu)(\bar{\mu}\Gamma\mu)$ have similar magnitudes.

Operator $\mathcal{O}_x$ induces the decay $\mu \to eee$ with a rate

$$\Gamma(\mu \to eee) = \frac{G_x^2 m_\mu^5}{768\pi^3}, \tag{6.10}$$

and we can use the bound on the branching ratio [29],

$$\frac{\Gamma(\mu \to eee)}{\Gamma(\mu \to e\nu\nu)} \simeq \frac{\Gamma(\mu \to eee)}{\Gamma(\mu \to \text{total})} < 10^{-12}, \tag{6.11}$$

to constrain G_x. We find

$$G_x < 2 \times 10^{-6} G_F, \tag{6.12}$$

(G_F is the Fermi constant [29]).

In order to find the contribution of $\mathcal{O}_x$ to the amplitude $\mu \to e\gamma$ we consider its mixing with the dipole operators in equation 6.2. We write the result as

$$g_x \bar{e}\,\sigma^{\mu\nu}(1+\gamma_5)\,\mu F_{\mu\nu}, \tag{6.13}$$

with

$$g_X = \frac{e m_\mu G_X}{16\pi^2} \frac{29}{18} \frac{\alpha}{\pi} \ln \frac{\Lambda}{m_\mu}, \tag{6.14}$$

where $e = \sqrt{4\pi\alpha} \simeq 0.3$. Finally, we would like to compare the effect of this four-fermion operator on the form-factors f_i $(i = E, M)$ with the effect of the QED correction in equation 6.4. For this purpose, we assume $f_E = f_M$ and consider the ratio R defined as

$$R = \frac{g_X}{f_i \frac{4\alpha}{\pi} \ln \frac{\Lambda}{m_\mu}}. \tag{6.15}$$

It follows that

$$R < e \frac{29\sqrt{3}}{8\pi \cdot 18} \cdot 10^{-6} \cdot \frac{1}{\sqrt{\mathrm{BR}(\mu \to e\gamma)}}, \tag{6.16}$$

where we have taken $f_i = \frac{G_F m_\mu}{4\sqrt{3}\pi} \sqrt{\mathrm{BR}(\mu \to e\gamma)}$ and used the bound 6.12. If $\mu \to e\gamma$ is discovered with a branching ratio between 10^{-11} and 10^{-14}, the upper bound on the ratio R of the four-fermion and dipole radiative effects will be between about 10^{-2} and 0.3.

The QED corrections we considered in this chapter will be relevant for the upcoming PSI experiment if it observes a fair number (of the order of a hundred or more) of decay events $\mu \to e\gamma$. This corresponds to the branching ratio of at least 10^{-12}, for which the ratio R is about 0.03. We conclude that the effects of the four-fermion operators are likely to be negligible for the next generation of the $\mu \to e\gamma$ searches.

6.3 Conclusions

The logarithmic suppression which we have discussed in section 6.1 affects not only $\mu \to e\gamma$ but also other lepton-flavour violating processes occurring via the dipole transition of the type (6.2). For example, the rates of the τ-lepton decays $\tau \to \mu\gamma$ and $\tau \to e\gamma$ are decreased by

$$1 - \frac{8\alpha}{\pi} \ln \frac{\Lambda}{m_\tau}, \tag{6.17}$$

which is between about 7.5% and 12% for Λ between 100 and 1000 GeV. On the other hand, the decays of the type $\mu^+ \to e^+e^+e^-$ and muon-electron conversion in the nuclear field, $\mu^- N \to e^- N$, can occur via a more general interaction, including monopole form-factors, which do not receive such logarithmic corrections.

To summarize, we have pointed out an electromagnetic short-distance effect which decreases the predicted rate of the lepton-flavour violating decay $\mu \to e\gamma$ by a factor $\left(1 - \frac{8\alpha}{\pi} \ln \frac{\Lambda}{m_\mu}\right)$, or between 12% and 17% for the new physics scale Λ between 100 and 1000 GeV. If the lepton-flavour non-conservation is observed by the next generation of experiments, the $\mu \to e\gamma$ search at

the Paul Scherrer Institute and the conversion $\mu^- N \rightarrow e^- N$ search by the MECO Collaboration [30] in Brookhaven, this correction will help disentangle the underlying new physics structure.

Bibliography

[1] Andrzej Czarnecki and Ernest Jankowski. Electromagnetic suppression of the decay mu → e gamma. *Phys. Rev.*, D65:113004, 2002.

[2] R. R. Crittenden, W. D. Walker, and J. Ballam. *Phys. Rev.*, 121:1823, 1961.

[3] H. Yukawa. *Rev. Mod. Phys.*, 21:474, 1949.

[4] T. D. Lee. in proc. of the 1960 Annual International Conference on High Energy Physics at Rochester. page 567, 1960. Interscience Publishers, New York, 1960.

[5] T. D. Lee and Chen-Ning Yang. Implications of the intermediate boson basis of the weak interactions: Existence of a quartet of intermediate bosons and their dual isotopic spin transformation properties. *Phys. Rev.*, 119:1410–1419, 1960.

[6] G. Danby et al. Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos. *Phys. Rev. Lett.*, 9:36–44, 1962.

[7] S. T. Petcov. The processes mu → e gamma, mu → e e anti-e, neutrino' → neutrino gamma in the Weinberg-Salam model with neutrino mixing. *Sov. J. Nucl. Phys.*, 25:340, 1977.

[8] W. J. Marciano and A. I. Sanda. Exotic decays of the muon and heavy leptons in gauge theories. *Phys. Lett.*, B67:303, 1977.

[9] B. W. Lee, S. Pakvasa, R. E. Shrock, and H. Sugawara. Muon and electron number nonconservation in a v-a six quark model. *Phys. Rev. Lett.*, 38:937, 1977.

[10] Benjamin W. Lee and Robert E. Shrock. Natural suppression of symmetry violation in gauge theories: Muon - lepton and electron lepton number nonconservation. *Phys. Rev.*, D16:1444, 1977.

[11] Riccardo Barbieri, Lawrence J. Hall, and Alessandro Strumia. Violations of lepton flavor and CP in supersymmetric unified theories. *Nucl. Phys.*, B445:219–251, 1995.

[12] Riccardo Barbieri and L. J. Hall. Signals for supersymmetric unification. *Phys. Lett.*, B338:212–218, 1994.

[13] Yoshitaka Kuno and Yasuhiro Okada. Muon decay and physics beyond the standard model. *Rev. Mod. Phys.*, 73:151–202, 2001.

[14] J. Hisano, Daisuke Nomura, Yasuhiro Okada, Yasuhiro Shimizu, and Minoru Tanaka. Enhancement of mu → e gamma in the supersymmetric SU(5) GUT at large tan(beta). *Phys. Rev.*, D58:116010, 1998.

[15] John R. Ellis, M. E. Gomez, G. K. Leontaris, S. Lola, and D. V. Nanopoulos. Charged lepton flavour violation in the light of the Super- Kamiokande data. *Eur. Phys. J.*, C14:319–334, 2000.

[16] M. L. Brooks et al. New limit for the family-number non-conserving decay mu+ - ¿ e+ gamma. *Phys. Rev. Lett.*, 83:1521–1524, 1999.

[17] T. Mori. Status and future of mu → e gamma: The PSI experiment. *Nucl. Phys. Proc. Suppl.*, 111:194–199, 2002.

[18] R. K. Ellis. The delta I = 1/2 rule with heavy quarks and righthanded currents. *Nucl. Phys.*, B108:239, 1976.

[19] Frank Wilczek and A. Zee. Delta I = 1/2 rule and right-handed currents: Heavy quark expansion and limitation on Zweig's rule. *Phys. Rev.*, D15:2660, 1977.

[20] A. I. Vainshtein, Valentin I. Zakharov, and Mikhail A. Shifman. Right-handed currents and sum rule delta t = 1/2 in nonleptonic strange particle decays. *JETP Lett.*, 23:602, 1976.

[21] Mikhail A. Shifman, A. I. Vainshtein, and Valentin I. Zakharov. On the weak radiative decays (effects of strong interactions at short distances). *Phys. Rev.*, D18:2583–2599, 1978.

[22] Andrzej Czarnecki and William J. Marciano. Electroweak radiative corrections to b → s gamma. *Phys. Rev. Lett.*, 81:277–280, 1998.

[23] T. V. Kukhto, E. A. Kuraev, Z. K. Silagadze, and A. Schiller. The dominant two loop electroweak contributions to the anomalous magnetic moment of the muon. *Nucl. Phys.*, B371:567–596, 1992.

[24] Andrzej Czarnecki, Bernd Krause, and William J. Marciano. Electroweak corrections to the muon anomalous magnetic moment. *Phys. Rev. Lett.*, 76:3267–3270, 1996.

[25] G. Degrassi and G. F. Giudice. QED logarithms in the electroweak corrections to the muon anomalous magnetic moment. *Phys. Rev.*, D58:053007, 1998.

[26] Marco Ciuchini, E. Franco, G. Martinelli, and L. Reina. The delta s = 1 effective hamiltonian including next-to- leading order QCD and QED corrections. *Nucl. Phys.*, B415:403–462, 1994.

[27] W. J. Marciano and A. Sirlin. Precise SU(5) predictions for sin**2-theta-w, m(W) and m(Z). *Phys. Rev. Lett.*, 46:163, 1981.

[28] W. J. Marciano and A. Sirlin. Radiative corrections to beta decay and the possibility of a fourth generation. *Phys. Rev. Lett.*, 56:22, 1986.

[29] D. E. Groom et al. Review of particle physics. *Eur. Phys. J.*, C15:1–878, 2000.

[30] http://meco.ps.uci.edu/.

Chapter 7

Conclusions

We discussed Optimal Jet Finder, a tool for analysis of hadronic jets in high energy physics experiments. Final jet configurations are determined by the global energy flow in the event, which significantly distinguishes this jet definition from the cone or binary recombination algorithms. The idea of using the global structure of an event is not new [1], but it involves a non-trivial optimization problem. To the best of our knowledge, Optimal Jet Finder is the first program of this kind working sufficiently fast to be of practical use in data analysis.

The large scale Monte Carlo benchmark test of Optimal Jet Finder, based on the W-boson mass extraction from fully hadronic decays of pairs of W-bosons at the center of mass energy 180 GeV, provides a clear evidence of the accuracy of Optimal Jet Definition. Indeed, Optimal Jet Definition is equivalent to Durham, which has been concluded [2] to be the best algorithm in a similar context.

A high efficiency of Optimal Jet Finder was confirmed. The software is more complex than similar conventional tools for reconstructing jets as it involves a large optimization problem. (However, Optimal Jet Definition is clear, transparent, and definite in contrast to many conventional schemes with their arbitrary prescriptions, where a small implementation detail constitutes a new jet definition.) Thus, it is slower than conventional schemes for a small number of input particles. However, the running time of Optimal Jet Finder scales linearly with the number of input data, making it an ideal tool for analyzing whole calorimeters or their substantial parts without resorting to a preclustering step.

The branching ratio for the see-saw induced $\mu \to e + \gamma$ decay computed in a framework of the Albright-Barr Grand Unification model [3] confirms the consistency of the very successful Albright-Barr scenario with the current experimental limits of the $\mu \to e + \gamma$ branching ratio (1.2×10^{-11}). (Contrary to what was suggested in [4].) We assumed the Constrained Minimal Supersymmetric Extension of the Standard Model with its parameter space restricted by the direct laboratory searches [5] and the recent cosmological observations from Wilkinson Microwave Anisotropy Probe [6, 7]. If our assumptions regarding supersymmetry breaking, and the interpretation of Wilkinson Microwave Anisotropy Probe are valid, and the Grand Unification

model itself is plausible, the $\mu \rightarrow e + \gamma$ decay is very likely to be seen in the current MEG experiment at the Paul Scherrer Institute. If $\mu \rightarrow e + \gamma$ is not seen up to the branching ratio of 5×10^{-14}, our conservative assumptions about the supersymmetry breaking have to be relaxed for the Albright-Barr model to be still valid.

If the $\mu \rightarrow e + \gamma$ decay is indeed observed by the MEG experiment, the evaluated result of QED suppression of the decay rate will assist in interpretation of the experimental data. The suppression result will enhance the precision with which the parameters of new physics models responsible for this lepton flavour violating decay channel can be extracted.

We are nearing very exciting times. The MEG experiment will announce its first results in 2006; the Large Hadron Collider will start operating in 2007; and the Next Linear Collider may be built in the more distant future. We are looking forward to seeing how the methods developed in this thesis will help to interpret the data from the experiments.

Bibliography

[1] S. Youssef. Clustering with local equivalence relations. *Comput. Phys. Commun.*, 45:423–426, 1987.

[2] Stefano Moretti, Leif Lonnblad, and Torbjorn Sjostrand. New and old jet clustering algorithms for electron positron events. *JHEP*, 08:001, 1998.

[3] Carl H. Albright and Stephen M. Barr. Construction of a minimal higgs SO(10) SUSY GUT model. *Phys. Rev.*, D62:093008, 2000.

[4] Xiao-Jun Bi and Yuan-Ben Dai. Lepton flavor violation and its constraints on the neutrino mass models. *Phys. Rev.*, D66:076006, 2002.

[5] K. Hagiwara et al. Review of particle physics. *Phys. Rev.*, D66:010001, 2002.

[6] C. L. Bennett et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Preliminary maps and basic results. *Astrophys. J. Suppl.*, 148:1, 2003.

[7] D. N. Spergel et al. First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Determination of cosmological parameters. *Astrophys. J. Suppl.*, 148:175, 2003.

Appendix A

FORTRAN 77 implementation

A version of this chapter has been published [1].

In this chapter, we describe a FORTRAN 77 implementation of Optimal Jet Definition [2], called Optimal Jet Finder [1]. We discuss the algorithm for minimization of the Ω function and explain the details of the software. This chapter is intended as a documentation for the users of the software. The program is available for download from [3].

A.1 Algorithm for minimizing Ω

The domain of the function $\Omega(\{z_{aj}\})$ is a $(n_{\text{parts}} \times n_{\text{jets}})$-dimensional product of simplices. That is, for a fixed a, the numbers z_{aj}, $j = 1, ..., n_{\text{jets}}$, satisfying conditions 3.5 and 3.6 define an n_{jets}-dimensional simplex. In typical application, $n_{\text{parts}} \sim 200$ (or more) and $n_{\text{jets}} \sim 5$, and therefore Ω is a function of ~ 1000 variables. The algorithm described below allows for efficient minimization of $\Omega(\{z_{aj}\})$.

The algorithm iteratively descends into local minimum of $\Omega(\{z_{aj}\})$ starting from a given initial value of $\{z_{aj}\}$. At each iteration, subsequently for each particle, $\{z_{aj}\}$ is moved into a new position that gives the smaller value of Ω. The iteration loop is terminated when no particle is moved at a single iteration, meaning that the local minimum has been found. (Or some safe number of maximal iterations has been exceeded.)

