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.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

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

Symbolic Computation of Electron-Proton to Slepton-Squark Scattering Cross Sections Based on a Left-Right Supersymmetric Extension of the Standard Model

Mark R. A. Adcock

A Thesis

in

The department of Physics

Presented in partial fulfillment of the requirements for

the Degree of Master of Science at

Concordia University

Montréal, Québec, Canada

March 1997

ABSTRACT

Symbolic Computation of Electron-Proton to Slepton-Squark Scattering Cross Sections Based on a Left-Right Supersymmetric Extension of the Standard Model

Mark R. A. Adcock

The calculation of the cross section of supersymmetric processes based on a left-right supersymmetric extension of the standard model is presented. The Feynman diagrams for the neutralino-quark-squark and chargino-quark-squark are written down, and the production of sleptons and squarks resulting from the t-channel exchange of neutralinos and charginos in electron-proton scattering is analyzed. The symbolic programming language Mathematica®, is used to develop a program to generate a symbolic representation of the invariant amplitude for these processes. Differential cross sections, based on numerical values of the free parameters of the left-right supersymmetric model, are generated.

ACKNOWLEDGEMENTS

Special thanks to my advisor Dr. C. S. Kalman for his advise and guidance during the preparation of this thesis. I would also like to thank him for his insistence that a deep understanding of physics is hard to be gained through the application of equations but is much is much easier to acquire through the act of writing about ones understanding of the principles behind the equations. His technique of having students prepare pre and post summaries of lecture materials without reference to a single equation is an effective one. I would also like to thank my employer, Computing Devices Canada, for their support.

Sincere thanks to V. Pramod for his comments and assistance in building the manuscript and to P. Cherrier who started me back on the path to further studies in theoretical physics.

I also wish to thank my family Lynn, Kathryn, Keith and Stephen for their support and patience during the many weekends spent in study.

TABLE OF CONTENTS

LIST OF FIGURES viii

INTRODUCTION 1

CALCULATING CROSS SECTIONS 3

1.1 Overview of Particle Physics 3

1.1.1 The Dirac Equation 4

1.1.2 Local Gauge Transformations 5

1.1.3 The Weak Interaction 7

1.1.4 Electroweak Unification 8

1.1.5 Gauge Boson Masses 10

1.1.6 Fermion Masses 13

1.2 Scattering Cross Section Calculations 14

1.2.1 S Matrix Treatment of Coulomb Scattering 15

1.2.2 Feynman Rules for Coulomb Scattering 20

1.2.3 Looking Ahead - SUSY Cross Sections 22

CHAPTER 2: SUPERSYMMETRY 24

2.1 Unifying the External and Internal Symmetries 25

2.2 The Supersymmetry Algebra 28

2.2.1 Breaking Supersymmetry 30

2.3 Left-Right Symmetry 30

2.4 Left-Right Supersymmetry 32

2.4.1 Mass Eigenstates of the L-R SUSY Model 35

2.4.2 Derivation of the Feynman Rules 39

2.4.3 The Feynman Rules 42

2.5 Calculation of ep -> slepton Cross Sections 47

2.5.1 Neutralino Contribution to Invariant Amplitude 47

2.5.2 Chargino Contribution to Invariant Amplitude 52

2.5.3 Supersymmetric Cross Section 55

CHAPTER 3: SUSY CROSS SECTION CALCULATOR 58

3.1 Description of Mathematica® 59

3.1.1 Symbolic Capabilities 59

3.1.2 Replacement Rules 61

3.2 Calculator Design 62

3.2.1 Design Overview 63

3.2.2 Detailed Design 64

3.3 Program Output 75

3.3.1 Neutralino Masses 75

3.3.2 Chargino Masses 76

3.3.3 Solution of Electron-Proton->Selectron-Squark Cross Section 77

3.3.4 Solution of Electron-Proton->Sneutrino-Squark Cross Section 79

3.3.5 Parametric Variation of Total Energy 82

3.4 Platform Matters 83

FUTURE CONSIDERATIONS 85

4.1 Limitations 85

4.2 Improvements 86

5. Conclusion 88

APPENDIX A: PROGRAM SOURCE CODE 89

REFERENCES 111

LIST OF FIGURES

FIG. 1 Particle physics road map 3

FIG. 2. Feynman rules for Rutherford scattering 20

FIG. 3. The reaction 1+2 -> 3+4 defining the scattering angle θ 21

FIG. 4. Feynman rules Neutralino -> Quark Squark (1-13) 43

FIG. 5. Feynman rules Neutralino -> Quark Squark (14-26) 44

FIG. 6. Feynman rules Chargino -> Quark Squark (1-13) 45

FIG. 7.Feynman rules Chargino -> Quark Squark (14-26) and Chargino -> Lepton Slepton (26-30) 46

FIG. 8. Neutralino contribution to ep->sleptons invariant amplitude 47

FIG. 9. Neutralino contribution to ep->sleptons invariant amplitude (cont'd) 48

FIG. 10 Chargino contribution to ep->sleptons invariant amplitude 52

FIG. 11. Neutralino contribution to ep->sleptons invariant amplitude (cont'd) 53

FIG. 12. The reaction 1+2->3+4 in the centre-of-mass frame 55

FIG. 13. Pictorial representation of the packages comprising the SUSY calculator 63

FIG. 14.SUSY calculator execution flow 66

FIG. 15 Neutralino mass states 75

FIG. 16 Chargino mass states 76

FIG. 17 Differential cross section $e^- p \to \tilde{e}\tilde{q}X$ for $\sqrt{s} = 314$ GeV 79

FIG. 18 Differential cross section $e^- p \to \tilde{\nu}\tilde{q}X$ for $\sqrt{s} = 314$ GeV 82

FIG. 19 Differential cross section $e^- p \to \tilde{\nu}\tilde{q}X$ for $\sqrt{s} = 410$ GeV 82

FIG. 20 Differential cross section $e^- p \to \tilde{\nu}\tilde{q}X$ for $\sqrt{s} = 1400$ GeV 83

INTRODUCTION

The standard model represents one of mankind's greatest achievements in describing the workings of nature. The model however has some fundamental flaws that indicate it is a low energy approximation of a more fundamental theory. But what is the nature of these higher energy unifying theories? Supersymmetry offers a tantalizing insight into such questions. The supersymmetric theories are based upon a wider range of particles than have actually been observed in nature. The reasoning behind the need for these "super" particles is so compelling that the theory has reached an impressive level of acceptance by the worldwide physics community despite no experimental evidence of their existence to date. If the universe is indeed supersymmetric and as the energy scales achievable by modern particle accelerators increases, there is more likelihood that we will be able to acquire such experimental evidence. But how will such evidence be uncovered ?

The nature of particles and how they interact with each other have historically been explored by probing the interaction through scattering experiments. Evidence of supersymmetry will most likely be uncovered in the same manner. It is widely believed that electron-proton scattering experiments will provide the environment in which the production of supersymmetric particles can be studied. Supersymmetric theories predict that the supersymmetric products in such experiments will either be undetectable or decay into undetectable by-products. Nevertheless, undetectable particles may be characterized by high missing transverse momentum. If we know the signature of the process, it is possible to extract the evidence from the background noise. The signature of the process is characterized by its cross section which can be calculated from the model.

Development of techniques to estimate the scattering cross section of the supersymmetric interaction have already been established, but the calculations are long and tedious, and

the number of free parameters in the model are many. It is desirable to develop computer-based techniques to assist in these calculations. With the advent of algebraic programming Languages such as Mathematica®, symbolic calculators have been developed to assist in calculations of interactions based on the standard model. It is only natural to extend these techniques to the calculations of supersymmetric interactions.

In this thesis in order to gain an appreciation of the techniques employed in developing such a supersymmetric calculator, we will first review the nature of the standard model and how scattering cross sections are calculated using the theory of quantum electrodynamics (QED). This leads naturally to the modifications to the standard model needed to make it left-right supersymmetric and to the modification to the Feynman rules governing the production of supersymmetric particles in electron-proton collisions. Finally, the design of the calculator is presented and the results of the calculation of supersymmetric process are discussed.

CALCULATING CROSS SECTIONS

1.1 Overview of Particle Physics

Figure 1 is a simplified road map of the major concepts in particle physics which lead to the development of the supersymmetric theories. The Dirac equation is a natural starting point because it provides a natural way of introducing internal and external symmetry groups that can only be merged by the principle of supersymmetry.

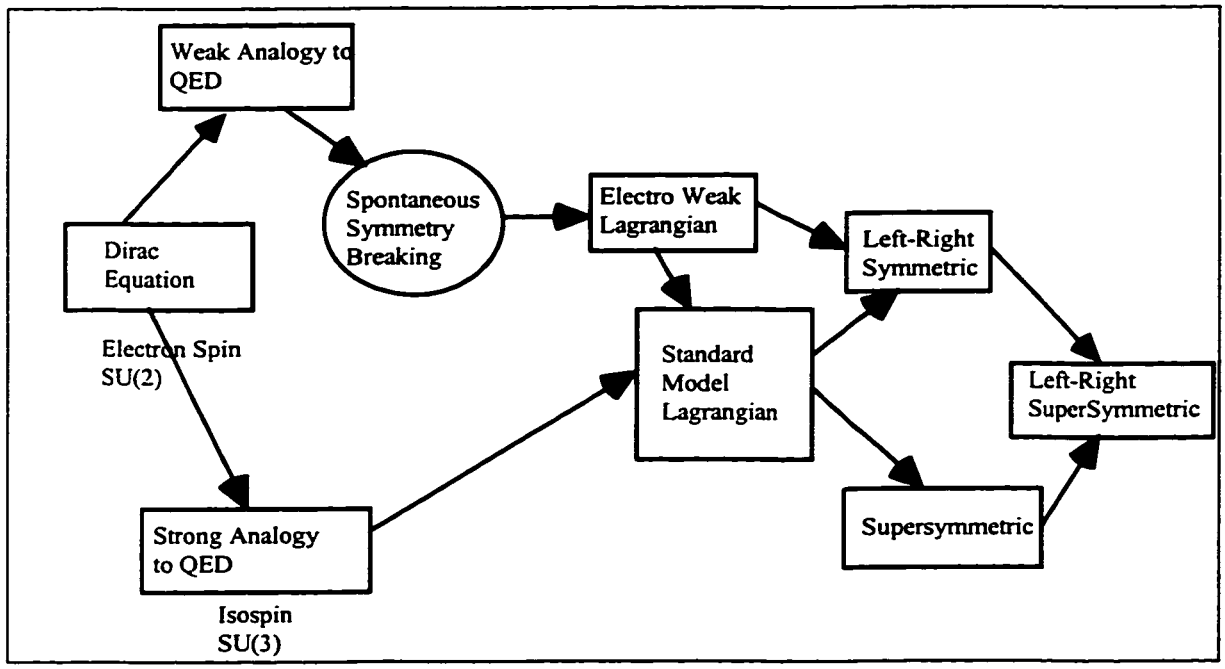

FIG. 1 Particle physics road map

From the Dirac equation we proceed to a discussion of local gauge invariance and the concept of symmetry breaking. The concepts of internal symmetries and local gauge invariance provide the framework on which the supersymmetry is built..

1.1.1 The Dirac Equation

The Dirac equation was discovered because of the need for an equation invariant under a Lorentz transformation. The Dirac equation may be derived from the requirements that the equation be linear in the gradient operator and satisfy the relativistic energy momentum relation:

$$\begin{aligned} H\psi &= (\alpha \cdot P + \beta m)\psi \qquad \text{Linearity Requirement} \\ H^2\psi &= \left(P^2 + m^2\right)\psi \qquad \text{Energy / Momentum Requirement} \end{aligned} \tag{1.1}$$

These two equations lead to the realization that the α's and the β's are not simply numbers but that they are hermitian traceless matrices of at least 4 dimensionality. The Pauli matrices are most frequently used to represent them:

$$\alpha_i = \begin{pmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{pmatrix}, \quad \beta = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \tag{1.2}$$

Thus the Dirac equation is represented:

$$\begin{aligned} & i\beta\frac{\partial\psi}{\partial t} = -i\beta\alpha \cdot \nabla\psi + m\psi \\ & \text{with } \gamma^\mu = (\beta, \beta\alpha) \text{ this may be re-written:} \\ & \left(i\gamma^\mu\partial_\mu - m\right)\psi = 0 \end{aligned} \tag{1.3}$$

The γ's represent the 4x4 Dirac gamma matrices. The Dirac-Pauli representation of the gamma matrices are used throughout this thesis. The wave function solutions to this equation are typically of the form:

$$\psi = u(p,s)e^{-ip\cdot x} \tag{1.4}$$

where the first term in equation 1.4 represents four-component entities called spinors. Further insight can be gained by noting the different symmetries inherent in the Dirac equation. Equations 1.1 result from the requirement that the Dirac equation be invariant under Lorentz transformations. The Lorentz transformation consists of Lorentz boosts and ordinary rotations and is an external symmetry of the Dirac equation. The algebraic entities represented by the α's and β's in equations 1.1 obey the algebra of SU(2) and represent an internal symmetry. It has long been known that the Lorentz subgroup of ordinary rotations O(3) and the group SU(2) are isomorphic, but all attempts to unify these external and internal symmetries into one symmetry group failed until the supersymmetric theories were put forth. Another key concept to understanding supersymmetry is that of local gauge invariance.

1.1.2 Local Gauge Transformations

Invariance under external group transformations leads to the conservation of energy, linear momentum etc., but what is conserved under internal group transformations? That is: what conserved current is required to ensure that the different internal fields do not mix with different space-time properties? In order to answer this question, an important first step is to understand the concept of gauge transformations. The Dirac equation may be defined by the following Lagrangian:

$$\mathrm{L} = i\overline{\psi}\gamma_\mu\partial^\mu\psi - m\overline{\psi}\psi \tag{1.5}$$

This Lagrangian is invariant under global gauge transformations, but is not invariant under the local gauge transformation:

$$\psi(x) \rightarrow e^{i\alpha(x)}\psi(x), \tag{1.6}$$

because of the presence of ∂_μ:

$$\partial_\mu \psi \rightarrow e^{i\alpha(x)}\partial_\mu \psi(x) + e^{i\alpha(x)}\psi(x)\partial_\mu \alpha \tag{1.7}$$

If we insist that this Lagrangian be invariant under local gauge transformations, we must introduce a vector field A_μ.to cancel the extra term. Careful analysis of this vector field demonstrates that it satisfies Maxwell's equations, and indeed these famous equations can be derived from this principle of gauge invariance alone[1]. It is illuminating to observe that Maxwell's equations, which were the crowning unification of more than two hundred years of empirical work, can be derived by requiring this basic invariance of the Dirac Lagrangian. Thus the vector field A_μ, is interpreted as the photon field, and its introduction leads to the famous Lagrangian of quantum electrodynamics (QED):

$$\mathrm{L} = \overline{\psi}\left(i\gamma_\mu \partial^\mu - m\right)\psi + e\overline{\psi}\gamma^\mu A_\mu \psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \tag{1.8}$$

The imposition of the natural requirement of local phase invariance on the free fermion Lagrangian leads to an interacting Lagrangian of two fields each which transform according to two different groups. The photon field is invariant under Poincaré transformations while the Dirac spinors of the fermion field are invariant under SU(2)

transformations. As mentioned above, supersymmetric theories offer a means of unifying these two groups. Armed with the concept of the QED Lagrangian we can now embrace the unification of the electromagnetic and the weak interactions through the concept of spontaneous symmetry breaking.

1.1.3 The Weak Interaction

The weak interaction can be classified into three categories

i) leptonic decays e.g. $\mu^- \rightarrow e^- \bar{\nu}_e \nu_\mu$

ii) semileptonic decays e.g. $^{10}C \rightarrow {}^{10}B^* + e^- + \nu_e \quad (p \rightarrow n e^+ \nu_e)$

iii) non-leptonic decays e.g. $K^+ \rightarrow \pi^+ \pi^0$

The famous β decay involves the transformation of a proton into a neutron (or vice versa), and it is an example of a semi-leptonic decay. When it was discovered that weak interactions exhibit parity violation, it necessitated describing the invariant amplitude of a weak process in a vector-axial (V-A) form. This form is realized by replacing the bilinear covariant γ^μ found in the electromagnetic invariant amplitude with $\gamma^\mu(1-\gamma^5)$. This V-A form automatically violates parity conservation. The V-A structure of the weak current is directly exposed by scattering neutrinos off electrons. This process always selects a left-handed neutrino (or a right-handed antineutrino) and indeed there is no empirical evidence for the existence of right-handed neutrinos (or left-handed antineutrinos). The last is only true if the neutrino mass is strictly zero otherwise a

Lorentz transformation could transfer one into the other. The zero mass neutrino mass prediction of the standard model is considered a problem and will be discussed in chapter 2 when we investigate a left-right symmetric model.

The weak current was at first observed to have a charge-raising or charge-lowering structure. Extending the electromagnetic analogy, it was postulated that the weak interactions are generated by the emission and absorption of weak bosons called $W^{\pm}$ .The existence of an electrically neutral weak current was not revealed until 1973. The observation that weak neutral current effects occurred at a much higher rate than could be explained by combined electromagnetic and weak charged currents resulted in the definition of weak neutral currents. This current was defined as a four-vector current-current form to fit the experimental data. In general, the weak neutral currents are not pure V-A currents because they have right-handed currents. The weak neutral currents are associated with the weak neutral boson Z_0. It was the existence of the right-handed components of the neutral currents that led to the unification of the weak and electromagnetic forces as will be discussed.

1.1.4 Electroweak Unification

The principle question asked in the quest for electroweak unification is whether the two charged currents and the weak current form a symmetry group of weak interactions. An isospin triplet of weak currents should generate an $SU(2)_L$ algebra, but the neutral current can't complete the triplet because it has a right-handed component. However, the

electromagnetic current is neutral with right-handed as well as left-handed components. Neither the neutral nor the electromagnetic current respect the $SU(2)_L$ symmetry, but the idea is to form two orthogonal combinations which do have definite transformation properties under $SU(2)_L$.

One of these combinations completes the isospin triplet; the other, called the weak hypercharge current, is unchanged by $SU(2)_L$ operations. In order to complete the unification, the current-current form of the weak interaction is modified:

In QED, the em current is coupled to the photon:

$$-ie(j^{em})^{\upsilon} A_{\upsilon} \tag{1.9}$$

In the electroweak interaction we assume a similar interaction:

$$-ig(j^{j})^{\mu} W_{\mu}^{i} - i\frac{g'}{2}(j^{Y})^{\mu} B_{\mu} \tag{1.10}$$

The electromagnetic interaction is embedded in this expression. The W_{μ}^{i} is an isotriplet of vector fields coupled with strength g to the weak isospin current J_{μ}^{i}, together with a single vector field B_{μ} coupled to the weak hypercharge current j_{μ}^{Y} with strength $g'/2$. When the masses of the weak bosons are generated by symmetry breaking, the neutral fields W_{μ}^{3} and B_{μ} mix in such a way to produce the physical states:

$$\begin{aligned} A_{\mu} &= B_{\mu}\cos\theta_{w} + W_{\mu}^{3}\sin\theta_{w} \quad (massless) \\ Z_{\mu} &= -B_{\mu}\sin\theta_{w} + W_{\mu}^{3}\cos\theta_{w} \quad (massive) \end{aligned} \tag{1.11}$$

θ_W is called the weak mixing angle, and is expressed as:

$$g\sin\theta_w = g'\cos\theta_w = e \tag{1.12}$$

This mixing angle is given by the ratio of the two independent group coupling constants which are determined by experiment. Finally, the observed neutral currents are expressed as:

$$\begin{aligned} j_\mu^{em} &= J_\mu^3 + \tfrac{1}{2} j_\mu^Y \\ J_\mu^{NC} &= J_\mu^3 - \sin^2\theta_w j_\mu^{em} \end{aligned} \tag{1.13}$$

The right-handed component of the weak neutral current is canceled by the correct projection (through the weak mixing angle) of the electromagnetic current leaving J_μ^3 , a pure left-handed current belonging to the symmetry group $SU(2)_L$. Similarly, the electron magnetic current is made up of the weak neutral current and the hypercharge current. Electroweak unification is achieved through these definitions. In order for the standard model to be successful, sin θ_W must be the same for all weak interactions. As noted above, the physical states result from the breaking of a symmetry. In order to introduce the masses of the weak bosons we investigate this process.

1.1.5 Gauge Boson Masses

The Goldstone theorem[2] states that massless scalars occur whenever a continuous symmetry of a system is spontaneously broken. The Higgs mechanism is a way of turning this massless Goldstone boson into a longitudinal polarization of a massive gauge

particle. The Higgs mechanism is a choice of a particular gauge.

Spontaneous breaking of a local SU(2) symmetry provides a way of viewing how the weak bosons acquire mass. We want to formulate the Higgs mechanism so that the $W^{\pm}$ and Z^0 become massive and the photon remains massless. The Weinberg-Salam model achieves this by starting with an SU(2) invariant Lagrangian for example, of the electron-neutrino lepton pair:

$$\begin{aligned} L_1 &= \bar{\chi}_L \gamma^\mu [i\partial_\mu - g\tfrac{1}{2}\tau \cdot W_\mu - ig'(-\tfrac{1}{2})B_\mu]\chi_L \\ &+ \bar{e}_R \gamma^\mu [i\partial_\mu - g'(-\tfrac{1}{2})B_\mu]e_R - \tfrac{1}{4}W_{\mu\nu}W^{\mu\nu} - \tfrac{1}{4}B_{\mu\nu}B^{\mu\nu} \end{aligned} \tag{1.14}$$

An SU(2) X U(1) invariant Lagrangian, L_2, for the 4 scalar fields ϕ_i (i = 1,2,3,4) is added, and the gauge boson masses are generated by the Higgs Mechanism.

$$L_2 = \left|(i\partial_\mu - gT \cdot W_\mu - ig'(\tfrac{Y}{2})B_\mu)\phi\right|^2 - V(\phi) \tag{1.15}$$

To keep L_2 gauge invariant the ϕ_i must belong to the SU(2) X U(1) gauge group. The most economical choice is to arrange ϕ in an isospin doublet. It is this minimal choice of the Higgs doublet that is the cornerstone of the Weinburg Salam model:

$$\phi = \tfrac{1}{\sqrt{2}}\begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix} \tag{1.16}$$

To generate the gauge boson masses, we use the Higgs potential and choose a vacuum expectation value for one of the fields; the standard choice is to set :

Higgs potential:

$$V(\phi) = -\mu^2 \phi^+ \phi + \lambda(\phi^+ \phi)^2, \quad \mu^2 > 0$$

minumum of V occurs at:

$$\phi_1 = \phi_2 = \phi_4 = 0 \quad \phi_3 = v$$

$$\phi_0 = {}^1\!/\!_{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix} \quad \text{where} \quad v = \frac{\mu}{\sqrt{\lambda}}$$

Due to gauge invariance, we can substitute (1. 17)

$$\phi(x) = {}^1\!/\!_{\sqrt{2}} \begin{pmatrix} 0 \\ v + h(x) \end{pmatrix}$$

back into the Lagrangian to generate the boson masses.

Any choice of ϕ_0 will break a symmetry operation and will generate a mass for the corresponding gauge boson. Indeed, different choices of the Higgs potential itself result in different interpretations of the vacuum as we shall see when we discuss left-right symmetric models. If the vacuum ϕ_0 is still left invariant by a subgroup of gauge transformations, the gauge boson associated with that subgroup will remain massless. Hence, we choose a doublet of complex scalar fields (that will leave $U(1)_{em}$ intact) about which we expand the Lagrangian. The resulting expansion of the vacuum breaks the SU(2) and $U(1)_Y$ symmetries but leaves the $U(1)_{em}$ symmetry intact. Thus the vacuum remains invariant under $U(1)_{em}$ transformations and the photon remains massless. The weak gauge boson masses are identified by substituting this vacuum expectation of ϕ_0 into the Lagrangian L_2. The charged weak bosons are predicted to have mass $M_W = \frac{1}{2} vg$. The neutral fields are described by a 2x2 matrix whose eigenvalues are the neutral boson masses. This matrix is diagonalized to obtain the mass of the neutral boson. Upon normalizing the fields this process results in:

$$A_\mu = \frac{g'W_\mu^3 + gB_\mu}{\sqrt{g^2 + g'^2}} \qquad with \quad M_A = 0$$
$$Z_\mu = \frac{gW_\mu^3 - g'B_\mu}{\sqrt{g^2 + g'^2}} \qquad with \quad M_Z = \tfrac{1}{2}v\sqrt{g^2 + g'^2} \tag{1. 18}$$

The ratio $\frac{M_W}{M_Z} = \cos\theta_W$. is a prediction of the standard model. In 1983, CERN collider results were used to measure these boson masses, and the results were found to be in impressive agreement with the standard model[3].