We describe now in detail how $\{z_{aj}\}$ is moved in a single iteration step for a given particle. Denote $\mathbf{z} \equiv (z_1, z_2, ..., z_{n_{\text{jets}}})$ with $z_j = z_{aj}$ and $\Omega(\mathbf{z}) \equiv \Omega(\{z_{aj}\})$ with fixed a in both definitions. The change in Ω when we change $\mathbf{z}$ to $\mathbf{z} + \tau\mathbf{d}$ can be described locally as

$$\Omega(\mathbf{z} + \tau\mathbf{d}) = \Omega(\mathbf{z}) + \tau\mathbf{f} \cdot \mathbf{d} + O(\tau^2), \tag{A.1}$$

where $\mathbf{f} = (f_1, ..., f_{n_{\text{jets}}})$, $f_j \equiv \partial\Omega(\mathbf{z})/\partial z_j$, $\mathbf{f} \cdot \mathbf{d} = \sum_{j=1}^{n_{\text{jets}}} f_j d_j$, and $\mathbf{d} = (d_1, ..., d_{n_{\text{jets}}})$ describes

some direction. If **z** were not constrained to the simplex, we could take $\mathbf{d} = -\mathbf{f}$ and some $\tau > 0$ to decrease Ω. But choosing τ and **d** we have to ensure that $\mathbf{z}+\tau\mathbf{d}$ is within the simplex. Rewrite

$$\mathbf{f}\cdot\mathbf{d} = \sum_{j=1}^{n_{\text{jets}}} f_j d_j = \sum_{j=1}^{n_{\text{jets}}} \bar{f}_j d_j + \bar{f}_0 d_0 \tag{A.2}$$

with the following definitions

$$\bar{f}_j \equiv f_j - f_J, \quad \bar{f}_0 \equiv -f_J, \tag{A.3}$$

$$d_0 \equiv -\sum_{j=1}^{n_{\text{jets}}} d_j, \tag{A.4}$$

where J is any of $1,...,n_{\text{jets}}$ for which $z_J > 0$ (there always must be such J). Now **d** can be chosen as follows

$$d_j = \begin{cases} \max\left(0, -\bar{f}_j\right) & \text{for all } j = 0,...,n_{\text{jets}}, \text{ for which } z_j = 0 \\ -\bar{f}_j & \text{for all } j = 0,...,n_{\text{jets}},\ j \neq J \text{ for which } z_j > 0 \end{cases}, \tag{A.5}$$

and d_J is chosen so that (A.4) is satisfied. With such choice of **d** and the proper parameter τ the new candidate minimum $\mathbf{z}+\tau\mathbf{d}$ will belong to the simplex and $\Omega(\mathbf{z}+\tau\mathbf{d}) < \Omega(\mathbf{z})$. In the above prescription the choice of J is arbitrary. We found it advantageous to choose J ($z_J > 0$) such that the norm

$$|(\mathbf{d}, d_0)| \equiv \max\{|d_j| :\ j = 0,1,...,n_{\text{jets}}\} \tag{A.6}$$

is maximal. The choice of step length τ is determined by the experimental finding that the minimum tends to be located at the boundary of the simplex. We find

$$\tau = \min\left(\left\{-\frac{z_j}{d_j} :\quad j = 0,...,n_{\text{jets}},\ z_j > 0 \text{ and } d_j < 0\right\}\right) \tag{A.7}$$

from the requirement that the new candidate minimum $\mathbf{z}+\tau\mathbf{d}$ should be located at the boundary of the simplex; and if this results in an increase of the value of Ω, then τ is iteratively divided by a constant factor (~ 3) until minimum is found.

An important technical implementation detail is so-called "snapping". If some z_{aj} is small enough (that is, **z** is close enough to a boundary of the simplex) then it is set to zero. A similar snapping is used for the direction **d**.

We state here the explicit formulas for derivatives $f_j \equiv \partial\Omega(\{z_{aj}\})/\partial z_{aj}$ used within the program (derived from the definitions given in the previous sections).

Spherical kinematics:

$$f_j = 2p_a\tilde{q}_j - E_a. \tag{A.8}$$

Cylindrical kinematics:

$$f_j = 2p_a\tilde{q}_j - \frac{2E_a^{\perp}}{\sum_{j=1}^{n_{\rm jets}} z_{aj}E_a^{\perp}}\left(\eta_a - \eta_j\right)\left(q_j^0\sinh\eta_j - q_j^z\cosh\eta_j\right) - E_a^{\perp}. \tag{A.9}$$

A.2 Code and data structure

The code (file `ojf_014.f`) consists of subroutines (and functions) which can be divided in three logical groups: (i) interface subroutines, (ii) core subroutines and (iii) example jet search or result printing subroutines. In addition block data `ojf_lock` contains default values of some program parameters. The interface subroutines allow the user to enter input data, to read output or already entered input, to set or change program parameters and to obtain information about current program parameters. *All parameters that are supposed to be set or changed by the user can be accessed by these subroutines. The same applies to all input and output data.* The user is not supposed to write directly to common blocks. The core subroutines (functions) perform $\Omega\left(\{z_{aj}\}\right)$ minimization and conversion between various data forms. The user is not supposed to call them directly except for `Q_minimize`. The subroutine `Q_search` is an example application of OJF frame to simple jet search (see section A.6.7). The user may want to modify it or write their own subroutines if needed.

All floating point variables within the program are defined as
`DOUBLE PRECISION`. If the user employs `REAL` type variables they should ensure that a proper conversion of the parameter values is made in the calls of the OJF subroutines.

The file `ojf_com.fh` contains common block definitions of internal data structures for OJF, for instance, matrices for parameters of input particles, output jets parameters, and recombination matrix $\{z_{aj}\}$. The file `ojf_par.fh` contains the definitions of constants used within the program. The file `ojf_kin.fh` contains the definitions of two constants: `sphere=1, cylinder=2`. The file can be contained in user programs whenever reference to kinematics type is made, for example,

```
...
INCLUDE 'ojf_kin.fh'
INTEGER kinematics
...
kinematics=sphere
event_setup_begin( kinematics )
...
```

The other two files (`ojf_com.fh` and `ojf_par.fh`) normally do not need to be contained in user programs.

A.3 Normalization of energy units

The energies E_a or $E_a^\perp$ of input particles and the corresponding four-momenta p_a are normalized (after being entered) according to

$$E_a \to \frac{E_a}{\sum_{a=1}^n E_a}, \quad p_a \to \frac{p_a}{\sum_{a=1}^n E_a} \tag{A.10}$$

for spherical kinematics or according to

$$E_a^\perp \to \frac{E_a^\perp}{\sum_{a=1}^n E_a^\perp}, \quad p_a \to \frac{p_a}{\sum_{a=1}^n E_a^\perp} \tag{A.11}$$

for cylindrical kinematics. The normalization constant $\sum_{a=1}^n E_a$ or $\sum_{a=1}^n E_a^\perp$ is stored to interpret properly the final output. The normalization allows to make the implementation independent of energy units and scale.

A.4 Error messages

Significant part of the code consists of various checks. For example:

```
...
IF (.NOT. ojf_event_begin) THEN
   WRITE(6,*) 'add_particle: 20: wrong call sequence'
   WRITE(6,*) 'call event_setup_begin first'
   STOP 'add_particle: 20'
END IF
...
```

The checks are used to assure that subroutines are not called in inappropriate order, chosen parameters or input data do not have pathological values and that the program runs properly. The check can generate an error message and terminate the program. Messages with numbers 20-29 are due to the user errors. Messages with numbers ≥ 30 are generated by program failures, so should you get such a message, please inform the authors; please include the corresponding event in text form.

A.5 Key minimization subroutine `Q_minimize`

Subroutine `Q_minimize` minimizes $\Omega(\{z_{aj}\})$ for a given number of jets starting from the existing configuration of $\{z_{aj}\}$. An example program that uses `Q_minimize` is given in section A.8.

The subroutine performs iteratively the minimization algorithm described in section A.1. The iteration loop is terminated when no particle is moved in a single iteration or the maximal number of iterations is exceeded. (We regard that the minimum is found only in the former case.)

Default value of the maximal number of iterations is set 1000 which corresponds to $\sim$ 1 second of computing time on a modest computer. It can be changed with `set_maxiter ( maxiter )`, see section A.6.3. In each iteration, a loop over all particles is run ($a = 1, 2, ..., n$). For each particle separately, new candidate $\{z_{aj}\}$ for the minimum is found. The direction, $\mathbf{d}$, and step , τ, are computed according to the procedure described in section A.1. Unless the step is zero or "infinity", indicating that the particle should not be moved, the condition

$$\Omega(\mathbf{z} + \tau\mathbf{d}) < \Omega(\mathbf{z}). \tag{A.12}$$

is checked. If the condition is met, the recombination matrix $\{z_{aj}\}$ is moved into the new position. If not, the step is reduced 3 times, and (A.12) is checked again. If (A.12) is not true, τ is reduced again, and so on. If τ falls below some small parameter ($\tau\,|(\mathbf{d}, d_0)| \leq$ `eps_dist`), the particle is not moved and the program proceeds to the next particle.

A.6 User callable subroutines

We describe all user callable subroutines other than `Q_minimize`, explained above.

A.6.1 Event setup

`event_setup_begin ( kinematics )`

input:

`INTEGER kinematics` kinematics type

The subroutine begins initialization of a new event. It must be called before event data is entered. The parameter `kinematics` informs the program what type of kinematics is used: spherical (center of mass collisions), `kinematics`=1 or cylindrical (hadron collisions), `kinematics`=2. If the file `ojf_kin.fh` is included, constants `sphere` and `cylinder` can be used to assign value to `kinematics`:

```
...
INCLUDE 'ojf_kin.fh'
INTEGER kinematics
...
kinematics=sphere
event_setup_begin( kinematics )
...
```

Kinematics ought to be set once for all events in a job.

`add_particle ( energy, theta, phi )`

input:

DOUBLE PRECISION	energy	energy E_a
DOUBLE PRECISION	theta	angle θ_a
DOUBLE PRECISION	phi	angle ϕ_a

The subroutine is used to enter input data. It must be called between `event_setup_begin` and `event_setup_end`. Each call adds a particle (=detector cell) to the event. The energy E_a of the particle can be in any units, for example GeV. The direction of the particle is described by the standard angles θ_a (measured from beam axis) and ϕ_a.

`add_particle_raw ( px, py, pz )`

input:

DOUBLE PRECISION	px, py, pz	3-momentum components

The subroutine is used to enter input data, as an alternative to `add_particle`. It must be called between `event_setup_begin` and `event_setup_end`. Each call adds a particle (=detector cell) to the event. The parameters px, py, pz are 3-momentum components in the same units as energy in `add_particle`. The beam axis is in z-direction. The subroutine is useful with output of Monte Carlo event generators. It can be freely mixed with `add_particle`.

`event_setup_end`

must be called after all input particles are entered and before the jet search can be undertaken. No particles can be added to the event afterwards. This subroutine is needed for internal housekeeping. For instance, it provides the proper normalization of the energies of the particles.

A.6.2 Setup of initial jet configuration

`jets_setup_begin ( njets, Radius )`

input:

INTEGER	njets	number of jets, n_{jets}
DOUBLE PRECISION	Radius	parameter R

The subroutine has to be called to begin setup of the initial jet configuration - the initial value of the recombination matrix $\{z_{aj}\}$, necessary for the iterative minimization of Ω, as explained in section A.1. It is called automatically by `Q_search` but it must be called explicitly if `Q_minimize` is used instead. The number of jets, n_{jets}, must be positive. (The event will be reconstructed to the number of jets entered here.) R is the parameter in equations 3.9 and 3.17. It has to be positive and not too close to zero. The bigger R is, the less energy is left outside jets. New configurations of jets can be set up any number of times for the same event. The value of the seed from which the random number generator will start for this jet configuration is stored at this point. (From this point until the first invocation of anything random, the seed can be reset by `set_seed`.) If you need only to change R and proceed with minimization starting from the current configuration, use `reset_Radius`.

`set_seed ( seed )`

input:

`INTEGER seed`

This is to allow variation in random initial configurations of jets in case there are several local minima. It may be called once for a whole sequence of events - each event starts with a seed set up by the internal random number generator. The seed can be read (see `get_seed`) and used as a key to regenerate the corresponding configuration of jets (that is, local minimum; so the local minimum is completely determined by its seed). It must be called after `jets_setup_begin` but cannot be called after the first invocation of `init_z_random` or `init_random_all` or `jets_setup_end` and until the next `jets_setup_begin`.

`reset_Radius ( Radius )`

input:

`DOUBLE PRECISION Radius` parameter R

The subroutine changes the value of the parameter R in equations 3.9 and 3.17. R has to be positive and not too close to zero. A large value of R means less energy is left outside jets. The subroutine can be called at any time - the current configuration of jets is not affected (only Ω is recalculated properly). This may be useful for setting up interesting variations of the algorithm ("annealing") in which one starts from some small value of R and then changes it gradually, fine-tuning the resulting jet configurations by calls to `Q_minimize`. With infinitesimal values of R, the global minimum occurs for jet configurations with the most energetic particles playing the role of jets, so this can be used to obtain the most energetic (narrow clusters of) particles.

`init_z_random_all`

The subroutine can only be called between `jets_setup_begin` and `jets_setup_end`. It is the simplest way to initialize the recombination matrix $\{z_{aj}\}$: completely and uniformly random in the direct product of all the simplices corresponding to particles. If only specific particles need to be randomized, `init_z_random( a )` should be used. If only specific particles need to be set non-randomly, `init_z( a, z_in)` or `assign_to_jet( a, j )` should be called for those particles. Then `init_z_random_all` can be called to randomize the remaining particles. If this is not called explicitly, the particles not explicitly initialized are set to "neutral" positions (democratically shared between all jets and the soft energy).

`assign_to_jet ( a, j )`

input:

`INTEGER a` index of the particle

`INTEGER j` index of the jet

The subroutine can only be called between `jets_setup_begin` and `jets_setup_end`. It can be used to set the initial configuration of jets explicitly, for instance, when the output of another jet algorithm is to be fine-tuned. It sets the value $z_{aj} = 1$ for the given a, j, that is, directly assigns the a-th particle to the j-th jet. It must have $1 \leq a \leq n$

and $0 \leq j \leq n_{\text{jets}}$; $j = 0$ corresponds to soft energy. The subroutine only sets the initial configuration. No elements of the recombination matrix are protected from being changed by subsequent minimizations.

`init_z_from ( a, z_in )`

input:

INTEGER	a	index of the particle
DOUBLE PRECISION	z_in(0:njets_max)	components z_{a0}, z_{a1}, ..., $z_{an_{\text{jets}}}$

The subroutine can only be called between `jets_setup_begin` and `jets_setup_end`. It can be used to set the initial configuration of jets explicitly, for instance, to fine-tune the output of another jet algorithm. It initializes the recombination matrix $\{z_{aj}\}$ for the a-th particle, that is, sets z_{a0}, z_{a1}, ..., $z_{an_{\text{jets}}}$. Only $n_{\text{jets}} + 1$ components of the vector `z_in(0:njets_max)` are used. The components must be all non-negative but do not need to be normalized correctly - correct normalization will be imposed automatically; `z_in(0)` is the particle's fraction relegated to soft energy. For instance, `z_in(j)` can be some measure of distance between the a-th particle and the j-th jet from another jet algorithm. The subroutine only sets the initial configuration. No elements of the recombination matrix are protected from being changed by subsequent minimizations.

`init_z_random ( a )`

input:

INTEGER	a	index of the particle

The subroutine can only be called between `jets_setup_begin` and `jets_setup_end`. It does random initialization of the recombination matrix $\{z_{aj}\}$ for the a-th particle.

`jets_setup_end`

The subroutine must be called prior to minimization. It does housekeeping such as initialization of the particles whose recombination matrix elements have not been explicitly initialized by calls from `init_z_random`.

A.6.3 Setting algorithm control parameters

`set_maxiter ( maxiter )`

input:

INTEGER	maxiter	maximal number of iterations

The subroutine can be called to change the maximal number of iteration, see A.5. Default value of the maximal number of iterations is set to 1000 which corresponds to $\sim$ 1 second of computing time on a modest PC. It can be called at any time.

`set_njets_limits ( nstart, nstop )`

input:

INTEGER	nstart	starting number of jets
INTEGER	nstop	maximal number of jets

The subroutine is needed in conjunction with `Q_search` only. It sets the starting and the final number of jets in `Q_search` (see the end of section A.1 and description of `Q_search` in section A.6.7). The parameters must obey $1 \leq$ `nstart` $\leq$ `nstop` and `nstop` $\leq$ `njets_max` (constant `njets_max`, set in `ojf_par.fh`, defines the dimension of matrices and is the maximal allowed number of jets). The default values are: `nstart`=1 and `nstop`= `njets_max`=20. The subroutine can be called any time.

`set_ntries ( n )`

input:

INTEGER	n	number of tries

The subroutine is needed in conjunction with `Q_search` only. It sets the number of tries to find the minimum with different random initial configurations for each number of jets (see the end of section A.1 and description of `Q_search`, section A.6.7). The parameter `n` must be positive. The larger `n`, the higher the probability that the found configuration is the global minimum. Note that number of local minima correlates positively with number of hard partons. Usually values ~ 10 should suffice. The subroutine can be called at any time.

`set_trace_nmoved ( bool )`

input:

LOGICAL	bool	see text

The subroutine with parameter `bool=.TRUE.` turns on the option in which `Q_minimize` prints how many particles were moved at each iteration; with `bool=.FALSE.` it switches the option off (default). The subroutine can be called any time.

A.6.4 Access to parameters

`get_kinematics ( kinematics )`

output:

INTEGER	kinematics	type of kinematics

The subroutine returns the type of kinematics. The possible values are 1 (spherical kinematics, center of mass collisions) and 2 (cylindrical kinematics, hadron collisions), which is equivalent to constants `sphere` and `cylinder` if the header file `ojf_kin.fh` is included (see also section A.6.1). The subroutine cannot be called prior to the very first call of `event_setup_begin`.

`get_nparts ( nparts, e_scale )`

output:

INTEGER	nparts	number of particles in the event
DOUBLE PRECISION	e_scale	total energy of the event

The subroutine returns the number of particles, n, and the total energy in the event. The total energy is the sum of the usual energies of the particles for spherical kinematics $\sum_{a=1}^{n} E_a$ and the sum of transverse energies for cylindrical kinematics $\sum_{a=1}^{n} E_a^{\perp}$ in physical units, that is, prior to the normalization $\sum_{a=1}^{n} E_a = 1$ or $\sum_{a=1}^{n} E_a^{\perp} = 1$. In other words, `e_scale` is the normalization constant. Energy/momentum parameters returned by some other subroutines are normalized by the value of `e_scale`. The subroutine cannot be called between `event_setup_begin` and `event_setup_end`.

`get_particle ( a, e, xta, phi, p, ephys, pphys )`

input:		
`INTEGER`	`a`	index of the particle
output:		
`DOUBLE PRECISION`	`e`	normalized energy E_a or $E_a^{\perp}$
`DOUBLE PRECISION`	`xta`	angle θ_a or pseudorapidity η_a
`DOUBLE PRECISION`	`phi`	angle ϕ_a
`DOUBLE PRECISION`	`p(0:3)`	normalized four-momentum p_a
`DOUBLE PRECISION`	`ephys`	energy E_a or $E_a^{\perp}$ not normalized
`DOUBLE PRECISION`	`pphys(0:3)`	four-momentum p_a not normalized

The subroutine returns parameters of the a-th particle. For spherical kinematics the parameters are the usual energy, E_a, and the standard angles θ_a (from the beam axis) and ϕ_a. For cylindrical kinematics the parameters are the transverse energy, $E_a^{\perp}$, pseudorapidity, η_a, and the angle ϕ_a. The value of `e` is normalized and the `ephys` is in the same units as used in the input, that is, `ephys` = `e` · `e_scale` (see the previous subroutine for `e_scale`). All angles are in degrees. In both kinematics, `p(0:3)` and `pphys(0:3)` are the normalized and non-normalized four-momenta, p_a, of the particle. The subroutine cannot be called between `event_setup_begin` and `event_setup_end`.

`get_njets ( njets )`

output:		
`INTEGER`	`njets`	number of jets

The subroutine returns the number of jets in the current configuration of jets. It cannot be called before the first configuration of jets is setup.

`get_seed ( seed )`

input:		
`INTEGER`	`seed`	seed for random generator

The subroutine returns the value of the seed for the random generator, used for setting up the current random jet configuration. The value of the seed is "locked" (causing attempts to reset it to result in program termination) by the first invocation of anything "random" and retained until "unlocked" and reset by `jets_setup_begin`.

`get_Radius ( R )`

output:

`DOUBLE PRECISION R` parameter R in equations 3.9 and 3.17

The subroutine returns current value of the parameter R in equations 3.9 and 3.17.

`get_maxiter ( maxiter )`

output:

`INTEGER maxiter` maximal number of iteration

The subroutine returns the maximal number of iterations (see section A.5).

`get_njets_limits ( nstart, nstop )`

output:

`INTEGER nstart` starting number of jets

`INTEGER nstop` maximal number of jets

The subroutine returns the current values of the starting number of jets and the maximal number of jets in subroutine `Q_search` (see the end of section A.1 and description of `Q_search` in section A.6.7).

`get_ntries ( n )`

output:

`INTEGER n` number of tries

The subroutine returns the current number of tries in `Q_search`, the number of attempts to find minimum with different random initial configurations for each number of jets (see the end of section A.1 and the description of `Q_search`, section A.6.7).

A.6.5 Access to results

`get_criterion ( omega, y, esoft )`

output:

`DOUBLE PRECISION omega` value of Ω

`DOUBLE PRECISION y` value of Y

`DOUBLE PRECISION esoft` value of E_{soft}

The subroutine returns the value of Ω, Y and E_{soft}. Whenever a jet configuration is set up or modified, the corresponding values of Ω, Y and E_{soft} are recalculated and can be retrieved using this subroutine.

`get_jet ( j, e, xta, phi, q, qtilde, ephys, qphys )`

input:		
INTEGER	j	index of the jet
output:		
DOUBLE PRECISION	e	normalized energy or normalized transverse energy
DOUBLE PRECISION	xta	angle θ_j or pseudorapidity η_j
DOUBLE PRECISION	phi	angle ϕ_j
DOUBLE PRECISION	q(0:3)	normalized four-momentum q_j
DOUBLE PRECISION	qtilde(0:3)	four-direction $\tilde{q}_j$
DOUBLE PRECISION	ephys	energy (or transverse energy) in physical units
DOUBLE PRECISION	qphys(0:3)	four-momentum q_j in physical units

The subroutine returns parameters of the j-th jet, where j obeys $0 \leq j \leq n_{\rm jets}$ and $j = 0$ is the zeroth "jet", name for the fractions of particles that do not belong to any jet (that is, soft energy). For spherical kinematics the parameters are the usual energy, E_j, normalized e and non-normalized ephys (that is, in the units of energy used in the input), the standard angles θ_j (from the beam axis) and ϕ_j. For cylindrical kinematics the parameters are the transverse energy, $E_j^{\perp}$, normalized e and non-normalized ephys (that is, in the units of energy used in the input), pseudorapidity η_j and the standard angle ϕ_j. All angles are in degrees. For both kinematics, the parameters q(0:3) and qtilde(0:3) are the normalized and non-normalized four-momentum of the jet. $\tilde{q}_j$ is the four-direction defined in chapter 3.

`get_z ( a, z_out )`

input:		
INTEGER	a	index of the particle
output:		
DOUBLE PRECISION	z_out(0:njets_max)	components $z_{a0}, z_{a1}, ..., z_{an_{\rm jets}}$

The subroutine returns the components $z_{a0}, z_{a1}, ..., z_{an_{\rm jets}}$ of the recombination matrix for the a-th particle. a must satisfy $1 \leq a \leq n$. The value of z_out(j) is the a-th particle contribution to the j-th jet, and $j = 0$ corresponds to the soft energy. Note: $\sum_{\rm j=0}^{\rm njets}$ z_out(j) $= 1$.

`get_particle_split ( a, total_jets, jet, zj )`

input:		
INTEGER	a	index of the particle
output:		
INTEGER	total_jets	number of jets the particle belongs to
INTEGER	jet(0:njets_max)	indices of the jets
DOUBLE PRECISION	zj(0:njets_max)	corresponding z_{aj}

For the a-th particle, the subroutine returns: the number of jets (including soft energy 0-th "jet") which include a non-zero fraction of the particle ($z_{aj} \neq 0$), the labels of the jets and the corresponding values of z_{aj} in such an order that `zj(k)` $\geq$ `zj(j+1)`. In other words, the vector `zj(0:njets_max)` is the collection of $z_{a0}, z_{a1}, ..., z_{an_{\rm jets}}$ ordered by their value (descending from the left to right); only the components `(0:total_jets-1)` are different from zero.

`get_jet_split ( j, nwhole, whole_a, nfract, fract_a, fract_z )`

input:		
INTEGER	j	index of the jet
output:		
INTEGER	nwhole	number of particles entirely belonging to the jet
INTEGER	whole_a(0:nparts_max)	labels of the particles
INTEGER	nfract	number of particles belonging in some fraction
INTEGER	fract_a(0:nparts_max)	labels of the particles
DOUBLE PRECISION	fract_z(0:nparts_max)	the fraction z_{aj}

For the j-th jet ($0 \leq j \leq n_{\rm jets}$, $j = 0$ is the soft energy) the subroutine returns:

- number of the particles wholly in the jet, that is, $z_{aj} = 1$
- vector `whole_a(0:nparts_max)` with labels of such particles (indices a)
- number of particles partially in the jet, i.e $0 < z_{aj} < 1$
- vector `fract_a(0:nparts_max)` with labels of such particles (indices a)
- vector `fract_z(0:nparts_max)` with corresponding z_{aj} for such particles.

The latter two vectors are synchronously ordered so that subsequent components of `fract_z(0:nparts_max)` do not increase.

A.6.6 Sample print routines

`print_z_raw`
is an example subroutine to print the recombination matrix $\{z_{aj}\}$. A possible output may look like:

```
 a    background      1          2          3
---   ---------- ---------- ---------- ----------
  1    0.0000     0.0000     0.0000     1.0000
  2    0.0000     0.0000     0.0000     1.0000
  3    0.0000     0.0000     0.0000     1.0000
  4    0.0000     0.0000     0.0000     1.0000
...
```

`print_z_nice`
is an example subroutine to print the recombination matrix $\{z_{aj}\}$. A possible output may look like:

```
recombination matrix z by particle label a:

                          j e t   n u m b e r s
 a    background      1           2           3
---   ----------- ----------- ----------- -----------
  1        -           -           -          1.
  2        -           -           -          1.
  3        -           -           -          1.
  4        -           -           -          1.
...
```

`print_jets`
is an example subroutine to print properties of jets. See the output of the example program in section A.8.2.

`print_particles`
is an example subroutine to print properties of particles. A possible output may look like:

```
Configuration by particle:

                                          (soft energy is denoted as jet=0)
 a      E           E(%)    theta     phi      jet [ fraction ]; ...
---   ----------   -------  -------  -------   ---
  1    0.5100       6.7194  70.0000   0.0000     3
  2    0.4000       5.2701  90.0000   0.0000     3
  3    0.4000       5.2701  85.0000  10.0000     3
  4    0.2000       2.6350  84.0000 -10.0000     3
...
 24    0.2000       2.6350 170.0000  -7.0000     1
 25    7.0000E-02   0.9223  90.0000 170.0000     0
--------------------------
TOTAL: 7.5900      100.0000
```

A.6.7 Example subroutine of straightforward jet search `Q_search`

This is a simple jet search subroutine using `Q_minimize` as a key component. It is possible that the user may want to modify it, for example, when trying to do something with local minima. This subroutine uses only interface routines; it does not access internal data.

The subroutine tries to find the configuration of jets which minimizes Ω and ensures that $\Omega < \omega_{\rm cut}$ with the minimal number of jets (`njets`) starting from the number of jets previously

set via `set_njets_start` (usually the same for all events). For each number of jets, the search is repeated `ntries` times, each time with a different random initial value of the recombination matrix $\{z_{aj}\}$ and the configuration with the lowest value of Ω is retained as a result. Failure of the search is signaled by the condition `njets=0`.

Note that `Q_search` randomizes the initial value of $\{z_{aj}\}$, so it is meaningless to use it if one wants to specify the initial configuration for $\{z_{aj}\}$. In this case, the user should use `Q_minimize` directly. We comment that some other control options could be to continue attempts until a specified number of attempts fails to yield a better configuration or to stop the search for new minimum if, for example, the first three random initial configurations yielded the same configurations (the event has a single local minimum which is automatically the global one; this is quite likely and may be useful if CPU time is an issue).

A.7 Compilation

Optimal Jet Finder consists of the following files:

- `ojf_014.f` main file contains all subroutines and functions
- `ojf_com.fh` contains definitions of common blocks
- `ojf_par.fh` contains definitions of parameters
- `ojf_kin.fh` contains definition of kinematics type parameters

Example programs `example.f`, `ww160.f`, `ww160a.f` with input or output files `example.in`, `ww160.in`, `ww160.out`, `ww160a.out` are added.

To compile and run any of example programs with OJF under Linux equipped with g77 the user can type:

```
g77 user_program.f ojf_014.f -o executable_file   (enter)
executable_file                                   (enter)
```

where `user_program.f` is the name of the user own program applying OJF. Each example program `example.f`, `ww160.f` or `ww160a.f` can be used in its place. Files `ojf_com.fh`, `ojf_par.fh`, and `ojf_kin.fh` should be available in the current directory (but not compiled).

A.8 Example

The simplest possible example, file `example.f` below, should give the idea how Optimal Jet Finder can be used. File `example.in` contains input data. Each line corresponds to one particle and consists of E_a, θ_a and ϕ_a for that particle. The user is encouraged to study subroutine `Q_search` and programs `ww160.f`, `ww160a.f` providing additional, more advanced examples.

A.8.1 Source code of example.f

```
      PROGRAM simplest_example

      INCLUDE 'ojf_kin.fh'

      DOUBLE PRECISION  Radius
      DOUBLE PRECISION  e, theta, phi
      DOUBLE PRECISION  o_fin, y_fin, e_fin
      INTEGER           a, seed, nparts, njets, kinematics
      LOGICAL           success
```

number of jets is chosen

```
      njets = 3
```

seed for random generation of the recombination matrix

```
      seed  = 13
```

R parameter from equations 3.9 and 3.17

```
      Radius = 1.0
```

choose spherical (lepton collisions) kinematics

```
      kinematics = sphere
```

file with input data is opened

```
      OPEN(10, FILE='example.in', FORM='formatted', STATUS = 'old')
```

input event setup starts

```
      CALL event_setup_begin ( kinematics )
```

loop over all particles in the event

```
      nparts = 0
      DO a = 1, 1999
         READ(10,*, end=1000, err=1000) e, theta, phi
         CALL add_particle ( e, theta, phi )
         nparts = nparts + 1
      ENDDO

 1000 CLOSE(10)
```

input of the event ends

```
      CALL event_setup_end
```

set up random the initial value of the recombination matrix

```
      CALL jets_setup_begin ( njets, Radius )
      CALL set_seed ( seed )
      CALL init_z_random_all
      CALL jets_setup_end
```

minimize Ω

```
      CALL Q_minimize ( success )
      IF (.NOT. success) STOP 'minimum not found'
```

get and print the values of Ω, Y *and* E_{soft} *for the final jet configuration*

```
      CALL get_criterion ( o_fin, y_fin, e_fin )

      WRITE(*,*) 'Omega  =', o_fin
      WRITE(*,*) 'Y      =', y_fin
      WRITE(*,*) 'E_soft =', e_fin
```

prints properties of the resulting jets

```
      call print_jets

      END
```

A.8.2 Output of the example

```
Omega  =  0.293404849
Y      =  0.0338528071
E_soft =  0.259552042

SPHERE:  3 jets processed

Configuration by jet:

jet       E           E(%)     theta      phi
---   -----------   -------   -------   -------
  1     1.380       18.1818  138.3848  -52.8876
  2     1.220       16.0738  124.4338  -26.6115
  3     3.020       39.7892   81.0226    0.3566
---------------------------
TOTAL: 5.6200       74.0448
```

```
Particle content by jet:

jet label  1 (  3 particle(s) ):
        E(%)  =   18.18
       theta  =   138.4
         phi  =  -52.89
  3 particle(s) in jet as a whole:  21  22  24

jet label  2 (  4 particle(s) ):
        E(%)  =   16.07
       theta  =   124.4
         phi  =  -26.61
  4 particle(s) in jet as a whole:  17  18  19  20

jet label  3 (  8 particle(s) ):
        E(%)  =   39.79
       theta  =   81.02
         phi  =  0.3566
  8 particle(s) in jet as a whole:   1   2   3   4
 5   6   7   8

soft energy ( 10 particle(s) ):
 10 whole particle(s) in soft energy:   9  10  11
12  13  14  15  16  23  25
no particles partially in soft energy
```

A.9 Definitions of constants: ojf_par.fh

In this section we explain the meaning of the parameters defined in the header file ojf_par.fh and give their default values.

INTEGER njets_max=50
The maximal number of jets; used for example to define the size of matrices.

INTEGER nparts_max=2000
The maximal number of particles in the event; used for example to define the size of matrices.

DOUBLE PRECISION zero=0

DOUBLE PRECISION one=1
DOUBLE PRECISION inf=10^{100}
are the numerical constants.