1.1.6 Fermion Masses

The gauge boson masses were generated by adding an SU(2) X U(1) gauge invariant Lagrangian for the scalar fields ϕ. Similarly, to generate the electron masses we include the following SU(2) X U(1) gauge invariant term:

$$L_3 = -G_e\left[(\bar{\nu}_e, \bar{e})_L \begin{pmatrix} \phi^+ \\ \phi^o \end{pmatrix} e_R + \bar{e}_R(\phi^-, \bar{\phi}^o)\begin{pmatrix} \nu_e \\ e \end{pmatrix}_L\right] \tag{1. 19}$$

Conveniently, the same Higgs doublet that generates the boson masses also give masses to the fermions. For example, the Higgs doublet has the required SU(2) X U(1) quantum numbers to couple the electrons $\bar{e}_L e_R$. We spontaneously break the symmetry with the same expansion and on substitution of ϕ, we can generate the electron mass. The Lagrangian now becomes:

$$L_3 = -\frac{G_e}{\sqrt{2}} v(\bar{e}_L e_R + \bar{e}_R e_L) - \frac{G_e}{\sqrt{2}} h(\bar{e}_L e_R + \bar{e}_R e_L)$$

$$if \ \ m_e = \frac{G_e v}{\sqrt{2}} \tag{1.20}$$

$$L_3 = -m_e \bar{e} e - \frac{m_e}{v} \bar{e} e h$$

Note that since the electron coupling is arbitrary the electron mass is not predicted. The Lagrangian also contains an interaction term coupling the Higgs scalar h to the electron. This coupling is very small and so far has not produced a detectable result. The Higgs particle has never been observed and any evidence of its existence is eagerly awaited. Before proceeding to a discussion of the Higgs sector in supersymmetry, we provide a detailed example of the powerful techniques developed to calculate particle interactions within the standard model. Calculation of the scattering cross section of an electron off a static nucleus provides a simple, straightforward example in which these sophisticated techniques can be presented.

1.2 Scattering Cross Section Calculations

Scattering experiments involve directing a beam of particles at a target and observing how the incident particles become scattered by the target. Using scattering experiments to explore the relationship between incident particle energy and the interaction between target and incident beam has a wide range of applicability. The quantity measured in such experiments is called the scattering cross section and has units of area. The most likely means of gaining experimental evidence of supersymmetry will be through scattering experiments. In order to get an understanding of the calculation techniques

required to obtain a theoretical prediction of scattering cross sections, we will proceed with a discussion of the relativistic generalization of classical Rutherford cross section using the theory of QED and Feynman rules. The cross section discussion will end with a generalization of these techniques for the purpose of supersymmetric cross section calculations.

1.2.1 S Matrix Treatment of Coulomb Scattering

In quantum mechanics we cannot speak in terms of certainties, only in terms of probabilities. Thus quantum mechanical cross section calculations are based upon the transition probability per particle into the number of final states within the range of momentum uncertainty of the measuring apparatus.

The number of final states within in the range of the final momentum is given by:

$$dN_f = V\frac{d^3pf}{(2\pi)^3} \tag{1.21}$$

where V denotes the normalization volume. The square of the S matrix element gives us the probability of particle transition. The transition probability per particle into the final states is thus written:

$$dW = |S_{fi}|^2 \frac{Vd^3p_f}{(2\pi)^3} \tag{1.22}$$

The transition probability per particle per unit time is denoted:

$$dR = \frac{dW}{T} = \frac{|S_{fi}|^2}{T} \frac{V d^3 p_f}{(2\pi)^3} \tag{1.23}$$

Finally, the scattering cross section is defined as the transition probability per particle per unit time divided by the magnitude of the incoming current of particles:

$$d\sigma = \frac{dR}{|J_{inc}|} = \frac{dR}{\frac{|v_i|}{V}} = \frac{|S_{fi}|^2}{T\frac{|v_i|}{V}} \frac{V d^3 p_f}{(2\pi)^3} \tag{1.24}$$

Noting that:

$$d^3 p_f = p_f^2 d|p_f| d\Omega, \tag{1.25}$$

the differential cross section per unit solid angle is expressed:

$$\frac{d\sigma}{d\Omega} = \frac{|S_{fi}|^2}{T\frac{|v_i|}{V}} \frac{V p_f^2 d|p_f|}{(2\pi)^3} \tag{1.26}$$

All of the physics lies in the square of the S matrix element. To calculate the Rutherford scattering cross section for an electron scattered by a fixed Coulomb potential, we start with the S matrix to the lowest order based on the covariant electromagnetic potential :

$$S_{fi} = -ie \int d^4 x \overline{\psi}_f(x) A(x) \psi_i(x) \tag{1.27}$$

For Coulomb scattering the potential is static, so the covariant potential is simply the coulomb potential times γ^0:

$$A(x) = A_\mu \gamma^\mu = A_0(x)\gamma^0 = -\frac{Ze}{|x|}\gamma^0 \tag{1.28}$$

We can write the S matrix element as:

$$S_{fi} = iZe^2 \int d^4x \bar{\psi}_f(x)\left[\frac{\gamma^0}{|x|}\right]\psi_i(x) \tag{1.29}$$

Now we have to insert the expressions for the fermion fields. We approximate the incoming electron as a plane wave with its initial momentum and spin and the outgoing electron with its final momentum and spin:

$$\begin{aligned} \psi_i &= \sqrt{\frac{m}{E_i V}}u(p_i,s_i)e^{-ip_i\cdot x} \\ \psi_f &= \sqrt{\frac{m}{E_f V}}u(p_f,s_f)e^{-ip_f\cdot x} \end{aligned} \tag{1.30}$$

V denotes the normalization volume over which the fields are normalized to probability 1. The Dirac spinors u and $\bar{u}$ are normalized for Lorentz boosts[4]. With the proceeding definitions, the S matrix element now reads:

$$S_{fi} = \frac{iZe^2}{V}\sqrt{\frac{m^2}{E_f E_i}}\bar{u}(p_f,s_f)\gamma^0 u(p_i,s_i)\int d^4x e^{i(p_f-p_i)x}\frac{1}{|x|} \tag{1.31}$$

Details of the integration may be found in reference [2]. Inserting the results of the integration, the S Matrix becomes:

$$S_{fi} = \frac{iZe^2}{V}\sqrt{\frac{m^2}{E_f E_i}}\bar{u}(p_f,s_f)\gamma^0 u(p_i,s_i)\frac{4\pi}{q^2}2\pi\delta(E_f - E_i) \tag{1.32}$$

where $q = (p_f - p_i)$ is the transferred momentum. Squaring the magnitude of this expression and inserting it into the differential cross section we arrive at:

$$\frac{d\sigma}{d\Omega} = Z^2e^4\frac{m^2}{E_f E_i}\left|\bar{u}(p_f,s_f)\gamma^0 u(p_i,s_i)\right|^2\frac{64\pi^4}{|q|^4}\left|\delta(E_f - E_i)\right|^2\frac{p_f^2 d|p_f|}{T|v_i|(2\pi)^3} \tag{1.33}$$

In order to further simplify this expression, it can be shown that:

$$\left|\delta(E_f - E_i)\right|^2 = \frac{T}{2\pi}\delta(E_f - E_i) \tag{1.34}$$

Inserting this into the expression for the S matrix, we arrive at the expression for the differential cross section per unit of solid angle.

$$\frac{d\sigma}{d\Omega} = \frac{4Z^2e^4m^2}{|q|^4}\left|\bar{u}(p_f,s_f)\gamma^0 u(p_i,s_i)\right|^2\int_{\Delta p}\delta(E_f - E_i)\frac{p_f^2 d|p_f|}{E_f E_i|v_i|} \tag{1.35}$$

In order to complete this last integration we have to change variables:

$$\begin{aligned} &E = \frac{p^2}{2m} \rightarrow dE = \frac{p}{m}dp \\ &\int_{\Delta p}\delta(E_f - E_i)\frac{p_f m^2 dE_f}{E_f E_i p_i} = 1 \end{aligned} \tag{1.36}$$

This results in the following expression for the differential cross section per solid angle:

$$\frac{d\sigma}{d\Omega} = \frac{Z^2 e^4 \cdot m^2}{|q|^4} \left| \bar{u}(p_f, s_f) \gamma^0 u(p_i, s_i) \right|^2 \tag{1.37}$$

Now the tricky part of this calculation is to calculate the contributions form the spinor sums. To calculate this entity we rely on the Trace theorems of the products of gamma matrices. Explanations of why we can replace them are well detailed in the literature[5]. Application of trace theorems results in the relativistic Rutherford cross section also known as the Mott cross section:

$$\frac{d\sigma}{d\Omega} = \frac{Z^2 e^4}{4|p|^4 \sin^4 \frac{\theta}{2}} \left[E^2 - \beta^2 E^2 \sin^2 \frac{\theta}{2} \right] \tag{1.38}$$

In the non-relativistic limit $E^2 \to m^2$ *and* $\beta^2 \to 0$ this reduces to the classical cross section :

$$\frac{d\sigma}{d\Omega} = \frac{Z^2 \alpha^2 \cdot m^2}{4|p|^4 \sin^4 \frac{\theta}{2}} \tag{1.39}$$

The additional term in the relativistic cross section results from the magnetic moment of the Dirac electron interacting with the magnetic field of the scattering centre. This effect is negligible at small velocities. This rather torturous calculation can be treated in a more natural way by expressing the cross section in terms of the invariant amplitude. This amplitude can be read right off the Feynman diagram for the interaction.

1.2.2 Feynman Rules for Coulomb Scattering

The Feynman rules for scattering of electrons off a static charge are presented in the following figure:

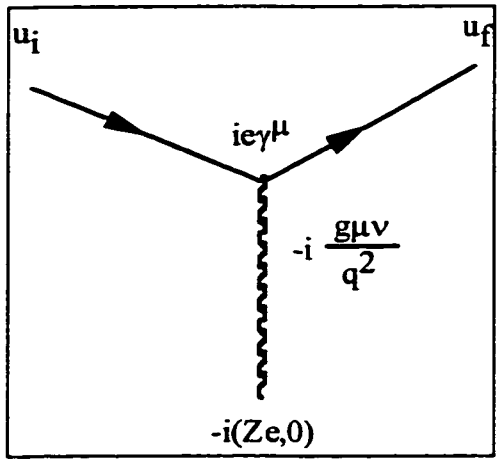

FIG. 2. Feynman rules for Rutherford scattering

The invariant amplitude can be read directly off the diagram:

$$-iM = \left(ie\bar{u}_f\gamma^\mu u_i\right)\left(\frac{-i4\pi g_{\mu\nu}}{q^2}\right)\left(-ij^\nu(q)\right) \tag{1.40}$$

Equating the electromagnetic current to the static coulomb potential the invariant amplitude now reads:

$$-iM = \left(ie\bar{u}_f\gamma^0 u_i\right)\left(\frac{-i4\pi}{q^2}\right)\left(-iZe\right) \tag{1.41}$$

In order to express the scattering cross section in terms of the invariant amplitude we rely

on the techniques of De Wit and Smith[6]. For a scattering process in the laboratory frame we have the following geometry:

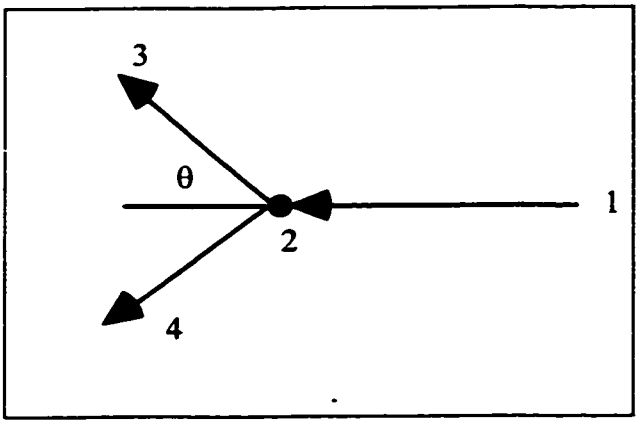

FIG. 3. The reaction 1+2 -> 3+4 defining the scattering angle θ

The Mandelstam variables t and s represent the transferred momentum. The cross section in terms of t and s and the invariant amplitude M may be expressed:

$$\frac{d\sigma}{dt} = \frac{1}{16\pi}\frac{|M|^2}{\lambda(s,m_1^2,m_2^2)} \qquad (1.42)$$

where $\lambda(s,m_1^2,m_2^2) = \left(s-(m_1+m_2)^2\right)\left(s-(m_1-m_2)^2\right)$

The subscripts 1 and 2 refer to Figure 1.4. In the case of Coulomb scattering 1 represents the incoming electron, 3 represents the outgoing electron, 2 and 4 represent the nucleus. For Coulomb scattering the nucleus does not recoil hence p_4 is zero. Making this substitution and simplifying by noting energy conservation lead to the following representation of the differential cross section in the laboratory frame:

$$\frac{d\sigma}{d\Omega} == \frac{|M|^2}{4\pi^2} \tag{1.43}$$

Squaring the invariant amplitude and inserting it into the expression for the differential cross section results in the same expression as 1.40.

$$\frac{d\sigma}{d\Omega} = \frac{Z^2 e^4 \cdot m^2}{|q|^4} \left|\bar{u}(p_f, s_f)\gamma^0 u(p_i, s_i)\right|^2 \tag{1.44}$$

We see that using the Feynman rules and the invariant amplitude is a straight forward way to arrive at the scattering cross section. We will employ these techniques in the calculation of supersymmetric cross sections

1.2.3 Looking Ahead - SUSY Cross Sections

In this chapter we have noted several important points about the standard model, supersymmetry and the calculation of cross sections:

i) The masses of the gauge bosons and the fermions in the standard model are generated by the process of symmetry breaking. The mechanism for breaking the symmetry is the Higgs potential. Often referred to as the Higgs sector, the choice of the Higgs potential is a degree of freedom in the standard model, and the particular choice of the potential affects the form of the particles.

ii) The GHP of the standard model could be improved by choosing different Higgs potentials for the generation of boson masses. We explore this in the next chapter in

our discussion of supersymmetry.

iii) Scattering experiments are a powerful means by which particles and their interactions can be studied. Physical evidence of supersymmetry, if indeed the universe is supersymmetric, will most likely be found in scattering experiments[7]. The heart of the scattering calculation lies in the invariant amplitude which is derived from the Feynman rules for the interaction. In the next chapter we will present the Feynman rules for two specific supersymmetric cross section calculations.

CHAPTER 2: SUPERSYMMETRY

Attempts to determine a general group that encompasses external symmetry groups such as the Lorentz group with internal symmetry groups such as SU(2) using Lie algebras have not been successful. Theorems called No-Go theorems show that a connected group that contains both groups is at best a simple product of the external group and any one of the internal groups[8]. This is not a satisfactory solution since generality is a requirement. The commutation relationships of a group's generators form a Lie algebra that defines certain symmetries; these symmetries are related by Noether's theorem to conserved currents[9]. It follows that the field operators which quantize these fields have commutation rules that are directly related to the group generators. Since we require a field theoretical formulation that encompasses supersymmetry, we require a way out of the No-Go theorems. The only way out is to extend the concept of a Lie algebra.

A superalgebra is a type of Lie algebra that includes both commutators and anti-commutators in its definition. Supersymmetry is such a superalgebra. The anti commutation relations transform bosons into fermions and vice versa. If one allows these relations, then a unification of the external and the internal symmetries can be achieved. Supersymmetry requires that all fermions have boson partners and vice versa. Since this spectrum of particles has not been observed, it is assumed that the symmetry is badly broken at the currently available energies. Future high-energy experiments may confirm that Supersymmetry does indeed exist in nature.

2.1 Unifying the External and Internal Symmetries

The Lorentz group is the group of rotations and boosts in 4-dimensional Minkowski space. The restricted Lorentz group, also referred to as the proper orthochronous Lorentz group, does not contain time or space reflections. The generators of the restricted Lorentz group are the K_i which are the generators of the Lorentz boost and the J_i which are the generators of the Lorentz rotations. A non-unitary representation of the Lorentz group may be formed by changing the basis of K_i and J_i by introducing a complex transformation. The intriguing result of this operation is a pair of generators that obey the commutation relations of the Lie algebra of SU(2) which as noted in chapter 1, is an internal symmetry of the Dirac equation. Although the act of complexifying the group doubles the group size, it can be seen that the close relationship between the imposition of external symmetries of the Lorentz group and the internal symmetries that result from such an imposition hints of the possibility of unification.

The Poincaré Group is an extension of the Lorentz group. It is a semi-direct product of the Lorentz group and the translation group. The commutation relations of the Poincaré Group represent a 10 dimensional Lie algebra. There are 6 infinitesimal parameters leading to a Lorentz transformation and 4 infinitesimal parameters leading to an infinitesimal translation.

Internal symmetry groups are representative of internal degrees of freedom of a particle. In particular the SU(2) group is the lowest-dimensional nontrivial representation of the

rotation group. This group describes the spin angular momentum of the electron and arises in a natural way in the solution of the Dirac equation. There are other important internal symmetry groups such as SU(3) which is represented by, among other things, the colour charges of a quark. There is a connection between the spinor algebra of the SU(2) group and the restricted Lorentz group. This connection can be seen by looking from either the internal or the external points of view.

As noted in chapter 1, the Dirac gamma matrices result from the algebra necessary for non-trivial solutions of the Dirac equation presented in equation 1.1. These four-dimensional matrices are usually represented in either the Dirac-Pauli or the Weyl forms. Both forms use the 2-dimensional Pauli matrices. In order to investigate the connection between SU(2) and the restricted Lorentz group from the point of view of the Dirac equation, it is instructive to study the extreme relativistic limit of the Dirac equation in the Weyl representation. This limit decouples the 4x4 matrix equation into two 2x2 matrix equations which are wave equations for spin ½ particles. Since this is the massless limit, these equations describe massless neutrinos. The result of this limit is the result that the equations are no longer invariant under the parity transformation. Thus we have left and right handed Weyl spinors. These Weyl spinors are related to the helicity eigenstates. The helicity is a Poincaré invariant quantity. It is defined as the projection of total angular momentum onto the momentum directions. The invariance of the helicity operator under Lorentz transformations implies that left-handed neutrinos are left-handed in any inertial system. Thus we see that the two groups are implicitly related. It is the Lorentz group

which defines the Dirac equation, but this equation can only be solved by turning to the internal symmetries (SU(2)) implicit in the Dirac algebra[8].

The No-Go theorems prove that a connected symmetry group of the external and internal groups is only realizable if the following requirements are met by the connected group[8]:

i) Contains a sub-group that is Lorentz invariant.

ii) Has a finite number of particles.

iii) Elastic scattering functions are analytic.

iv) Scattering for two particle state occurs at (almost) any energy.

v) Generators of this group have distributions for their kernels.

The Coleman - Mandula theorem[10] proves that there are Lie groups that contain the Poincaré group and internal symmetry groups in a non-trivial manner; however, these groups lead to trivial physics. In this spirit, if one were to use such a group, then the S-matrix for all processes would be zero. Clearly if genuine unification is to be achieved, a different type of algebra is required.

Supersymmetries avoid the restriction of the Coleman-Mandula Theorem by relaxing one condition, they include anti-commutators in the definitions of the algebra. This is achieved through a generalization of the notion of a Lie algebra. The defining relations of these super algebras involve both commutators and anti-commutators. Superalgebras are

also know as graded Lie algebras.

In order to generalize the Poincaré algebra to a superalgebra, a Majorana spinor charge (also derived from the Dirac representation) is added to the algebra. Since these are spinors under the Lorentz Group, very strong constraints are placed on the resulting algebra. The theorem of Haag, Lopuszanski and Sohnius[11] now states that there is a symmetry that includes the Poincaré group and an internal symmetry group in a non-trivial way. The super algebra contains commutating and anti-commutating relations between this spinor and the generators of the Lorentz transformations. Since the Majorana is in the spin - ½ representation of the Lorentz group, its action on a state of spin j will result in a state of spin j +/- ½. Hence we have a symmetry which mixes particles of different spins - that is it mixes fermions and bosons.

2.2 The Supersymmetry Algebra

The Supersymmetric algebra has the following 14 generators:

4 Translation generators P_μ - The Energy Momentum Operator

6 Lorentz generators $M_{\mu\nu}$ - The Lorentz Rotation Generator

4 Spinor charges Q_a - The Majorana Spinor Operator

The algebra is defined by the following commutation and anti-commutation relationships between the generators which includes the algebra of the Poincaré group. It is apparent that the algebra will transform spin states.

$$
\begin{aligned}
&[P_\mu, P_\nu] = 0 \\
&[M_{\mu\nu}, P_\rho] = -i(\eta_{\mu\rho} P_\nu - \eta_{\nu\rho} P_\mu) \\
&[M_{\mu\nu}, M_{\rho\sigma}] = -i(\eta_{\mu\rho} M_{\nu\sigma} + \eta_{\nu\sigma} M_{\mu\rho} - \eta_{\mu\sigma} M_{\nu\rho} - \eta_{\nu\rho} M_{\mu\sigma}) \\
&[P_\mu, Q_a] = 0 \\
&[M_{\mu\nu}, Q_a] = -\left(\frac{i}{4}[\gamma_\mu, \gamma_\nu]\right)_{ab} Q_b \\
&\{Q_a, \bar{Q}_b\} = 2(\gamma^\mu)_{ab} P_\mu \\
&\{Q_a, Q_b\} = -2(\gamma^\mu C)_{ab} P_\mu \\
&\{\bar{Q}_a, \bar{Q}_b\} = -2(C^{-1}\gamma^\mu)_{ab} P_\mu
\end{aligned}
\tag{2.1}
$$

Since the commutator bracket for the spinor charge and the momentum shows that Q commutes with the Hamiltonian, it can be concluded that states of non-zero energy are paired by the action of Q. This implies that supersymmetric multiplets contain an equal number of bosonic and fermionic particles of the same mass.

The Wess-Zumino model[12] is a field theoretical realization of the Super-Poincaré algebra. The lowest dimensional representation of the supersymmetric algebra corresponds to a system of two spin-0 fields (1 scalar and the other pseudo-scalar) and one spin-½ field. Symbolically we present this as :

A is a scalar field

B is a pseudo scalar field

Ψ is a spin-1/2 Majorana field

In the Supersymmetric Lagrangian all three fields couple with the same mass m and the

same coupling constant The defined spinor charges and charge density that necessarily result from the conserved current can be shown to satisfy the commutation and anti-commutation relations of the Super symmetric algebra; thus the Wess-Zumino model is indeed a valid field theoretical realization of such an algebra. The Supersymmetry transformations transform the scalar and pseudo scalar fields into spin-½ fields and the spin-½ fields into linear combinations of the scalar and pseudo scalar fields.

2.2.1 Breaking Supersymmetry

Since the supersymmetric partners have never been observed, it is assumed that supersymmetry is broken at the currently available energies. Theoretical mechanisms for breaking of supersymmetry have proven cumbersome, so it is preferred practice to break the symmetry by manually introducing explicit mass terms into the Lagrangian[13]. Although the introduction of these "soft-breaking" terms remove some of aesthetic appeal of supersymmetry, it enables techniques to be developed to facilitate the analysis of interactions at energies where the supersymmetry is broken. The addition of the extra mass terms must not destroy the renormalizability of the model. Before writing down a left-right, supersymmetric Lagrangian, we will discuss the implications of left-right symmetry.

2.3 Left-Right Symmetry

One of the unsolved problems of the standard model is understanding the origin of parity violation in low-energy physics. An interesting approach[14] is to assume that the

interaction Lagrangian is both left-right (L-R) symmetric but that this parity symmetry is broken at low energies. This formulation produces all the effects of the SM, but as we move up in energies, new effects associated with parity invariance of the Lagrangian are observed. Another important reason to consider a L-R symmetric model is the question of neutrino mass. We do not know whether the neutrino has mass or not.

The L-R symmetric model considered here is based on the gauge group $SU(2)_L$ x $SU(2)_R$ x $U(1)_{B-L}$. For the first generation leptons an quarks these doublets are:

$$L_L = \begin{pmatrix} \nu \\ e \end{pmatrix}_L ; L_R = \begin{pmatrix} \nu \\ e \end{pmatrix}_R ; \; Q_L = \begin{pmatrix} u \\ d \end{pmatrix}_L ; \; Q_R = \begin{pmatrix} u \\ d \end{pmatrix}_R ; \tag{2.2}$$

The L-R Symmetry can be broken in three steps:

i) Parity symmetry breaking the equality between g_L and g_R

ii) Breaks $SU(2)_R$. This is chosen to break at the same scale as the parity symmetry.

iii) By the non-zero expectation of a Higgs bi-doublet.

This L-R symmetric model offers a refinement of the standard electroweak theory; most notably it predicts a small mass for the neutrino. This symmetry will be considered in conjunction with supersymmetry in the next section.

2.4 Left-Right Supersymmetry

In this section a supersymmetric version of $SU(2)_L$ x $SU(2)_R$ x $U(1)_{B-L}$ is presented. The fields are listed and the Lagrangian is written down. Only a very sketchy outline of the fields and the symmetry breaking is given. For a more complete discussion see Francis [13]. Table 1 is a summary of the fields, the quantum numbers and their names.