DOUBLE PRECISION eps_snap=10^{-3}
If z_{aj} < eps_snap than z_{aj} is set to zero, that is, the particle is snapped to the boundary of the simplex. The parameter is used in subroutines z_snap and z_assert.

DOUBLE PRECISION eps_round=10^{-6}
DOUBLE PRECISION eps_sum=10^{-8}
DOUBLE PRECISION eps_sum0=10^{-6}
DOUBLE PRECISION eps_sum1=10^{-4}
The constants are used to keep control of rounding errors. If some variable exceeds the allowing range of values more than eps_, the error message is generated and the program is terminated. The constants are used in subroutines d_minus_snap, z_snap, d_assert, z_assert, z_force_to_simplex and in the function pos_prod.

DOUBLE PRECISION eps_norm=10^{-6}
The constant is used to determine whether the norm of the 3-vector $\mathbf{q}_j$ (or transverse part of the norm in case of cylindrical dynamics) is zero. It is used in subroutine j_eval_nonlinear.

DOUBLE PRECISION eps_Et=10^{-6}
The constant is used to handle small values of the transverse energy of a jet. It is used in subroutines grad_Y and j_eval_nonlinear.

DOUBLE PRECISION eps_dist=10^{-6}
See section A.5. The constant is used to determine when to stop subsequent reductions of the step τ. The constant is used in subroutine Q_minimize_wrt.

DOUBLE PRECISION eps_radius=10^{-3}
The constant sets the limit on the smallest value of R, the parameter from equations 3.9 and 3.17. It is used in subroutines jet_setup_begin and reset_Radius.

DOUBLE PRECISION inf_step=10^{30}
See section A.5. "Infinite" step means that the particle should not be moved. The constant is used in subroutines Q_minimize_wrt, d_eval_step and
z_move_by.

It is imaginable that some of the parameters above may need to be changed but the user is advised to be careful when doing this. In particular, smaller values of some parameters would

enhance sensitivity to rounding errors, causing the safety checks to generate error messages and terminate the program. One may change eps_snap to a smaller value, say 10^{-5}, and see if the results would change; for a small fraction of events this may slow the finding of jets but help to better identify local minima.

INTEGER random_m=259200
The constant is used by the random number generator, subroutine seed and the function random().

The constants below play a technical role and are not supposed to be changed. The reason for defining them is cleared in the next section.

```
INTEGER par_Et=4
INTEGER par_eta=5
INTEGER par_Eteta=6
INTEGER par_y=7
INTEGER par_p0shmpzch=8
INTEGER par_tilde=9
```

A.10 Common block definitions: ojf_com.fh

The header file ojf_com.fh contains common block definitions and is included in most of the subroutines. The user is not supposed to write to common blocks directly but to use interface subroutines. Data that cannot be accessed that way is not supposed to be used by the user.

A.10.1 Input of the event

```
      COMMON /ojf_event/
     &     ojf_event_begin, ojf_event_set,
     &     ojf_kinematics, ojf_nparts,
     &     ojf_p, ojf_e, ojf_e_scale
```

LOGICAL ojf_event_begin, ojf_event_set
The two logical values bracket the event setup:
FALSE, FALSE - at start of program, no event has been set up;
TRUE , FALSE - event setup in progress, adding particles;
FALSE, TRUE - event setup completed, can search for jets.

`INTEGER ojf_kinematics`
The variable marks the type of kinematics: 1 - spherical kinematics of lepton collisions, 2 - cylindrical kinematics of hadron collisions.

`INTEGER ojf_nparts`
The number of particles in the event.

`DOUBLE PRECISION ojf_p(0:6, 1:nparts_max)`
The matrix stores the properties of particles:

`ojf_p(0, particle_label)`	energy E_a
`ojf_p(1, particle_label)`	x-component of momentum p_a
`ojf_p(2, particle_label)`	y-component of momentum p_a
`ojf_p(3, particle_label)`	z-component of momentum p_a
`ojf_p(4, particle_label)`	transverse energy $E_a^{\perp}$
`ojf_p(5, particle_label)`	pseudorapidity η_a
`ojf_p(6, particle_label)`	combination $E_a^{\perp} \cdot \eta_a$

and `particle_label` is the index a of the particle. The constants `par_Et=4`, `par_eta=5`, `par_Eteta=6` are defined to access the components of the matrix,
for example, `ojf_p(par_Eteta, particle_label)`.

`DOUBLE PRECISION ojf_e(1:nparts_max)`
The vector stores the energies of the particles.

`DOUBLE PRECISION ojf_e_scale`
The variable stores the energy scaling factor (see section A.3).

A.10.2 Configuration of jets (output)

```
      COMMON /ojf_jets/
     &      ojf_jets_begin, ojf_jets_set,
     &      ojf_njets, ojf_seed, ojf_Radius,
     &      ojf_z, ojf_b, ojf_q,
     &      ojf_Omega, ojf_Y, ojf_Esoft
```

`LOGICAL ojf_jets_begin, ojf_jets_set`
The two logical values bracket setup of initial jet configuration:
`FALSE, FALSE` - at start of program, or after event set up;
`TRUE, FALSE` - jets setup in progress, change anything;
`FALSE, TRUE` - jets setup complete, can do minimization.

`INTEGER ojf_njets`

The number of jets in the current configuration.

`INTEGER ojf_seed`
The seed used to generate the current (random) jet configuration.

`DOUBLE PRECISION ojf_Radius`
The value of R, the parameter in (3.9) and (3.17).

`DOUBLE PRECISION ojf_z(0:njets_max,1:nparts_max)`
The recombination matrix, $\{z_{aj}\}$.

`LOGICAL ojf_b(0:njets_max,1:nparts_max)`
It is used to indicate that the particle belongs to (`TRUE`) or does not belong (`FALSE`) to the boundaries of the simplex, that is, $z_{aj} = 0$.

`DOUBLE PRECISION ojf_q(0:12, 1:jets_max)`
The matrix stores the properties of particles:

`ojf_q( 0, jet_label)`	energy E_j
`ojf_q( 1, jet_label)`	x-component of momentum q_j
`ojf_q( 2, jet_label)`	y-component of momentum q_j
`ojf_q( 3, jet_label)`	z-component of momentum q_j
`ojf_q( 4, jet_label)`	transverse energy $E_j^\perp$
`ojf_q( 5, jet_label)`	pseudorapidity η_j
`ojf_q( 6, jet_label)`	combination $E_j^\perp \cdot \eta_j$
`ojf_q( 7, jet_label)`	fuzziness Y
`ojf_q( 8, jet_label)`	combination $(q_j)^0 \cdot (\tilde{q}_j)^3 - (q_j)^3 \cdot (\tilde{q}_j)^0$
`ojf_q( 9, jet_label)`	0-component of four-direction $\tilde{q}_j$
`ojf_q(10, jet_label)`	x-component of four-direction $\tilde{q}_j$
`ojf_q(11, jet_label)`	y-component of four-direction $\tilde{q}_j$
`ojf_q(12, jet_label)`	z-component of four-direction $\tilde{q}_j$

`jet_label` is the index j of the jet. The constants `par_Et`=4, `par_eta`=5, `par_Eteta`=6, `par_y`=7, `par_p0shmpzch`=8, `par_tilde`=9 are defined to access the components of the matrix, for example, `ojf_q(par_y, jet_label)`.

`DOUBLE PRECISION ojf_Omega, ojf_Y, ojf_Esoft`
The variables store the values of Ω, Y and E_{soft}.

The remaining common block `/ojf_work/` contains the definitions of "work" variables, mainly of the types explained above. The "work" variables are used at the intermediate stages of computations.

Bibliography

[1] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder. *Comput. Phys. Commun.*, 155:42–64, 2003.

[2] Fyodor V. Tkachov. The definition of jets. *Int. J. Mod. Phys.*, A17:2783–2884, 2002.

[3] http://www.inr.ac.ru/~ftkachov/projects/jets/.

Appendix B

C++ Implementation

A version of this chapter has been submitted for publication [1].

This appendix is intended as a documentation for the users of an object-oriented C++ implementation [1] of Optimal Jet Definition [2]. The source code is available for downloading from [3].

The C++ implementation is based on the verification version [4].

The program is self-contained: it requires only standard C++ libraries and should compile with any standard C++ compiler.

The C++ implementation has been verified against the FORTRAN 77 version: ojf_015, available from [3]. The details can be found in section B.4.

B.1 User interface classes and methods

All classes are contained within the `OptimalJetFinder namespace`. In this section, we describe several classes and methods most likely to be needed by the user. The reader may find it more practical to study `example.cpp` in the next section before browsing through this section.

B.1.1 class Event

This class represents a high energy physics event: a collection of input particles (calorimeter cells, preclusters, et cetera).

- `Event( Kinematics k )`
  - constructor. `Kinematics = enum { sphere, cylinder }`, where `sphere` applies to the center of mass kinematics (lepton collisions), and `cylinder` applies to the cylindrical kinematics of hadron collisions.

- `void AddParticleRaw( double px, double py, double pz )`
  adds a particle to the event. `px`, `py`, `pz` are the components of the momentum of the particle in arbitrary units.

- `void AddParticle( double E, double theta, double phi )`
  adds a particle to the event. `E` is the energy of the particle in arbitrary units and the standard angles `theta` and `phi` describe the direction of the particle. The angles are measured in degrees.

- `void Normalize()`
  has to be called before jets are searched. It normalizes the four-momenta of the particles so that the sum of all energies or transverse energies of all particles is equal to one.

- `void Clear()`
  removes all particles from the event and releases memory accordingly.

- `Kinematics GetKinematics() const`
  returns the type of kinematics; see the constructor above.

- `Particle* GetFirst() const`
  returns the pointer to the first particle in the event or 0 if there are no particles.

- `bool IsNormalized() const`
  returns `true/false` depending whether the event is already normalized; see `Normalize()` above.

- `double GetXEnergy() const`
  returns the sum of energy (for the spherical kinematics) or sum of transverse energy (for the cylindrical kinematics) of all particles in the event.

- `int GetNumber() const`
  returns the number of particles in the event.

B.1.2 class JetSearch

This is a simple jet search class.

- `JetSearch(const Event* P, double R, int ntries = 10)`
  - constructor. Initializes jet search. `P` is a pointer to the object of the `Event` class. `R` is the radius parameter R of equations 3.9 and 3.17. `ntries` is the number of different random initial jet configurations tried.

- `bool FindJetsForFixedNJets(int njets)`
  finds the final jet configuration with the number of jets equal to `njets` and returns `true` if successful and `false` otherwise. For each "try", it starts with a random initial jet

configuration and finds a local minimum of the Ω function. After a number of tries (set with
`void SetNTries( int ntries )`; default = 10) the best jet configuration is chosen, that is, the one that gives the smallest value of Ω (the deepest local minimum).

- `int FindJetsForOmegaCut(double omegaCut)`
  finds the final jet configuration for `omegaCut`=ω_{cut} of relation 3.10 and returns the number of jets in the final jet configuration or 0 if the search is not successful. It runs
  `bool JetSearch::FindJetsForFixedNJets(int njets)` increasing the number of jets between the values set by
  `void JetSearch::SetNJetsBegin( int nBegin )`
  and `void JetSearch::SetNJetsEnd( int nEnd )`.
  The final jet configuration is the one with the smallest number of jets for which the value of Ω function (equations 3.9 and 3.17) is smaller than ω_{cut} parameter.

- `Jets* GetJets() const`
  can be used to access the final jet configuration.

- `void SetNTries( int ntries )`
  sets the number of different random initial jet configurations tried.

- `int GetNTries() const`
  returns the number of different random initial jet configurations tried.

- `void SetMaxIter( int MaxIter )`
  sets the maximal number of iterations in the minimization algorithm. The default value is 2000. If the local minimum is not found within the maximal number of iterations the current jet search is terminated and `bool FindJetsForFixedNJets(int njets)` returns `false`, or
  `int FindJetsForOmegaCut(double omegaCut)` returns 0.

- `int GetMaxIter() const`
  returns the maximal number of iterations in the minimization algorithm.

- `void SetNJetsBegin( int nBegin )`
  sets the initial number of jets in
  `int FindJetsForOmegaCut(double omegaCut)`.

- `int GetNJetsBegin() const`
  returns the initial number of jets in
  `int FindJetsForOmegaCut(double omegaCut)`.

- `void SetNJetsEnd( int nEnd )`
  sets the maximal allowed number of jets in
  `int FindJetsForOmegaCut(double omegaCut)`.

- `int GetNJetsEnd() const`
  returns the maximal allowed number of jets in
  `int FindJetsForOmegaCut(double omegaCut)`.

B.1.3 class Particle

Objects of this class correspond to particles (or calorimeter cells, preclusters, et cetera.) in the event. In most cases, the user will not need to create instances of this class directly, but only use pointers to this class to access information about particles.

- `Particle(int Label, Kinematics k, const Event* P)`
  - constructor. In most cases, the user does not need to call the constructor directly but only through
  `Event::AddParticleRaw(double px, double py, double pz)`
  or `Event::AddParticle(double px, double py, double pz)`. If particles are entered using either of the two just mentioned methods, the first particle has label 1, the next 2, et cetera. Otherwise the `label` has an arbitrary value specified by the user.

- `double GetE() const`
  returns the energy of the particle in the same units as used in the input.

- `double GetPx() const`

- `double GetPy() const`

- `double GetPz() const`
  return the x(y,z)-component of the momentum of the particle in the same units as used in the input.

- `double GetXEnergy() const`
  returns the energy of the particle (for the spherical kinematics) or transverse energy of the particle (for the cylindrical kinematics) in the same units as used in the input.

- `double GetXEta() const`
  returns the standard angle θ in degrees for the spherical kinematics or pseudorapidity η for the cylindrical kinematics.

- `double GetPhi() const`
  returns the standard angle ϕ in degrees.

- `double GetESoft() const`
  for the spherical kinematics, it returns the fraction of the energy of the particle that does not belong to any jet; for the cylindrical kinematics, it returns the fraction of the transverse energy of the particle that does not belong to any jet; in normalized units (see `Event::Normalize()`).

- `double GetFractionInJet(int j) const`
  returns the fraction of the particle that belongs to the `j`-th jet.

- `int GetLabel() const`
  returns the label of the particle. If particles are entered using
  `Event::AddParticleRaw(double px, double py, double pz)` or
  `Event::AddParticle(double px, double py, double pz)`, the first particle has label 1, the next 2, et cetera. Otherwise, the label has the value that was used in the constructor call.

- `Particle* GetNext() const`
  returns the pointer to the next particle in the event. This method allows to loop over all particles in the event.

B.1.4 class Jets

This class represents a configuration of jets. In most cases, the user will not need to create instances of this class directly, but only use pointers to this class to access information about the jet configuration.

- `Jets(int njets, const Event* P, double R)`
  - constructor. `njets` is the number of jets, `P` is a pointer to the object of the class `Event`, `R` is the radius parameter R of equations 3.9 and 3.17.

- `const Event* GetEvent() const`
  returns the pointer the event with which the jets are associated.

- `double GetR() const`
  returns the radius parameter R of equations 3.9 and 3.17.

- `int GetNumber() const`
  returns the number of jets.

- `Jet* GetFirst() const`
  returns the pointer to the first jet.

- `double GetESoft() const`
  For the spherical kinematics, it returns the soft energy in normalized units, which is the part of the energy of the event that does not belong to any jet. For the cylindrical kinematics, it returns the fraction of the transverse energy of the event that does not belong to any jet.

- `Jet* GetJet( int n ) const`
  returns the pointer to the `n`-th jet.

- `double GetY() const`
  returns the value of the fuzziness, the first term in equations 3.9 and 3.17.

- `double GetOmega() const`
  returns the value of Ω of equations 3.9 and 3.17.

B.1.5 class `Jet`

This class represents a single jet. In most cases, the user will not need to create instances of this class directly, but only use pointers to objects of this class to access the information about the jets.

- `Jet(int label, Jets* Q, Kinematics k)`
  - constructor. `label` is the index of the jet, `Q` is the pointer to the jet configuration (to an object of the class `Jets`).
  `Kinematics = enum { sphere, cylinder }`, where `sphere` applies to the center of mass kinematics (lepton collisions), and `cylinder` applies to the cylindrical kinematics of hadron collisions.

- `double GetE() const`
  returns the energy of the jets in the same units as used in the input.