Table 1. Supersymmetry particle suite

Superfield	Component Fields	Quantum Numbers $SU(2)_L x SU(2)_R x U(1)_{B-L}$	Name
Matter Fields			
Q_L	$\begin{pmatrix} u_L \\ d_L \end{pmatrix} \equiv Q_L$	½ 0 $^1/_3$	L-h up quark L-h down quark
Q_R	$\begin{pmatrix} u_R \\ d_R \end{pmatrix} \equiv Q_R$	0 ½ $^1/_3$	R-h up quark R-h down quark
L_L	$\begin{pmatrix} \nu_L \\ e_L^- \end{pmatrix} \equiv L_L$	½ 0 -1	L-h electron L-h neutrino
L_R	$\begin{pmatrix} \nu_R \\ e_R^- \end{pmatrix} \equiv L_R$	0 ½ -1	R-h electron R-h neutrino
$\tilde{Q}_L$	$\begin{pmatrix} \tilde{u}_L \\ \tilde{d}_L \end{pmatrix} \equiv Q_L$	½ 0 $^1/_3$	L-h up squark L-h down squark
$\tilde{Q}_R$	$\begin{pmatrix} \tilde{u}_R \\ \tilde{d}_R \end{pmatrix} \equiv Q_R$	0 ½ $^1/_3$	R-h up squark R-h down squark
$\tilde{L}_L$	$\begin{pmatrix} \tilde{\nu}_L \\ \tilde{e}_L^- \end{pmatrix} \equiv \tilde{L}_L$	½ 0 -1	L-h up selectron L-h down sneutrino
$\tilde{L}_R$	$\begin{pmatrix} \tilde{\nu}_R \\ \tilde{e}_R^- \end{pmatrix} \equiv \tilde{L}_R$	½ 0 -1	R-h up selectron R-h down sneutrino
Gauge Fields			

Table 1. Supersymmetry particle suite

Superfield	Component Fields	Quantum Numbers $SU(2)_L x SU(2)_R x U(1)_{B-L}$	Name
W_L	$W_L^+ ; W_L^- ; W_L^0$	Triplet Singlet Singlet	Gauge boson
W_R	$W_R^+ ; W_R^- ; W_R^0$	Singlet Triplet Singlet	Gauge boson
V	V	Singlet Singlet Singlet	Gauge boson
λ_L	$\lambda_L^+ ; \lambda_L^- ; \lambda_L^0$	Triplet Singlet Singlet	Gaugino
λ_R	$\lambda_R^+ ; \lambda_R^- ; \lambda_R^0 ;$	Singlet Triplet Singlet	Gaugino
λ_v	λ_v	Singlet Singlet Singlet	Gaugino
Higgs			
$\Phi_{u,d}$	$\begin{pmatrix} \phi_1^0 & \phi_1^+ \\ \phi_2^- & \phi_2^0 \end{pmatrix}$	½ ½ 0	Higgs Boson
Δ_L	$\begin{pmatrix} \frac{1}{\sqrt{2}}\Delta^+ & \Delta^{++} \\ \Delta^0 & -\frac{1}{\sqrt{2}}\Delta^+ \end{pmatrix}_L$	1 ½ 2	Higgs Boson
Δ_R	$\begin{pmatrix} \frac{1}{\sqrt{2}}\Delta^+ & \Delta^{++} \\ \Delta^0 & -\frac{1}{\sqrt{2}}\Delta^+ \end{pmatrix}_R$	0 1 2	Higgs Boson
δ_L	$\begin{pmatrix} \frac{1}{\sqrt{2}}\delta^- & \delta^0 \\ \delta^{--} & -\frac{1}{\sqrt{2}}\delta^- \end{pmatrix}_L$	1 0 -2	Higgs Boson
δ_R	$\begin{pmatrix} \frac{1}{\sqrt{2}}\delta^- & \delta^0 \\ \delta^{--} & -\frac{1}{\sqrt{2}}\delta^- \end{pmatrix}_R$	0 1 -2	Higgs Boson
$\tilde{\Phi}_{u,d}$	$\begin{pmatrix} \tilde{\phi}^0 & \tilde{\phi}^+ \\ \tilde{\phi}^- & \tilde{\phi}^0 \end{pmatrix}$	0 1 2	Higgsino
$\tilde{\Delta}_L$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\tilde{\Delta}^+ & \tilde{\Delta}^{++} \\ \Delta^0 & \tilde{\Delta}^+ \end{pmatrix}_L$	1 0 -2	Higgsino
$\tilde{\Delta}_R$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\tilde{\Delta}^+ & \tilde{\Delta}^{++} \\ \Delta^0 & \tilde{\Delta}^+ \end{pmatrix}_R$	0 1 -2	Higgsino
$\tilde{\delta}_L$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\tilde{\delta}^- & \tilde{\delta}^0 \\ \tilde{\delta}^{--} & \frac{1}{\sqrt{2}}\tilde{\delta}^- \end{pmatrix}_L$	½ ½ 0	Higgsino
$\tilde{\delta}_R$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\tilde{\delta}^- & \tilde{\delta}^0 \\ \tilde{\delta}^{--} & \frac{1}{\sqrt{2}}\tilde{\delta}^- \end{pmatrix}_R$	½ ½ 0	Higgsino

The breaking of the L-R supersymmetry follows that of section 2.1.2 with the addition of the SSBT (Soft supersymmetry breaking terms):

$$(SUSY)\cdot SU(2)_L xU(1)_Y \xrightarrow{\{\Phi_{u,d}\}\neq 0,\&SSBT} U(1)_{em} \tag{2.3}$$

The resulting Lagrangian consists of five pieces: the gauge Lagrangian which contains the kinetic and self interaction terms for the vector field and the Dirac Lagrangian for the gaugino fields; the matter Lagrangian which contains the kinetic terms for the fermionic and bosonic matter fields; the Yukawa Lagrangian which involves the self interactions of the matter multiplets, the scalar potential and the soft-breaking Lagrangian.

The aim of this thesis is to develop the techniques to enable us to actually calculate supersymmetric interactions. In order to be able to do this, we need to have a procedure that permits us to write down the invariant amplitude for a process. The most natural way of doing this is to have the Feynman rules for the process. In order to arrive at the Feynman rules for a supersymmetric process we start with the interaction Lagrangian. After the breaking of supersymmetry, we determine the mass states of the Lagrangian that results from the mixing of the supersymmetric gauge particles. Armed with the definitions of the mass states, the Lagrangian for a given interaction may be written in terms of them, and the Feynman rules are read directly from the resulting expression. This procedure is further detailed in the following section.

2.4.1 Mass Eigenstates of the L-R SUSY Model

Analogously to the mechanism in the electroweak standard model discussed in section 1.1.5, after the spontaneous breaking of supersymmetry, all the gauge bosons acquire mass. Diagonalization of the mass matrix then yields the mass eigenstates of the bosons. However in the breaking of supersymmetry in parallel to the rearrangement of the vector fields after the breaking of the gauge symmetries, the gauginos interact with the Higgs bosons and the higgsinos to produce new particles referred to as neutralinos and charginos. The neutralinos and charginos are of interest because these are the states that interact with the sleptons, squarks and leptons, quarks in the low energy production of sleptons in the interaction:

$$ep \rightarrow sleptons \tag{2.4}$$

The physical neutralino states are determined by a rather convoluted procedure the outline of which is given here. The Lagrangian for the light neutralinos is shown by Francis[13] to be:

$$L_{NML} = -i\lambda_L^0 \frac{1}{\sqrt{2}} g_L \kappa_u \tilde{\Phi}_{1u}^0 + \frac{i\lambda_B^0 \frac{1}{\sqrt{2}} g_v g_R \kappa_u \tilde{\Phi}_{1u}^0}{\left(g_r^2 + 4g_v^2\right)^{\frac{1}{2}}}$$

$$+ i\lambda_L^0 \frac{1}{\sqrt{2}} g_L \kappa_u \tilde{\Phi}_{2d}^0 - \frac{i\lambda_B^0 \frac{1}{\sqrt{2}} g_v g_R \kappa_u \tilde{\Phi}_{2d}^0}{\left(g_r^2 + 4g_v^2\right)^{\frac{1}{2}}}$$

$$+ m_L \lambda_L^0 \lambda_L^0 + \frac{m_r 4g_v^2 \lambda_B^0 \lambda_B^0}{g_r^2 + 4g_v^2} + \frac{m_v g_v^2 \lambda_B^0 \lambda_B^0}{g_r^2 + 4g_v^2}$$

$$+ 2\mu_1 \tilde{\Phi}_u^0 \tilde{\Phi}_d^0 + h.c. \qquad (2.5)$$

The mass eigenstates are identified by finding and diagonalizing the mass matrix M^n defined by:

$$L_{NML} = -\frac{1}{2}\left(\Omega^0\right)^T M^n \Omega^0 + h.c.$$

$$\text{where } \left(\Omega^0\right)^T = \left(-1\lambda_L^0, -1\lambda_B^0, \tilde{\Phi}_{1u}^0, \tilde{\Phi}_{2d}^0\right) \qquad (2.6)$$

The mass matrix may be read directly off the Lagrangian and is in general a complex symmetric matrix given by:

$$M^n = \begin{bmatrix} M_L & 0 & -\frac{1}{\sqrt{2}} g_L \kappa_u & \frac{1}{\sqrt{2}} g_L \kappa_u \\ 0 & \frac{4M_R g_v^2 + M_v g_R^2}{g_1^2} & \frac{\sqrt{2} g_v g_R \kappa_u}{g_1} & \frac{-\sqrt{2} g_v g_R \kappa_d}{g_1} \\ -\frac{1}{\sqrt{2}} g_L \kappa_u & \frac{\sqrt{2} g_v g_R \kappa_u}{g_1} & 0 & -2\mu \\ \frac{1}{\sqrt{2}} g_L \kappa_u & \frac{-\sqrt{2} g_v g_R \kappa_d}{g_1} & -2\mu & 0 \end{bmatrix} \qquad (2.7)$$

The diagonalizing procedure results in the physical neutralino states:

$$\chi_i^0 = M_{ij}\Omega_j^0, \quad (i,j = 1,..,4) \tag{2.8}$$

whose masses are the positive square roots of:

$$M_D^2 = MM^{n+}M^nM^{-1} \tag{2.9}$$

The values of M and M_D are unknown and will be determined by assuming values for the couplings and through variation of the higgsino mass parameter μ, in the computer model developed in chapter 3. The Feynman rules will be written in terms of these parameters. The Feynman rules for the neutralino-lepton-slepton and the neutralino-quark-squark interactions are required to calculate the cross section for the interaction represented by equation 2.4 above, but first we determine the chargino mass states.

In a similar manner, the physical chargino mass states are determined. The Lagrangian for the charged gaugino-higgsino mixing is:

$$\begin{aligned} L_{CH} = {} & \frac{i}{\sqrt{2}} g_R v_R \bar{\Delta}_R^+ \lambda_R^- + ig_R \kappa_u \tilde{\phi}_d^{\,+} \lambda_R^- + ig_L \kappa_d \tilde{\phi}_d^{\,+} \lambda_L^- \\ & + ig_R \kappa_u \tilde{\phi}_d^{\,-} \lambda_R^+ + ig_L \kappa_d \tilde{\phi}_d^{\,-} \lambda_L^+ + M_L \lambda_L^+ \lambda_L^- \\ & + M_R \lambda_R^+ \lambda_R^- + \mu \tilde{\phi}_u^{\,+} \tilde{\phi}_d^{\,-} + \mu \tilde{\phi}_u^{\,-} \tilde{\phi}_d^{\,+} \\ & + h.c. \end{aligned} \tag{2.10}$$

The mass eigenstates are identified by finding and diagonalizing the mass matrix M^c defined by:

$$L_{CH} = -\frac{1}{2}\left(\psi^{+}\psi^{-}\right)M^{c}\begin{pmatrix}\psi^{+}\\ \psi^{-}\end{pmatrix} + h.c.$$

where $\psi^{+} = \left(-i\lambda_{L}^{+}, -i\lambda_{R}^{+}, \tilde{\phi}_{u}^{+}, \tilde{\phi}_{d}^{+}\right)$

$$\psi^{-} = \left(-i\lambda_{L}^{-}, -i\lambda_{R}^{-}, \tilde{\phi}_{u}^{-}, \tilde{\phi}_{d}^{-}\right) \qquad (2.11)$$

The mass matrix is read directly off the Lagrangian and is given by:

$$M^{c} = \begin{bmatrix} M_{L} & 0 & 0 & \sqrt{2}M_{W}\cos\theta_{\kappa} \\ 0 & M_{R} & 0 & \sqrt{2}M_{W}\cos\theta_{\kappa} \\ \sqrt{2}M_{W}\sin\theta_{\kappa} & \sqrt{2}M_{W}\sin\theta_{\kappa} & 0 & -\mu \\ 0 & 0 & -\mu & 0 \end{bmatrix} \qquad (2.12)$$

where

$g\kappa_{u} = \sqrt{2}M_{W}\sin\theta_{\kappa}$

$g\kappa_{d} = \sqrt{2}M_{W}\cos\theta_{\kappa}$

Since M^{c} is an asymmetric matrix, we require two matrices U and V to diagonalize the elements of M_{D}. The diagonalizing procedure results in the physical chargino states given by:

$$\chi_{i}^{+} = V_{ij}\psi_{j}^{+}, \quad \chi_{i}^{-} = U_{ij}\psi_{j}^{-} \quad (i,j = 1,\ldots,4) \qquad (2.13)$$

whose masses are the positive square roots of:

$$M_{D}^{2} = VM^{c+}M^{c}V^{-1} = UM^{c}M^{c+}U^{-1} \qquad (2.14)$$

The values of V, U and M_{D} are unknown and will be determined by assuming values for the gauge boson masses, the couplings and through variation of the higgsino mass parameter μ, in the computer model developed in chapter 3. The Feynman rules will be

written in terms of these parameters. The Feynman rules for the chargino-lepton-slepton and the chargino-quark-squark interactions are required to calculate the cross section for the interaction represented by equation 2.4 above.

2.4.2 Derivation of the Feynman Rules

The neutralino-lepton-slepton rules have been defined by Francis[13] and are listed in Figure 5 as rules 18-26. The neutralino-quark-squark rules are determined in a similar manner: The piece of the Lagrangian that determines this interaction is called L_{SI}, defined as:

$$\begin{aligned} L_{SI} = & \frac{i}{\sqrt{2}}\tilde{u}_L^{\dagger}\left(g_L\lambda_L^0 + \frac{g_V}{3}\lambda_V^0\right)u_L - \frac{i}{\sqrt{2}}\tilde{d}_L^{\dagger}\left(g_L\lambda_L^0 - \frac{g_V}{3}\lambda_V^0\right)d_L \\ & + \frac{i}{\sqrt{2}}\tilde{u}_R^{\dagger}\left(g_R\lambda_R^0 + \frac{g_V}{3}\lambda_V^0\right)u_R - \frac{i}{\sqrt{2}}\tilde{d}_L^{\dagger}\left(g_R\lambda_R^0 - \frac{g_V}{3}\lambda_V^0\right)d_L \\ & + h_u^Q\left(\tilde{u}_L^{\dagger}\tilde{\phi}_{1u}^0 u_R + \tilde{d}_L^{\dagger}\tilde{\phi}_{2u}^0 d_R + \tilde{u}_R^{\dagger}\tilde{\phi}_{1u}^0 u_R + \tilde{d}_R^{\dagger}\tilde{\phi}_{2u}^0 d_L\right) \\ & + h_d^Q\left(\tilde{u}_L^{\dagger}\tilde{\phi}_{1d}^0 u_R + \tilde{d}_L^{\dagger}\tilde{\phi}_{2d}^0 d_R + \tilde{u}_R^{\dagger}\tilde{\phi}_{1d}^0 u_R + \tilde{d}_R^{\dagger}\tilde{\phi}_{2d}^0 d_L\right) \end{aligned} \tag{2.15}$$

The conversion to Neutralino mass states requires the conversion to four-component notation and the recognition that weak interaction eigenstates can be represented in terms of the mass eigenstates as follows:

$$\tilde{W}_L^0 \equiv \begin{pmatrix} -i\lambda_L^0 \\ i\bar{\lambda}_L^0 \end{pmatrix} \; ; \; \tilde{W}_B^0 \equiv \begin{pmatrix} -i\lambda_B^0 \\ i\bar{\lambda}_B^0 \end{pmatrix}$$

$$\tilde{F}_{1u}^0 \equiv \begin{pmatrix} \tilde{\Phi}_{1u}^0 \\ \bar{\tilde{\Phi}}_{1u}^0 \end{pmatrix} \; ; \; \tilde{F}_{1d}^0 \equiv \begin{pmatrix} \tilde{\Phi}_{1d}^0 \\ \bar{\tilde{\Phi}}_{1d}^0 \end{pmatrix} \; ; \; \tilde{F}_{2u}^0 \equiv \begin{pmatrix} \tilde{\Phi}_{2u}^0 \\ \bar{\tilde{\Phi}}_{2u}^0 \end{pmatrix} \; ; \; \tilde{F}_{2d}^0 \equiv \begin{pmatrix} \tilde{\Phi}_{2d}^0 \\ \bar{\tilde{\Phi}}_{2d}^0 \end{pmatrix} \; ;$$

$$\tilde{D}_L^0 \equiv \begin{pmatrix} \tilde{\Delta}_L^0 \\ \bar{\tilde{\Delta}}_L^0 \end{pmatrix} \; ; \; \tilde{D}_R^0 \equiv \begin{pmatrix} \tilde{\Delta}_R^0 \\ \bar{\tilde{\Delta}}_R^0 \end{pmatrix} \qquad (2.16)$$

$$\bar{\tilde{W}}_L^0 P_L = \left(M_{11}\bar{\tilde{\chi}}_1^0 + M_{21}\bar{\tilde{\chi}}_2^0 + M_{31}\bar{\tilde{\chi}}_3^0 + M_{41}\bar{\tilde{\chi}}_4^0\right)P_L$$

$$\bar{\tilde{W}}_L^0 P_L = \left(M_{12}\bar{\tilde{\chi}}_1^0 + M_{22}\bar{\tilde{\chi}}_2^0 + M_{32}\bar{\tilde{\chi}}_3^0 + M_{42}\bar{\tilde{\chi}}_4^0\right)P_L$$

$$P_L\tilde{W}_L^0 = P_L\left(M_{12}^*\tilde{\chi}_1^0 + M_{22}^*\tilde{\chi}_2^0 + M_{32}^*\tilde{\chi}_3^0 + M_{42}^*\tilde{\chi}_4^0\right)$$

$$P_L\tilde{F}_{1u}^0 = P_L\left(M_{13}^*\tilde{\chi}_1^0 + M_{23}^*\tilde{\chi}_2^0 + M_{33}^*\tilde{\chi}_3^0 + M_{43}^*\tilde{\chi}_4^0\right)$$

$$\bar{\tilde{F}}_{1u}^0 P_L = \left(M_{13}\bar{\tilde{\chi}}_1^0 + M_{23}\bar{\tilde{\chi}}_2^0 + M_{33}\bar{\tilde{\chi}}_3^0 + M_{43}\bar{\tilde{\chi}}_4^0\right)P_L$$

$$P_L\tilde{F}_{2d}^0 = P_L\left(M_{14}^*\tilde{\chi}_1^0 + M_{24}^*\tilde{\chi}_2^0 + M_{34}^*\tilde{\chi}_3^0 + M_{44}^*\tilde{\chi}_4^0\right)$$

$$\bar{\tilde{F}}_{2d}^0 P_L = \left(M_{14}\bar{\tilde{\chi}}_1^0 + M_{24}\bar{\tilde{\chi}}_2^0 + M_{34}\bar{\tilde{\chi}}_3^0 + M_{44}\bar{\tilde{\chi}}_4^0\right)P_L \qquad (2.17)$$

where M_{ij} are defined in equation 2.9. Substituting equation 2.16 and 2.17 into the interaction Lagrangian 2.15 permits the Feynman rules to be read directly. These rules are graphically presented in figures 4 and 5. The next step is to derive the rules for the chargino interactions.

The chargino-lepton-slepton rules have been defined by Francis[13] and are listed in figure 7 as rules 26-30. The chargino-quark-slepton rules are determined in a similar manner to the neutralino rules. The piece of the Lagrangian that determines this interaction is called L_{chqs}, defined as:

$$
\begin{aligned}
L_{chqs} = i\Big\{ & g_L\left(\lambda_L^- u_L \tilde{d}_L^* - \bar{\lambda}_L^- \bar{u}_L \tilde{d}_L + \lambda_L^+ d_L \tilde{u}_L^* - \lambda_L^+ \bar{d}_L \tilde{u}_L\right) \\
& + g_R\left(\lambda_R^- u_R \tilde{d}_R^* - \bar{\lambda}_R^- \bar{u}_R \tilde{d}_R + \lambda_R^+ d_R \tilde{u}_R^* - \lambda_R^+ \bar{d}_R \tilde{u}_R\right) \\
& + h_u^Q\left(\tilde{\phi}_u^- u_R \tilde{d}_L^* + \bar{\tilde{\phi}}_u^- \bar{u}_R \tilde{d}_L^* + \tilde{\phi}_u^+ d_R \tilde{u}_L^* + \bar{\tilde{\phi}}_u^+ \bar{d}_R \tilde{u}_L^*\right) \\
& + h_d^Q\left(\tilde{\phi}_d^- u_R \tilde{d}_L^* + \bar{\tilde{\phi}}_d^- \bar{u}_R \tilde{d}_L^* + \tilde{\phi}_d^+ d_R \tilde{u}_L^* + \bar{\tilde{\phi}}_d^+ \bar{d}_R \tilde{u}_L^*\right) \\
& + h_u^Q\left(\tilde{\phi}_u^- u_L \tilde{d}_R^* + \bar{\tilde{\phi}}_u^- \bar{u}_L \tilde{d}_R^* + \tilde{\phi}_u^+ d_L \tilde{u}_R^* + \bar{\tilde{\phi}}_u^+ \bar{d}_L \tilde{u}_R^*\right) \\
& + h_d^Q\left(\tilde{\phi}_d^- u_L \tilde{d}_R^* + \bar{\tilde{\phi}}_d^- \bar{u}_L \tilde{d}_R^* + \tilde{\phi}_d^+ d_L \tilde{u}_R^* + \bar{\tilde{\phi}}_d^+ \bar{d}_L \tilde{u}_R^*\right)\Big\}
\end{aligned}
\tag{2.18}
$$

The conversion to chargino mass states requires the conversion to four-component notation and the recognition that weak interaction eigenstates can be represented in terms of the mass eigenstates as follows:

$$
\begin{aligned}
& \tilde{W}_L \equiv \begin{pmatrix} -i\bar{\lambda}_L^+ \\ i\bar{\lambda}_L^- \end{pmatrix} \; ; \quad \tilde{W}_R \equiv \begin{pmatrix} -i\bar{\lambda}_R^+ \\ i\bar{\lambda}_R^- \end{pmatrix} \\
& \tilde{F}_u \equiv \begin{pmatrix} \tilde{\Phi}_{1u}^0 \\ \bar{\tilde{\Phi}}_{1u}^0 \end{pmatrix} \; ; \quad \tilde{F}_d \equiv \begin{pmatrix} \tilde{\Phi}_{1d}^0 \\ \bar{\tilde{\Phi}}_{1d}^0 \end{pmatrix} \; ; \quad \tilde{D}_L^1 \equiv \begin{pmatrix} \tilde{\Delta}_L^+ \\ \bar{\tilde{\Delta}}_L^- \end{pmatrix} \; ; \\
& \tilde{D}_R^1 \equiv \begin{pmatrix} \tilde{\Delta}_L^+ \\ \bar{\tilde{\Delta}}_L^- \end{pmatrix} \; ; \quad \tilde{D}_L^2 \equiv \begin{pmatrix} \tilde{\Delta}_L^{++} \\ \bar{\tilde{\Delta}}_L^{--} \end{pmatrix} \; ; \quad \tilde{D}_R^2 \equiv \begin{pmatrix} \tilde{\Delta}_L^{++} \\ \bar{\tilde{\Delta}}_L^{--} \end{pmatrix}
\end{aligned}
\tag{2.19}
$$

$$
\begin{aligned}
\bar{\tilde{W}}_L P_L &= \left(U_{11}^* \tilde{\chi}_1 + U_{21}^* \tilde{\chi}_2 + U_{31}^* \tilde{\chi}_3 + U_{41}^* \tilde{\chi}_4\right) P_L \\
\bar{\tilde{W}}_L^c P_L &= \left(V_{11}^* \bar{\tilde{\chi}}_1^c + V_{21}^* \bar{\tilde{\chi}}_2^c + V_{31}^* \bar{\tilde{\chi}}_3^c + V_{41}^* \bar{\tilde{\chi}}_4^c\right) P_L \\
P_R \bar{\tilde{W}}_L^c &= P_R\left(V_{11} \bar{\tilde{\chi}}_1^c + V_{21} \bar{\tilde{\chi}}_2^c + V_{31} \bar{\tilde{\chi}}_3^c + V_{41} \bar{\tilde{\chi}}_4^c\right) \\
P_L \bar{\tilde{W}}_R^c &= P_L\left(U_{12}^* \bar{\tilde{\chi}}_1^c + U_{22}^* \bar{\tilde{\chi}}_2^c + U_{32}^* \bar{\tilde{\chi}}_3^c + U_{42}^* \bar{\tilde{\chi}}_4^c\right) \\
\bar{\tilde{W}}_R^c P_R &= \left(U_{12} \bar{\tilde{\chi}}_1^c + U_{22} \bar{\tilde{\chi}}_2^c + U_{32} \bar{\tilde{\chi}}_3^c + U_{42} \bar{\tilde{\chi}}_4^c\right) P_R \\
P_L \bar{\tilde{W}}_R &= P_L\left(V_{12}^* \tilde{\chi}_1 + V_{22}^* \tilde{\chi}_2 + V_{32}^* \tilde{\chi}_3 + V_{42}^* \tilde{\chi}_4\right) \\
\bar{\tilde{W}}_R P_R &= \left(V_{12} \tilde{\chi}_1 + V_{22} \tilde{\chi}_2 + V_{32} \tilde{\chi}_3 + V_{42} \tilde{\chi}_4\right) P_R
\end{aligned}
\tag{2.20}
$$

where V_{ij} and U_{ij} are defined in equation 2.14. Substituting equation 2.19 and 2.20 into the interaction Lagrangian 2.18 permits the Feynman rules to be read directly. These rules are graphically represented in figures 6 and 7.