- `double GetPx() const`

- `double GetPy() const`

- `double GetPz() const`
  return the x(y,z)-component of the momentum of the jet in the same units as used in the input.

- `double GetXEnergy() const`
  returns the energy of the jet for the spherical kinematics or transverse energy of the jet for the cylindrical kinematics.

- `double GetXEta() const`
  returns the standard angle θ of the jet direction (in degrees) for the spherical kinematics or the pseudorapidity η of the jet for the cylindrical kinematics.

- `double GetPhi() const`
  returns the standard angle ϕ of the jet direction (in degrees).

- `int GetLabel() const`
  returns the label of the jet (the index of the jet).

- `Jets* GetJets() const`
  returns the pointer to the jet configuration to which the jet belongs.

- `Jet* GetNext() const`
  returns the pointer to the next jet. It allows to loop over jets.

B.2 Compilation

The program is self-contained and requires only a standard C++ compiler and the standard C++ libraries. It consists of the implementation files: `OJFZD.cpp`, `OJFKinematics.cpp`, `OJFJets.cpp`, `OJFSearch.cpp`, header files: `OJFZD.h`, `OJFKinematics.h`, `OJFJets.h`, `OJFSearch.h`, example program: `example.cpp`, input data for the example program `inputWW.dat`, and the `Makefile`. To compile and run the example program (with g++ under Linux)

```
>make example
>example
```

can be used or alternatively

```
>g++ OJFZD.cpp OJFKinematics.cpp OJFJets.cpp OJFSearch.cpp
 example.cpp -o example
>example
```

In the last three lines, the example program `example.cpp` can be replaced by the user's own program.

B.3 Usage example

The usage of Optimal Jet Finder is best explained with the following example.

B.3.1 Source code of the `example.cpp`

```
#include "OJFKinematics.h"
#include "OJFJets.h"
#include "OJFSearch.h"
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cstdlib>
using namespace std;
using namespace OptimalJetFinder;

int main() {

  //input data

  ifstream in( "inputWW.dat" );

  //create a new event
  Event *P = new Event( sphere );
```

```
//use "cylinder" instead of "sphere" for cylindrical kinematics

double px, py, pz;
while( in>>px>>py>>pz ) {
  P->AddParticleRaw( px, py, pz ); //input a particle
}

in.close();

//input data ends


//normalize input momenta so that the sum of input energies = 1
//(or the sum of transverse energies for cylindrical kinematics = 1)
P->Normalize();


cout << P->GetNumber() << " particles in the event." << endl;

//set the seed for the random number generator
OJFRandom::SetSeed( 13 );

double radius = 1.0; // R parameter
unsigned ntries = 3; // number of tries

//new jet search created
JetSearch* js = new JetSearch( P, radius, ntries );

//find jets for a given value of Omega_cut
unsigned njets = js->FindJetsForOmegaCut(0.05);
if( njets == 0 ) { cout << "Jets lost." << endl; exit(1); }

//alternatively,
//find jet configuration for a fixed number of jets
//bool success = js->FindJetsForFixedNJets(4);
//if (  ! success  ) { cout << "Jets lost." << endl; exit(1); }

//get the jet configuration
Jets* Q = js->GetJets();
```

```
  //display the number of jets
  // and parameters /Omega, Y, Esoft/ of the jet configuration
  cout << Q->GetNumber() << " jets found." << endl;
  cout << "Omega: " << Q->GetOmega() << ",   "
       << "Y: " << Q->GetY() << ",   "
       << "Esoft (normalized): " << Q->GetESoft() << "." << endl;

  //display the details of the jets
  cout << "The details of the jets (E px py pz):" << endl;

  Jet* jet = Q->GetFirst();
  while( jet ) {
    cout << setw( 10 ) << jet->GetE() << "  "
<< setw( 10 ) << jet->GetPx() << "  "
<< setw( 10 ) << jet->GetPy() << "  "
<< setw( 10 ) << jet->GetPz() << endl;
    jet = jet->GetNext();
  }

  //the user is responsible for deleting
  //what they created themselves with new
  delete P;
  delete js;

}
```

B.3.2 Output of example.cpp

```
65 particles in the event.
4 jets found.
Omega: 0.0464792,   Y: 0.0382961,   Esoft (normalized): 0.00818312.
The details of the jets (E px py pz):
   49.1723      45.698    -9.17652    -11.8903
   29.8772     10.4448     26.5263     6.43505
   38.1886    -18.5112     25.5879     19.5718
   59.2424    -37.2752    -43.3007    -12.9766
```

B.4 Comparison between FORTRAN 77 and C++ version

We have run several test programs to compare the output of the FORTRAN ojf_015 version (appendix A) and the C++ version described in this appendix (both compiled with GNU gcc 3.4.2 on Linux Fedora Core 3, Intel i686).

In each test, we compute

$$\Delta = \begin{cases} \left| \dfrac{x_{\text{C++}} - x_{\text{FORTRAN}}}{x_{\text{FORTRAN}}} \right| & (x_{\text{FORTRAN}} \neq 0) \\ \\ |x_{\text{C++}} - x_{\text{FORTRAN}}| & (x_{\text{FORTRAN}} = 0) \end{cases}, \tag{B.1}$$

where x is any of the following quantities: Ω, Y, E_{soft}, E_j, $p_j^{(x)}$, $p_j^{(y)}$, $p_j^{(z)}$, and E_j, θ_j, ϕ_j (spherical kinematics) or $E_j^{\perp}$, η_j, ϕ_j (cylindrical kinematics); j runs over all reconstructed jets. We characterize each event by Δ_{max}, the maximal value of all Δ's calculated for this event. Tables B.1 and B.2 present the distribution of Δ_{max}'s for two multi event tests. Tables B.3 and B.4 show the parameters and the results of single-event tests.

All events were generated with Pythia 6.222 [5].

Note that different stochastic minimum search algorithms must find the same set of local minima – but not necessarily in the same order (if only because of different floating point machine codes generated by different compilers). However, it proved possible to adjust the current implementation (the control parameters, et cetera) so as to ensure that even the order of the local minima found is the same as with the FORTRAN 77 version for the same seed of the random number generator – without spoiling the high precision of the computations. Whatever minor numerical differences remain (see the comparison tables) must be attributed to the observed differences in computation of hyperbolic sines, et cetera by the different routines provided by the C++ and FORTRAN 77 compilers.

Bibliography

[1] S. Chumakov, E. Jankowski, and F. V. Tkachov. Optimal Jet Finder (C++ v1.0). Submitted for publication in Comput. Phys. Commun.

[2] Fyodor V. Tkachov. The definition of jets. *Int. J. Mod. Phys.*, A17:2783–2884, 2002.

[3] http://www.inr.ac.ru/~ftkachov/projects/jets/.

[4] Fyodor V. Tkachov. A verification of the Optimal Jet Finder. 2001.

[5] Torbjorn Sjostrand et al. High-energy-physics event generation with PYTHIA 6.1. *Comput. Phys. Commun.*, 135:238–259, 2001.

Table B.1: Distribution of $\Delta_{\max}$ for a sample of 10^6 $e^+e^- \rightarrow WW \rightarrow$ hadrons events at 180 GeV. Spherical kinematics. Three-momenta used in the input. $R = 1.0$, $n_{\text{tries}} = 1$, $n_{\text{jets}} = 4$, seed = 13.

$\Delta_{\max}$ RANGE	FRACTION OF EVENTS IN THE RANGE
$10^{-18} - 10^{-17}$	0.000002
$10^{-17} - 10^{-16}$	0.068718
$10^{-16} - 10^{-15}$	0.508784
$10^{-15} - 10^{-14}$	0.409418
$10^{-14} - 10^{-13}$	0.010992
$10^{-13} - 10^{-12}$	0.001616
$10^{-12} - 10^{-11}$	0.000369
$10^{-11} - 10^{-10}$	0.000074
$10^{-10} - 10^{-9}$	0.000021
$10^{-9} - 10^{-8}$	0.000005
$10^{-8} - 10^{-7}$	0.000001

Table B.2: Distribution of $\Delta_{\max}$ for a sample of 10^5 $pp \rightarrow tt + X \rightarrow$ hadrons events at 14 TeV. Cylindrical kinematics. Three-momenta used in the input. $R = 1.0$, $n_{\text{tries}} = 1$, $n_{\text{jets}} = 6$, seed = 13. Two events yielded different jet configurations in the FORTRAN and C++ versions, corresponding to different local minima. The value of Ω was smaller for the C++ version by approximately 10^{-4} and 0.25.

Δ_{max} RANGE	FRACTION OF EVENTS IN THE RANGE
$< 10^{-18}$	0.00004
$10^{-18} - 10^{-17}$	0.00051
$10^{-17} - 10^{-16}$	0.00336
$10^{-16} - 10^{-15}$	0.11745
$10^{-15} - 10^{-14}$	0.22692
$10^{-14} - 10^{-13}$	0.24575
$10^{-13} - 10^{-12}$	0.23479
$10^{-12} - 10^{-11}$	0.13154
$10^{-11} - 10^{-10}$	0.03411
$10^{-10} - 10^{-9}$	0.00536
$10^{-9} - 10^{-8}$	0.00015

Table B.3: A single $e^+e^- \rightarrow WW \rightarrow$ hadrons event at 180 GeV. Spherical kinematics. In the input, three-momenta are used for tests B01-B17, and angles are used for tests C01-C04.

TEST ID	R	n_{tries}	n_{jets}	ω_{cut}	seed	Δ_{max}
B01	1.0	1	2	-	13	$8.5 \cdot 10^{-15}$
B02	1.0	1	4	-	13	$1.6 \cdot 10^{-16}$
B03	1.0	1	12	-	13	$3.2 \cdot 10^{-15}$
B04	1.0	1	20	-	13	$5.4 \cdot 10^{-15}$
B05	0.1	1	4	-	13	$8.3 \cdot 10^{-14}$
B06	0.2	1	4	-	13	$1.4 \cdot 10^{-16}$
B07	0.7	1	4	-	13	$1.4 \cdot 10^{-16}$
B08	10.0	1	4	-	13	$2.3 \cdot 10^{-15}$
B09	1.0	2	4	-	13	$2.9 \cdot 10^{-15}$
B10	1.0	3	4	-	13	$2.9 \cdot 10^{-15}$
B11	1.0	100	4	-	13	$2.9 \cdot 10^{-15}$
B12	1.0	3	4	-	6969	$2.9 \cdot 10^{-15}$
B13	1.0	50	-	0.005	13	$5.6 \cdot 10^{-15}$
B14	1.0	50	-	0.01	13	$2.4 \cdot 10^{-16}$
B15	1.0	50	-	0.02	13	$8.2 \cdot 10^{-15}$
B16	1.0	50	-	0.04	13	$3.3 \cdot 10^{-15}$
B17	1.0	50	-	0.06	13	$2.9 \cdot 10^{-15}$
C01	0.7	3	-	0.021	1313	$1.9 \cdot 10^{-14}$
C02	1.5	50	-	0.04	1313	$2.4 \cdot 10^{-16}$
C03	0.7	1	4	-	1313	$1.9 \cdot 10^{-16}$
C04	1.5	2	5	-	1313	$1.9 \cdot 10^{-15}$

Table B.4: A single $pp \rightarrow tt + X \rightarrow$ hadrons event at 14 TeV. Cylindrical kinematics. In the input, three-momenta are used for tests D01-D13, and angles are used for tests E01-E04.

TEST ID	R	n_{tries}	n_{jets}	ω_{cut}	seed	Δ_{max}
D01	1.0	1	2	-	13	$4.2 \cdot 10^{-16}$
D02	1.0	1	6	-	13	$6.2 \cdot 10^{-14}$
D03	1.0	1	12	-	13	$4.1 \cdot 10^{-13}$
D04	1.0	1	20	-	13	$1.5 \cdot 10^{-11}$
D05	0.1	1	6	-	13	$4.2 \cdot 10^{-13}$
D06	10.0	1	6	-	13	$3.8 \cdot 10^{-13}$
D07	1.0	2	6	-	13	$4.2 \cdot 10^{-13}$
D08	1.0	3	6	-	13	$4.2 \cdot 10^{-13}$
D09	1.0	100	6	-	13	$3.4 \cdot 10^{-14}$
D10	1.0	3	6	-	6969	$3.3 \cdot 10^{-14}$
D11	1.0	50	-	0.05	13	$2.4 \cdot 10^{-13}$
D12	1.0	50	-	0.1	13	$3.4 \cdot 10^{-14}$
D13	1.0	50	-	0.2	13	$9.6 \cdot 10^{-16}$
E01	0.7	3	-	0.1	1313	$1.9 \cdot 10^{-12}$
E02	1.5	50	-	0.2	1313	$4.5 \cdot 10^{-16}$
E03	0.7	1	6	-	1313	$7.3 \cdot 10^{-16}$
E04	1.5	2	7	-	1313	$1.4 \cdot 10^{-12}$

Appendix C

See-saw Induced $\mu \longrightarrow e + \gamma$ Branching Ratio from Albright-Barr SO(10) Grand Unified Theory - Technical Details

A version of this chapter has been published [1].

In this appendix, we clarify some of the calculational details. We carefully explain our notation and conventions. Also, we include the full one loop amplitude for the rate $\mu \rightarrow e\gamma$ that we used in our calculations. Formulas similar to those given in sections C.2-C.6 can be found in [2].

We express the supersymmetric Lagrangian using the 2-component Weyl formalism. $\mathbf{L}^\alpha = (L_1^\alpha, L_2^\alpha, L_3^\alpha)^{\mathrm{T}}$ denotes a column vector in generation space containing the SU(2) doublet lepton chiral superfields; 1,2,3 are generation labels, and $\alpha = 1, 2$ are the SU(2) indices. $\mathbf{E} = (E_1, E_2, E_3)$ denotes a row vector in generation space containing SU(2) singlet charged lepton superfields. The gauge singlet neutrino chiral superfields are denoted by $\mathbf{N} = (N_1, N_2, N_3)$. Similarly, for the quark superfields: $\mathbf{Q}^\alpha = (Q_1^\alpha, Q_2^\alpha, Q_3^\alpha)^{\mathrm{T}}$ denotes the SU(2) doublet, $\mathbf{Q}^1 = \mathbf{u} = (u_1, u_2, u_3)^{\mathrm{T}}$, $\mathbf{Q}^2 = \mathbf{d} = (d_1, d_2, d_3)^{\mathrm{T}}$; and the SU(2) singlet quark superfields are $\mathbf{U} = (U_1, U_2, U_3)$, $\mathbf{D} = (D_1, D_2, D_3)$. H_{d}^α, H_{u}^α are the SU(2) Higgs doublet superfields of opposite hypercharge with the standard components: $H_{\mathrm{d}}^{\alpha=1} = H_{\mathrm{d}}^0$, $H_{\mathrm{d}}^{\alpha=2} = H_{\mathrm{d}}^-$, $H_{\mathrm{u}}^{\alpha=1} = H_{\mathrm{u}}^+$, $H_{\mathrm{u}}^{\alpha=2} = H_{\mathrm{u}}^0$. The corresponding scalar components of the superfields are written respectively as $\tilde{\mathbf{L}}^\alpha$, $\tilde{\mathbf{L}}^1 = \tilde{\nu}$, $\tilde{\mathbf{L}}^2 = \tilde{\mathbf{e}}$; $\tilde{\mathbf{E}}$; $\tilde{\mathbf{N}}$; $\tilde{\mathbf{Q}}^\alpha$, $\tilde{\mathbf{Q}}^1 = \tilde{\mathbf{u}}$, $\tilde{\mathbf{Q}}^2 = \tilde{\mathbf{d}}$; $\tilde{\mathbf{D}}$; $\tilde{\mathbf{U}}$; (all are vectors in generation space). The fermionic components of the