2.4.3 The Feynman Rules

The Lagrangian presented by equation 2.15 may be translated into the neutralino states by the application of the equations 2.16 and 2.17. The Feynman rules for neutralino interactions may be read directly off the transformed Lagrangian. The transformed Lagrangian is presented here as a collection of Feynman diagrams with the corresponding Feynman rules written directly underneath each diagram. Figures 4 and 5 each present 13 diagrams numbered sequentially from 1 to 26. These numbers will employed as an index into the list of rules in the calculator developed in chapter 3.

The Lagrangian presented by equation 2.18 may be translated into the chargino states by the application of the equations 2.19 and 2.20. The Feynman rules for chargino interactions may be read directly off the transformed Lagrangian. The transformed Lagrangian is presented here as a collection of Feynman diagrams with the corresponding Feynman rules written directly underneath each diagram. Figures 6 and 7 each present 15 diagrams numbered sequentially from 1 to 30. These numbers will be used as an index into the chargino rules by the calculator presented in chapter 3.

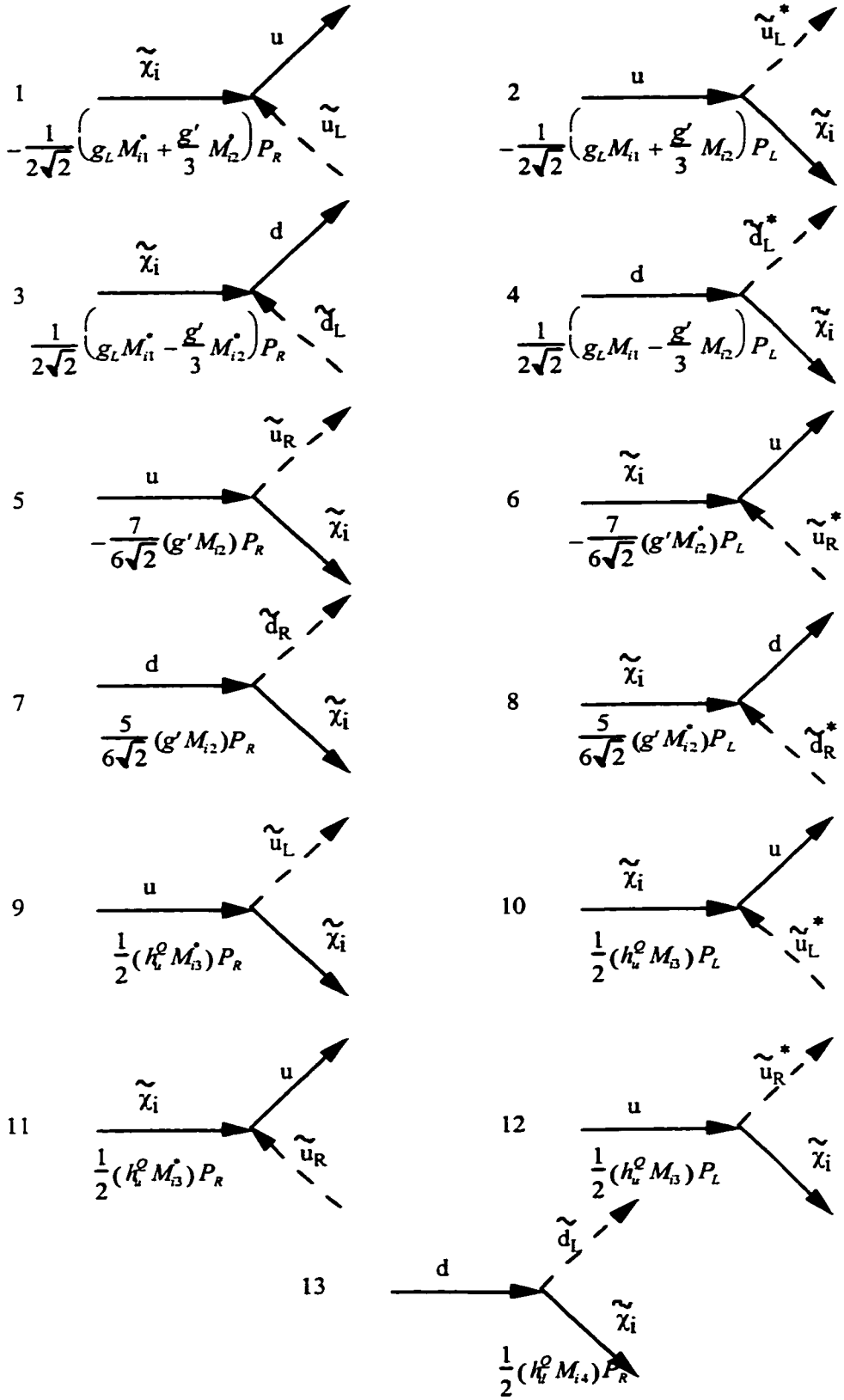

FIG. 4. Feynman rules Neutralino -> Quark Squark (1-13)

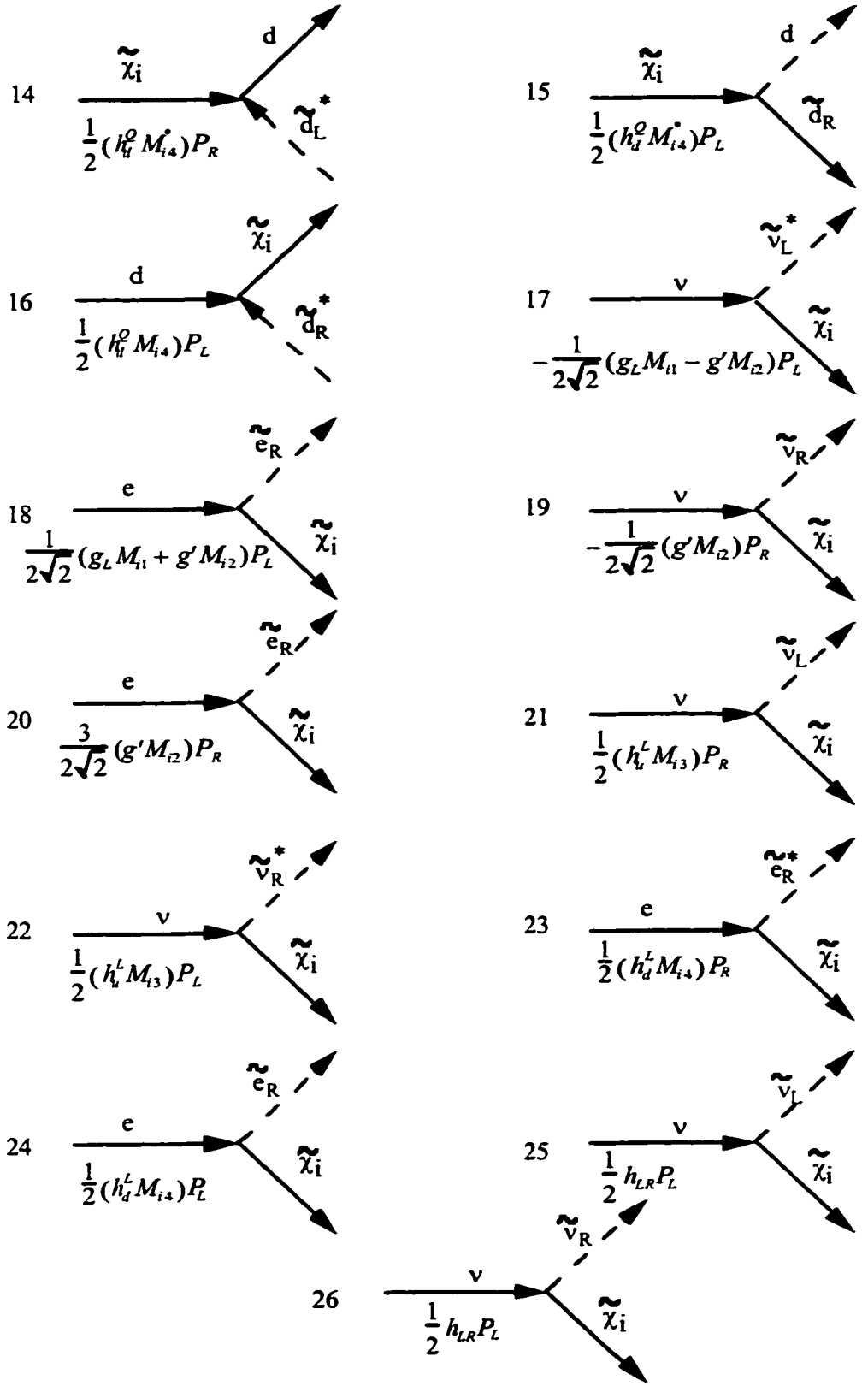

FIG. 5. Feynman rules Neutralino -> Quark Squark (14-26)

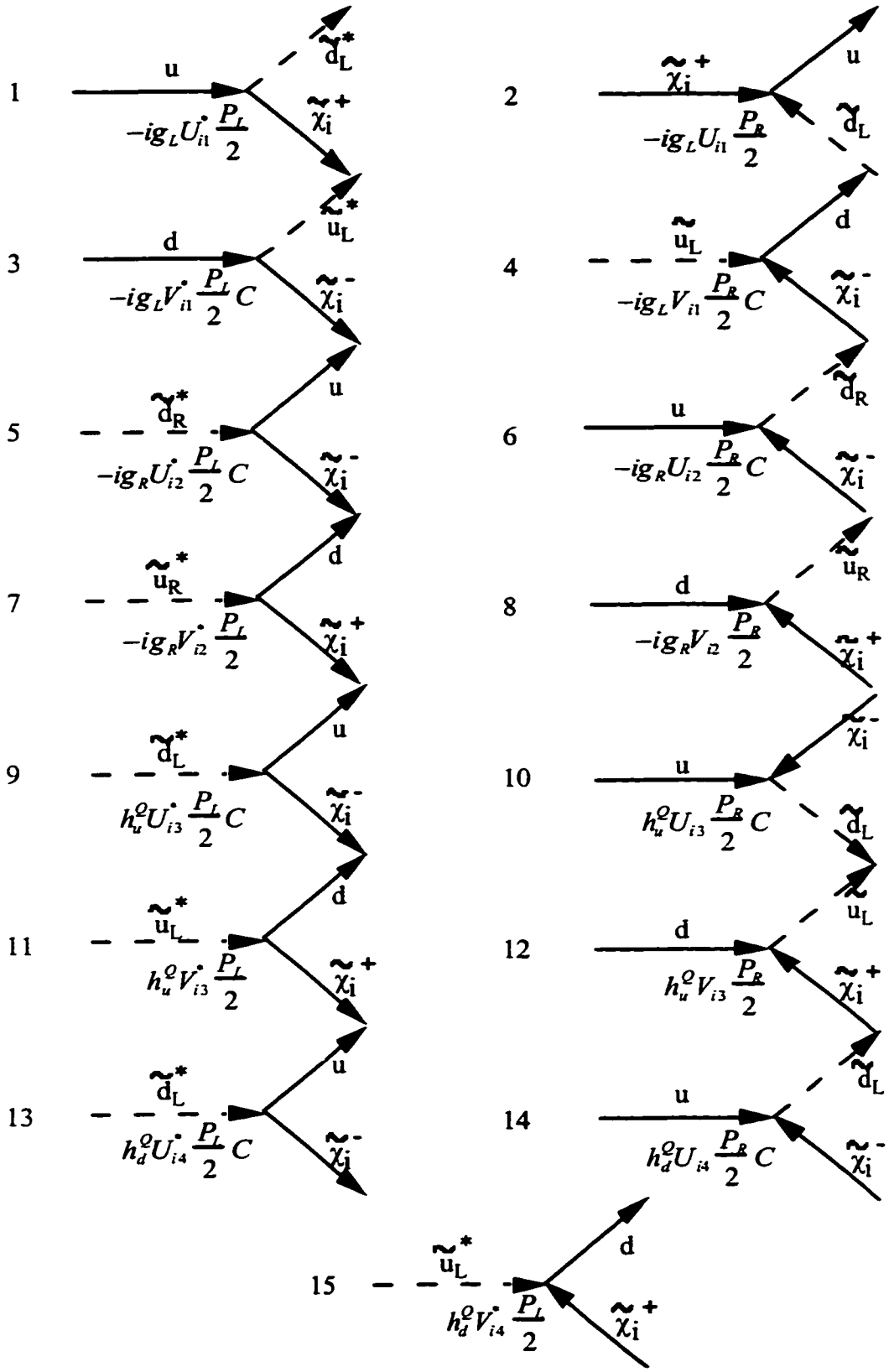

FIG. 6. Feynman rules Chargino -> Quark Squark (1-13)

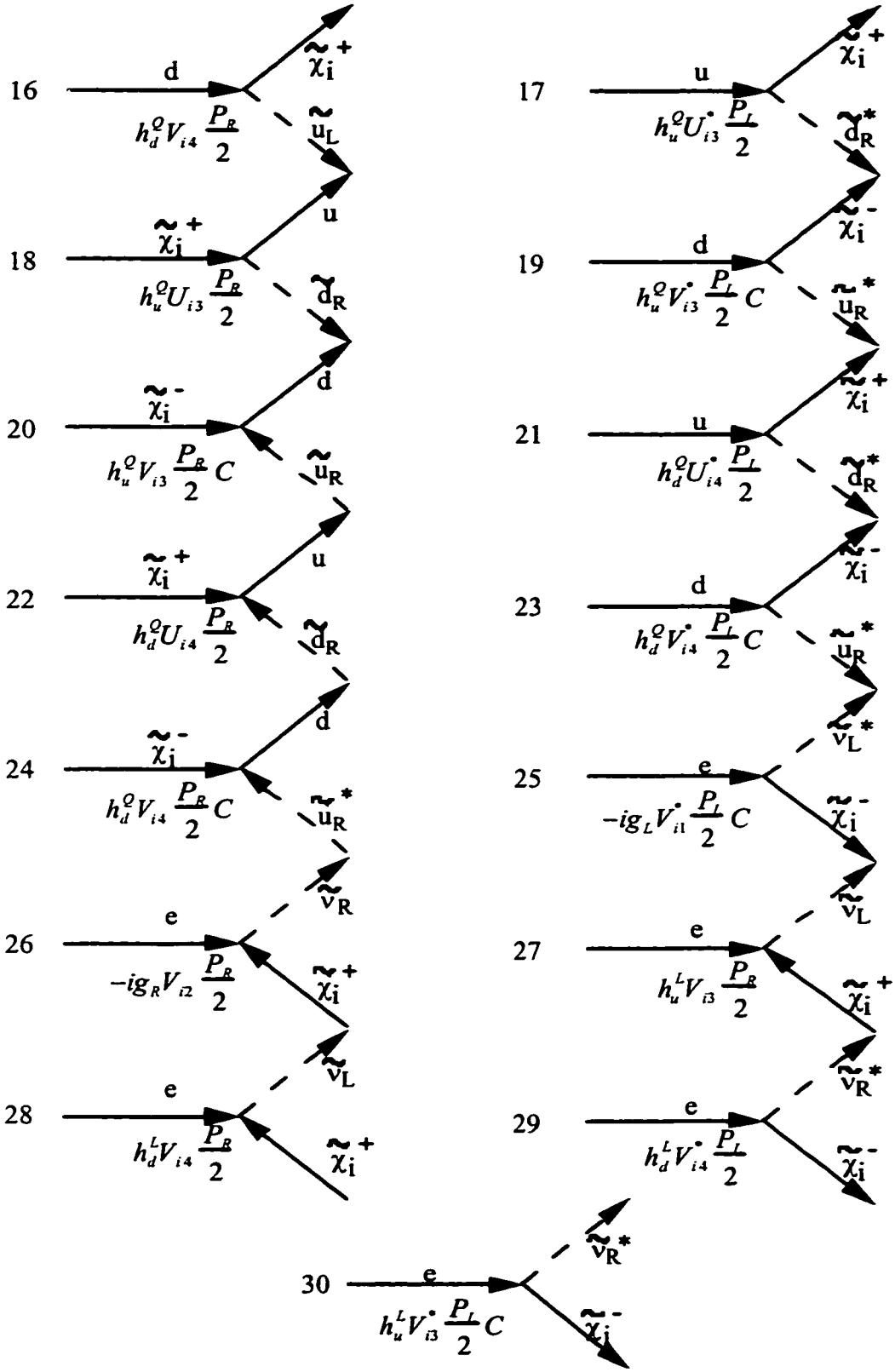

FIG. 7. Feynman rules Chargino -> Quark Squark (14-26) and Chargino -> Lepton Slepton (26-30)

2.5 Calculation of ep -> slepton Cross Sections

As discussed in chapter 1, the key part of a calculation of a scattering cross section, is determining an expression for the square of the invariant amplitude M. The reaction ep -> sleptons can be broken into two portions - the first dealing with neutralino exchange; the second with chargino exchange.

2.5.1 Neutralino Contribution to Invariant Amplitude

The neutralino portion of M consists of contributions from 8 separate Feynman diagrams so that amplitude may be written:

$$M_{Neutralino} = \sum_{i=1}^{i=8} M_i \tag{2.21}$$

The 8 diagrams correspond to the permutations of electrons interacting with either up or down quarks to produce the sleptons and squarks of different handedness. The diagrams are:

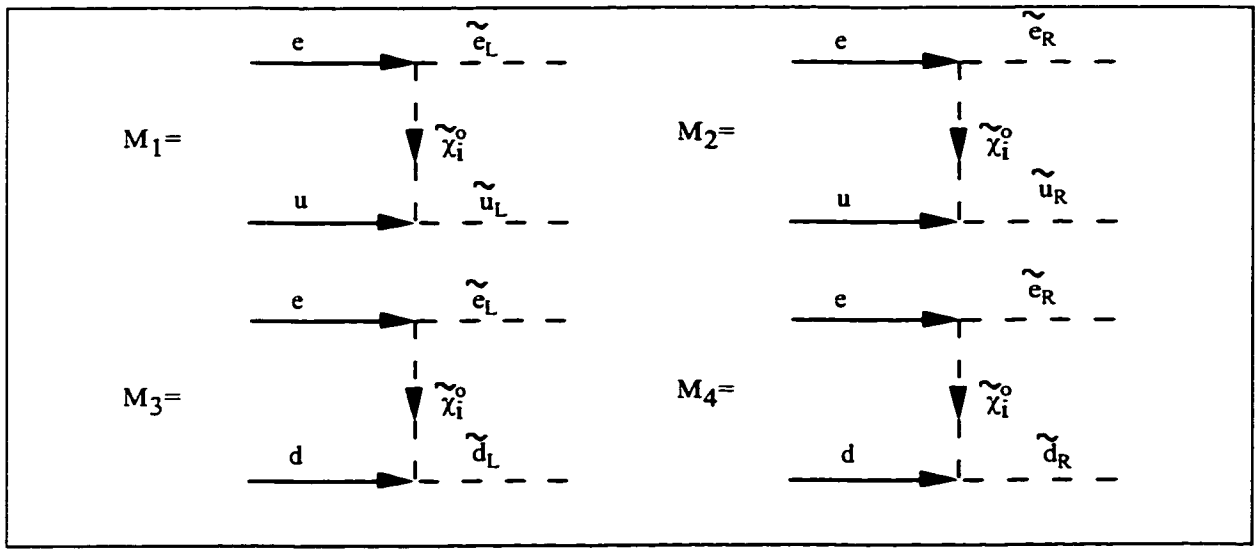

FIG. 8. Neutralino contribution to ep->sleptons invariant amplitude

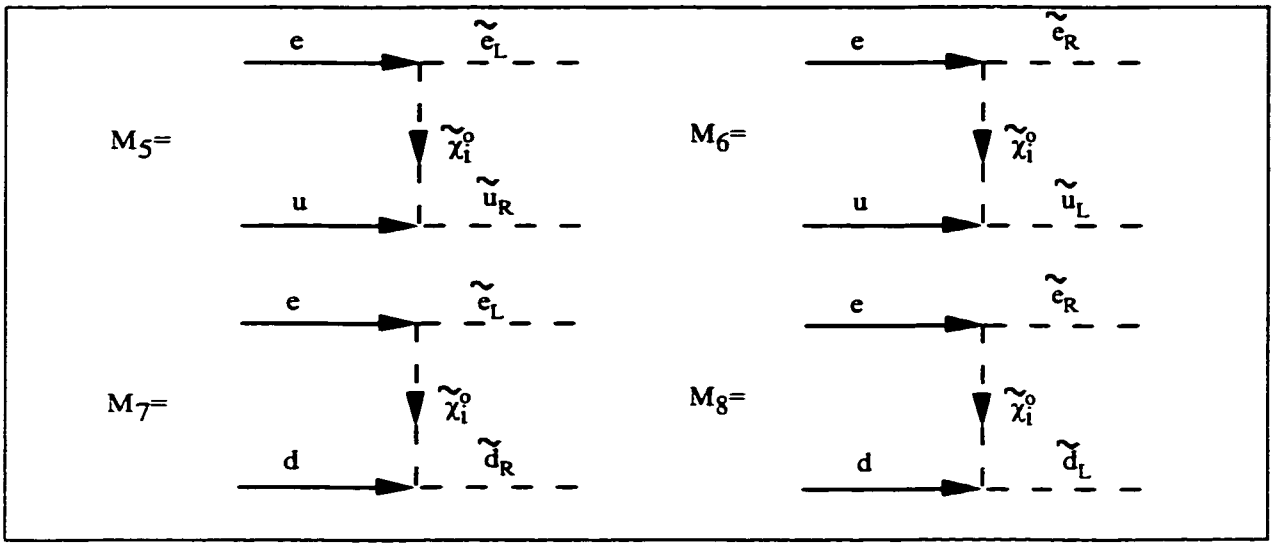

FIG. 9. Neutralino contribution to ep->sleptons invariant amplitude (cont'd)

Treating the neutralino as a standard fermion propagator, the invariant amplitude for each of the contributions to the neutralino invariant amplitude can be written down in terms of the Feynman rules acting at each of the vertices of the above diagrams:

$$M_1 = e\left\{\frac{i}{2\sqrt{2}}\left(g_L M_{i1} + g' M_{i2}\right)P_R + \frac{ih_L^d}{2} M_{i4} P_L\right\}\frac{i(q + m_x)}{q^2 - m_x^2}$$
$$\left\{-\frac{1}{2\sqrt{2}}\left(g_L M_{i1} + \frac{g'}{3} M_{i2}\right)P_L + \frac{ih_u^Q}{2} M_{i3}^{*} P_R\right\}u \qquad (2.22)$$

$$M_2 = e\left\{\frac{3i}{2\sqrt{2}} g' M_{i2} P_R + \frac{ih_L^d}{2} M_{i4} P_L\right\}\frac{i(q + m_x)}{q^2 - m_x^2}$$
$$\left\{-\frac{7}{6\sqrt{2}} g' M_{i2} P_R + \frac{ih_u^Q}{2} M_{i3} P_L\right\}u \qquad (2.23)$$

$$M_3 = e\left\{\frac{i}{2\sqrt{2}}\left(g_L M_{i1} + g' M_{i2}\right)P_L + \frac{ih_L^d}{2} M_{i4} P_R\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{-\frac{1}{2\sqrt{2}}\left(g_L M_{i1} - \frac{g'}{3} M_{i2}\right)P_L + \frac{ih_d^Q}{2} M_{i4} P_R\right\}d \tag{2.24}$$

$$M_4 = e\left\{\frac{3i}{2\sqrt{2}} g' M_{i2} P_R + \frac{ih_L^d}{2} M_{i4} P_L\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{\frac{5}{6\sqrt{2}} g' M_{i2} P_R + \frac{h_d^Q}{2} M_{i3} P_L\right\}d \tag{2.25}$$

$$M_5 = e\left\{\frac{i}{2\sqrt{2}}\left(g_L M_{i1} + g' M_{i2}\right)P_L + \frac{ih_L^d}{2} M_{i4} P_R\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{-\frac{7}{6\sqrt{2}} g' M_{i2} P_R + \frac{h_u^Q}{2} M_{i3} P_L\right\}u \tag{2.26}$$

$$M_6 = e\left\{\frac{3i}{2\sqrt{2}} g' M_{i2} P_R + \frac{ih_L^d}{2} M_{i4} P_L\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{-\frac{1}{2\sqrt{2}}\left(g_L M_{i1} + \frac{g'}{3} M_{i2}\right)P_L + \frac{h_u^Q}{2} M_{i3}^{\bullet} P_R\right\}u \tag{2.27}$$

$$M_7 = e\left\{\frac{i}{2\sqrt{2}}\left(g_L M_{i1} + g' M_{i2}\right)P_L + \frac{ih_L^d}{2} M_{i4} P_R\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{\frac{5}{6\sqrt{2}} g' M_{i2} P_R + \frac{h_d^Q}{2} M_{i4} P_L\right\}d \tag{2.28}$$

$$M_8 = e\left\{\frac{3i}{2\sqrt{2}} g' M_{i2} P_R + \frac{ih_L^d}{2} M_{i4} P_L\right\}\frac{i\left(q + m_x\right)}{q^2 - m_x^2}$$
$$\left\{\frac{1}{2\sqrt{2}}\left(g_L M_{i1} - \frac{g'}{3} M_{i2}\right)P_L + \frac{h_d^Q}{2} M_{i4} P_R\right\}d \tag{2.29}$$

Hand calculation of the expectation of the square of the magnitude of the full expression for M is a daunting task. This type of calculation is very well suited to being performed using an algebraic program language such as Mathematica®. After squaring M and calculating the expectation via application of trace theorems and other simplifications, $\langle |M|^2 \rangle$ is simply written here for comparison with later program output:

$$\begin{aligned}
\langle |M_1|^2 \rangle &\approx \frac{1}{2}\left(g_L M_{i1} + g' M_{i2}\right)^2 \left(g_L M_{i1} + \frac{g'}{3} M_{i2}\right)^2 \left(\frac{m_x}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_2|^2 \rangle &\approx \frac{49}{2}(g')^4 |M_{i2}|^4 \left(\frac{m_x}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_3|^2 \rangle &\approx \frac{1}{2}\left(g_L M_{i1} + g' M_{i2}\right)^2 \left(g_L M_{i1} - \frac{g'}{3} M_{i2}\right)^2 \left(\frac{m_x}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_4|^2 \rangle &\approx \frac{25}{2}(g')^4 |M_{i2}|^4 \left(\frac{m_x}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_5|^2 \rangle &\approx \frac{49}{18}\left(g_L M_{i1} + g' M_{i2}\right)^2 (g')^2 |M_{i2}|^2 \left(\frac{q^2}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_6|^2 \rangle &\approx \frac{9}{2}(g')^2 |M_{i2}|^2 \left(g_L M_{i1} + \frac{g'}{3} M_{i2}\right)^2 \left(\frac{q^2}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_7|^2 \rangle &\approx \frac{25}{18}\left(g_L M_{i1} + g' M_{i2}\right)^2 (g')^2 |M_{i2}|^2 \left(\frac{q^2}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \\
\langle |M_8|^2 \rangle &\approx \frac{9}{2}(g')^2 |M_{i2}|^2 \left(g_L M_{i1} - \frac{g'}{3} M_{i2}\right)^2 \left(\frac{q^2}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u)
\end{aligned} \tag{2.30}$$

The equations for each of the invariant amplitudes listed above are actually expressed in shorthand. For example the portion of the second component M_2, written in terms of the ith neutralino, is fully expressed:

$$\left\langle |M_2|^2 \right\rangle = \frac{49}{2}(g')^4 \left\{ |M_{12}|^4 \left(\frac{m_{x_1^0}}{q^2 - m_{x_1^0}^2} \right)^2 + |M_{22}|^4 \left(\frac{m_{x_2^0}}{q^2 - m_{x_2^0}^2} \right)^2 + \right.$$
$$\left. |M_{32}|^4 \left(\frac{m_{x_3^0}}{q^2 - m_{x_3^0}^2} \right)^2 + |M_{42}|^4 \left(\frac{m_{x_4^0}}{q^2 - m_{x_4^0}^2} \right)^2 \right\} (P_e \cdot P_u) \tag{2.31}$$

where M_{ij} are the elements of the diagonalizing matrix of eigenvectors and $m_{\chi_i^0}$ are the positive square roots of the eigenvalues defined in equation 2.9. Some word on the algebra involved in arriving at the full expression for the expectation of the square of the invariant amplitude is in order. The trace theorems for gamma matrices, the spinor completeness relationships and the projection operator identities have been used extensively. The projection operator identities are summarized as follows:

$$P_R = 1 + \gamma^5$$
$$P_L = 1 - \gamma^5$$
$$P_R P_R = 2P_R$$
$$P_L P_L = 2P_L$$
$$P_R P_L = P_L P_R = 0$$
$$P_R \not{q} P_R = P_L \not{q} P_L = 0$$
$$P_R \not{q} P_L = 2\not{q} P_L$$
$$P_L q P_R = 2\not{q} P_R \tag{2.32}$$

In addition, some order of magnitude approximations have been made; these are:

$$h_u^{Q^2} \approx h_d^{Q^2} \approx h_d^{L^2} \approx m_e m_q \approx 0 \tag{2.33}$$

A similar approach is used to arrive at expressions for the chargino invariant amplitude..

2.5.2 Chargino Contribution to Invariant Amplitude

The chargino portion of M also consists of contributions from 8 separate Feynman diagrams so that amplitude may be written:

$$M_{Chrgino} = \sum_{i=1}^{i=8} M_i \tag{2.34}$$

The 8 diagrams correspond to the permutations of electrons interacting with either up or down quarks to produce the sleptons and squarks of different handedness. The diagrams are:

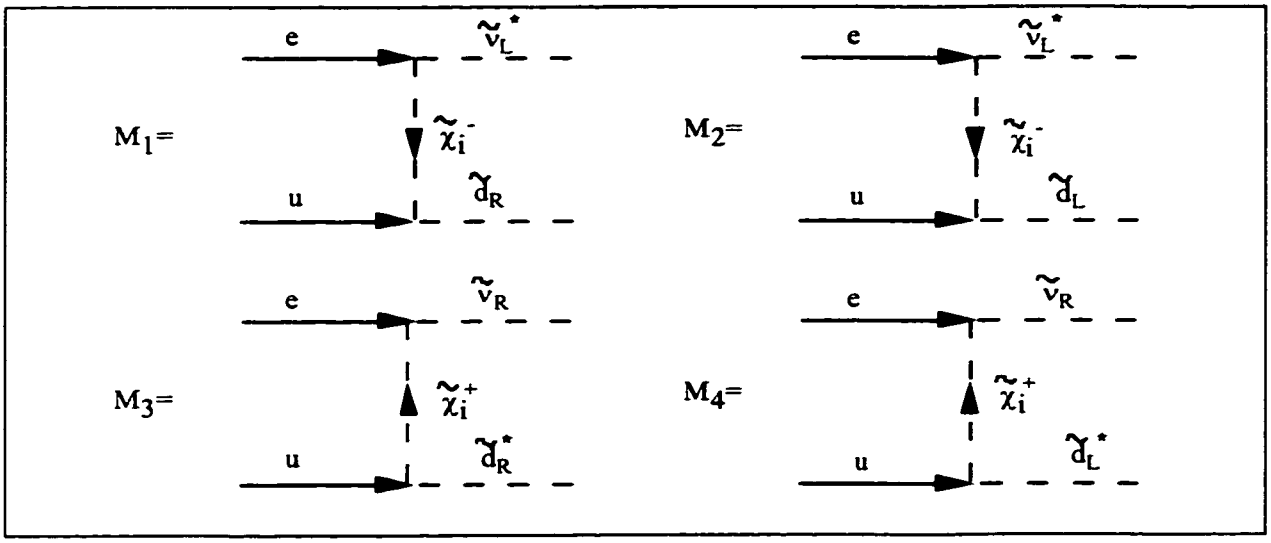

FIG. 10 Chargino contribution to ep->sleptons invariant amplitude

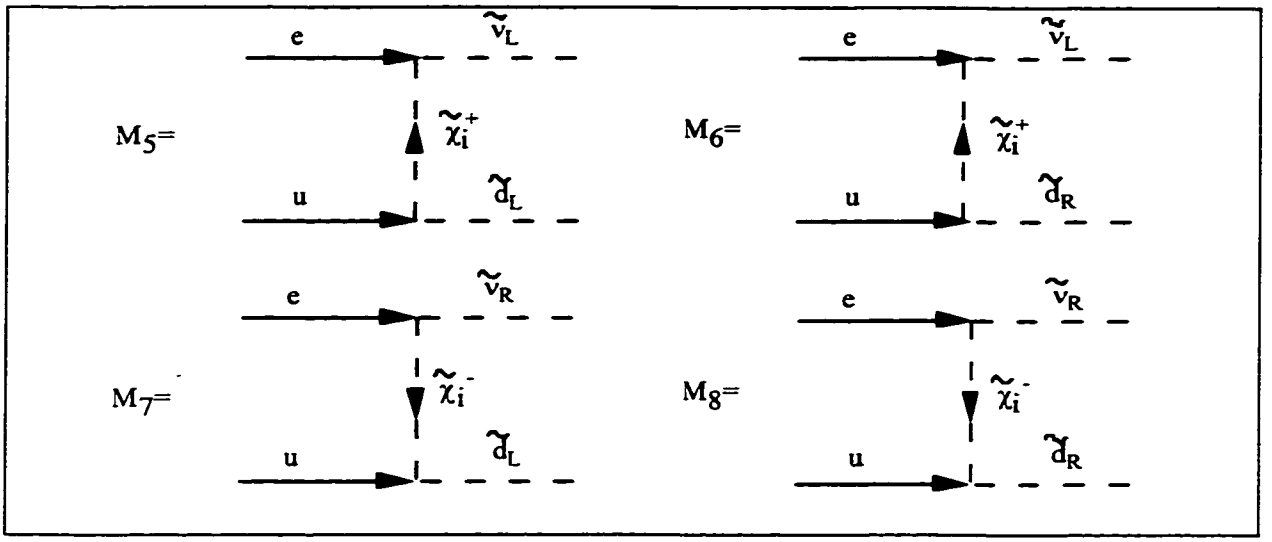

FIG. 11. Neutralino contribution to ep->sleptons invariant amplitude (cont'd)

Treating the chargino as a standard fermion propagator, the invariant amplitude for each of the contributions to the neutralino invariant amplitude can be written down in terms of the Feynman rules acting at each of the vertices of the above diagrams:

$$M_1 = e\left\{-\frac{i}{2}g' V_{i1}^{*} P_L\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{-\frac{1}{2}g_R U_{i2}^{*} P_R\right\}u \tag{2.35}$$

$$M_2 = e\left\{-\frac{i}{2}g' V_{i1}^{*} P_L\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{\frac{1}{2}h_u^Q U_{i3} P_R + \frac{1}{2}h_d^Q U_{i4} P_R\right\}u \tag{2.36}$$

$$M_3 = e\left\{-\frac{i}{2}g_R V_{12} P_R\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{-\frac{1}{2}g_L U_{i1}^{*} P_L\right\}u \tag{2.37}$$

$$M_4 = e\left\{-\frac{i}{2}g_R V_{12} P_R\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{\frac{1}{2}h_d^Q U_{i4}^{*} P_L\right\}u \tag{2.38}$$

$$M_5 = e\left\{\frac{1}{2}h_u^L V_{i3} P_R + \frac{1}{2}h_d^L V_{i4} P_R\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{-\frac{i}{2}g_L U_{i1}^* P_L\right\}u \tag{2.39}$$

$$M_6 = e\left\{\frac{1}{2}h_u^L V_{i3}^* P_R + \frac{1}{2}h_d^L V_{i4}^* P_R\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{\frac{1}{2}h_u^Q U_{i3} P_L + \frac{1}{2}h_d^Q U_{i4} P_L\right\}u \tag{2.40}$$

$$M_7 = e\left\{\frac{1}{2}h_u^L V_{i3}^* P_L + \frac{1}{2}h_d^L V_{i4}^* P_L\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{\frac{1}{2}h_u^Q U_{i3} P_R + \frac{1}{2}h_d^Q U_{i4} P_R\right\}u \tag{2.41}$$

$$M_8 = e\left\{\frac{1}{2}h_u^L V_{i3}^* P_L + \frac{1}{2}h_d^L V_{i4}^* P_L\right\}\frac{i(q+m_x)}{q^2-m_x^2}\left\{-\frac{i}{2}g_R U_{i2} P_R\right\}u \tag{2.42}$$

After squaring M and calculating the expectation via application of trace theorems, $\langle|M|^2\rangle$ is simply written here for comparison with later program output:

$$\begin{aligned}
\langle|M_1|^2\rangle &\approx 2(g_L)^2|V_{i1}|^2(g_R)^2|U_{i2}|^2\left(\frac{q^2}{q^2-m_x^2}\right)^2(P_e\cdot P_u)\\
\langle|M_3|^2\rangle &\approx 2(g_R)^2|V_{i2}|^2(g_L)^2|U_{i1}|^2\left(\frac{q^2}{q^2-m_x^2}\right)^2(P_e\cdot P_u)\\
\langle|M_2|^2\rangle &\approx \langle|M_4|^2\rangle \approx \langle|M_5|^2\rangle \approx \langle|M_6|^2\rangle \approx \langle|M_7|^2\rangle \approx \langle|M_8|^2\rangle \approx 0
\end{aligned} \tag{2.43}$$

where U_{ij} and V_{ij} are the elements of the diagonalizing matrices of eigenvectors and $M_{\chi_i^0}$ are the positive square roots of the eigenvalues defined in equation 2.14. We now proceed with writing down the expression for the cross section.

2.5.3 Supersymmetric Cross Section

The calculation for the cross section is performed in the centre of mass frame. Figure 12 is a graphical representation of the incoming and outgoing particles defining the centre of mass scattering angle.

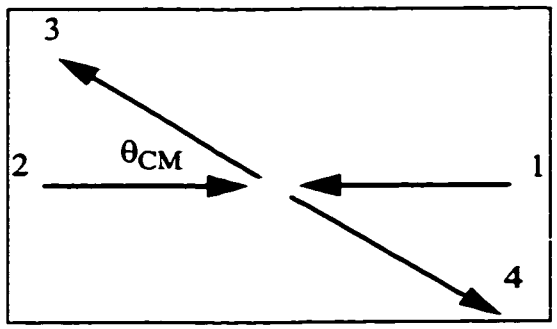

FIG. 12. The reaction 1+2->3+4 in the centre-of-mass frame

The expression for the cross section is shown by De Wit and Smith[6] to be:

$$\frac{d\sigma}{d\cos\theta_{CM}} = \frac{1}{32\pi}\frac{1}{s}\sqrt{\frac{\lambda\left(s,m_3^2,m_4^2\right)}{\lambda\left(s,m_1^2,m_2^2\right)}}|M|^2 \qquad (2.44)$$

where $\lambda\left(s,m_i^2,m_j^2\right)$ is defined in equation 1.42. Since we are really interested in calculating the cross section for ep -> sleptons and because M is an expression for eq -> sleptons, we must integrate over an empirical distribution for the quarks known as quark structure functions. The relationship between the two cross sections is expressed as follows:

$$\frac{d\sigma}{d\cos\theta_{CM}}(ep \to \tilde{e}\tilde{q}) = \int dx F(x,q) \frac{d\hat{\sigma}}{d\cos\theta_{CM}}(eq \to \tilde{e}\tilde{q}) \tag{2.45}$$

where $F(x,q)$ are the quark structure functions[15]:

$$F\left(x = \frac{\hat{s}}{s}, u\right) = 1.78x^{-0.5}\left(1 - x^{1.51}\right)^{3.5}$$
$$F\left(x = \frac{\hat{s}}{s}, d\right) = 0.67x^{-0.6}\left(1 - x^{1.51}\right)^{4.5} \tag{2.46}$$

, and

$$\frac{d\hat{\sigma}}{d\cos\theta_{CM}} = \frac{1}{32\pi}\frac{1}{sx}\sqrt{\frac{\lambda\left(sx, m_{\tilde{e}}^2, m_{\tilde{q}}^2\right)}{\lambda\left(sx, m_e^2, m_q^2\right)}}|M|^2 \tag{2.47}$$

takes into consideration the requirement that the total energy, s be integrated over the quark spatial distributions. In this expression, we have defined the incoming particles as the electron and quark and outgoing the particles as the slepton and squark. In order to perform the integration specified in 2.45, we must express the invariant amplitude in terms of the integration variable x. This is achieved by recognizing that the momentum transferred is the difference between the incoming and outgoing momenta which in the centre-of-mass frame, is written:

$$t = q^2 = 2|p||p'|\cos\theta_{CM} - 2\sqrt{\left(p'^2 + m_{\tilde{q}}^2\right)\left(p^2 + m_{\tilde{e}}^2\right)} + m_{\tilde{q}}^2 + m_{\tilde{e}}^2 \tag{2.48}$$

where the magnitudes of the momenta are expressed in terms of the product sx:

$$|p| = \frac{1}{2\sqrt{sx}}\lambda^{\frac{1}{2}}\left(xs, m_{\tilde{e}}^2, m_{\tilde{q}}^2\right)$$
$$|p'| = \frac{1}{2\sqrt{sx}}\lambda^{\frac{1}{2}}\left(xs, m_e^2, m_q^2\right) \tag{2.49}$$

Inserting the expression for the transferred momentum into the invariant amplitude for the process, results in an expression that can be numerically integrated. The form of this equation can be readily written down by contracting the many constant terms in the invariant amplitude into two constants:

$$\left\langle |M|^2 \right\rangle = C_1\left(M_i, g_i\right)\left(\frac{xs}{2}\right)\left(\frac{m_x}{t - m_x^2}\right)^2 + C_2\left(M_i, g_i\right)\left(\frac{xs}{2}\right)\frac{t}{\left(t - m_x^2\right)^2} \tag{2.50}$$

The complete expression for the differential cross section is thus:

$$\frac{d\sigma}{d\cos\theta_{CM}} = \frac{1}{32\pi}\int dx F(x,q)\frac{1}{sx}\sqrt{\frac{\lambda\left(sx, m_{\tilde{e}}^2, m_{\tilde{q}}^2\right)}{\lambda\left(sx, m_e^2, m_q^2\right)}}\left\{C_1\left(M_i, g_i\right)\left(\frac{xs}{2}\right)\left(\frac{m_x}{t - m_x^2}\right)^2\right\}$$
$$+\frac{1}{32\pi}\int dx F(x,q)\frac{1}{sx}\sqrt{\frac{\lambda\left(sx, m_{\tilde{e}}^2, m_{\tilde{q}}^2\right)}{\lambda\left(sx, m_e^2, m_q^2\right)}}\left\{C_2\left(M_i, g_i\right)\left(\frac{xs}{2}\right)\frac{t}{\left(t - m_x^2\right)^2}\right\} \tag{2.51}$$

This expression may be numerically integrated over the range of $\cos\theta_{cm}$ from -1 to 1 and the resulting differential cross section plotted. It is valid for both the neutralino and the chargino cross sections with the appropriate expectation of the square of the invariant amplitude. Since it is desirable to parametrically vary the total energy and other constants in this expression, the calculation is best performed by computer. The implementation of an algebraic calculator and numerical integrator to solve this expression is the subject of chapter 3.

CHAPTER 3: SUSY CROSS SECTION CALCULATOR

As noted in chapters 1 and 2, the calculation of supersymmetric interactions are long, tedious and thus subject to errors at many stages. For this reason alone, it is sensible to implement the calculations in a computer program. Using a symbolic programming language provides the added benefit of making the program easier to write and understand since many of the symbols employed in the calculations represent complex algebraic entities such as Dirac gamma matrices or iterated sums that are more difficult to implement in standard languages such as Fortran or C. There is a trade off between simplicity and speed when comparing symbolic program and standard program performance. The approach adopted in this thesis is compromise between generality and speed within the confines of a symbolic language. Compromises are discussed in the sections describing the detailed design as the need arises.

The SUSY calculator is a program implemented in the symbolic programming language Mathematica®. The calculator performs the algebraic operations to determine the invariant amplitudes for the neutralino and chargino exchanges in the interaction $ep \rightarrow sleptons$ based on the Feynman rules derived in chapter 2. The final output of the calculator is plots of the differential cross sections for the interaction. An overview of Mathematica® and the calculator design is detailed in the following sections.

3.1 Description of Mathematica®

Mathematica® is a powerful programming language as well as an interactive mathematical workshop. Mathematica® performs numerical, symbolic and graphical computation all of which are employed in the SUSY calculator. The calculator employs custom programs called packages that are stored in appropriately named files with a .m extension. These packages must be loaded in order to access the functions they contain. The SUSY calculator utilizes many features of the programming language; some of the design features are the result of the peculiarities of Mathematica®. Mathematica® references[16] give excellent examples of using Mathematica®, but some detail of some the languages features are given here because of their effect on the design.

3.1.1 Symbolic Capabilities

The symbolic capability of Mathematica® is used extensively in determining the expression for the invariant amplitude of an interaction. The algebraic manipulations are fairly straightforward and easy to follow with one or two exceptions - one of them being that it is necessary to deal with what Mathematica® terms the *standard form*. Mathematica® always stores and displays output in the standard form which is essentially alphabetical order. For example if the expression:

In[1]:= b a (3.1)

is entered into Mathematica® (a space represents multiplication), the output will read:

```
Out[1] = a b .                                                    (3.2)
```

This creates a problem when a and b represent entities which don't commute. There are several ways around this idiosyncratic feature. The standard approach is to employ the non-commutative operator "**" or the matrix multiplication operator ".", but with either of theses approaches, long expressions cannot be simplified until the symbols are defined. This is not surprising since Mathematica® has no way of "knowing" if a symbol represents a scalar, a vector or a matrix. For computational efficiency, it is not desirable to define matrices until as late as possible (or not at all) since repeated matrix multiplication is costly in terms of performance. Sophisticated techniques can be employed to change the Mathematica® output form[16], but these are not necessary if the program employs specific symbol labeling techniques that work in concert with the standard output to preserve the order of multiplication. This technique is further explored in the detailed design section.

Another obtuse but necessary feature of Mathematica® involves the use of logic tests. Consider the following dialogue from a Mathematica® session:

```
In[2]:= 2==2
Out[2]=True
In[3]:= 3==2
Out[3]=False
In[4]:=a==a                                                       (3.3)
Out[4]=True
In[4]:=a==b
Out[4]=a == b
```

We see that while Mathematica® can determine that the statement a==a is True, it cannot

determine that a==b is False. The Mathematica® test a===b returns the desired result since it tests the structure of the expression as opposed to mathematical equality. Since it is desirable to test for equality, the SUSY calculator uses the unusual If construction provided by Mathematica®:

If[*statement, True body, False body, Neither body*] (3.4)

When performing conditional tests against undefined symbols as is the case that is required to access the list of Feynman rules, it is necessary to execute either the *True body* or the *Neither body* since the *False body* can only be reached be if numbers are tested. This feature is employed in the package susyrule.m detailed in section 3.2.2.5.

3.1.2 Replacement Rules

Replacement rules are a powerful feature of Mathematica® that permit, among other things, identities to be implemented in an efficient manner. Replacement rules are used frequently in the calculator to avoid the overhead of lengthy matrix calculations. For example the matrix operation:

$$A = P_R P_R = \left(I + \gamma^5\right)\left(I + \gamma^5\right), \tag{3.5}$$

can be calculated by brute force:

$$A = \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right\} \times \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right\}$$

$$= \begin{bmatrix} 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix} \qquad (3.6)$$

$$= 2PR$$

Alternatively, the same equation can be much more efficiently implemented in Mathematica® using replacement rules:

$$A = P_R * P_R /. \{P_R * P_R -> 2 * P_R\} \qquad (3.7)$$

where /. is the Mathematica® replacement operator which is read as *given that*. In English, the above expression reads A equals P_R times P_R given that P_R times P_R is replaced with 2 times P_R . In this expression, no matrix multiplication is performed and indeed P_R is not even defined. Replacement rules permit symbols to be carried through a lengthy calculation and only converted into a matrix or indexed sum when necessary. This ensures that intermediate program output is readable and that unnecessary calculations are kept to a minimum. This approach does however represent a trade-off between generality and speed and complexity of the code.

3.2 Calculator Design

In order to assist in the ability for the reader to understand the file printouts of the

Mathematica® packages included in Appendix A, the high-level design description and execution flow is given. This is followed by a detailed design of each of the packages. Packages are written in the style recommended by the developers of Mathematica®[17].

3.2.1 Design Overview

The SUSY calculator consists of nine Mathematica® packages. These are called susycalc.m, susycons.m, susyamp.m, susykine.m susyexp.m, susyrule.m, susymass.m cfr.m and nfr.m. The names are based upon the unfortunate, eight character limitation of the MS-DOS operating system only for the sake of portability. Figure 13 is a pictorial representation of the packages making up the calculator.

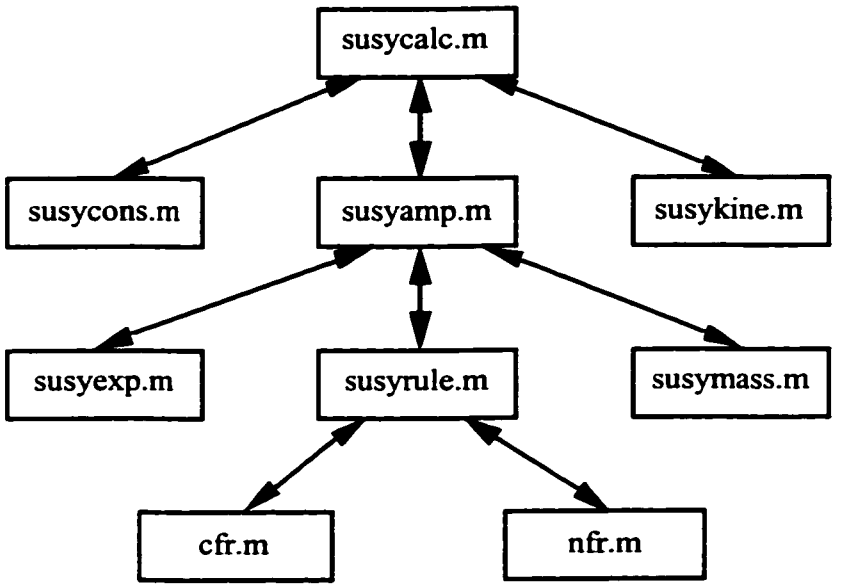

FIG. 13. Pictorial representation of the packages comprising the SUSY calculator

The arrows between the rectangles represent the calling of functions contained in the packages. The called functions are contained in packages represented by the

lower of two rectangles joined by a double-headed arrow. The functions are called from within the packages represented by the upper rectangles, and the result is returned directly to that package before further instructions are executed. The main package is susycalc.m. It loads all the other packages and calls functions from the indicated packages in the process of calculating the cross sections. The calculator output consists of a graphical object that is a plot of the differential cross section as well as algebraic expressions of the invariant amplitude and its expectation value.