Higgs superfield, the Higgsinos, are denoted as $\tilde{H}_d^\alpha$, $\tilde{H}_u^\alpha$. The superpotential W is given by

$$\begin{aligned} W = & \quad \epsilon_{\alpha\beta} H_d^\alpha \mathbf{E} \mathbf{Y}_E \mathbf{L}^\beta + \epsilon_{\alpha\beta} H_u^\alpha \mathbf{N} \mathbf{Y}_N \mathbf{L}^\beta + \frac{1}{2} \mathbf{N} \mathbf{M}_N \mathbf{N} \\ & + \epsilon_{\alpha\beta} H_d^\alpha \mathbf{D} \mathbf{Y}_D \mathbf{Q}^\beta + \epsilon_{\alpha\beta} H_u^\alpha \mathbf{U} \mathbf{Y}_U \mathbf{Q}^\beta \\ & + \mu \epsilon_{\alpha\beta} H_d^\alpha H_u^\beta \end{aligned} \tag{C.1}$$

where $\mathbf{Y}_E$, $\mathbf{Y}_N$, $\mathbf{Y}_D$, $\mathbf{Y}_U$ are Yukawa matrices $\mathbf{M}_N$ is the singlet Majorana neutrino mass matrix, μ is the Higgs parameter that breaks the U(1) Pecci-Quinn symmetry, and the totally antisymmetric symbol is defined $\epsilon_{12} = +1$. The soft supersymmetry breaking Lagrangian is

$$\begin{aligned} \mathcal{L}_{\text{breaking}} = & -\delta_{\alpha\beta} \tilde{\mathbf{L}}^{\alpha\dagger} \mathbf{m}_{\tilde{L}}^2 \tilde{\mathbf{L}}^\beta - \tilde{\mathbf{E}} \mathbf{m}_{\tilde{E}}^2 \tilde{\mathbf{E}}^\dagger - \tilde{\mathbf{N}} \mathbf{m}_{\tilde{N}}^2 \tilde{\mathbf{N}}^\dagger \\ & -\delta_{\alpha\beta} \tilde{\mathbf{Q}}^{\alpha\dagger} \mathbf{m}_{\tilde{Q}}^2 \tilde{\mathbf{Q}}^\beta - \tilde{\mathbf{D}} \mathbf{m}_{\tilde{D}}^2 \tilde{\mathbf{D}}^\dagger - \tilde{\mathbf{U}} \mathbf{m}_{\tilde{U}}^2 \tilde{\mathbf{U}}^\dagger \\ & -m_{H_d}^2 \delta_{\alpha\beta} H_d^{\alpha *} H_d^\beta - m_{H_u}^2 \delta_{\alpha\beta} H_u^{\alpha *} H_u^\beta \\ & + \left(-b \epsilon_{\alpha\beta} H_d^\alpha H_u^\beta - \frac{1}{2} \tilde{\mathbf{N}} \mathbf{B}_{\tilde{N}} \tilde{\mathbf{N}} + \text{c. c.} \right) \\ & + \left(-\epsilon_{\alpha\beta} H_d^\alpha \tilde{\mathbf{E}} \mathbf{A}_E \tilde{\mathbf{L}}^\beta - \epsilon_{\alpha\beta} H_u^\alpha \tilde{\mathbf{N}} \mathbf{A}_N \tilde{\mathbf{L}}^\beta + \text{c. c.} \right) \\ & + \left(-\epsilon_{\alpha\beta} H_d^\alpha \tilde{\mathbf{D}} \mathbf{A}_D \tilde{\mathbf{Q}}^\beta - \epsilon_{\alpha\beta} H_u^\alpha \tilde{\mathbf{U}} \mathbf{A}_U \tilde{\mathbf{Q}}^\beta + \text{c. c.} \right) \\ & + \left(-\frac{1}{2} M_1 \tilde{B} \tilde{B} - \frac{1}{2} M_2 \tilde{W}^a \tilde{W}^a - \frac{1}{2} M_3 \tilde{G}^b \tilde{G}^b + \text{c. c.} \right) \end{aligned} \tag{C.2}$$

where $\tilde{B}$ denotes electroweak U(1) gaugino field; $\tilde{W}^a$, $a = 1, 2, 3$, denote electroweak SU(2) gaugino fields; $\tilde{G}^b$, $b = 1, ..., 8$, denote strong interaction, SU(3), gaugino fields; $\mathbf{m}_{\tilde{L}}^2$, $\mathbf{m}_{\tilde{E}}^2$, $\mathbf{m}_{\tilde{N}}^2$, $\mathbf{m}_{\tilde{Q}}^2$, $\mathbf{m}_{\tilde{D}}^2$, $\mathbf{m}_{\tilde{U}}^2$, $\mathbf{B}_\nu$, $\mathbf{A}_E$, $\mathbf{A}_N$, $\mathbf{A}_D$, $\mathbf{A}_U$, $m_{H_d}^2$, $m_{H_u}^2$, b, M_1, M_2, M_3 are the supersymmetry breaking parameters, and at the GUT scale:

$$\mathbf{m}_{\tilde{L}}^2 = \mathbf{m}_{\tilde{E}}^2 = \mathbf{m}_{\tilde{N}}^2 = \mathbf{m}_{\tilde{Q}}^2 = \mathbf{m}_{\tilde{D}}^2 = \mathbf{m}_{\tilde{U}}^2 = m_0^2 \cdot \mathbf{I}, \tag{C.3}$$

$$m_{H_d}^2 = m_{H_u}^2 = m_0^2, \tag{C.4}$$

$$\mathbf{A}_E = \mathbf{A}_N = \mathbf{A}_D = \mathbf{A}_U = 0, \tag{C.5}$$

$$M_1 = M_2 = M_3 = m_{1/2} \tag{C.6}$$

where m_0 and $m_{1/2}$ denote the universal scalar mass and the universal gaugino mass respectively ($\mathbf{I}$ is the 3×3 unit matrix). After running the CMSSM RGEs (see section C.6), we rotate all the Yukawa couplings to a diagonal basis, and in particular the lepton sector,

$$\mathbf{Y}_E \quad \rightarrow \quad \mathbf{U}_E^* \mathbf{Y}_E \mathbf{V}_E^\dagger = \text{diagonal}, \tag{C.7}$$

$$\mathbf{m}_{\tilde{L}}^2 \quad \rightarrow \quad \mathbf{V}_E \mathbf{m}_{\tilde{L}}^2 \mathbf{V}_E^\dagger, \tag{C.8}$$

$$\mathbf{m}_{\tilde{E}}^2 \quad \rightarrow \quad \mathbf{U}_E^* \mathbf{m}_{\tilde{E}}^2 \mathbf{U}_E^T, \tag{C.9}$$

$$\mathbf{A}_E \quad \rightarrow \quad \mathbf{U}_E^* \mathbf{A}_E \mathbf{V}_E^\dagger. \tag{C.10}$$

Not all of the bi-unitary rotation matrices can be absorbed away through the field re-definitions as the left-handed neutrinos become massive below the see-saw scale and after electroweak symmetry breaking.

C.1 μ parameter

The scalar potential of the Higgs fields is given at its minimum by

$$\begin{aligned} V &= \left(\mu^2 + m_{\mathrm{H_d}}^2\right)\left\langle H_{\mathrm{d}}^0\right\rangle^2 + \left(\mu^2 + m_{\mathrm{H_u}}^2\right)\left\langle H_{\mathrm{u}}^0\right\rangle^2 \\ &+ b\left\langle H_{\mathrm{d}}^0\right\rangle\left\langle H_{\mathrm{u}}^0\right\rangle + b^*\left\langle H_{\mathrm{d}}^0\right\rangle\left\langle H_{\mathrm{u}}^0\right\rangle \\ &+ \frac{g_1^2 + g_2^2}{8}\left(\left\langle H_{\mathrm{u}}^0\right\rangle^2 - \left\langle H_{\mathrm{d}}^0\right\rangle^2\right)^2 \end{aligned} \tag{C.11}$$

where g_1, g_2 are respectively U(1) and SU(2) gauge coupling constants. We can use the SU(2) gauge transformation freedom to choose the vacuum expectation value of the charged Higgs field $\left\langle H_{\mathrm{d}}^-\right\rangle = 0$; then it follows that also $\left\langle H_{\mathrm{u}}^+\right\rangle = 0$ at the minimum of the Higgs potential. Therefore, we are left with only the neutral Higgs fields of equation C.11. The conditions that the minimum of the potential V breaks the electroweak symmetry properly are

$$\mu^2 + m_{\mathrm{H_d}}^2 + b\tan\beta = -\frac{1}{2}m_{\mathrm{Z}}^2\cos 2\beta, \tag{C.12}$$

$$\mu^2 + m_{\mathrm{H_u}}^2 + b\cot\beta = \frac{1}{2}m_{\mathrm{Z}}^2\cos 2\beta \tag{C.13}$$

where m_{Z} is the mass of the Z-boson. After eliminating the terms containing b we obtain the tree level μ parameter relation,

$$\mu^2 = -\frac{1}{2}m_{\mathrm{Z}}^2 - \frac{m_{\mathrm{H_d}}^2 - m_{\mathrm{H_u}}^2\tan^2\beta}{1-\tan^2\beta}. \tag{C.14}$$

C.2 Neutralinos

The neutralinos $\tilde{\chi}_1^0$, $\tilde{\chi}_2^0$, $\tilde{\chi}_3^0$, $\tilde{\chi}_4^0$ are mass eigenstates of the neutral gauginos $\tilde{B}$, $\tilde{W}^3$ and neutral Higgsinos $\tilde{H}_{\mathrm{d}}^0$, $\tilde{H}_{\mathrm{u}}^0$. The neutralino mass Lagrangian is given by

$$\mathcal{L} = -\left(\tilde{B}\ \tilde{W}^3\ \tilde{H}_{\mathrm{d}}^0\ \tilde{H}_{\mathrm{u}}^0\right)\mathrm{M_{ne}}\begin{pmatrix}\tilde{B}\\ \tilde{W}^3\\ \tilde{H}_{\mathrm{d}}^0\\ \tilde{H}_{\mathrm{u}}^0\end{pmatrix} + \text{c. c.} \tag{C.15}$$

where

$$\mathbf{M}_{\rm ne} = \begin{pmatrix} M_1 & 0 & -m_{\rm Z}\cos\beta\sin\theta_{\rm W} & m_{\rm Z}\sin\beta\sin\theta_{\rm W} \\ 0 & M_2 & m_{\rm Z}\cos\beta\cos\theta_{\rm W} & -m_{\rm Z}\sin\beta\cos\theta_{\rm W} \\ -m_{\rm Z}\cos\beta\sin\theta_{\rm W} & m_{\rm Z}\cos\beta\cos\theta_{\rm W} & 0 & -\mu \\ m_{\rm Z}\sin\beta\sin\theta_{\rm W} & -m_{\rm Z}\sin\beta\cos\theta_{\rm W} & -\mu & 0 \end{pmatrix}. \tag{C.16}$$

An orthonormal rotation leads to the mass eigenstates:

$$\begin{pmatrix} \tilde{\chi}_1^0 \\ \tilde{\chi}_2^0 \\ \tilde{\chi}_3^0 \\ \tilde{\chi}_4^0 \end{pmatrix} = \mathbf{O}_{\rm ne} \begin{pmatrix} \tilde{B} \\ \tilde{W}^3 \\ \tilde{H}_{\rm d}^0 \\ \tilde{H}_{\rm u}^0 \end{pmatrix} \tag{C.17}$$

where $\mathbf{O}_{\rm ne}$ is a real, orthogonal matrix. The mass matrix (C.16) can therefore be decomposed in terms of real mass eigenvalues, $M_{\tilde{\chi}_a^0}$, $a = 1, 2, 3, 4$,

$$\mathbf{M}_{\rm ne} = \mathbf{O}_{\rm ne}^{\rm T} {\rm diag}\left(M_{\tilde{\chi}_1^0}\, M_{\tilde{\chi}_2^0}\, M_{\tilde{\chi}_3^0}\, M_{\tilde{\chi}_4^0}\right) \mathbf{O}_{\rm ne}, \tag{C.18}$$

and (C.15) can be rewritten as

$$\mathcal{L} = -\frac{1}{2}\sum_{a=1}^{4} M_{\tilde{\chi}_a^0}\tilde{\chi}_a^0\tilde{\chi}_a^0. \tag{C.19}$$

C.3 Charginos

The charginos are mass eigenstates of the charged SU(2) gauginos and charged Higgsinos,

$$\mathcal{L} = -\left(\tilde{W}^+\, \tilde{H}_{\rm u}^+\right) \mathbf{M}_{\rm C} \begin{pmatrix} \tilde{W}^- \\ \tilde{H}_{\rm d}^- \end{pmatrix} + {\rm c.\,c.} \tag{C.20}$$

where

$$\tilde{W}^\pm = \frac{\tilde{W}^1 \mp i\tilde{W}^2}{\sqrt{2}}, \tag{C.21}$$

and the mass matrix is

$$\mathbf{M}_{\rm C} = \begin{pmatrix} M_2 & \sqrt{2}m_{\rm W}\cos\beta \\ \sqrt{2}m_{\rm W}\sin\beta & \mu \end{pmatrix} \tag{C.22}$$

($m_{\rm W}$ is the W-boson mass). The mass eigenstates are given by

$$\begin{pmatrix} \tilde{\chi}_1^- \\ \tilde{\chi}_2^- \end{pmatrix} = \mathbf{O}_{\rm L} \begin{pmatrix} \tilde{W}^- \\ \tilde{H}_{\rm d}^- \end{pmatrix}, \qquad \begin{pmatrix} \tilde{\chi}_1^+ \\ \tilde{\chi}_2^+ \end{pmatrix} = \mathbf{O}_{\rm R} \begin{pmatrix} \tilde{W}^+ \\ \tilde{H}_{\rm u}^+ \end{pmatrix} \tag{C.23}$$

where $\mathbf{O_R}$ and $\mathbf{O_L}$ are real orthogonal matrices, and they can be chosen so that the mass eigenvalues $M_{\tilde{\chi}_1^-}$, $M_{\tilde{\chi}_2^-}$ are positive, and

$$\mathbf{M_C} = \mathbf{O_R^T} \mathrm{diag}\left(M_{\tilde{\chi}_1^-}\ M_{\tilde{\chi}_2^-}\right) \mathbf{O_L}, \tag{C.24}$$

Equation C.20 can be written as

$$\mathcal{L} = -M_{\tilde{\chi}_1^-} \tilde{\chi}_1^+ \tilde{\chi}_1^- - M_{\tilde{\chi}_2^-} \tilde{\chi}_2^+ \tilde{\chi}_2^- \ + \text{ c. c.} \tag{C.25}$$

C.4 Sleptons

Masses of the charged sleptons are given by the Lagrangian

$$\mathcal{L} = -\tilde{\mathbf{e}}^\dagger \mathbf{m}_{\mathrm{LL}}^2 \tilde{\mathbf{e}} - \tilde{\mathbf{e}}^\dagger \mathbf{m}_{\mathrm{RL}}^{2\dagger} \tilde{\mathbf{E}}^\dagger - \tilde{\mathbf{E}} \mathbf{m}_{\mathrm{RL}}^2 \tilde{\mathbf{e}} - \tilde{\mathbf{E}} \mathbf{m}_{\mathrm{RR}}^2 \tilde{\mathbf{E}}^\dagger \tag{C.26}$$

with the mass matrices

$$\mathbf{m}_{\mathrm{LL}}^2 = \mathbf{m}_{\mathrm{l}}^2 + \mathbf{m}_{\tilde{\mathrm{L}}}^2 + m_{\mathrm{Z}}^2 \cos 2\beta \left(\sin^2 \theta_{\mathrm{W}} - \frac{1}{2}\right) \cdot \mathbf{I}, \tag{C.27}$$

$$\mathbf{m}_{\mathrm{RR}}^2 = \mathbf{m}_{\mathrm{l}}^2 + \mathbf{m}_{\tilde{\mathrm{E}}}^2 - m_{\mathrm{Z}}^2 \cos 2\beta \sin^2 \theta_{\mathrm{W}} \cdot \mathbf{I}, \tag{C.28}$$

$$\mathbf{m}_{\mathrm{RL}}^2 = -\mu \mathbf{m}_{\mathrm{l}} \tan\beta + \frac{v \cos\beta}{\sqrt{2}} \mathbf{A}_{\mathrm{E}} \tag{C.29}$$

where

$$\mathbf{m}_{\mathrm{l}} = \mathrm{diag}\left(m_{\mathrm{l}_1}\ m_{\mathrm{l}_2}\ m_{\mathrm{l}_3}\right), \tag{C.30}$$

and m_{l_1}, m_{l_2}, m_{l_3} are electron, muon, and tau masses respectively. The above Lagrangian written in terms of mass eigenstates $\tilde{f}_1, ..., \tilde{f}_6$ (six complex scalar fields) is

$$\mathcal{L} = -\sum_{b=1}^{6} m_{\tilde{\mathrm{f}}_b}^2 \tilde{f}_b^* \tilde{f}_b \tag{C.31}$$

with

$$\begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \\ \tilde{f}_4 \\ \tilde{f}_5 \\ \tilde{f}_6 \end{pmatrix} = \mathbf{U}_{\tilde{\mathrm{f}}} \begin{pmatrix} \tilde{e}_1 \\ \tilde{e}_2 \\ \tilde{e}_3 \\ \tilde{E}_1^* \\ \tilde{E}_2^* \\ \tilde{E}_3^* \end{pmatrix}, \tag{C.32}$$

and $\mathbf{U}_{\tilde{\mathrm{f}}}$ is a complex unitary matrix defined by

$$\begin{pmatrix} \mathbf{m}_{\mathrm{LL}}^2 & \mathbf{m}_{\mathrm{RL}}^{2\dagger} \\ \mathbf{m}_{\mathrm{RL}}^2 & \mathbf{m}_{\mathrm{RR}}^2 \end{pmatrix} = \mathbf{U}_{\tilde{\mathrm{f}}}^\dagger \mathrm{diag}\left(m_{\tilde{\mathrm{f}}_1}^2\ m_{\tilde{\mathrm{f}}_2}^2\ m_{\tilde{\mathrm{f}}_3}^2\ m_{\tilde{\mathrm{f}}_4}^2\ m_{\tilde{\mathrm{f}}_5}^2\ m_{\tilde{\mathrm{f}}_6}^2\right) \mathbf{U}_{\tilde{\mathrm{f}}}. \tag{C.33}$$

Similarly, the light sneutrinos (the heavy singlet sneutrinos are ignored since they have decoupled well above the weak scale)

$$\mathcal{L} = -\tilde{\nu}^{\dagger}\mathbf{M}^2_{\tilde{\nu}}\tilde{\nu} \tag{C.34}$$

where

$$\mathbf{M}^2_{\tilde{\nu}} = \mathbf{m}^2_{\tilde{\mathrm{L}}} + \frac{1}{2}m^2_{\mathrm{Z}}\cos 2\beta \cdot \mathbf{I}. \tag{C.35}$$

The sneutrino mass Lagrangian written in terms of mass eigenstates $\tilde{n}_1$, $\tilde{n}_2$, $\tilde{n}_3$ (three complex scalar fields) reads

$$\mathcal{L} = -\sum_{b=1}^{3} m^2_{\tilde{\mathrm{n}}_b}\tilde{n}^*_b\tilde{n}_b \tag{C.36}$$

with the mass eigenstates defined by

$$\begin{pmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{pmatrix} = \mathbf{U}_{\tilde{\mathrm{n}}} \begin{pmatrix} \tilde{\nu}_1 \\ \tilde{\nu}_2 \\ \tilde{\nu}_3 \end{pmatrix}, \tag{C.37}$$

and $\mathbf{U}_{\tilde{\mathrm{n}}}$ is a complex unitary matrix satisfying

$$\mathbf{M}^2_{\tilde{\nu}} = \mathbf{U}^{\dagger}_{\tilde{\mathrm{n}}}\mathrm{diag}\left(m^2_{\tilde{\mathrm{n}}_1}\ m^2_{\tilde{\mathrm{n}}_2}\ m^2_{\tilde{\mathrm{n}}_3}\right)\mathbf{U}_{\tilde{\mathrm{n}}}. \tag{C.38}$$

C.5 Lepton Flavour Violating Interactions

The interactions leading to the lepton flavour violating process $l_j \rightarrow l_i + \gamma$ involve two effective Lagrangians: neutralino-lepton-slepton and chargino-lepton-sneutrino. Written in the mass eigenbasis they are

$$\mathcal{L} = \sum_{i=1}^{3}\sum_{a=1}^{4}\sum_{b=1}^{6} N^{\mathrm{L}}_{iab}\tilde{f}_b E_i\tilde{\chi}^0_a + N^{\mathrm{R}*}_{iab}\tilde{f}^*_b e_i\tilde{\chi}^0_a \ + \ \mathrm{c.\,c.} \tag{C.39}$$

and

$$\mathcal{L} = \sum_{i=1}^{3}\sum_{a=1}^{2}\sum_{b=1}^{3} C^{\mathrm{L}}_{iab}\tilde{\nu}_b E_i\tilde{\chi}^-_a + C^{\mathrm{R}*}_{iab}\tilde{\nu}^*_b e_i\tilde{\chi}^+_a \ + \ \mathrm{c.\,c.} \tag{C.40}$$

where

$$N^{\mathrm{L}}_{iab} = -\frac{g_2}{\sqrt{2}}\left(2\tan\theta_{\mathrm{W}}\left(\mathbf{U}_{\tilde{\mathrm{f}}}\right)^*_{b(i+3)}\left(\mathbf{O}_{\mathrm{ne}}\right)_{a1} + \frac{m_{\mathrm{l}_i}}{m_{\mathrm{W}}\cos\beta}\left(\mathbf{U}_{\tilde{\mathrm{f}}}\right)^*_{bi}\left(\mathbf{O}_{\mathrm{ne}}\right)_{a3}\right), \tag{C.41}$$

$$N^{\mathrm{R}}_{iab} = \frac{g_2}{\sqrt{2}}\left(\tan\theta_{\mathrm{W}}\left(\mathbf{U}_{\tilde{\mathrm{f}}}\right)^*_{bi}\left(\mathbf{O}_{\mathrm{ne}}\right)_{a1} + \left(\mathbf{U}_{\tilde{\mathrm{f}}}\right)^*_{bi}\left(\mathbf{O}_{\mathrm{ne}}\right)_{a2} - \frac{m_{\mathrm{l}_i}}{m_{\mathrm{W}}\cos\beta}\left(\mathbf{U}_{\tilde{\mathrm{f}}}\right)^*_{b(i+3)}\left(\mathbf{O}_{\mathrm{ne}}\right)_{a3}\right), \tag{C.42}$$

and

$$C^{\mathrm{L}}_{iab} = \frac{g_2 m_{l_i}}{\sqrt{2} m_{\mathrm{W}} \cos\beta} \left(\mathbf{O}_{\mathrm{L}}\right)_{a2} \left(\mathbf{U}_{\tilde{\mathrm{n}}}\right)^*_{bi}, \tag{C.43}$$

$$C^{\mathrm{R}}_{iab} = -g_2 \left(\mathbf{O}_{\mathrm{R}}\right)_{a1} \left(\mathbf{U}_{\tilde{\mathrm{n}}}\right)^*_{bi}. \tag{C.44}$$

The on-shell amplitude for $l_j \to l_i + \gamma$ can be written in the general form

$$\mathcal{M} = e\epsilon^*_\mu \bar{l}_i \left(p-q\right) \left(i m_{l_j} \sigma^{\mu\nu} q_\nu \left(A_{\mathrm{L}}\mathrm{L} + A_{\mathrm{R}}\mathrm{R}\right)\right) l_j \left(p\right); \tag{C.45}$$

here we have used Dirac spinors $l_i\left(p-q\right)$ and $l_j\left(p\right)$ for the charged leptons i and j with momenta $p-q$ and p, respectively; $\mathrm{L} = \left(1-\gamma^5\right)/2$ and $\mathrm{R} = \left(1+\gamma^5\right)/2$. Each of the dipole coefficients A_{L} and A_{R} have contributions from the neutralino-lepton-slepton and the chargino-lepton-sneutrino interaction, namely,

$$A_{\mathrm{L}} = A^{(\mathrm{n})}_{\mathrm{L}} + A^{(\mathrm{c})}_{\mathrm{L}}, \tag{C.46}$$

$$A_{\mathrm{R}} = A^{(\mathrm{n})}_{\mathrm{R}} + A^{(\mathrm{c})}_{\mathrm{R}} \tag{C.47}$$

where $A^{(\mathrm{n})}_{\mathrm{L}}$, $A^{(\mathrm{n})}_{\mathrm{R}}$, $A^{(\mathrm{c})}_{\mathrm{L}}$, $A^{(\mathrm{c})}_{\mathrm{R}}$ can be evaluated from the Feynman diagrams in figure 5.2;

$$A^{(\mathrm{n})}_{\mathrm{L}} = \frac{1}{32\pi^2} \sum_{a=1}^{4} \sum_{b=1}^{6} \frac{1}{m^2_{\tilde{l}_b}} \left(N^{\mathrm{L}}_{iab} N^{\mathrm{L}*}_{jab} J_1\left(\frac{M^2_{\tilde{\chi}^0_a}}{m^2_{\tilde{l}_b}}\right) + N^{\mathrm{L}}_{iab} N^{\mathrm{R}*}_{jab} \frac{\left|M_{\tilde{\chi}^0_a}\right|}{m_{l_j}} J_2\left(\frac{M^2_{\tilde{\chi}^0_a}}{m^2_{\tilde{l}_b}}\right)\right), \tag{C.48}$$

$$A^{(\mathrm{c})}_{\mathrm{L}} = -\frac{1}{32\pi^2} \sum_{a=1}^{2} \sum_{b=1}^{3} \frac{1}{m^2_{\tilde{\nu}_b}} \left(C^{\mathrm{L}}_{iab} C^{\mathrm{L}*}_{jab} J_3\left(\frac{M^2_{\tilde{\chi}^-_a}}{m^2_{\tilde{\nu}_b}}\right) + C^{\mathrm{L}}_{iab} C^{\mathrm{R}*}_{jab} \frac{M_{\tilde{\chi}^-_a}}{m_{l_j}} J_4\left(\frac{M^2_{\tilde{\chi}^-_a}}{m^2_{\tilde{\nu}_b}}\right)\right), \tag{C.49}$$

$$A^{(\mathrm{n})}_{\mathrm{R}} = A^{(\mathrm{n})}_{\mathrm{L}}\Big|_{L\leftrightarrow R}, \tag{C.50}$$

$$A^{(\mathrm{c})}_{\mathrm{R}} = A^{(\mathrm{c})}_{\mathrm{L}}\Big|_{L\leftrightarrow R}. \tag{C.51}$$

The functions $J_1\left(x\right)$, $J_2\left(x\right)$, $J_3\left(x\right)$, $J_4\left(x\right)$ are defined as

$$J_1\left(x\right) = \frac{1 - 6x + 3x^2 + 2x^3 - 6x^2 \ln x}{6\left(1-x\right)^4}, \tag{C.52}$$

$$J_2\left(x\right) = \frac{1 - x^2 + 2x \ln x}{\left(1-x\right)^3}, \tag{C.53}$$

$$J_3\left(x\right) = \frac{2 + 3x - 6x^2 + x^3 + 6x \ln x}{6\left(1-x\right)^4}, \tag{C.54}$$

$$J_4\left(x\right) = \frac{-3 + 4x - x^2 + 2\ln x}{\left(1-x\right)^3}. \tag{C.55}$$

Finally, the decay rate for $l^-_j \to l^-_i + \gamma$ is given by

$$\Gamma\left(l^-_j \to l^-_i + \gamma\right) = \frac{e^2}{16\pi} m^5_{l_j} \left(\left|A_{\mathrm{L}}\right|^2 + \left|A_{\mathrm{R}}\right|^2\right), \tag{C.56}$$

and $i = 1, j = 2$ for $\mu \to e + \gamma$.

C.6 Renormalization group equations (RGEs)

The general form of the supersymmetric renormalization group equations [3, 4, 2] is

$$\frac{dX}{dt} = \frac{1}{16\pi^2}\dot{X} \tag{C.57}$$

where X is any of g_1, g_2, g_3, $\mathbf{Y_N}$, $\mathbf{Y_E}$, $\mathbf{Y_U}$, $\mathbf{Y_D}$, M_1, M_2, M_3, $m^2_{H_u}$, $m^2_{H_d}$, $\mathbf{m^2_{\tilde{L}}}$, $\mathbf{m^2_{\tilde{N}}}$, $\mathbf{m^2_{\tilde{E}}}$, $\mathbf{m^2_{\tilde{Q}}}$, $\mathbf{m^2_{\tilde{U}}}$, $\mathbf{m^2_{\tilde{D}}}$, $\mathbf{A_N}$, $\mathbf{A_E}$, $\mathbf{A_U}$, $\mathbf{A_D}$, and the dotted quantities are listed below:

$$\dot{g}_1 = 11g_1^3, \tag{C.58}$$

$$\dot{g}_2 = g_2^3, \tag{C.59}$$

$$\dot{g}_3 = -3g_3^3, \tag{C.60}$$

(C.61)

$$\dot{\mathbf{Y}}_\mathrm{N} = \mathbf{Y}_\mathrm{N}\left(-g_1^2\mathbf{I} - 3g_2^2\mathbf{I} + 3\mathrm{Tr}\left(\mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U}\right)\mathbf{I} + \mathrm{Tr}\left(\mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N}\right)\mathbf{I} + 3\mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N} + \mathbf{Y}_\mathrm{E}^\dagger\mathbf{Y}_\mathrm{E}\right), \tag{C.62}$$

$$\dot{\mathbf{Y}}_\mathrm{E} = \mathbf{Y}_\mathrm{E}\left(-3g_1^2\mathbf{I} - 3g_2^2\mathbf{I} + 3\mathrm{Tr}\left(\mathbf{Y}_\mathrm{D}^\dagger\mathbf{Y}_\mathrm{D}\right)\mathbf{I} + \mathrm{Tr}\left(\mathbf{Y}_\mathrm{E}^\dagger\mathbf{Y}_\mathrm{E}\right)\mathbf{I} + 3\mathbf{Y}_\mathrm{E}^\dagger\mathbf{Y}_\mathrm{E} + \mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N}\right), \tag{C.63}$$

$$\begin{aligned}\dot{\mathbf{Y}}_\mathrm{U} = \mathbf{Y}_\mathrm{U}\Big(&-\frac{13}{9}g_1^2\mathbf{I} - 3g_2^2\mathbf{I} - \frac{16}{3}g_3^2\mathbf{I} + 3\mathrm{Tr}\left(\mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U}\right)\mathbf{I} + \mathrm{Tr}\left(\mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N}\right)\mathbf{I} \\ &+ 3\mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U} + \mathbf{Y}_\mathrm{D}^\dagger\mathbf{Y}_\mathrm{D}\Big),\end{aligned} \tag{C.64}$$

$$\begin{aligned}\dot{\mathbf{Y}}_\mathrm{D} = \mathbf{Y}_\mathrm{D}\Big(&-\frac{7}{9}g_1^2\mathbf{I} - 3g_2^2\mathbf{I} - \frac{16}{3}g_3^2\mathbf{I} + 3\mathrm{Tr}\left(\mathbf{Y}_\mathrm{D}^\dagger\mathbf{Y}_\mathrm{D}\right)\mathbf{I} + \mathrm{Tr}\left(\mathbf{Y}_\mathrm{E}^\dagger\mathbf{Y}_\mathrm{E}\right)\mathbf{I} \\ &+ 3\mathbf{Y}_\mathrm{D}^\dagger\mathbf{Y}_\mathrm{D} + \mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U}\Big),\end{aligned} \tag{C.65}$$

$$\dot{M}_1 = 22g_1^2M_1, \tag{C.66}$$

$$\dot{M}_2 = 2g_2^2M_2, \tag{C.67}$$

$$\dot{M}_3 = -6g_3^2M_3, \tag{C.68}$$

$$S = m^2_{H_u} - m^2_{H_d} + \mathrm{Tr}\left(\mathbf{m^2_{\tilde{Q}}} - 2\mathbf{m^2_{\tilde{U}}} + \mathbf{m^2_{\tilde{D}}} - \mathbf{m^2_{\tilde{L}}} + \mathbf{m^2_{\tilde{E}}}\right), \tag{C.69}$$