3.2.2 Detailed Design

This section supplements the source code listings provided in Appendix A. Every statement in the program is not explained; only those that are germane to understanding the essence of how the program works are discussed in detail. More information on the built-in functions supplied with Mathematica® can be found in the appropriate sections in Mathematica 2nd Edition. It is recommended that a copy of this text be handy when reading this section or the program listings in Appendix A.

3.2.2.1 susycalc.m

This is the main package of the SUSY calculator. It loads the three topmost packages and then calls functions from them as the calculation progresses. Figure 14 presents the execution flow of this package. After loading the packages, a call is made to the function CrossSection[] found in the package susyamp.m. The function call passes the argument Neutralino or Chargino depending on the process to be calculated. The function returns a

list of the expectation of the square of each of the amplitudes contributing to the cross section. Details of these calculations are provided in section 3.2.2.2 describing the package susyamp.m

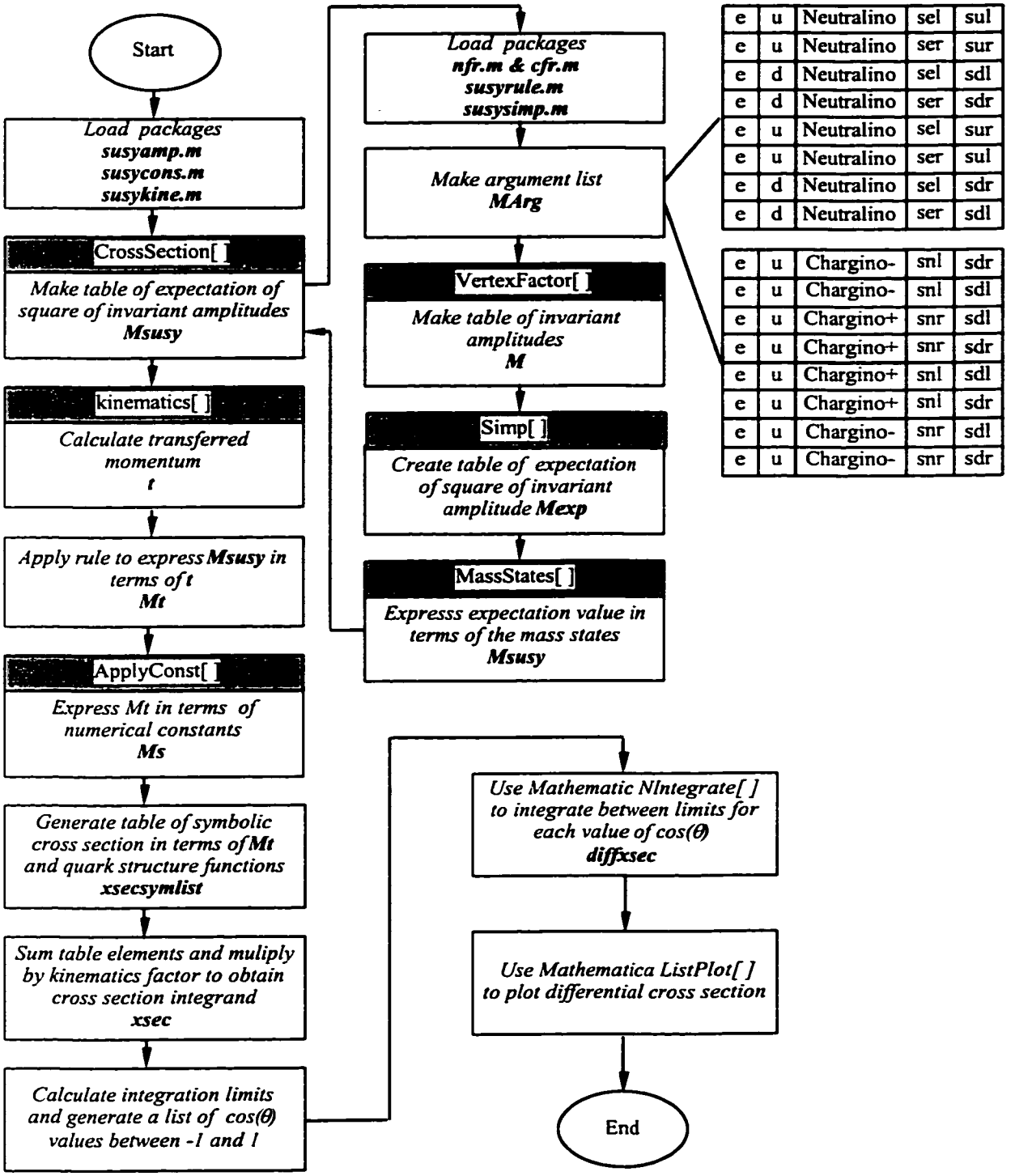

FIG. 14.SUSY calculator execution flow

The next step is a call to the function kinematics[] found in the package susykine.m to

determine the value of the transferred momentum t defined in equation 2.48. We then apply the rule which replaces the symbol q^2 in the expectation of the invariant amplitude with the expression for t. The function ApplyConst[] found in susycons.m applies the rules converting the mass symbols and other symbols into numerical values.

The final step prior to integration is to make a table corresponding to the product of each of the eight expectation values of the invariant amplitude and its respective quark structure function (equation. 2.46); sum these together and multiply by the factors defining the kinematics. This results in the integrand of the expression representing the differential cross section expressed by equation 2.51. The differential cross section is calculated using the Mathematica® supplied function that performs numerical integration, NIntegrate[]. This is implemented in the following complex expression that requires some explanation:

```
diffxsec = Table[NIntegrate[xsec/.{Cos[thetacm] → costheta[[i]]},{x,IntLimit,1}],
          {i,Length[costheta]}];
```
(3.8)

The arguments of the NIntegrate[] function are enclosed in square brackets. These arguments state that the expression for the integrand of the differential cross section, xsec, with the symbol Cos[thetacm] (t is expressed in terms of this angle - see equation 2.48) replaced by the elements of costheta[i] be integrated from IntLimit to 1. This is all embedded in the Mathematica Table[] function which will create a list consisting of the integration result for each of the elements of the list costheta[i]. This list varies Cos[thetacm] from -1.0 to 1.0 in variable increments. This expression shows the

power of the Mathematica® programming language in which one expression replaces what, in most programming languages, would require a while or a for loop with attendant loop counters and tests. The resulting list of integration results is plotted against the values of Cos[thetacm]. Example plots are presented in section 3.3. Before presenting details of the program output, we will discuss the details of each of the packages invoked in the process of calculating the cross section.

3.2.2.2 susyamp.m

Susyamp.m contains the function CrossSection[] which returns a list of the expectation of the square of the amplitude. It generates this list by first creating a list of eight arguments of the function call VertexFactor[]. These arguments are the input, exchange and output particles of either of the eight Feynman diagrams comprising the neutralino contribution to the invariant amplitude listed in figure 11 or the chargino invariant amplitude listed in figure 12. The function VertexFactor[] resides in the package susyrule.m. The function call is embedded in the Mathematica Table[] function and returns a table listing each of the eight algebraic expressions of the corresponding invariant amplitudes.

The next step determines the expectation of the square of the amplitude. This is achieved by the function call Simp[] which resides in the package susysimp.m. This is followed by a call to either NMassM[] for the Neutralino case or ChMassM[] which both reside in the package susymass.m. The argument passed in this function call is the higgsino mass parameter discussed in section 2.4.1. These functions return the neutralino or chargino

mass matrices and mass vectors corresponding to the diagonalization of the matrices represented by equations 2.9 and 2.15. The final step is to apply the mass states to the symbolic expectation Mexp by a call to the function AppMStates[] which also resides in the package susymass.m The details of this final calculation are provided in the susymass.m section below.

3.2.2.3 susycons.m

The package susycons.m is used to replace algebraic symbols with numerical values so that the cross section can be calculated. The function ApplyConst[] replaces each symbol with its numerical value in a straightforward and easy to follow manner. In some cases this value is well known as is the case of the electron mass and the weak mixing angle, but in other cases it may be desirable to vary the parameter in a parametric manner. It is difficult to foresee, at this time exactly which parameters are desirable to vary automatically and over what range. The total energy symbol s represents the square of the collider energy making the currently available collider energy an obvious candidate for variation. Frank, Kalman and Saif[18] have performed calculations for $\sqrt{s} = 314$ GeV (HERA), 410 GeV (HERA upgrade) and 1400 GeV (future LEP/LHC collider). Since the cross section varies in a non-trivial manner with this parameter, it is included as an automatically varied parameter. If in the future it is desirable to vary any other parameter in a similar manner, the collider energy variation can be used as a prototype.

3.2.2.4 susykine.m

The package susykine.m contains the kinematics of the interaction. It provides two functions KalKin[] and Lam[]. The function KalKin[] simply returns the transferred momentum in terms of momenta of the incoming and outgoing particles. It is a straightforward implementation of equation 2.48. The function Lam[] is an implementation of equation 1.42. This function is also called directly from susycalc.m

3.2.2.5 susyrule.m

The package susyrule.m performs the key algebraic manipulations required to arrive at an expression for the invariant amplitude. The package consists of two functions VertexFactor[] and ExpandVF[]. Each of these functions are further detailed below. The first step is to load the Feynman rules for neutralino (nfr.m) and chargino (cfr.m) interactions. These files contain indexed lists whose numbering corresponds to the Feynman rules presented in figures 4 and 5 and figures 6 and 7 respectively.

The function VertexFactor[] is implemented using the Mathematica® If construction discussed in section 3.1.1. Several If statements are nested. Successful termination of these nested tests results when the input arguments of the function call match the combination of particles needed to point to the correct indices in the list of Feynman rules.

Each set of five correct input arguments will result in four separate Feynman rules being

selected. Two of these will reside on the left-hand side of the fermion propagator, and two will reside on the right. Since each of these terms contains a projection operator that does not commute with the Dirac slash component of the fermion propagator, steps are taken to avoid the problems with Mathematica *Standard Form* discussed in section 3.1.1. This is achieved by suffixing the projection operators with the letter *a* for the operators on the left-hand side of the propagator and the letter *z* for the operators on the right-hand side. This ensures that after expanding and simplifying expressions through successive Mathematica® operations, the order of multiplication can still be discerned. Once the four Feynman rule components have been selected, the fermion propagator is introduced and the expression is simplified. Since this operation is common to all components of the invariant amplitude, it is carried out in a separate function ExpandVF[].

The function ExpandVF[] employs the Mathematica® function Expand[]. This powerful function carries out the multiplication of the sum of the two pairs of Feynman rules and the fermion propagator. The resulting expression is simplified by the application of the replacement rules for the projection operator identities shown in equations 2.32 and 3.7. The simplified expression is returned as the result of the function call. Note that in susyamp.m this function call is embedded in a table command causing all eight of the neutralino or chargino components to be generated. This function can also be called directly by the user when it is desirable to generate a single component of an interaction.

3.2.2.6 susyexp.m

The package susysimp.m calculates the expectation of the square of the invariant amplitude. It contains the function Simp[]. This function first simply squares and expands the amplitude. Note that the only hermitian conjugate symbol noted is for the Dirac slash matrix. The other terms can be carried as squares until it comes time to apply rules defining individual mass terms. This is carried out later in the package susycnst.m. The expression is simplified by further application of projection operator identities. Note that conjugation ensures that the Dirac slash matrix does not come in between the projection operators, so the precautions taken in ExpandVF[] are not required.

The expectation is calculated by application of trace theorems and the spinor completeness relations implemented as replacement rules. No matrix calculations are required which greatly improves performance. There is however, a penalty paid in generality. If matrix equations are used, the result is always right; where symbols are used, the rules must always be tailored for the symbols. Symbolic calculators developed for calculations in the standard model[19] [20]use the matrix equations for simplifying expressions and tend to be very slow. Resolution of this tradeoff could be the subject of future research. Further simplification is achieved by order of magnitude simplification. These simplifications are made here in keeping with the assumptions made by Kalman[21]. It must be noted that it may be desirable to parametrically vary the supersymmetric coupling constants in the future. This can be easily implemented using the construction employed in susyplot.m The resulting expression for the expectation of the square of the

invariant amplitude is returned in response to the function call.

3.2.2.7 susymass.m

This package contains four functions: NMassM[], ChMassM[], GetEV[] and AppMStates[]. These functions implement the techniques presented in section 2.4.1. The function NMassM[] generates the matrix M^n presented in equation 2.7 while ChMassM[] generates the matrix M^c presented in equation 2.12. These functions both terminate with a call to the function GetEV[] which uses the Mathematica® functions Eigenvalues[] and Eigenvectors[] to diagonalize the neutralino or chargino mass matrices. This function returns the matrices of eigenvectors and the eigenvalues. Since both function NMassM[] and ChMassM[] are invoked with the higgsino mass parameter, μ, they can be used to plot the variation of the neutralino and chargino masses with this parameter. Plots are presented in sections 3.3.1 and 3.3.6 below.

The function AppMStates[] implements the techniques discussed in section 2. Each of the Feynman rules listed in figures 4 through 7 are written in terms of the neutralino and chargino mass states. These equations require summing over the relevant indices of the matrix solutions to equations 2.9 and 2.14. This summing of indices is implemented as a replacement rule exchanging a symbol for a summed table. For example, it replaces the product of symbols $M_{i1}\frac{m_\chi^2}{\left(q^2-m_\chi^2\right)}$ with the sum of the indexed products of neutralino matrix components and masses:

$$M_{11}\frac{m_{\chi_1}^2}{\left(q^2-m_{\chi_1}^2\right)}+M_{21}\frac{m_{\chi_2}^2}{\left(q^2-m_{\chi_2}^2\right)}+M_{31}\frac{m_{\chi_3}^2}{\left(q^2-m_{\chi_3}^2\right)}+M_{41}\frac{m_{\chi_4}^2}{\left(q^2-m_{\chi_4}^2\right)} \qquad (3.9)$$

It makes a similar replacement of the chargino symbols.

3.2.2.8 cfr.m

This package contains the list of the chargino Feynman rules presented in figures 6 and 7 in Mathematica® symbolic notation. The rules are accessed by reference to the numbering presented in the figures. For example, typing ChargFeynRule[[1]] returns:

((-I/2)*HoldForm[gL*Ui1Conj]*PLz (3.10)

which is the Mathematica® representation of the first rule of figure 6. Feynman rules can be readily added to this list for easy access by the package susyrule.m for the calculation of other interactions.

3.2.2.9 nfr.m

This package contains the list of the neutralino Feynman rules presented in figures 4 and 5 in Mathematica® symbolic notation. The rules are accessed by reference to the numbering presented in the figures. For example, typing NeutFeynRule[[5]] returns:

((-7/6)*HoldForm[gp*Mi2]*PRz (3.11)

which is the Mathematica® representation of the fifth rule of figure 4.

3.3 Program Output

Key parts of the actual output from the Mathematica® sessions used to demonstrate the program are presented here.

3.3.1 Neutralino Masses

In this section of the neutralino masses versus the higgsino mass parameter is presented. Figure 15 is a plot of the neutralino mass states calculated by the package susymass.m as the mass parameter is varied. This plot was generated by the package plotmass.m. This package is not a part of the SUSY calculator but is included in Appendix A to provide the facility for investigating the mass spectrum as the mass parameters of the L-R supersymmetric model are varied.

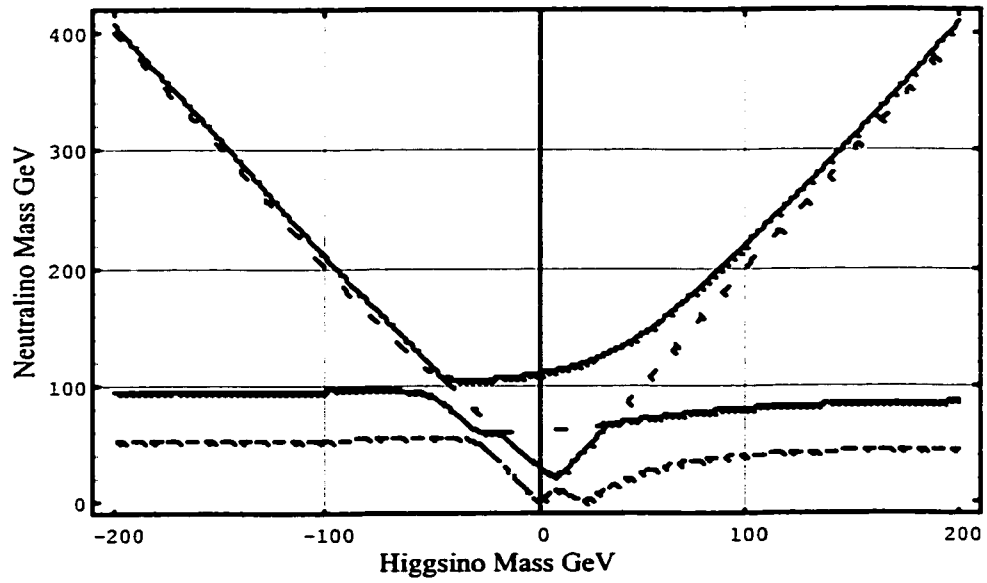

FIG. 15 Neutralino mass states

Figure 15 has been generated for the case where M_R=300 GeV, M_L=50Gev, M_Z=91Gev $\tan\theta_k$=1.6 and $g \cong g_L \cong g_R$. These values are used for all sample calculations involving neutralino interactions presented in this thesis. The four curves in figure 15 are: $\bar{\chi}_1^0$; dotted, $\bar{\chi}_2^0$; light solid, $\bar{\chi}_3^0$; dashed, $\bar{\chi}_4^0$; heavy solid.

3.3.2 Chargino Masses

In this section of the chargino masses versus the higgsino mass parameter is presented. Figure 16 is a plot of the chargino mass states calculated by the package susymass.m as the mass parameter is varied. This plot was also generated by the package plotmass.m.

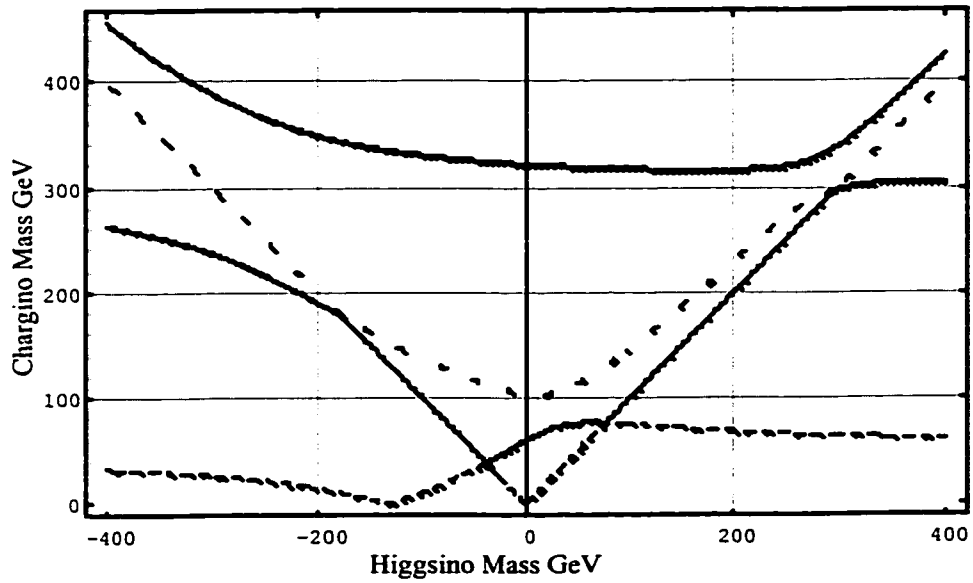

FIG. 16 Chargino mass states

Figure 16 has been generated for the case where M_R=300 GeV, M_L=50Gev, M_w=80Gev and $\tan\theta_k$=1.6. These values are used for all sample calculations involving chargino

interactions presented in this thesis. The four curves in figure 16 are: $\overline{\chi}_1^{\pm}$; dotted, $\overline{\chi}_2^{\pm}$; light solid, $\overline{\chi}_3^{\pm}$; dashed, $\overline{\chi}_4^{\pm}$; heavy solid.

3.3.3 Solution of Electron-Proton->Selectron-Squark Cross Section

In this section the results of a sample $e^- p \to \tilde{e}\tilde{q}X$ cross section calculation, based on neutralino exchange, are presented. The key elements of this calculation are: the symbolic representation of expectation of the invariant amplitude squared; the mass states; the presentation of the expectation value in terms of the neutralino mass states and finally, a plot of the differential cross section. The cross section is calculated for the same mass parameters presented in section 3.3.1 with the higgsino mass parameter set to zero and $\sqrt{s} = 314.$ GeV (HERA). Rather than listing all of the Mathematica® output here, key portions are recreated here for comparison with equations presented in chapter 2. The expectation of the square of the invariant amplitude has eight components, and the second component, because it is relatively simple, is recreated here:

```
In[2]:=
   Mexp[[2]]
Out[2]=
          4     4    2
   49 gp   Mi2   mx   Pe . Pu
   ------------------------------
                2     2 2
       2 (-mx   + q )
```

(3.12)

The above expression may be compared with the second component of equation 2.30 repeated here for ease of reference:

$$\left\langle |M_2|^2 \right\rangle \approx \frac{49}{2}(g')^4 |M_{i2}|^4 \left(\frac{m_x}{q^2 - m_x^2}\right)^2 (P_e \cdot P_u) \tag{3.13}$$

Application of the rules replacing the symbols found in this expression with the components of the diagonalizing matrix and the mass eigenstates results in the following expression:

In[16]:=

```
Mexpsusy[[2]]
```

Out[16]=

$$\frac{178475.\ gp^4\ Pe\ .\ Pu}{(-12251.6 + q^2)^2} +$$

$$\frac{1779.65\ gp^4\ Pe\ .\ Pu}{(-3720.93 + q^2)^2} +$$

$$\frac{181.7\ gp^4\ Pe\ .\ Pu}{(-932.467 + q^2)^2} +$$

$$\frac{1.33769\ 10^{-48}\ gp^4\ Pe\ .\ Pu}{(-8.81853\ 10^{-14} + q^2)^2} \tag{3.14}$$

where the mass eigenstates are:

```
In[7]:=
   Mchi
Out[7]=
   {0, 30.5363, 60.9994, 110.687}
```

(3.15)

As indicated in equations 2.50 and 2.51, this expression for the expectation may be expressed in terms of the transferred momentum and numerical values and integrated. The resulting differential cross section for the sum of all eight components of the expectation of the square of the invariant amplitude is calculated and plotted in figure 17.

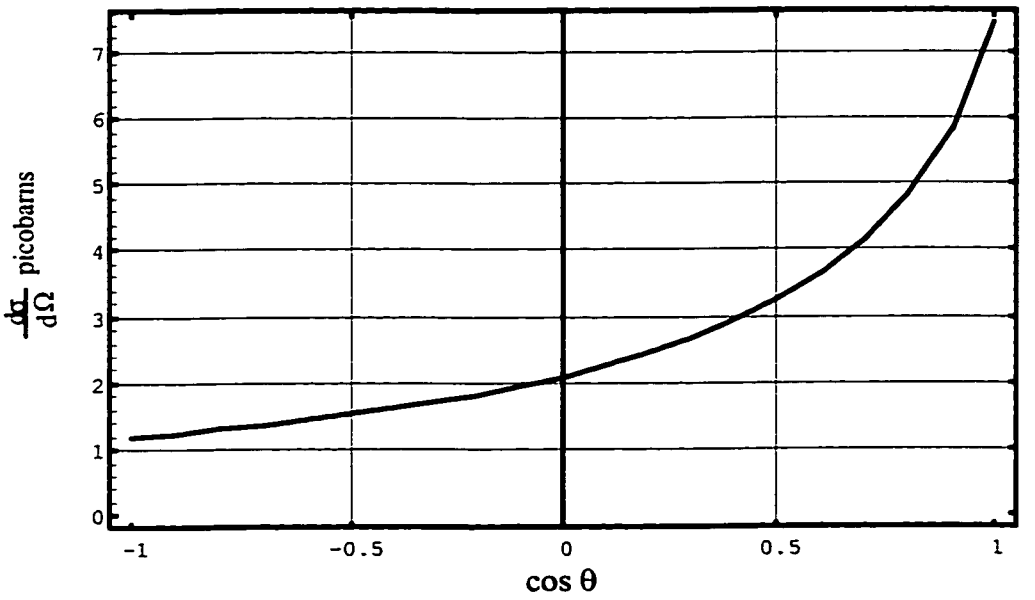

FIG. 17 Differential cross section $e^- p \to \tilde{e}\tilde{q}X$ for $\sqrt{s} = 314$ GeV

3.3.4 Solution of Electron-Proton->Sneutrino-Squark Cross Section

In this section the results of a sample $e^- p \to \tilde{\nu}\tilde{q}X$ cross section calculation, based on chargino exchange, are presented. The cross section is calculated for the same mass parameters presented in section 3.3.2 with the higgsino mass parameter set to zero and

$\sqrt{s} = 314.\ \text{GeV}$ (HERA). Rather than listing all of the Mathematica® output here, key portions are listed for comparison with equations presented in chapter 3.

In[22]:=

```
Mexp[[1]]
```

Out[22]=

$$\frac{2\ gL^2\ gR^2\ q^2\ Ui2Conj^2\ Vi1Conj^2\ Pe \cdot Pu}{(-mx^2 + q^2)^2} \tag{3.16}$$

The above expression may be compared with the second component of equation 2.43 repeated here for ease of reference:

$$\left\langle |M_1|^2 \right\rangle \approx 2(g_L)^2 |V_{i1}|^2 (g_R)^2 |U_{i2}|^2 \left(\frac{q^2}{q^2 - m_x^2} \right)^2 (P_e \cdot P_u) \tag{3.17}$$

Application of the rules replacing the symbols found in this expression with the components of the diagonalizing matrix and the mass eigenstates results in the following expression:

In[24]:=

```
Mexpsusy[[1]]
```

Out[24]=

$$\frac{0.0193026\ gL^2\ gR^2\ q^2\ Pe \cdot Pu}{(-103599. + q^2)^2} + \frac{0.167147\ gL^2\ gR^2\ q^2\ Pe \cdot Pu}{(-10915.5 + q^2)^2} +$$

$$\frac{0.00112061\ gL^2\ gR^2\ q^2\ Pe \cdot Pu}{(-3585.06 + q^2)^2} + \frac{0.\ gL^2\ gR^2\ q^2\ Pe \cdot Pu}{(0. + q^2)^2} \tag{3.18}$$

where the mass eigenstates:

In[23]:=

```
Mchi=MStates[[3]]
```

Out[23]=

```
{0., 59.8754, 104.477, 321.869}
```

(3.19)

have been used. As indicated in equations 2.50 and 2.51, the expression for the expectation may be expressed in terms of the transferred momentum and numerical values and integrated. The resulting differential cross section for the sum of all eight components of the expectation of the square of the invariant amplitude is calculated and plotted in figure 18.

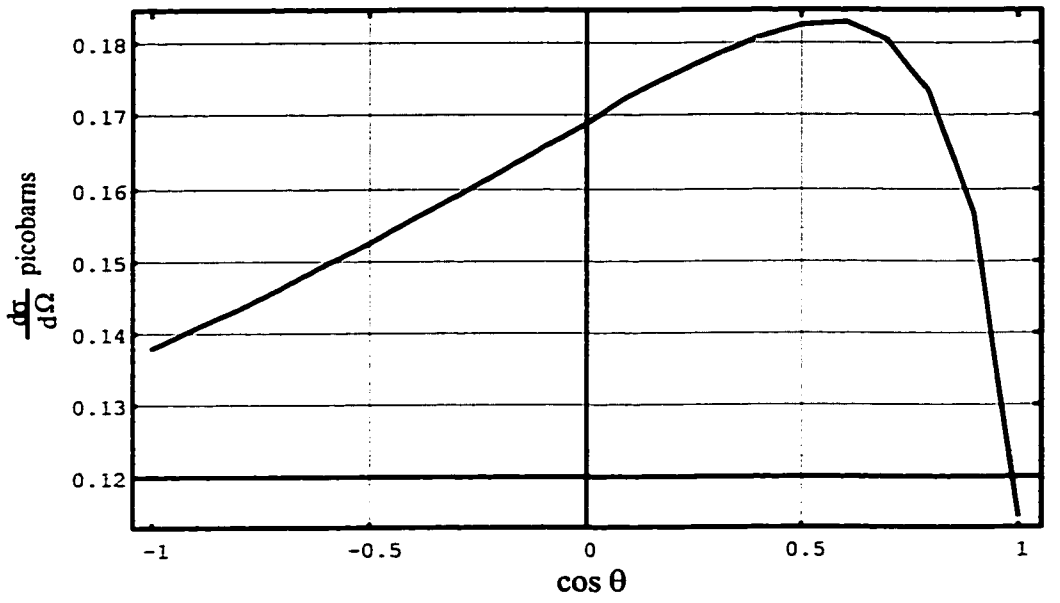

FIG. 18 Differential cross section $e^- p \to \tilde{v}\tilde{q}X$ for $\sqrt{s} = 314$ GeV

3.3.5 Parametric Variation of Total Energy

The cross section based on chargino is presented for here for $\sqrt{s} = 410$ GeV (HERA upgrade) and $\sqrt{s} = 1400$ GeV (future LEP/LHC collider).

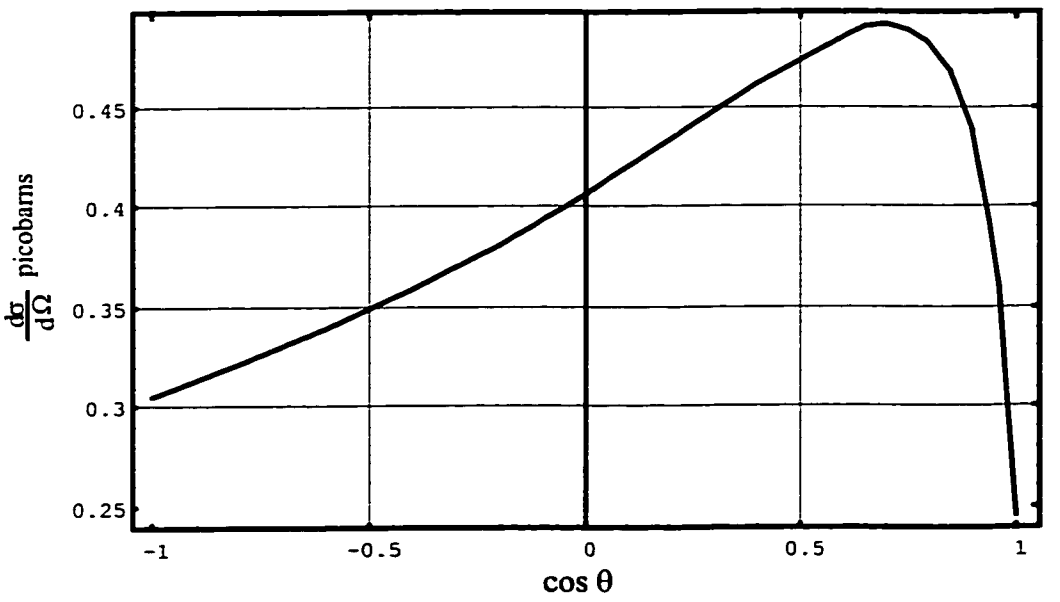

FIG. 19 Differential cross section $e^- p \to \tilde{v}\tilde{q}X$ for $\sqrt{s} = 410$ GeV

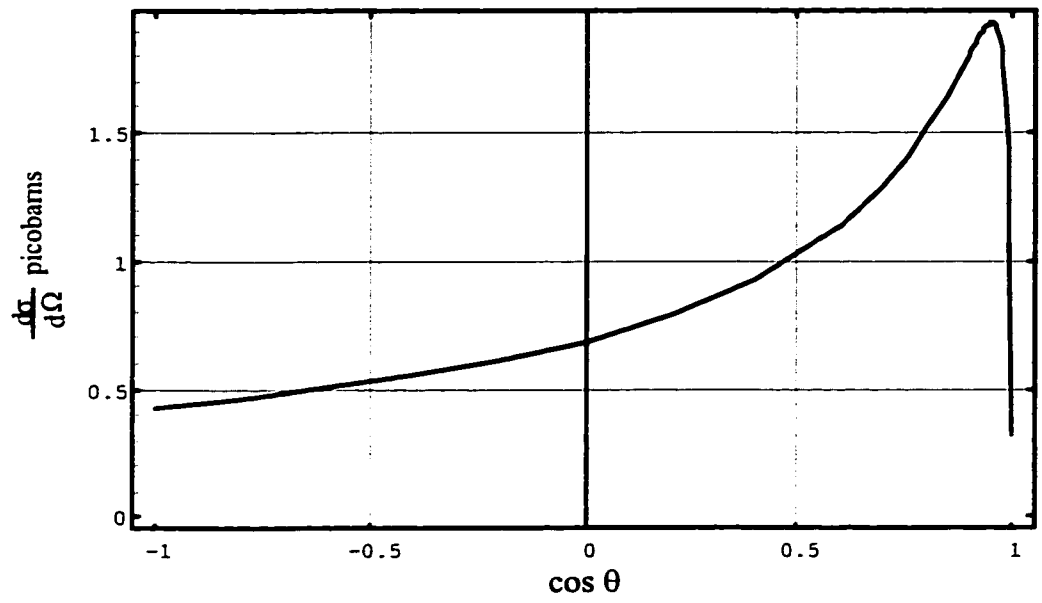

FIG. 20 Differential cross section $e^- p \to \tilde{\nu}\tilde{q}X$ for $\sqrt{s} = 1400$ GeV

3.4 Platform Matters

A brief discussion on performance is in order. All development of the SUSY calculator was done using Mathematica® version 2.1. for Windows 95 running on a Pentium 75 machine with 8 megabytes of RAM. For this configuration, it takes approximately 40 minutes to calculate the neutralino differential cross section integration for 6 values of cosθ and approximately 10 minutes to calculate the chargino differential cross section integration for 12 values of cosθ. The reason for the difference in time is that the neutralino cross section has 8*4 = 32 individual components to the expectation value while the chargino cross section only has 2*4=8 . In addition the neutralino components tend to be more complex.

The ultimate purpose of this calculator is to provide an insight into the physics of

supersymmetric interactions. Excessive amount of time to calculate the outcome of a variation of process is very frustrating; hence it is recommended that a faster machine with significantly more RAM be employed.

FUTURE CONSIDERATIONS

Developing useful, user-friendly software is a difficult task. The development of a SUSY calculator is no different. The version produced in this thesis is the first attempt at such a calculator, and many lessons have been learned that need to be included in later revisions to increase its utility. This section consists of two parts. The first identifies the weaknesses uncovered in the current design and provides a vision of the ultimate calculator; the second provides detailed guidance as to how the author would implement such improvements.

4.1 Limitations

The current calculator is an implementation of a specific problem. Although a knowledgeable Mathematica® user should have little difficulty in tailoring the calculator to another specific problem, the calculator is not general enough to be employed by someone with little Mathematica® knowledge. Other standard model calculators[22] investigated as a part of this thesis suffer from this problem as well.

Perhaps the ultimate vision of a useful calculator is one where the front-end of the calculator is similar to a drawing tool interface (e.g., the picture editor in Microsoft Word). Icons representing the various incoming, outgoing and exchanged particles may be clicked and dragged to form a Feynman diagram of the interaction to be studied. The diagram can be exported in various formats for later documentation. Once the interaction

is satisfactorily entered, the tool can be directed to automatically generate the symbolic form of the invariant amplitude and the expectation of its square. The ideal tool would also provide a kinematics editing tool and a tool for defining the numerical value and form (e.g., scalar or parametrically varied list) of all symbols employed. The tool could be commanded to produce graphical output as would be case for the investigation of scattering cross sections or to produce high precision numerical output as would be desirable in calculating the supersymmetric effects on the anomalous magnetic moment of the muon. This vision of the ultimate tool would be straightforward but time consuming to implement. It could be a collaborative project for the Computer Science and the Physics departments.

4.2 Improvements

The implementation of the vision of the ideal calculator would have Mathematica® at its core. The front-end could be implemented in a variety of ways. Perhaps the most straightforward being to develop a custom front-end using commercially available GUI builders such as Visual C++ for Microsoft® Windows based machines; TeleUSE for X-windows on UNIX based machines or Code Warrior for Macintosh machines. The program developed by this builder would use the Mathlink feature provided by Mathematica®. Using this capability, the GUI can send requests to Mathematica® to evaluate expressions. The Mathematica® core would be essentially the same as presented in this thesis.

In the current SUSY calculator, the Feynman rules are implemented as a manually entered list. The difference in the ideal calculator would be that the indices into the list of Feynman rules would be based on attributes of the drawing elements rather than by hard-coded symbols.

Another limitation of the current calculator is its lack of generality in the application of trace theorems. It is felt that it is essential that these theorems be implemented as symbolic replacement rules as employed here rather than by brute force multiplication since there is a geometrical increase in the number of operations that have to be executed when matrices are involved. In order to be able to avoid matrix multiplication, a general set of Mathematica® symbols needs to be defined. For example, it is important to keep track of the number of Dirac gamma matrices involved in a given product in order to apply trace theorems. In the existing calculator, these theorems have been applied in specific cases through an a priori knowledge of the interaction. The definition of an appropriate Mathematica® predicate function[16,17] appears to be the appropriate means for such a set symbolic entities.

Perhaps the most challenging and interesting problem would be to implement symbolic symmetry operators using predicate functions and the strict definitions of a Lie algebra.

5. Conclusion

The calculation of processes involving supersymmetric particles is very complex. The complexity arises in the algebra and from the number of free parameters in the models. The ultimate aim is to gather experimental evidence of supersymmetry. Since the most likely means of achieving this is by pinpointing the signatures of the production of supersymmetric particles through the identification of missing momentum, it is desirable to be able to estimate the predicted signatures as key model parameters are varied. In view of the large number of possible outcomes and the complexity of each calculation, it is advantageous to use an algebraic program such as the one developed in this thesis.

This calculator permits the invariant amplitudes of the two reactions having the highest cross sections, $ep \rightarrow \tilde{e}\tilde{q}X$ and $ep \rightarrow \tilde{\nu}\tilde{q}X$ to be expressed in symbolic form based on the Feynman rules derived form the left-right supersymmetric Lagrangian. The free parameters in the left-right supersymmetric extension of the standard model can be readily varied within the calculator to investigate, for example, the effect of different gaugino-higgsino mixing scenarios or the value of the higgsino mass parameters on the scattering cross sections calculated. This capability should prove useful for narrowing down the number of possible signatures of the production of supersymmetric particles in future searches at HERA and LHP/LEC, when they become operational.