$$\begin{aligned}\dot{m}^2_{H_u} = &\ 6\mathrm{Tr}\left(\mathbf{m^2_{\tilde{Q}}}\mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U} + \mathbf{Y}_\mathrm{U}^\dagger\mathbf{m^2_{\tilde{U}}}\mathbf{Y}_\mathrm{U} + m^2_{H_u}\mathbf{Y}_\mathrm{U}^\dagger\mathbf{Y}_\mathrm{U} + \mathbf{A}_\mathrm{U}^\dagger\mathbf{A}_\mathrm{U}\right) \\ &+2\mathrm{Tr}\left(\mathbf{m^2_{\tilde{L}}}\mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N} + \mathbf{Y}_\mathrm{N}^\dagger\mathbf{m^2_{\tilde{N}}}\mathbf{Y}_\mathrm{N} + m^2_{H_u}\mathbf{Y}_\mathrm{N}^\dagger\mathbf{Y}_\mathrm{N} + \mathbf{A}_\mathrm{N}^\dagger\mathbf{A}_\mathrm{N}\right)\end{aligned}$$

$$-2g_1^2M_1^2 - 6g_2^2M_2^2 + g_1^2S, \tag{C.70}$$

$$\begin{aligned}\dot{m}^2_{\mathrm{H_d}} &= 2\mathrm{Tr}\left(\mathbf{m}^2_{\tilde{\mathrm{L}}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}} + \mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{m}^2_{\tilde{\mathrm{E}}}\mathbf{Y}_{\mathrm{E}} + m^2_{\mathrm{H_d}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}} + \mathbf{A}^\dagger_{\mathrm{E}}\mathbf{A}_{\mathrm{E}}\right)\\ &\quad +6\mathrm{Tr}\left(\mathbf{m}^2_{\tilde{\mathrm{Q}}}\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}} + \mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{m}^2_{\tilde{\mathrm{D}}}\mathbf{Y}_{\mathrm{D}} + m^2_{\mathrm{H_d}}\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}} + \mathbf{A}^\dagger_{\mathrm{D}}\mathbf{A}_{\mathrm{D}}\right)\\ &\quad -2g_1^2M_1^2 - 6g_2^2M_2^2 - g_1^2S,\end{aligned} \tag{C.71}$$

$$\begin{aligned}\dot{\mathbf{m}}^2_{\tilde{\mathrm{L}}} &= \mathbf{m}^2_{\tilde{\mathrm{L}}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}} + \mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}}\mathbf{m}^2_{\tilde{\mathrm{L}}} + \mathbf{m}^2_{\tilde{\mathrm{L}}}\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{Y}_{\mathrm{N}} + \mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{Y}_{\mathrm{N}}\mathbf{m}^2_{\tilde{\mathrm{L}}}\\ &\quad +2\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{m}^2_{\tilde{\mathrm{E}}}\mathbf{Y}_{\mathrm{E}} + 2m^2_{\mathrm{H_d}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}} + 2\mathbf{A}^\dagger_{\mathrm{E}}\mathbf{A}_{\mathrm{E}}\\ &\quad +2\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{m}^2_{\tilde{\mathrm{N}}}\mathbf{Y}_{\mathrm{N}} + 2m^2_{\mathrm{H_u}}\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{Y}_{\mathrm{N}} + 2\mathbf{A}^\dagger_{\mathrm{N}}\mathbf{A}_{\mathrm{N}}\\ &\quad -2g_1^2M_1^2\mathbf{I} - 6g_2^2M_2^2\mathbf{I} - g_1^2S\mathbf{I},\end{aligned} \tag{C.72}$$

$$\dot{\mathbf{m}}^2_{\tilde{\mathrm{N}}} = 2\mathbf{m}^2_{\tilde{\mathrm{N}}}\mathbf{Y}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{N}} + 2\mathbf{Y}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{m}^2_{\tilde{\mathrm{N}}} + 4\mathbf{Y}_{\mathrm{N}}\mathbf{m}^2_{\tilde{\mathrm{L}}}\mathbf{Y}^\dagger_{\mathrm{N}} + 4m^2_{\mathrm{H_u}}\mathbf{Y}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{N}} + 4\mathbf{A}_{\mathrm{N}}\mathbf{A}^\dagger_{\mathrm{N}}, \tag{C.73}$$

$$\begin{aligned}\dot{\mathbf{m}}^2_{\tilde{\mathrm{E}}} &= 2\mathbf{m}^2_{\tilde{\mathrm{E}}}\mathbf{Y}_{\mathrm{E}}\mathbf{Y}^\dagger_{\mathrm{E}} + 2\mathbf{Y}_{\mathrm{E}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{m}^2_{\tilde{\mathrm{E}}} + 4\mathbf{Y}_{\mathrm{E}}\mathbf{m}^2_{\tilde{\mathrm{L}}}\mathbf{Y}^\dagger_{\mathrm{E}} + 4m^2_{\mathrm{H_d}}\mathbf{Y}_{\mathrm{E}}\mathbf{Y}^\dagger_{\mathrm{E}} + 4\mathbf{A}_{\mathrm{E}}\mathbf{A}^\dagger_{\mathrm{E}}\\ &\quad -8g_1^2M_1^2\mathbf{I} + 2g_1^2S\mathbf{I},\end{aligned} \tag{C.74}$$

$$\begin{aligned}\dot{\mathbf{m}}^2_{\tilde{\mathrm{Q}}} &= \mathbf{m}^2_{\tilde{\mathrm{Q}}}\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}} + \mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}}\mathbf{m}^2_{\tilde{\mathrm{Q}}} + 2\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{m}^2_{\tilde{\mathrm{U}}}\mathbf{Y}_{\mathrm{U}} + 2m^2_{\mathrm{H_u}}\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}} + 2\mathbf{A}^\dagger_{\mathrm{U}}\mathbf{A}_{\mathrm{U}}\\ &\quad +\mathbf{m}^2_{\tilde{\mathrm{Q}}}\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}} + \mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}}\mathbf{m}^2_{\tilde{\mathrm{Q}}} + 2\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{m}^2_{\tilde{\mathrm{D}}}\mathbf{Y}_{\mathrm{D}} + 2m^2_{\mathrm{H_d}}\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}} + 2\mathbf{A}^\dagger_{\mathrm{D}}\mathbf{A}_{\mathrm{D}}\\ &\quad -\frac{2}{9}g_1^2M_1^2\mathbf{I} - 6g_2^2M_2^2\mathbf{I} - \frac{32}{3}g_3^2M_3^2\mathbf{I} + \frac{1}{3}g_1^2S\mathbf{I},\end{aligned} \tag{C.75}$$

$$\begin{aligned}\dot{\mathbf{m}}^2_{\tilde{\mathrm{U}}} &= 2\mathbf{m}^2_{\tilde{\mathrm{U}}}\mathbf{Y}_{\mathrm{U}}\mathbf{Y}^\dagger_{\mathrm{U}} + 2\mathbf{Y}_{\mathrm{U}}\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{m}^2_{\tilde{\mathrm{U}}} + 4\mathbf{Y}_{\mathrm{U}}\mathbf{m}^2_{\tilde{\mathrm{Q}}}\mathbf{Y}^\dagger_{\mathrm{U}} + 4m^2_{\mathrm{H_u}}\mathbf{Y}_{\mathrm{U}}\mathbf{Y}^\dagger_{\mathrm{U}} + 4\mathbf{A}_{\mathrm{U}}\mathbf{A}^\dagger_{\mathrm{U}}\\ &\quad -\frac{32}{9}g_1^2M_1^2\mathbf{I} - \frac{32}{3}g_3^2M_3^2\mathbf{I} - \frac{4}{3}g_1^2S\mathbf{I},\end{aligned} \tag{C.76}$$

$$\begin{aligned}\dot{\mathbf{m}}^2_{\tilde{\mathrm{D}}} &= 2\mathbf{m}^2_{\tilde{\mathrm{D}}}\mathbf{Y}_{\mathrm{D}}\mathbf{Y}^\dagger_{\mathrm{D}} + 2\mathbf{Y}_{\mathrm{D}}\mathbf{Y}^\dagger_{\mathrm{D}}\mathbf{m}^2_{\tilde{\mathrm{D}}} + 4\mathbf{Y}_{\mathrm{D}}\mathbf{m}^2_{\tilde{\mathrm{Q}}}\mathbf{Y}^\dagger_{\mathrm{D}} + 4m^2_{\mathrm{H_d}}\mathbf{Y}_{\mathrm{D}}\mathbf{Y}^\dagger_{\mathrm{D}} + 4\mathbf{A}_{\mathrm{D}}\mathbf{A}^\dagger_{\mathrm{D}}\\ &\quad -\frac{8}{9}g_1^2M_1^2\mathbf{I} - \frac{32}{3}g_3^2M_3^2\mathbf{I} + \frac{2}{3}g_1^2S\mathbf{I},\end{aligned} \tag{C.77}$$

$$\begin{aligned}\dot{\mathbf{A}}_{\mathrm{N}} &= -g_1^2\mathbf{A}_{\mathrm{N}} - 3g_2^2\mathbf{A}_{\mathrm{N}} + 3\mathrm{Tr}\left(\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}}\right)\mathbf{A}_{\mathrm{N}} + \mathrm{Tr}\left(\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{Y}_{\mathrm{N}}\right)\mathbf{A}_{\mathrm{N}}\\ &\quad -2g_1^2M_1\mathbf{Y}_{\mathrm{N}} - 6g_2^2M_2\mathbf{Y}_{\mathrm{N}} + 6\mathrm{Tr}\left(\mathbf{Y}^\dagger_{\mathrm{U}}\mathbf{A}_{\mathrm{U}}\right)\mathbf{Y}_{\mathrm{N}} + 2\mathrm{Tr}\left(\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{A}_{\mathrm{N}}\right)\mathbf{Y}_{\mathrm{N}}\\ &\quad +4\mathbf{Y}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{A}_{\mathrm{N}} + 5\mathbf{A}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{N}}\mathbf{Y}_{\mathrm{N}} + 2\mathbf{Y}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{A}_{\mathrm{E}} + \mathbf{A}_{\mathrm{N}}\mathbf{Y}^\dagger_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}},\end{aligned} \tag{C.78}$$

$$\begin{aligned}\dot{\mathbf{A}}_{\mathrm{E}} &= -3g_1^2\mathbf{A}_{\mathrm{E}} - 3g_2^2\mathbf{A}_{\mathrm{E}} + 3\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{Y}_{\mathrm{D}}\right)\mathbf{A}_{\mathrm{E}} + \mathrm{Tr}\left(\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{Y}_{\mathrm{E}}\right)\mathbf{A}_{\mathrm{E}} \\ &\quad -6g_1^2M_1\mathbf{Y}_{\mathrm{E}} - 6g_2^2M_2\mathbf{Y}_{\mathrm{E}} + 6\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{A}_{\mathrm{D}}\right)\mathbf{Y}_{\mathrm{E}} + 2\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{A}_{\mathrm{E}}\right)\mathbf{Y}_{\mathrm{E}} \\ &\quad +4\mathbf{Y}_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{A}_{\mathrm{E}} + 5\mathbf{A}_{\mathrm{E}}\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{Y}_{\mathrm{E}} + 2\mathbf{Y}_{\mathrm{E}}\mathbf{Y}_{\mathrm{N}}^\dagger\mathbf{A}_{\mathrm{N}} + \mathbf{A}_{\mathrm{E}}\mathbf{Y}_{\mathrm{N}}^\dagger\mathbf{Y}_{\mathrm{N}}, \end{aligned} \tag{C.79}$$

$$\begin{aligned}\dot{\mathbf{A}}_{\mathrm{U}} &= -\frac{13}{9}g_1^2\mathbf{A}_{\mathrm{U}} - 3g_2^2\mathbf{A}_{\mathrm{U}} - \frac{16}{3}g_3^2\mathbf{A}_{\mathrm{U}} + 3\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{Y}_{\mathrm{U}}\right)\mathbf{A}_{\mathrm{U}} + \mathrm{Tr}\left(\mathbf{Y}_{\mathrm{N}}^\dagger\mathbf{Y}_{\mathrm{N}}\right)\mathbf{A}_{\mathrm{U}} \\ &\quad -\frac{26}{9}g_1^2M_1\mathbf{Y}_{\mathrm{U}} - 6g_2^2M_2\mathbf{Y}_{\mathrm{U}} - \frac{32}{3}g_3^2M_3\mathbf{Y}_{\mathrm{U}} + 6\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{A}_{\mathrm{U}}\right)\mathbf{Y}_{\mathrm{U}} + 2\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{N}}^\dagger\mathbf{A}_{\mathrm{N}}\right)\mathbf{Y}_{\mathrm{U}} \\ &\quad +4\mathbf{Y}_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{A}_{\mathrm{U}} + 5\mathbf{A}_{\mathrm{U}}\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{Y}_{\mathrm{U}} + 2\mathbf{Y}_{\mathrm{U}}\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{A}_{\mathrm{D}} + \mathbf{A}_{\mathrm{U}}\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{Y}_{\mathrm{D}}, \end{aligned} \tag{C.80}$$

$$\begin{aligned}\dot{\mathbf{A}}_{\mathrm{D}} &= -\frac{7}{9}g_1^2\mathbf{A}_{\mathrm{D}} - 3g_2^2\mathbf{A}_{\mathrm{D}} - \frac{16}{3}g_3^2\mathbf{A}_{\mathrm{D}} + 3\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{Y}_{\mathrm{D}}\right)\mathbf{A}_{\mathrm{D}} + \mathrm{Tr}\left(\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{Y}_{\mathrm{E}}\right)\mathbf{A}_{\mathrm{D}} \\ &\quad -\frac{14}{9}g_1^2M_1\mathbf{Y}_{\mathrm{D}} - 6g_2^2M_2\mathbf{Y}_{\mathrm{D}} - \frac{32}{3}g_3^2M_3\mathbf{Y}_{\mathrm{D}} + 6\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{A}_{\mathrm{D}}\right)\mathbf{Y}_{\mathrm{D}} + 2\mathrm{Tr}\left(\mathbf{Y}_{\mathrm{E}}^\dagger\mathbf{A}_{\mathrm{E}}\right)\mathbf{Y}_{\mathrm{D}} \\ &\quad +4\mathbf{Y}_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{A}_{\mathrm{D}} + 5\mathbf{A}_{\mathrm{D}}\mathbf{Y}_{\mathrm{D}}^\dagger\mathbf{Y}_{\mathrm{D}} + 2\mathbf{Y}_{\mathrm{D}}\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{A}_{\mathrm{U}} + \mathbf{A}_{\mathrm{D}}\mathbf{Y}_{\mathrm{U}}^\dagger\mathbf{Y}_{\mathrm{U}}. \end{aligned} \tag{C.81}$$

Bibliography

[1] Ernest Jankowski and David W. Maybury. Lepton flavour violation in a class of lopsided SO(10) models. *Phys. Rev.*, D70:035004, 2004.

[2] J. Hisano, T. Moroi, K. Tobe, and Masahiro Yamaguchi. Lepton-flavor violation via right-handed neutrino Yukawa couplings in supersymmetric standard model. *Phys. Rev.*, D53:2442–2459, 1996.

[3] Norbert K. Falck. Renormalization group equations for softly broken supersymmetry: The most general case. *Z. Phys.*, C30:247, 1986.

[4] Stephen P. Martin and Michael T. Vaughn. Two loop renormalization group equations for soft supersymmetry breaking couplings. *Phys. Rev.*, D50:2282, 1994.

Appendix D

List of Acronyms

AB	Albright-Barr
ATLAS	A Toroidal LHC Apparatus
ATLFAST	Atlas Fast Simulation Package
BR	branching ratio
CDF	Collider Detector at Fermilab
CKM	Cabibbo-Kobayashi-Maskawa
CMSSM	Constrained Minimal Supersymmetric Standard Model
COBE	Cosmic Background Explorer
GUT	Grand Unification Theory
LEP	Large Electron Positron Collider
LFV	Lepton Flavour Violation
LHC	Large Hadron Collider
LMA	Large Mixing Angle
LSP	Lightest Supersymmetric Particle
MECO	Muon to Electron Conversion
MNS	Maki-Nakagawa-Sakata
MSSM	Minimal Supersymmetric Standard Model
OPAL	Omni-Purpose Apparatus at LEP
QCD	Quantum Chromodynamics
QED	Quantum Electrodynamics
OJD	Optimal Jet Definition
OJF	Optimal Jet Finder
RGEs	Renormalization Group Equations
SM	Standard Model
SNO	Sudbury Neutrino Observatory
SUSY	supersymmetry
UV	ultraviolet
WMAP	Wilkinson Microwave Anisotropy Probe