APPENDIX A: PROGRAM SOURCE CODE

```
(*: Title:    susycalc.m   *)
(*: Author: M. R. Adcock    *)
(*: Revision History:
    Date Created: 02 March 1997
*)

(*: Summary:
This package is the main program for calculating
cross-sections of the ep to sleptons process.
All other packages used in the program are loaded
by this package. *)

(* Leave the package public for now 02/03/97 *)
(* We may wish to protect it later on. *)
BeginPackage["susycalc`"]
EndPackage[]

(* Load Packages *)

Print[" Loading Packages "]
t1=TimeUsed[];

<< susyamp.m;
<< susycons.m;
<< susykine.m;

Print[" Calculating Invariant Amplitude M"]
xsecvar=Neutralino;
Print[" for reaction ep-> sleptons"]
Print[" for ", xsecvar," exchange"]
Mexpsusy=CrossSection[xsecvar];
t2=TimeUsed[];
Print[ " Expectation of M^2 Done "]
Print[" Session Elapsed Time = ", t2-t1," seconds"]

(* KalKin[me,mq]is found in the package susykine.m. It returns the
value of the transferred momentum t = q2 in terms of me, mq
x and s.*)

t=KalKin[me,mq];
```

```
(* Apply the rule changing replacing q^2 with t *)

Mt=Mexpsusy/.{q^2 -> t};

(* ApplyConst[] is found in susycons.m which contains all
the constants. Constants need be changed in this file only *)

Print[ " Converting Symbols to Numerical Values "]
Ms=Table[ApplyConst[Mt[[i]]],{i,Length[MArg]}];
Print[ " Converting Symbols Done "];
t2=TimeUsed[];
Print[" Session Elapsed Time = ", t2-t1," seconds"]

Print[ " Performing 11 Numerical Integrations "]
Print[ " For range of Cos[Theta] from -1 -> 1 "]
Print[ " Plot to Follow ... (~2200 secs on Pentium 75)"]

(* These are the up and down quark structure function
found on page 14-93-24 *)

Fqu = (1.78 x^-.5) (1-x^1.51)^3.5;
Fqd =(0.67 x^-.6) (1-x^1.51)^4.5;
If[xsecvar===Neutralino,
QSF ={Fqu, Fqu, Fqd, Fqd, -Fqu, -Fqu, -Fqd, -Fqd},
QSF ={-Fqu, -Fqu, -Fqu, -Fqu, -Fqu, -Fqu, -Fqu, -Fqu}];

(* Xsection is calculated per 21-93-30 *)
(* Lam[] is found in prepint.m *)

y = me^2/(x s);
z = mq^2/(x s);
xsecsymlist = (.389/(32 Pi x s)) Sqrt[Lam[1,y,z]] *
Table[QSF[[i]] Mt[[i]],{i,Length[Mt]}];
xsecsym = Apply[Plus,xsecsymlist];
xsec = ApplyConst[xsecsym];

(* Calculate the Lower Integration Limit *)

IntLimitSym = (me + mq)^2/s;
IntLimit = ApplyConst[IntLimitSym];

(* Perform Integrations for Different Values of
  Cos[thetacm] and make plot *)
```

```
(*theta=Range[-N[Pi],N[Pi],.2];*)
costheta=Range[-1,1,.1];

diffxsec=Table[NIntegrate[xsec/.{Cos[thetacm]->
costheta[[i]]},{x,IntLimit,1}],{i,Length[costheta]}];

millitopico=10^-9
plotdata=Table[{costheta[[i]],diffxsec[[i]]/millitopico},
{i,Length[costheta]}];

Print[ " Cross Section Integration Done "];
t2=TimeUsed[];
Print[" Session Elapsed Time = ", t2-t1," seconds"]

ListPlot[plotdata,AxesLabel->{"Diff Xsection pb",
"Cos[theta]"},PlotJoined->True,Frame->True,
GridLines->Automatic]

Print[ " Type M to see invariant mplitude "]
Print[ " Type Mexp to see <|M|^2> "]
Print[ " Type t to see transferred momentum "]
Print[ " Type Mt to see M in terms of t = q^2"]
Print[ " Type Ms to see full M numerical expression"]
Print[ " Type xsecsym to see symbolic form of Xsection Integrand "]
Print[ " Type xsec to see numerical expression "]
Print[ " "]

(* The End *)
```

```
(*: Title:    susyamp.m  *)
(*: Author: M. R. Adcock    *)
(*: Revision History:
    Date Created: 02 March 1997
*)

(*: Summary:
This package is the package that calls the
functions specific to either the neutralino or the
chargino invariant amplitude calculations
*)

(* Leave the package public for now 02/03/97 *)
(* We may wish to protect it later on. *)
BeginPackage["susyamp`"]
EndPackage[]

CrossSection::usage="CrossSection[ExchangeP] calculates cross
section for either neutralino or chargino Exchange in
ep-> slepton scattering"

CrossSection;

CrossSection[ExP_]:=(
Block[{out},
    (*Load Package containing Feynman rules *)
    (*Load Package for calculating expectation *)
    (*Load Package containing mass states *)
    << susyrule.m;
    << susyexp.m;
    << susymass.m;
    If[ExP==Neutralino,
        (* Produce an Argument List for VertexFactor[] *)
        MArg={{e,u,Neutralino,sel,sul},{e,u,Neutralino,ser,sur},
        {e,d,Neutralino,sel,sdl},{e,d,Neutralino,ser,sdr},
        {e,u,Neutralino,sel,sur},{e,u,Neutralino,ser,sul},
        {e,d,Neutralino,sel,sdr},{e,d,Neutralino,ser,sdl}};
        (* VertexFactor[] is found in susyrule.m *)
        (* Generate Symbolic Invariant Amplitudes *)
        (* M1 ... M8 For Neutralino Scattering *)
        M=Table[VertexFactor[MArg[[i,1]],MArg[[i,2]],MArg[[i,3]],
        MArg[[i,4]],MArg[[i,5]]],{i,Length[MArg]}];
        Print[ " Calculating Expectation of M^2 "];
        Mexp=Table[Simp[M[[i]],MArg[[i,1]],MArg[[i,2]]],
```

```
        {i,Length[MArg]}];
        (*Apply Nuetralino Mass States
        and return Mexpsusy*)
        MStates=NMassM[0];
        Mexpsusy=Table[AppMStates[Mexp[[i]],MStates[[1]],
        MStates[[2]],MStates[[3]]],{i,Length[MArg]}],
        (*Else1 of 1st If*)
        Print["Mathematica Doesn't know this"],
        (*Else2 of 1st If*)
        If[ExP==Chargino,
            (* Produce an Argument List for VertexFactor[] *)
            MArg={{e,u,CharginoMinus,snl,sdr},{e,u,CharginoMinus,snl,sdl},
            {e,u,CharginoPlus,snr,sdl},{e,u,CharginoPlus,snr,sdr},
            {e,u,CharginoPlus,snl,sdl},{e,u,CharginoPlus,snl,sdr},
            {e,u,CharginoMinus,snr,sdl},{e,u,CharginoMinus,snr,sdr}};
            (* VertexFactor[] is found in susyrule.m *)
            (* Generate Symbolic Invariant Amplitudes *)
            (* M1 ... M8 For Chargino Scattering *)
            M=Table[VertexFactor[MArg[[i,1]],MArg[[i,2]],MArg[[i,3]],
            MArg[[i,4]],MArg[[i,5]]],{i,Length[MArg]}];
            Print[ " Calculating Expectation of M^2 "];
            (*minus until sort out conjugate*)
            Mexp=-Table[Simp[M[[i]],MArg[[i,1]],MArg[[i,2]]],
            {i,Length[MArg]}];
            (*Apply Chargino Mass States
            and return Mexpsusy*)
            MStates=ChMassM[0];
            Mexpsusy=Table[AppMStates[Mexp[[i]],MStates[[1]],
            MStates[[2]],MStates[[3]]],{i,Length[MArg]}],
            (*Else1 of 2nd If*)
            Print["Mathematica Doesn't know this"],
            (*Else2 of 2nd If*)
            Print["Error: Input particle Missplelt"]
        ](*End 2nd If *)
    ](*End 1st If*)
](*End Block*)
)(*End CrossSection[]*)

(*End*)
```

```
(*: Title:    susycons.m  *)
(*: Author: M. R. Adcock     *)
(*: Revision History:
    Date Created: 02 March 1997
 *)

(*: Summary:
This package contains the constants necessary to obtain
numerical results for the ep -> sleptons cross section.*)

(*BeginPackage["susycon`"]*)

ApplyConst::usage="ApplyConst[myexpr] applies rules to
exhange symbols to numerical values"

ApplyConst;

ApplyConst[input_]:=(
    Block[{out,alpha,elec,thetaw},
            out=input;
            (* These constant definitions on
            page 26-93-35 *)
            out = out/.{me -> 45, mq -> 130,
            Pe.Pu -> x s/2};
            out = out/.{s -> 314*314};
            (* Data from weak interaction page 297 H&M *)
            alpha=1/137;
            elec = Sqrt[4 Pi alpha];
            thetaw = ArcSin[Sqrt[.25]];
            out = out/. {gL -> elec/Sin[thetaw],gR ->
            elec/Sin[thetaw],gp -> elec/Cos[thetaw]};
            out = out/. {Pi -> N[Pi,5]}
        ](*End Block*)
)(*End ApplyConst[]*)
(*End[]*)
```

```
(*: Title:    susykine.m   *)
(*: Author: M. R. Adcock     *)
(*: Revision History:
    Date Created: 02 March 1997
*)

(*: Summary:
This package contains the kinematics for quasi-elastic
scattering. The development of the equations is described
in Dewitt and Smith, Field Theory in Particle Physics
pages 98-100*)

BeginPackage["kinnemat`"]
EndPackage[]

KalKin::usage="KalKin[SusyKin] returns the expression
for t = q^2 in terms of the integration variable s x."

Lam::usage="Lam[s, m3^2, m4^2] is a symbolic implementation
of the Lamda functions defined in eq 3.47 on Page 99 of
DeWitt and Smith, Field Theory in Particle Physics."

Kalkin;
Lam;

KalKin[m1_,m2_]:=(
    Block[{t,magp,magpp},
        sx = s x;
        magp=(1/(2 Sqrt[sx]))*Sqrt[Lam[sx,m1^2,m2^2]];
        magpp=Sqrt[sx]/2;
        t=2 magp magpp Cos[thetacm] - 2 magpp Sqrt[
        magp^2 + m1^2] + m1^2
        ](*End Block*)
)(*End KalKin[]*)

Lam[s_,m1sq_, m2sq_]:=(
    Block[{l},
        l=(s-(Sqrt[m1sq]+Sqrt[m2sq])^2)*
        (s-(Sqrt[m1sq]-Sqrt[m2sq])^2);
        l=l/.{Sqrt[x_ x_] -> x}
        ](*End Block*)
)(*End Lam[]*)

(*End[]*)
```

```
(*:  Title:      susyrule.m      *)
(*:  Author:    M. R. Adcock  *)
(*:  Revision History:
     Date Created: 02 March 1997
*)

(*:  Summary:
This package contains the the Feynman rules for the ep ->
sleptons interaction. The rules are called from two files
cfr.m and nfr.m. The package contains the function GetVertexFactor
which returns the portion of the invariant amplitude for
a specific Feynman diagram. This function is called by the
the package susycalc.m which calculates all the components of
an ep -> sletons interaction. This function can also be called
directly called by the user for the calculation of a specific Feynman
diagram.*)

BeginPackage["susyrule`"]
EndPackage[]

VertexFactor::usage="VertexFactor[inparticle1,inparticle2,
exhangedparticle,outparticle1,outparticle2] returns the
vertexfactor derived from the Feynman rules of the
supersymmetric Lagrangian"

ExpandVF::usage="ExpandVF[f1,f2,f3,f4] is used by VertexFactor[].
It performs the portion of the calculation common to all
interactions."

VertexFactor;
ExpandVF;

VertexFactor[ip1_,ip2_,xp_,op1_,op2_]:=(

Block[{f1,f2,f3,f4},
    (*Load Feynman Rules*)
    << nfr.m;
    << cfr.m;
    (*1st If*)
    If[xp==Neutralino,
        (*2nd If *)
        If[ip1==e && ip2==u,
            (*3rd If*)
            (*sel = Susy Electron Left*)
            (*sul = Susy Up quark Left *)
            If[op1==sel && op2==sul,
                (*These are the Components of M1*)
                (f1=NeutFeynRule[[18]];
                f2= NeutFeynRule[[23]];
                f3= NeutFeynRule[[2]];
                f4= NeutFeynRule[[9]];
                M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
            (*Else1 of 3rd If*)
```

```
(*Mathematica can not reach this because
it is unable to determine if a==b is True
or false unless a and b are numbers.
See page 93 and 288 of Mathematica 2nd
Edition*)
Print["Mathematica Doesn't know this"],
(*Else2 of 3rd If*)
(*4th If*)
(* ser = Susy Electron Right
 sur = Susy Up quark Right *)
If[op1==ser && op2==sur,
      (*These are the Components of M2*)
      (f1=NeutFeynRule[[20]];
      f2= NeutFeynRule[[24]];
      f3= NeutFeynRule[[5]];
      f4= NeutFeynRule[[12]];
      M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
(*Else1 of 4th If*)
      Print["Mathematica Doesn't know this"],
      (*Else2 of 4th If*)
      (*5th If*)
      If[op1==sel && op2==sur,
            (*These are the Components of M5*)
            (f1=NeutFeynRule[[18]];
            f2= NeutFeynRule[[23]];
            f3= NeutFeynRule[[5]];
            f4= NeutFeynRule[[12]];;
            M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
      (*Else1 of 5th If*)
            Print["Mathematica Doesn't know this"],
            (*Else2 of 5th If*)
            (*6th If*)
            If[op1==ser && op2==sul,
                  (*These are the Components of M6*)
                  (f1=NeutFeynRule[[20]];
                  f2= NeutFeynRule[[24]];
                  f3= NeutFeynRule[[2]];
                  f4= NeutFeynRule[[9]];
                  M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
            (*Else1 of 6th If*)
                  Print["Mathematica Doesn't know this"],
                  (*Else2 of 6th If*)
                  Print["Output Particles Dont Match e u"]
            ](*End 6th If*)
      ](*End 5th If*)
   ](*End 4th If*)
],(*End 3rd If*)
(*Else1 of 2nd If*)
Print["Mathematica Doesn't know this"],
(*Else2 of 2nd If*)
(*7th If*)
If[ip1==e && ip2==d,
      (*8th If*)
```

```
If[op1==sel && op2==sdl,
    (*These are the Components of M3*)
    (f1=NeutFeynRule[[18]];
    f2= NeutFeynRule[[23]];
    f3= NeutFeynRule[[4]];
    f4= NeutFeynRule[[13]];
    M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
    (*Else1 of 8th If *)
    Print["Mathematica Can't Reach This"],
    (*Else2 of 8th If*)
    (*9th If*)
    If[op1==ser && op2==sdr,
        (*These are the Components of M4*)
        (f1=NeutFeynRule[[20]];
        f2= NeutFeynRule[[24]];
        f3= NeutFeynRule[[7]];
        f4= NeutFeynRule[[16]];
        M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
        (*Else1 of 9th If*)
        Print["Mathematica Can't Reach This"],
        (*Else2 of 9th If*)
        (*10th If*)
        If[op1==sel && op2==sdr,
            (*These are the Components of M7*)
            (f1=NeutFeynRule[[18]];
            f2= NeutFeynRule[[23]];
            f3= NeutFeynRule[[7]];
            f4= NeutFeynRule[[16]];
            M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
            (*Else1 of 10th If*)
            Print["Mathematica Can't Reach This"],
            (*Else2 of 10th If*)
            (*11th If*)
            If[op1==ser && op2==sdl,
                (*These are the Components of M8*)
                (f1=NeutFeynRule[[20]];
                f2= NeutFeynRule[[24]];
                f3= NeutFeynRule[[4]];
                f4= NeutFeynRule[[13]];
                M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                (*Else1 of 11th If*)
                Print["Mathematica Can't Reach This"],
                (*Else2 of 11th If*)
                Print["Error - Output Particles Don't Match e d"]
            ](*End 11th If*)
        ](*End 10th If*)
    ](*End 9th If*)
],
(*Else1 of 7th If*)
Print["Mathematica Can't Reach This"],
(*Else2 of 7th If*)
Print["Error - Input Particles Incorrect"]
](*End 7th If*)
```

```
        ],(*End 2nd If*)
        (*Else1 of 1st If*)
        Print["Mathematica Doesn't know this"],
        (*Else2 of 1st If *)
(**********Chargino Claculations*****************************)
        (*Cminus If*)
        If[ip1==e && ip2==u && xp==CharginoMinus,
            (*snl = susy neutrino left*)
            (*sdr = susy down quark right *)
            (* 1st cm If *)
            If[op1==snl && op2==sdr,
                (*These are the Components of M1 *)
                (f1=ChargFeynRule[[25]];
                f2=0;
                f3= ChargFeynRule[[6]];
                f4=0;
                M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                (*Else1 of 1st cm If*)
                Print["Mathematica Doesn't know1 this"],
                (*Else2 of 1st cm If *)
                (*snl = susy neutrino left*)
                (*sdl = susy down quark left *)
                (* 2nd cm If *)
                If[op1==snl && op2==sdl,
                    (*These are the Components of M2*)
                    (f1=ChargFeynRule[[25]];
                    f2=0;
                    f3= ChargFeynRule[[10]];
                    f4= ChargFeynRule[[14]];
                    M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                    (*Else1 of 2nd cm If*)
                    Print["Mathematica Doesn't know1 this"],
                    (*Else2 of 2nd cm If *)
                    (*snr = susy neutrino right*)
                    (*sdl = susy down quark left *)
                    (* 3rd cm If *)
                    If[op1==snr && op2==sdl,
                        (*These are the Components of M7*)
                        (f1=ChargFeynRule[[30]];
                        f2=ChargFeynRule[[29]];
                        f3=ChargFeynRule[[10]];
                        f4=ChargFeynRule[[14]];
                        M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                        (*Else1 of 3rd cm If*)
                        Print["Mathematica Doesn't know1 this"],
                        (*Else2 of 3rd cm If *)
                        (*snr = susy neutrino right*)
                        (*sdr = susy down quark right*)
                        (* 4th cm If *)
                        If[op1==snr && op2==sdr,
                            (*These are the Components of M8*)
                            (f1=ChargFeynRule[[30]];
                                        f2=ChargFeynRule[[29]];
```

```
                    f3=0;
                    f4=ChargFeynRule[[6]];
                    M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                    (*Else1 of 4th cm If*)
                    Print["Mathematica Doesn't know1 this"],
                    (*Else2 of 4th cm If *)
                    Print["Error Out Doesn't Match e u C-"]
                ](*End 4th cm If*)
            ](*End 3rd cm If*)
        ](*End 2nd cm If*)
],(*End 1st cm If*)
(*Else1 of Cminus If*)
Print["Mathematica Doesn't know this"],
(*Else2 of Cminus If *)
(*Cplus If*)
If[ip1==e && ip2==u && xp==CharginoPlus,
    (*snr = susy neutrino right*)
    (*sdl = susy down quark left *)
    (* 1st cp If *)
    If[op1==snr && op2==sdl,
        (*These are the Components of M3*)
        (f1=ChargFeynRule[[26]];
        f2=0;
        f3= ChargFeynRule[[1]];
        f4=0;
        M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
        (*Else1 of 1st cp If*)
        Print["Mathematica Doesn't know this"],
        (*Else2 of 1st cp If *)
        (*snr = susy neutrino right*)
        (*sdr = susy down quark right *)
        (* 2nd cp If *)
        If[op1==snr && op2==sdr,
            (*These are the Components of M4*)
            (f1=ChargFeynRule[[26]];
            f2=0;
            f3=ChargFeynRule[[17]];
            f4=ChargFeynRule[[21]];
            M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
            (*Else1 of 2nd cp If*)
            Print["Mathematica Doesn't know this"],
            (*Else2 of 2nd cp If *)
            (*snl = susy neutrino left*)
            (*sdl = susy down quark left *)
            (* 3rd cp If *)
            If[op1==snl && op2==sdl,
                (*These are the Components of M5*)
                (f1=ChargFeynRule[[27]];
                f2=ChargFeynRule[[28]];
                f3=ChargFeynRule[[1]];
                f4=0;
                M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                (*Else1    of    3rd cp If*)
```

```
                                    Print["Mathematica Doesn't know this"],
                                    (*Else2 of 3rd cp If *)
                                    (*snl = susy neutrino left*)
                                    (*sdrl = susy down quark right *)
                                    (* 4th cp If *)
                                    If[op1==snl && op2==sdr,
                                        (*These are the Components of M6*)
                                        (f1=ChargFeynRule[[27]];
                                        f2=ChargFeynRule[[28]];
                                        f3=ChargFeynRule[[17]];
                                        f4=ChargFeynRule[[21]];
                                        M=ip1*ExpandVF[f1,f2,f3,f4]*ip2),
                                        (*Else1 of 4th cp If*)
                                        Print["Mathematica Doesn't know this"],
                                        (*Else2 of 4th cp If *)
                                        Print["Error Out Doesn't Match e u C+"]
                                    ](*End 4th cp If *)
                                ](*End 3rd cp If *)
                            ](*End 2nd cp If *)
                        ],(*End 1st cp If *)
                        (*Else1 of Cplus If*)
                        Print["Mathematica Doesn't know this"],
                        (*Else2 of Cplus If *)
                        Print["Error - No particles match rules"]
                    ](*End Cplus If *)
                ](*End Cminus If*)
            ](*End 1st If*)
        ](*End Block*)
        )(*End of VertexFactor*)

        ExpandVF[f1_,f2_,f3_,f4_] := (

            Block[{epropden,eprop,M},

                (* Calculations Common to all *)

                epropden=(q^2 - mx^2);
                eprop=I*(qslash+mxI)/epropden;
                M=(f1+f2)*eprop*(f3+f4);
                M=Expand[M];

                (* Apply Parity Operator Identities *)

                M=M /. {PRa PRz mxI -> 2 PR mxI, PLa PLz mxI -> 2 PL mxI,
                PRa PLz mxI -> 0, PLa PRz mxI -> 0};
                M=M /. {PRa PRz qslash -> 0, PLa PLz qslash -> 0,
                PRa PLz qslash -> 2 qslash PL, PLa PRz qslash ->
                2 qslash PR}
                ] (* End Block *)
        ) (* End of ExpandVF *)

        (*End[];*)
```

```
(*: Title:    susyexp.m    *)
(*: Author: M. R. Adcock     *)
(*: Revision History:
    Date Created: 02 March 1997
 *)

(*: Summary:
This package contains the replacement rules
for calculating the expectation of the square of
the invariant amplitude.*)

BeginPackage["susyexp`"]
EndPackage[]

Simp::usage="Simp[myexpr] simplifies the expression
by applying parity operator identities, trace theorems
and neglecting higher order terms."

Simp;

Simp[input_,ip1_,ip2_]:=(
    Block[{out,out2},
            out=input/.{qslash->qdagger};
            out2=Expand[input*out];

            (* Parity Operator Rules *)

            out2=out2/. {PR*PR -> 2 PR, PL*PL -> 2 PL,
             PR*PL -> 0, PL*PR -> 0};

            (* Trace of gamma5's vanish *)

            out2=out2/. {PR -> 1+gamma5, PL -> 1-gamma5};
            out2=out2/. {gamma5 -> 0};

            (*Trace Theorem Simplification*)

            out2=out2/.{mxI^2 -> 4 mx^2};
            out2=out2/.{mxI qdagger -> 0};
            out2=out2/.{mxI qslash -> 0};
            out2=out2/.{qdagger qslash -> 4 q^2};

            (* Kinematics *)
            out2=out2/. {ip1^2 ip2^2 -> Pe.Pu + Me Mu};
```

```
            (* Order of Magnitude Simplificatiion*)

            out2=out2/. {Me Mu -> 0};
            out2=ReleaseHold[out2];
            out2=out2/. {hdL^2 -> 0, huL^2->0, huQ^2 -> 0,hdQ^2->0};
            out2=out2/. {hdQ huQ -> 0, hdL huL -> 0}
      ](*End Block*)
)
(*End[];*)
```

```
(*: Title:susymass.m  *)
(*:  Author:  M. R. Adcock     *)
(*: Revision History:
     Date Created:  02 March 1997
*)

(*:  Summary:
This package contains functions for determining the
nuetralino and chargino mass states and for replacing
symbols with products of mass states and eigenvectors *)

(* Leave the package public for now 02/03/97 *)
(* We may wish to protect it later on. *)
BeginPackage["susymass`"]
EndPackage[]

NMassM::usage="NMassM[mu] calculates the neutralino
mass states and the diagonalizing matrix of eigenvectors
for the Lagrangian Mass Matrix Mn. See Thesis section
3.2.2.7"

ChMassM::usage="ChMassM[mu] calculates the neutralino
mass states and the diagonalizing matrix of eigenvectors
for the Lagrangian Mass Matrix Mc. See Thesis section
3.2.2.7"

GetEV::usage="GetEV[M] uses the Mathematica functions
Eigenvalues and Eigenvectors to calulate the eigenvalues
of the product of the matrix M and its transpose. It
returns the square root of the eigenvalues thus calculated
as well as the matrices of eigenvectors. See Thesis
section 3.2.2.7"

AppMStates::usage="AppMStates[amplitude,eigenvectormatrix1,
eigenvectormatrix2,masseigenstatesvector] replaces the mass
element symbols in the amplitude with the dot product of the
appropriate row of the eigenvectormatrix and the mass eigenstate
vector. See thesis section 3.2.2.7"

NMassM;
ChMassM;
GetEV;
AppMStates;
```

```
NMassM[input_]:=(
    Block[{
        (*Local Variables*)
        out,mu,MZ,ML,MR,MW,thetak,ck,sk,tw,t,
        M11,M12,M13,M14,M21,M22,M23,M24,M31,M32,
        M33,M34,M41,M42,M43,M44,Mc,alpha,elec,
        g,gp,gR,gL},
        (*Body*)
        mu=input;
        (* Zeros result in very small (10^-14) eigenvalues
        as a result of numerical precision *)
        If[mu==0,mu=.0000001];
        MZ=91.15;
        ML=50;
        MR=300;
        MW=80;
        thetak=ArcTan[1.6];
        ck=Cos[thetak];
        sk=Sin[thetak];
        (* Data from weak interaction page 297 H&M *)
        thetaw=ArcSin[Sqrt[.25]];
        tw=Tan[thetaw];
        t=Sqrt[1 + 4 tw^2];
        (*Coupling Constants*)
        alpha=1/137;
        elec = Sqrt[4 Pi alpha];
        g=elec/Sin[thetaw];
        gp=elec/Cos[thetaw];
        gR=g;
        gL=g;
        ku=(Sqrt[2] MZ sk)/t;
        kd=(Sqrt[2] MZ ck)/t;
        (*These are the elements of mass matrix Mn*)
        M11=ML;
        M12=0;
        M13=-(1/Sqrt[2]) gL ku;
        M14=(1/Sqrt[2]) gL kd;
        M21=0;
        M22=MZ;
        M23=Sqrt[2] tw gR ku;
        M24=-Sqrt[2] tw gR kd;
        M31=-(1/Sqrt[2]) gL ku;
        M32=Sqrt[2] tw gR ku;
        M33=0;
```

```
        M34=-2 mu;
        M41=(1/Sqrt[2]) gL kd;
        M42=-Sqrt[2] tw gR kd;
        M43=-2 mu;
        M44=0;
        Mc=N[{{M11,M12,M13,M14},
        {M21,M22,M23,M24},
        {M31,M32,M33,M34},
        {M41,M42,M43,M44}}];
        out=GetEV[Mc]
    ](*End Block*)
)(*End NMassM[]*)

ChMassM[input_]:=(
    Block[{
        (*Local Variables*)
        out,mu,MZ,ML,MR,MW,thetak,ck,sk,tw,t,
        M11,M12,M13,M14,M21,M22,M23,M24,M31,M32,
        M33,M34,M41,M42,M43,M44,Mc},
        (*Body*)
        mu=input;
        MZ=91.15;
        ML=50;
        MR=300;
        MW=80;
        thetak=ArcTan[1.6];
        ck=Cos[thetak];
        sk=Sin[thetak];
        (* Data from weak interaction page 297 H&M *)
        thetaw=ArcSin[Sqrt[.25]];
        tw=Tan[thetaw];
        t=Sqrt[1 + 4 tw^2];
        M11=ML;
        M12=0;
        M13=0;
        M14=Sqrt[2] MW ck;
        M21=0;
        M22=MR;
        M23=0;
        M24=Sqrt[2] MW ck;
        M31=Sqrt[2] MW sk;
        M32=Sqrt[2] MW sk;
        M33=0;
        M34=-mu;
```

```
        M41=0;
        M42=0;
        M43=-mu;
        M44=0;
        Mc=N[{{M11,M12,M13,M14},
        {M21,M22,M23,M24},
        {M31,M32,M33,M34},
        {M41,M42,M43,M44}}];
        out=GetEV[Mc]

    ](*End Block*)
)
GetEV[M_]:=(
    Block[{out,Mc,Mct,Mchi,U,V},
        Mc=M;
        Mct=Transpose[Mc];
        Mchi=Sqrt[Eigenvalues[Mct.Mc]];
        V=Eigenvectors[Mct.Mc];
        U=Eigenvectors[Mc.Mct];
        (*Mathematica Returns the eigenvalues in descending
        order; Susy analysis presents them in order of
        increasing mass. Therefore, must reverse order presented*)
        Mchi=Reverse[Mchi];
        V=Reverse[V];
        U=Reverse[U];
        out={U,V,Mchi}
    ](*End Block*)
)

AppMStates[input_,U_,V_,Mchi_]:=(
    Block[{out},
        out=input/. {Vi1Conj->Vi1,Vi2Conj->Vi2,Vi3Conj->Vi3,
        Vi4Conj->Vi4,Ui1Conj->Ui1,Ui2Conj->Ui2,Ui3Conj->Ui3,
        Ui4Conj->Vi4};
        out=Table[out/.{Vi1->V[[i,1]],Vi2->V[[i,2]],
        Vi3->V[[i,3]],Vi4->V[[i,4]],Ui1->U[[i,1]],Ui2->U[[i,2]],
        Ui3->U[[i,3]],Ui4->U[[i,4]],Mi1->U[[i,1]],Mi2->U[[i,2]],
        Mi3->U[[i,3]],Mi4->U[[i,4]]},{i,4}];
        out=Table[out[[i]]/.{mx->Mchi[[i]]},{i,4}];
        out=Apply[Plus,out]
    ](*End Block*)
)
(*End[];*)
```

```
(*: Title:cfr.m     *)
(*:  Author: M. R. Adcock     *)
(*: Revision History:
     Date Created: 02 March 1997
 *)

(*:  Summary:
This package contains the list of Feynman rules
for the chargino/quark/squark and chargino/lepton/splepton
Feynman diagrams defined in section 2.4.3 of thesis.
Note that only those used in the calculations have been
defined.*)

ChargFeynRule={((-I/2)*HoldForm[gL*Ui1Conj]*PLz),
(2),(3),(4),(5),
((-I/2)*HoldForm[gR*Ui2Conj]*PRz),
(7),(8),(9),
((1/2)*HoldForm[huQ*Ui3]*PRz),
(11),(12),(13),
((1/2)*HoldForm[hdQ*Ui4]*PRz),
(15),(16),
((1/2)*HoldForm[huQ*Ui3Conj]*PLz),
(18),(19),(20),
((1/2)*HoldForm[hdQ*Ui4Conj]*PLz),
(22),(23),(24),
((-I/2)*HoldForm[gL*Vi1Conj]*PLa),
((-I/2)*HoldForm[gR*Vi2]*PRa),
((1/2)*HoldForm[huL*Vi3]*PRa),
((1/2)*HoldForm[hdL*Vi4]*PRa),
((1/2)*HoldForm[hdL*Vi4Conj]*PLa),
((1/2)*HoldForm[huL*Vi3Conj]*PLa)};
```

```
(*: Title:nfr.m     *)
(*:  Author:  M. R. Adcock     *)
(*: Revision History:
     Date Created: 02 March 1997
 *)

(*:  Summary:
This package contains the list of Feynman rules
for the neutralino/quark/squark and neutralino/lepton/splepton
Feynman diagrams defined in section 2.4.3 of thesis.
Note that only those used in the calculations have been
defined.*)

NeutFeynRule={(1),
(-(1/(2*Sqrt[2]))*HoldForm[(gL*Mi1+(gp/3)*Mi2)]*PLz),
(3),((1/(2*Sqrt[2]))*HoldForm[(gL*Mi1-(gp/3)*Mi2)]*PLz),
(-(7/(6*Sqrt[2]))*HoldForm[gp*Mi2]*PRz),(6),
((5/(6*Sqrt[2]))*HoldForm[gp*Mi2]*PRz),
(8),((1/2)*HoldForm[huQ*Mi3Conj]*PRz),
(10),(11),((1/2)*HoldForm[huQ*Mi3]*PLz),
((1/2)*HoldForm[hdQ*Mi4]*PRz),(14),(15),
((1/2)*HoldForm[hdQ*Mi4]*PLz),(17),
((I/(2*Sqrt[2]))*HoldForm[(gL*Mi1+gp*Mi2)]*PLa),
(19),((3 I/(2*Sqrt[2]))*HoldForm[gp*Mi2]*PRa),(21),(22),
((I/2)*HoldForm[hdL*Mi4]*PRa),((I/2)*HoldForm[hdL*Mi4]*PLa),
(25),(26)};
```

```
(*: Title:plotmass.m   *)
(*:  Author:  M. R. Adcock     *)
(*: Revision History:
     Date Created:  02 March 1997
*)

(*:  Summary:
This package calls the functions in susymass.m and
makes a plot of the mass eigenstates versus the
higgsino mass parameter. Change the call form ChmMassM[]
to NMassM[] to change plot from charginos to neutralinos. *)

<< susymass.m
r=Range[-200,200,1];
Print["Generating Mass State Plot Data"];
m=Table[NMassM[r[[i]]],{i,1,Length[r]}];
plotdata=r;
For[j=1,j<5,j++,
plotdata[[j]]=Table[{r[[i]],m[[i,3,j]]},
{i,Length[r]}]];
p1=ListPlot[plotdata[[1]],PlotJoined->True,
     PlotStyle->Dashing[{.01,.01}],
     DisplayFunction->Identity];
p2=ListPlot[plotdata[[2]],PlotJoined->True,
     DisplayFunction->Identity];
p3=ListPlot[plotdata[[3]],PlotJoined->True,
     PlotStyle->Dashing[{.01,.04,.01,.04}],
     DisplayFunction->Identity];
p4=ListPlot[plotdata[[4]],PlotJoined->True,
     PlotStyle->Thickness[.005],
     DisplayFunction->Identity];
Show[{p1,p2,p3,p4},AxesLabel->{"Neutralino Mass GeV",
"Higgsino Mass Parameter GeV"},Frame->True,
GridLines->Automatic,
     DisplayFunction->$DisplayFunction]
```

REFERENCES

[1] D. H. Kobe, *American Journal of Physics*, **46** No. 4 (1978)

[2] J. Goldstone, Nuovo Cimento, **19** 154 (1961).

[3] F. Halzen and A. D. Martin, *Quarks and Leptons*, Wiley, New York 1984

[4] M. Kaku , *Quantum Field Theory*, Oxford University Press, New York, 1993

[5] G. Grenier and J. Reinhardt, *Quantum Electrodynamics 2nd Edition*, Springer-Verlag, Rensselaer, N.Y.,1994

[6] B. De Wit and J. Smith, *Field Theory in Particle Physics*, North Holland, New York, 1986

[7] M. Janssen, *Supersymmetry Searches at Electron-Proton Colliders*, Doctoral Thesis, Universiteit Van Amsterdam, 1991

[8] H. J. W. Muller-Kirsten, and A. Wiedemann, *Supersymmetry An Introduction with Conceptual and Calculational Details*, World Scientific, Singapore 1987.

[9] H. Goldstein, *Classical Mechanics*, Addison Wesley , Reading Mass., 1981.

[10] S. Coleman and J. Mandula, *All Possible Symmetries of the S-Matrix*, Phys. Rev. **159**, 1251 (1967)

[11] M. F. Sohnius, Phys. Rep., **128**, 39 (1985)

[12] J. Wess and B. Zumino, *Nucl. Phys.* **B70**, 39 (1974)

[13] R. M. Francis, *Gauge Fields and Feynman Rules in a Fully Left-Right Supersymmetric Extension of the Standard Model*, Master's Thesis, Concordia University Press, Montreal, 1989

[14] R. N. Mohapatra, *Unification and Supersymmetry: The Frontiers of Quark-Lepton Physics*, Springer-Verlag, New York, 1986

[15] D. W. Duke and J. F. Owens, *Phys Rev.* D30(1984)49.

[16] S. Wolfram, *Mathematica 2nd Edition*, Addison-Wesley, Redwood City, 1991

[17] N. Blachman, *Mathematica: A Practical Approach, Prentice Hall*, New Jersey, 1992

[18] M. Frank, C. S. Kalman and H. N. Saif, *Z. Phys. C*, **59**, 655-667, 1993

[19] R. Mertig, *Guide to FeynCalc 1.0*, Fermilab FTP: fnth06.fnal.gov, (1991)

[20] A. Hsieh and E. Yehudai *HIP - Symbolic High-Energy Physics Calculations*, SLAC-PUB-5576, 1991

[21] C. S. Kalman, *Nuovo Cimento*, **A107**, 2805-2812 (1994)

[22] H. Eck and S. Kubleck, *Generating Feynman Graphs and Amplitudes with FeynArts*, Fermilab FTP: fnth06.fnal.gov, (1991)